

Designer 4.0 and 5.0

► Optimizing object definitions with @Select and @Where functions

You can use @Functions in Designer to allow you to use more flexible methods to specify SQL in object definitions. The @Functions available in Designer are each described in the Designer's guide, but this article aims to give you more detailed information on the specific use of the @Select and @Where functions. You can use these two functions to save coding time and to help you create more accurate code. Future articles will be published with examples to describe in more detail the use of the other @Functions.

What are @Functions?

@Functions are special functions available in the Functions pane of the Edit Select Statement and Edit Where Clause dialog boxes for an object. @Functions provide more flexible methods for specifying SQL. For example, you can use @Functions to create interactive objects, optimize object Select statements and Where clauses, and use data from summary tables in the database.

This article describes two of the @Functions available in Designer;

@Function	Description
@Select	Allows you to reuse the Select statement of another object.
@Where	Allows you to reuse the Where clause of another object.

What are the advantages of using the @Select and @Where functions?

Using the @Select and @Where functions creates a dynamic link between an original object (parent object) and a referencing object (child object). When the Select statement or Where clause of the original object is modified, the Select statement or Where clause of the referencing object is automatically updated.

The @Select and @Where functions allow you to use existing code, which has the following advantages:

- You have to maintain only one instance of the SQL code.
- Ensures consistency of the code.

Using the @Select function

The @Select function allows you to re-use the Select statement of another object. You insert the @Select function in the Select statement of an object. You then specify the path of another object in the universe as a parameter of the @Select function as follows:

Class_Name\Object_Name.

This statement then acts as a pointer to the Select statement of the referenced object.

Syntax

The syntax of this function is the following:

```
@Select(Classname\Objectname)
```

Classname is the name of the class that contains the object that you want to reference.

Objectname is the name of the object whose Select statement you want to reference in the current object.

Example

An object called Car Model has the following code in the Select box of its Definition page:

```
MODEL.MODEL_NAME & ' ' & MODEL.MODEL_TRIM & ' ' & MODEL.MODEL_ENGINE
```

This same code is used in another object called Model for Rental. You can optimize the code usage by referencing the Car Model code in the Select box for the Model for Rental object as follows:

```
@Select(Car\Model)
```

To use the @Select function

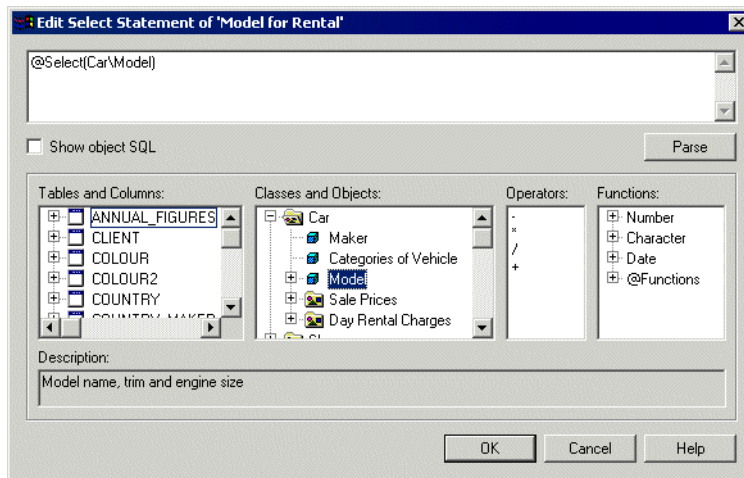
You can use the @Select function as follows:

1. Double click an object.
The Edit Properties dialog box for the object appears.
2. Click the >> button next to the Select box.
The Edit Select Statement dialog box appears.
3. Click in the Select box at the place where you want to add the @Select function.
4. In the Functions pane, expand the @Functions node and double click the @Select function.
The @Select function with empty parameters appears in the Select box.

5. In the Classes and Objects pane, expand the class folder that contains the object that you want to reference.
6. Double click the object that you want to reference.

The parameters for the @Select function are filled with the selected Classname/ Objectname values.

For example, the Edit Select dialog box for an object Model for Rental appears below. In the classes and Objects pane, the Car Class has been expanded and the Model object has been double clicked so that the function @Select(Car\Model) appears in the Select box.



7. Click OK in each of the dialog boxes and save the universe.

The Select statement for the child object (Model for Rental) is now automatically updated with any modifications made to the Select statement of the parent object (Model).

Using the @Where function

The @Where function allows you to re-use the Where clause of another object. You insert the @Where function in the Where clause box of an object. You then specify the path of another object in the universe as a parameter of the @Where function, in the form Class_Name\Object_Name. This then acts as a pointer to the Where statement of the referenced object.

Syntax

The syntax of this function is the following:

```
@Where (Classname\Objectname)
```

Classname\Objectname is the name of a class, and the name of the object whose Where clause you want to reference in the current object.

Example

An object called Resort Service Lines has the following code in the Where box of its Definition page:

```
Resort.resort = @Prompt('Enter a resort name','A','Resort\Resort','mono',)
```

This same code is used in another object called Resort. You can optimize the code usage by referencing the Resort object code in the Where box for the Resort Service Lines object as follows:

```
@Where(Resort\Resort)
```

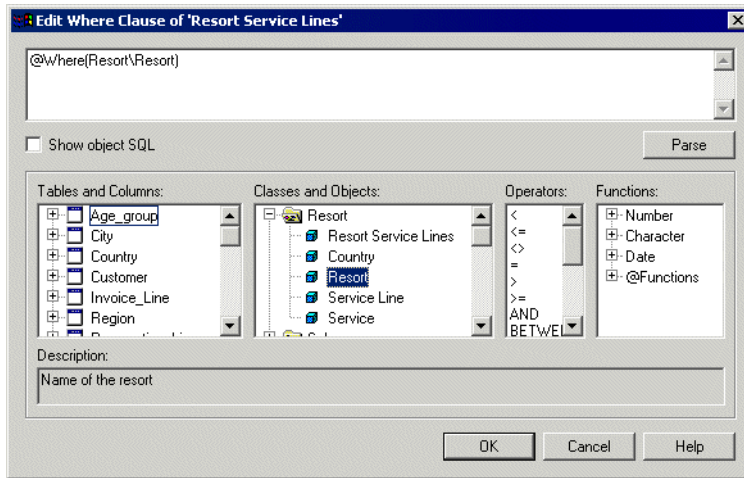
The object Resort Service Lines references the Where clause in the object Resort.

To use the @Where function

You can use the @Where function as follows:

1. Double click an object.
The Edit Properties dialog box for the object appears.
2. Click the >> button next to the Where box.
The Edit Where Clause dialog box appears.
3. Click in the Where box at the place where you want to add the @Where function.
4. In the Functions pane, expand the @Functions node and double click the @Where function.
The @Where function with empty parameters appears in the Where box.
5. In the Classes and Objects pane, expand the class folder that contains the object that you want to reference.
6. Double click the object that you want to reference.
The parameters for the @Where function are filled with the selected Classname/ Objectname values.

For example, the Edit Where Clause dialog box for an object called Resