

Pervasive Data Joiners Online Help - Table of Contents

Pervasive DataTools Data Joiner User's Guide



Pervasive Software Inc.

12365 Riata Trace Parkway

Building B

Austin, TX 78727 USA

About This Manual

This manual leads you through the operation of the Data Joiner user interface. The Pervasive Data Joiner allows you to join two heterogeneous data sets and export the joined results set to a CSV Text file. You must have a non-expired Data Joiner license to run this application.

Refer to the license.txt file in the default installation directory for disclaimers and information about trademarks and credits.

Table of Contents

Getting Started with Data Joiner

- 1 [Introduction to Data Joiner](#)
- 1 [Preferences](#)
- 1 [What do you want to do?](#)

Source Connections

- 1 [Source 1 Connection \(Tab 1\)](#)
- 1 [Source 2 Connection \(Tab 2\)](#)
- 1 [Browsing Source Data](#)

Join Settings

- 1 [Join Settings \(Tab 3\)](#)
- 1 [Testing the Join Design](#)

Exporting the Joined Data

- 1 [Exporting Joined Data](#)

Browsing the Data

- 1 [Browsing Source Data](#)
- 1 [Browsing the Exported Data](#)
- 1 [Browsing the Log File](#)

Saving and Reusing Connections

- 1 [Creating Reusable Source Connections](#)

Using the Query Builder

- 1 [Using the Query Builder](#)

Saving and Reusing Join Designs

Appendix

Introduction to Data Joiner

Data Joiner is an application used to join heterogeneous data. It allows you to do a full (or outer) join, an inner join, a left join or a right join. The Data Joiner user interface consists of three tabs for specifying source information and a run button for exporting the joined result set to an ASCII or Unicode delimited text file.

- 1 Source1 Connection
- 1 Source2 Connection
- 1 Join Settings

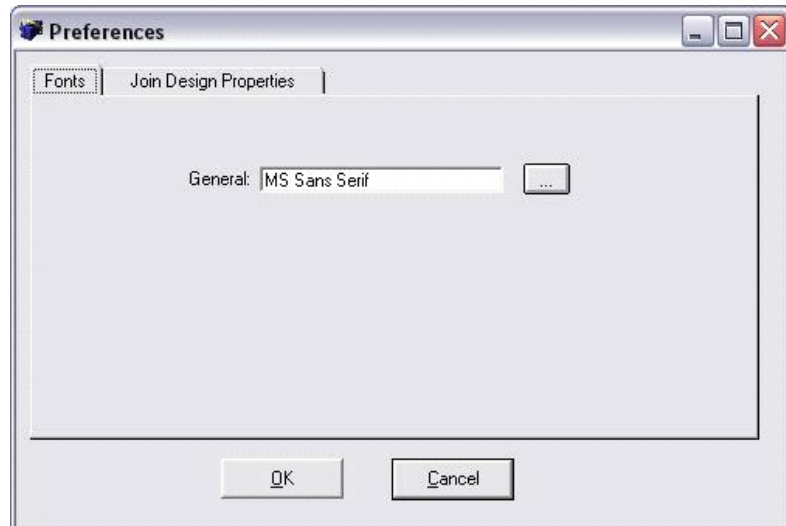
This application allows you to join only two source data sets. However, each source can be data from a file, a table, a view, or a SQL statement. When a source data set is described using a view or a SQL statement, the view or SQL statement itself could be joining multiple tables from the same database.

To successfully export joined data, you will first have to connect to Source 1, connect to Source 2, specify the keys to join on, and which fields from each source you want in the resulting data set. The Data Joiner allows you to browse each of your source data sets, test the join results and browse the exported result set.

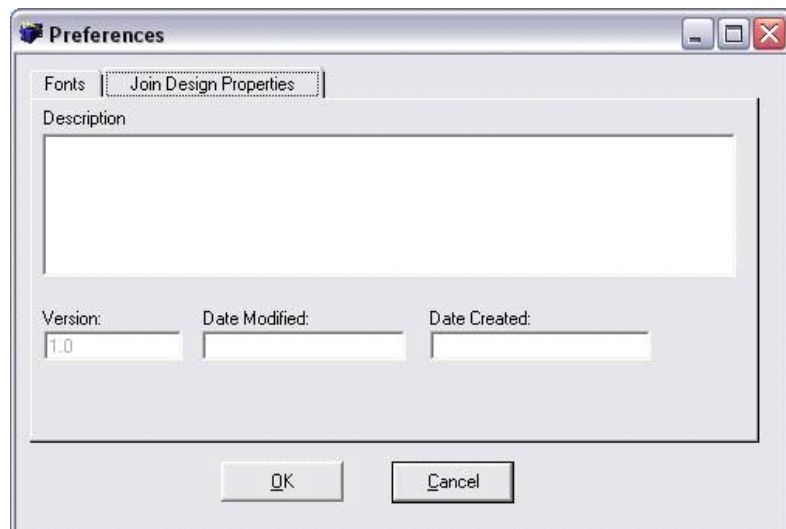
Preferences

You can access Join Designer preferences by selecting the View > Preference menu item. The preferences dialog allows you to set the application font and set a description for a join before you save the join design.

Join Designer Preferences - Fonts



Join Designer Preferences - Join Design Properties



What do you want to do?

For each task, we have provided a quick link to the section of the Data Joiner documentation that describes how to perform the specific task.

- Define a Join Source [Go To Source 1 Connection](#)
- Browse Source Data [Go To Browsing Source Data](#)
- Set Join Properties [Go To Join Settings](#)
- Test a Join Design [Go To Testing the Join Design](#)
- Export Joined Data [Go To Exporting Joined Data](#)
- Browse Exported Data [Go To Browsing Exported Data](#)
- Build a Query Statement [Go To Using the Query Builder](#)
- Create a Reusable Source Connection [Go To Creating a ReUsable Source Connection](#)
- Save a Join Design [Go To Saving the Join Design](#)

Source 1 Connection (Tab 1)

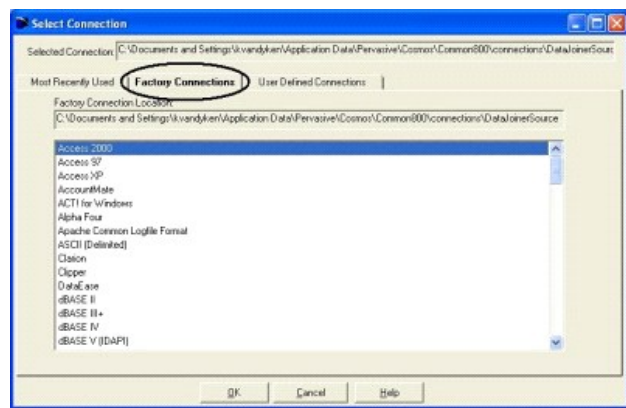
The first Data Joiner tab is for specifying one of your two sources. You can think of it as the left side of a join.

Source1 Connection Tab



Choose a source connector type by clicking on the down arrow to the right of the Source1 Connection textbox. The second tab of the connection selection dialog that pops up will display a list of Factory Connections. Once you have created a couple of designs, the first tab will display connections that were recently selected or used. The third tab of this dialog allows you to choose previously created User Defined Connections. See Also: Creating ReUsable Source Connections.

Select a Factory Connection

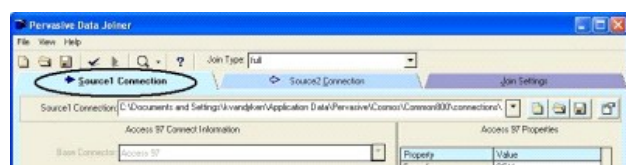


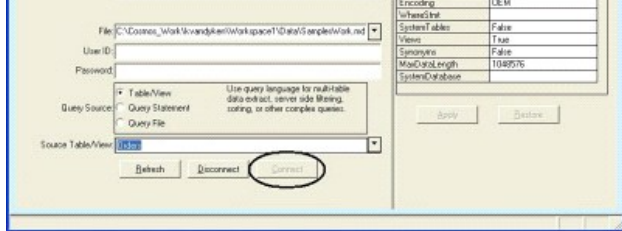
Once you have selected a connector type, you'll need to fill in the remaining source connection information. The connection information fields on Source1 Connection tab will vary depending on the type of connector selected. If your source connector is a SQL type source, you have the options of specifying a database table, an SQL statement, or an SQL file which is a file containing an SQL statement. For information on using the Query Builder to create an SQL statement, see Using the Query Builder.

You may also need to change the default connector properties in the grid to the right based on the particular source data you are trying to connect to. After changing any of the connector properties, press the Apply button to apply those changes.

After you enter enough information about your source connection, the application may automatically connect to the source. If it does not, press the connect button. You will know you are connected to the source data when the connect button is disabled and the connect arrow on the Source1 Connection turns solid.

Source1 Connection Completed





Source 2 Connection (Tab 2)

Data Joiner tab two is used to specify the connection information for your second source. You can think of this as the right side of a join. Select a connector type and fill in all appropriate information that allows a successful connection to the data.

Refer to Source 1 Connection (Tab 1) for details.

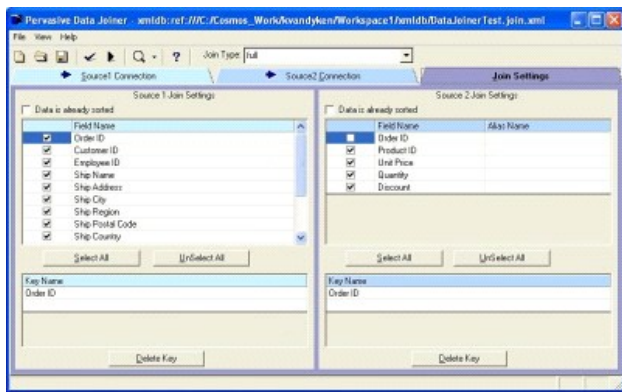
Join Settings (Tab 3)

Data Joiner tab three is used to specify additional join information about both sources. In particular, the join engine needs to know if your source data is already sorted. If it is not, the join engine will sort the data first. This will add additional processing time, especially for large data sets.

You can specify which fields in both sources you want in your joined results data set. You do not have to include all fields in the results. If source 2 contains a field with the same name as a source 1 field, the application will automatically enter an alias name for that field. You can change the alias name. The result set will contain the source 1 field name and the alias name for the source 2 field to avoid duplicate field names in the joined result set.

Finally, you will need to specify the key(s) for both sources. The keys will be used to join the data on the specified key fields.

Join Settings Tab



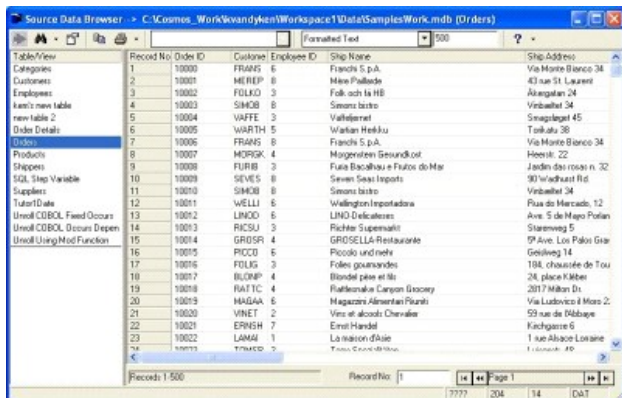
Browsing Source Data

Once you have successfully connected to a source data set, you can browse the data by selecting the View->Source1 Data or View->Source2 Data menu items or by clicking on the browse toolbar button drop down and selecting Browse Source1 Data or Browse Source2 Data. These menu items will only be enabled when a source connection has been made.

Browse Button



Browse Data Window



Testing the Join Design

At any time in the join design process, you can validate the join by pressing the validate toolbar button to see if you have included enough information to successfully join the two source data sets.

Validate Button



Upon successful validation of a join design, you can test the join by selecting the View>Joined Data (Test) file menu item or the browse toolbar>Browse Joined Data (Test) submenu item. Testing the join will do the following:

- 1 Save various join design files in a temporary directory
- 1 Join the data
- 1 Display the joined data in the data browser

Depending on the size of the source data sets as well as whether or not the source data sets need to be sorted, this could take some time.

Exporting Joined Data

Click the Export Joined Data toolbar button or select the File>Export Joined Data menu option to export the joined result set.

Export Joined Data Button



An export dialog will be displayed allowing you to choose the export target connector type and file location as well as set any target connector properties. In particular, if you want the first record in your target file to be a list of the result set field names, you'll want to set the Header property to True. You can also choose to export a subset of the joined record set either by choosing to export a sample number of joined records or by setting an export filter for exporting only certain joined records.

Export Options



When you have finished entering target export information, press the Export Joined Data button to export the data. A run status dialog will be presented. Depending on your source data set size and whether or not they are sorted, running the join and exporting the data can take some time.

Browsing the Exported Data

Once the export is complete, you can browse the exported data file by selecting the View->Exported Data menu item or the Browse toolbar button > Browse Exported Data sub menu.

Browse Button



Browsing the Log File

Exporting joined data performs a data transformation from the joined source to the export target. Statistics about this transformation will be logged to a file. Statistics include number of records read and number of records written. It will also include any errors that may have prevented the transformation from occurring. You can view the log file by selecting the View->Log File (DataJoiner.log) menu item or by clicking on the browse toolbar button and selecting the Browse Export Log File (Data Joiner.log) sub menu item.

Browse Button



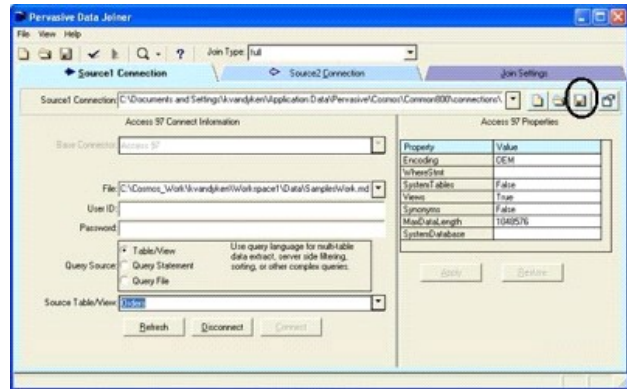
Creating Reusable Source Connections

If you find yourself using the same source in multiple join designs, you can enter the source connection information once, create a user defined connection, then select that user defined connection in future designs.

To create a user defined connection out of Source 1, fill in all the appropriate connection information on the first tab. Set whatever properties are needed for the

connection as well. After browsing the source data to verify the connection information is correct, press the save toolbar button that's to the right of the source connection textbox (top right).

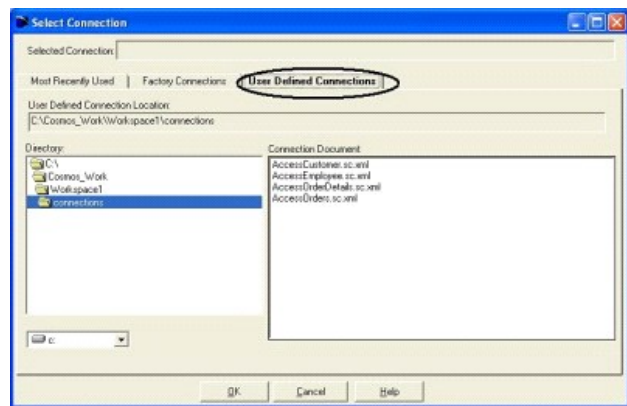
Save Connection Button



A dialog will appear asking you to name the user defined source connection. Connection files have a .sc.xml extension.

Once you have saved a user defined connection, you can use it in any future Join Design by clicking on the down arrow to the right of the Source1 Connection textbox, going to the third tab of the Select Connection dialog and choosing your saved connection. Tab 1 will automatically be populated with all connection information that was stored as a part of your user defined connection.

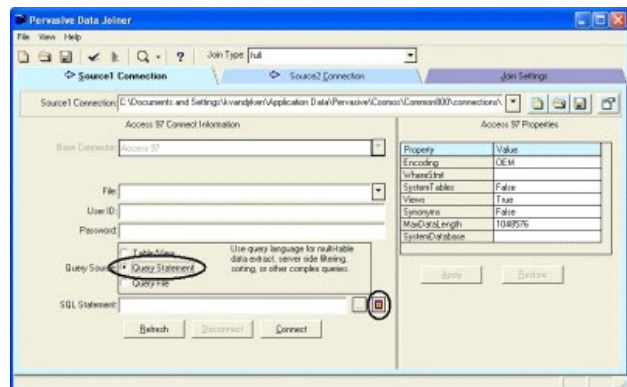
User Defined Source Connections



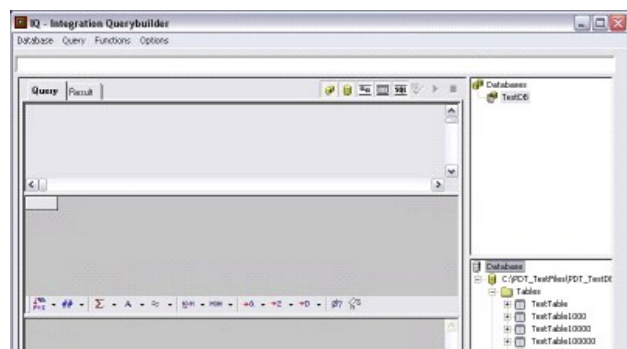
Using the Query Builder

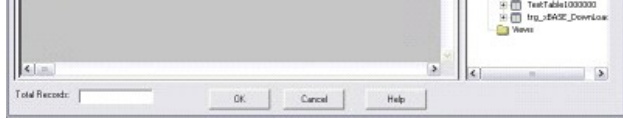
When creating a SQL statement to use for a source connection, you can use the Query Builder. You can access the Query Builder by selecting the Query Statement option and pressing the Build Query button.

Build Query Button



Query Builder Window



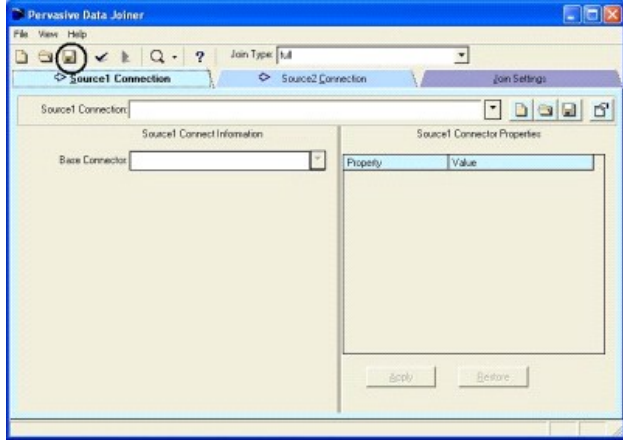


For documentation on how to use the Query Builder, see [Appendix A](#).

Saving the Join Design

Click the Save Join toolbar button or select the File->Save menu item to save a join design.

Save Join Button



Saving a join design saves a number of files needed by the join engine. Your saved join will consist of a JoinName.jd.xml file, JoinName_Source1.sv.xml and .svf.xml files, and JoinName_Source2.sv.xml and .svf.xml files. All of these will be saved a default workspace and repository created by Data Joiner.

Appendix A – Integration Query Builder

The Integration Query Builder consists of following five panes:

- 1 The upper-right Connections pane, where you set up database connections
- 1 The lower-right DB Browser pane, which displays the tables and views available for use in the currently active database
- 1 The upper-left Relations pane, in which relationships between database objects are defined
- 1 The center-left Columns pane, in which you select fields and define criteria, as well as grouping and sorting
- 1 The lower-left SQL pane, in which the finalized SQL command is displayed and more complicated editing can be done

Connecting to the Database

1. Select New Database Connection from the Database menu.
2. Enter a unique name for this connection in the Connection Name box.
3. Select the Data Access Type (see Table 1 below).

Table 1. Data Access Types

OLE DB Dynamic Data Access: ADO

If you select OLE DB Dynamic Data Access: ADO, the standard Windows Data Link Properties dialog box displays. Select the OLE DB provider you want to use from the list available. After selecting the appropriate OLE DB Provider, click the Next button to access the Connection tab to enter more details about the connection. The information required depends upon the provider selected. Click Test Connection to ensure you have entered the correct details.

OLE DB UDL link-file: ADO

If you already have the required connection information stored using a ".udl" file (Universal Data Link), select this type and navigate to the file using the Select Data Link File dialog box.

ODBC Machine Data Source: DSN or ODBC File Data Source: DAO

These ODBC data access options can be used when the data source does not support an adequate OLE DB Provider. They enable you to connect with data sources that have file data source names (DSNs). In general, it is recommended that you use the previously described ADO options rather than these.

In all cases when you have specified a valid connection and entered the required information, a new connection node is added in the upper-right Connections pane and you can click this in future to open this connection.

Integration Querybuilder SQL Functions

The following SQL functions are supported in the Integration Querybuilder:

Concatenate Functions



Highlight multiple rows in the Columns pane, and select the appropriate function from the menu. The selected fields are converted to string data type and concatenated as defined in the following table.>p>

Table 2. Concatenate and Formatting Descriptions

Fld1+", "+...+", "+FldN

Creates a string by concatenating all selected fields and inserting a comma followed by a space between them. For example: "Name1, Name2, Name3"

Fld1+", "+...+", "+FldN

Creates a string by concatenating all selected fields and inserting a comma between them. For example: "Name1,Name2,Name3"

Fld1+" "+...+" "+FldN

Creates a string by concatenating all selected fields and inserting a space between them. For example: "Name1 Name2 Name3"

Fld1+...+FldN

Creates a string by concatenating all selected fields without separating them. For example: "Name1Name2Name3"

Fld1+"-" "+...+"-" "+FldN

Creates a string by concatenating all selected fields and inserting a hyphen between them. For example: "Name1-Name2-Name3"

Fld1+" - "+...+" - "+FldN

Creates a string by concatenating all selected fields and inserting a space followed by a hyphen followed by space between them. For example: "Name1 - Name2 - Name3"

Fld1+", "+Fld2+" "+Fld3

Creates a string by concatenating the three selected files and inserting a comma and a space between the first and second fields and a space between the second and third. Used for last name, first name and middle name (or initial) format. For example: "Name1, Name2 Name3"

Fld1+" "+Fld2+"." "+Fld3

Creates a string by concatenating the three selected files and inserting a space between the first and second fields and a period followed by a space between the second and third. Used for first name, middle initial, last name format. For example: "Name1 N. Name3"

Mask Functions



Highlight a row from the Columns pane and select the appropriate function from the menu. The selected field is converted to string data type and formatted as described below:

Table 3. Masks and Formatting Descriptions

Phone [LO] : 1-999-999-9999

Converts the first 11 characters to a string using this phone number format.

Phone [LO] : 1 999-999-9999

Converts the first 11 characters to a string using this phone number format.

Phone [AC] : (999) 999-9999

Converts the first 10 characters to a string using this phone number format.

Phone [AC] : 999-999-9999

Converts the first 10 characters to a string using this phone number format.

Phone [AC] : 999.999.9999

Converts the first 10 characters to a string using this phone number format.

Phone [ACX]: (999) 999-9999 (ext 00000)

Converts the first 15 characters to a string using this phone number plus extension format.

Phone [SH] : 999-9999

Converts the first 7 characters to a string using this phone number format.

Area Code : (999)

Converts the first 3 characters to a string using this area code format.

Extension : (ext 00000)

Converts the first 5 characters to a string using this extension number format.

Extension : -X00000

Converts the first 5 characters to a string using this extension number format.

SS# : 999-99-9999

Converts the first 10 characters to a string using this social security number format.

Fed ID : 99-9999999

Converts the first 9 characters to a string using this federal ID number format.

Zip Code : 99999-9999

Converts the first 9 characters to a string using this zip code format.

Bank Acc : 99999-99999

Converts the first 10 characters to a string using this bank account number format.

Cred Card : 9999-9999-9999-9999

Converts the first 16 characters to a string using this credit card number format.

ISBN : 9-99-999999-9

Converts the first 10 characters to a string using this ISBN format.

ISBN : 9-999-99999-9

Converts the first 10 characters to a string using this ISBN format.

ISBN : 9-9999-9999-9

Converts the first 10 characters to a string using this ISBN format.

ISBN : 9-99999-999-9

Converts the first 10 characters to a string using this ISBN format.

ISBN : 9-999999-99-9

Converts the first 10 characters to a string using this ISBN format.

Aggregate Functions



Aggregate functions return a single value based on a set of other values, such as the sum of these values, including the maximum and minimum. The aggregate function can be applied to all the values in a particular field in the query, or to groups of these values by applying the Group By clause.

The aggregation functions available depend on the SQL variant and syntax used by the data source. For a full description of each function, refer to the vendor's documentation.

Other than COUNT, most aggregation functions are designed to work with numeric fields and unexpected results can occur if you try to aggregate strings.

String Functions



String functions operate on single text strings and are used to obtain and manipulate characters or character information, such as extracting a substring of one or more characters or trimming blank spaces from a field.

The functions available from this menu depend on the SQL variant and syntax used by the data source. Refer to the vendor's help or documentation for a full description of these functions.

Numeric Functions



Numeric functions operate on single numeric fields and are used to obtain simple mathematical information, such as the sign of a number, the next highest integer or its absolute value.

The functions available from this menu depend on the SQL variant and syntax used by the data source. Refer to the vendor's help or documentation for a full description of these functions.

Date Part Functions



Date Part functions operate on date fields and are used to extract and convert a portion of a date into a character string such as day of week, day of year, quarter, hour and minute.

The functions available from this menu depend on the SQL variant and syntax used by the data source. Those shown may only be the most commonly used of those available. Refer to the vendor's help or documentation that was provided for a full description of these functions.

Date Name Functions



Date Name functions operate on date fields and are used to extract and convert a portion of a date into a named date element such as month name or weekday name.

The functions available from this menu depend on the SQL variant and syntax used by the data source. Refer to the vendor's help or documentation for a full description of these functions.

Conversion to Character Functions



Conversion to Character functions convert numeric fields to text fields, then apply the selected formatting to enable different formats for currency and dates and times.

The functions available from this menu depend on the SQL variant and syntax used by the data source. Refer to the vendor's help or documentation for a full description of these functions.

Conversion to Number Functions



Conversion to Number functions convert numbers held as text strings to their numeric values. Strings containing non-numeric values will be converted to 0.

The functions available from this menu depend on the SQL variant and syntax used by the data source. Refer to the vendor's help or documentation for a full

description of these functions.

Conversion to Date Functions



Conversion to Date functions convert numeric fields to their date equivalent in the selected format.

The functions available from this menu depend on the SQL variant and syntax used by the data source. Refer to the vendor's help or documentation for a full description of these functions.

NULL Case



This button inserts code that will deal with a field containing a NULL value. Different SQL data sources will handle this differently, but should allow you to substitute some other value into the field if a NULL value is encountered. For example, in Access it might be:

```
IIf(IsNull(Fld), vNullCaseValue, Fld)
```

Whereas in Oracle it might be:

```
NVL(Fld, vNullCaseValue)
```

In both cases you replace vNullCaseValue with the value to be displayed (or further processing to be performed) if NULL is returned.

Simple Case



This button inserts code that deals with a simple either/or condition. Different SQL data sources handle this differently. For example, in Access it might be:

```
IIf(Fld = when_expression_1, result_expression_1, else_expression)
```

Whereas in Oracle it might be:

```
DECODE(Fld, when_expression_1, result_expression_1, else_expression)
```

Working in the Relations Pane

Once you have connected to a data source, your connection is displayed in the upper-right pane. You can set up and save as many data source connections as you need. Integration Querybuilder stores all connections you create unless you explicitly delete them.

Creating Queries on the Relations Pane

To create new queries on the Relations pane:

1. Double-click the connection you want to use. The DB Browser in the lower-right pane displays the database.
2. Click the database icon to display the icons for tables, views and procedures for this database.
3. Select New Query from the Query menu. A new query icon opens beneath the connection icon in the upper-right pane.
4. Drag the tables and views you want to use into the Relations (upper left) pane. SELECT... FROM statements are created in the SQL pane. If tables are already linked in the database, these links are displayed, although these can be changed or removed for the purpose of this particular query.

If you are using a table more than once, the second and further copies are renamed. For example, if you already have a Customer table in the Relations pane and you drag across another copy, it is automatically renamed Customer1.

NOTE: Although views can be used for creating relationships if supported by the database you are using, for ease of reading, this section only refer to tables.

Creating a Join in the Relations Pane

When two tables each contain a column that shares a common set of values, they can be linked on this column by a join. For example, a table of orders and a table of customers could be linked if both contain a Customer ID column.

To create a join between two tables in the Relations pane:

1. Click and drag the common column to the common column in the other table. A join line now connects the two columns and a JOIN statement inserted into the FROM clause in the SQL pane.

If the fields you have connected were set as primary and foreign keys in the original database, "key" and "infinity" icons appear at either end of the join. If these icons do not appear it may be that the two columns do not contain common values.

The default join between two tables is called an Inner Join. This means that the query only returns those records where the fields on both sides of the join

match. Unmatched records are discarded.

2. To change the join type, right-click the diamond icon in the middle of the join.

You can change the join to any of the following:

Available Joins and Their Descriptions

Left Outer Join

All records in the table on the left side of the join are returned, even if there is no matching record on the right side. To create, right-click the diamond icon and click the first Select All Rows From option.

Right Outer Join

All records in the table on the right side of the join are returned, even if there is no matching record on the left side. To create, right-click the diamond icon and click the second Select All Rows From... option.

Full Outer Join

If full joins are permitted by the database connection, then all records from both tables are returned, regardless of matching records in the other table. To select this, open the Join Line Properties dialog box by right-clicking the diamond icon and selecting Properties. Check both Include Rows boxes.

You can also create joins that are based on other logical conditions than the two selected columns being equal. Do this by right-clicking the diamond and selecting Properties. This is not recommended unless there is no other way to create a query and you know exactly what you are doing, as it can produce an extremely large number of records.

Selecting a Field in the Columns Pane

The Columns pane allows you to select the fields you want to use in your query. If field names are more than one word long they are displayed in the column cells surrounded by square brackets.

City
[Customer ID]

In the column list, fields are listed by the table they appear in and names are not bracketed.

Customers.City
Orders.Customer ID

An asterisk (*) is used to indicate all fields, so Employees.* means all fields in the Employees table.

To select a field in the Columns pane:

1. Click in the first empty cell beneath Column. Click the arrow to display the list of fields.
2. Scroll to the field you want to use and click it. The field name appears in the cell and in the SELECT statement in the SQL pane.
3. If you want to use a different name for this field when the query is generated, type the new name in the Alias cell.
4. Select or clear the Output check box.
5. If you want to sort the records by this field, click in the Sort Type cell and select either Ascending or Descending.
6. If you are sorting on more than one field, enter the sort order ranking for this field in the Sort Order cell. For example, if you were sorting by five fields, you would select "1" if this were the first field to sort on or "5" if it were the last.
7. Select the Group By check box to group identical occurrences of this field.
8. Specify the criterion, if any, to be applied to the Criteria cell. For example, if you had a Total Cost field and only wanted to return items where the total cost was more than \$30.00 you would enter ">30.00" here. The code generated by your criteria will be displayed in the SQL pane as a WHERE clause.

The value you use in a Criteria cell can be preceded by one of the following operators: "=", "<", ">", "<=", ">=", "!<", "!>", "!=", "<>", "IS", "IS NOT", "NOT", "IN", "LIKE", "BETWEEN". If no operator is entered, the default is "=".

Values following the LIKE operator will be put into quotes unless already so. The LIKE operator enables you to use regular expressions.

The wildcards and syntax available for use in regular expressions will depend upon your source database, so you should refer to the documentation supplied for further information. They should not be confused with those used in RIFL regular expressions.

Each time you define a criterion, another criterion cell opens, enabling you to set up a string of OR conditions for this column. For example, to select records where the Employee ID is less than 10000, between 10500 and 10750 or greater than 20000:

Column	Criteria	Or	Or
[Employee ID]	<10000	BETWEEN 10500 AND 10750	>20000

Which will result in the following SQL clause:

```
WHERE  
([Employee ID] < 10000 OR  
[Employee ID] BETWEEN 10500 AND 10750 OR  
[Employee ID] > 20000)
```

OR criteria are created horizontally, whereas AND criteria vertically. For example, to select records where the Employee ID is greater than 10000 and the Last Name is not Smith:

Column	Criteria
[Employee ID]	>10000

[Last Name] <> "Smith"

Which will result in the following SQL clause:

WHERE
([Employee ID] > 10000 AND
[Last Name] <> "Smith")

To set up an OR criteria that tests two separate fields, do the following:

Column	Criteria	Criteria
Balance	> 400	
[Last Payment]		> 100

Which will result in the following SQL clause:

WHERE
([Balance] > 400 OR
[Last Payment] > 100)

To combine OR and AND criteria, remember that criteria are grouped with the criteria in the same column. As in the following example:

Column	Criteria	Or...
[Employee ID]	>10000	
[Last Name]	= "Smith"	= "Jones"

WHERE
([Employee ID] > 10000 AND
[Last Name] = "Smith" OR
[Last Name] = "Jones")

[Employee ID] > 10000 is only used with [Last Name] = "Smith". Records will be returned when the [Last Name] = "Jones" regardless of employee ID.

To return all records where the customer ID is greater than 20000 and the company name is Smith, Jones or Featherstone you will need to set up the following criteria:

Column	Criteria	Or...	Or...
[Employee ID]	>20000	>20000	>20000
[Company Name]	= "Smith"	= "Jones"	= "Featherstone"

WHERE
([Employee ID] > 20000 AND
[Last Name] = "Smith" OR
[Employee ID] > 20000 AND
[Last Name] = "Jones" OR
[Employee ID] > 20000 AND
[Last Name] = "Featherstone")

Working in the SQL Pane


Integration Querybuilder's SQL pane displays the SQL statement as you create it using the Relations and Columns panes. However, it can also be used by more experienced users as a SQL editor to create queries that cannot be put together simply using the Relations and Columns panes.

Caution: If you are not familiar with writing SQL by hand, do not edit the code Querybuilder generates, as changes made in the SQL pane override the settings you created in the Relations and Columns panes, and you might have to begin the query again.

Incorporating SQL Functions

To use SQL functions:

1. In the SQL Pane, click the cursor at the appropriate point in the SQL statement where you want the new function to be inserted. You can create new lined, press the TAB key and insert blank spaces as with any text editor to improve the readability of the code.
2. In the Column pane, select the required field or fields. Hold down the CTRL key and click each field if you are using a function that concatenates multiple fields.
3. Select the required function from the SQL functions menu. Caution: If you have not selected a field or fields in the Column pane, the field or fields highlighted in the Relations pane are used even if you have not selected this field or fields. If you select any code in the SQL Pane prior to selecting a function, this code is overwritten.
4. If you have put a new statement after an existing statement in the SELECT section, insert a comma after the existing code. Because of the way Querybuilder validates code, you have to add the comma after you have created the function. The selected function, with the field name(s) inserted, is created at the cursor point.
5. Test the validity of your SQL query by clicking the Check SQL Syntax icon.

 If the syntax is valid, a new line containing this function is created in the Columns pane. Enter a short, meaningful name for it in the Alias cell. If the syntax is invalid an error message displays. The Relations and Columns pane is grayed-out until the SQL code is corrected and validated again.

Defining SQL Manually

To manually define SQL:


If you are an experienced SQL user, type query statements directly into Integration Querybuilder using this pane as an editor and the Check SQL Syntax button to validate the code as you build it. You can also paste text into the SQL pane from external programs, online help and e-mails.

If you copy existing material into the SQL pane, you must remove any comments, as these will cause an error when the query is exported to Map Designer.

If you set up a join conditions in the SQL pane rather than in the Relations pane and the join condition has an expression instead of a field on one side, the join line (in the Relations pane) will be drawn from a field of one object to the title bar of the other. The diamond icon will display "f*". If both sides of the join condition are expressions rather than fields the join line is drawn between the title bars of both objects.

Querybuilder does not validate or run any query containing DJX. Test and validate your statement using dummy information and substitute the DJX code in Map Designer's Query Statement box afterwards.

To run the query:

1. Click the Execute Query button.
 The Result tab displays showing the result of this query. If the query takes a noticeable time to execute, the Stop button becomes active so that the query can be canceled if required.
2. Toggle between the Query and Result tabs of Integration Querybuilder if you need to fine-tune the query and check the result after each adjustment.
3. Once you are satisfied with the query you have created, click OK (located at the bottom of the window). Save the query with a meaningful name.
4. Integration Querybuilder closes and the generated SQL code is copied into the SQL Statement box. To see the code formatted, click the ellipsis to display it in a separate window.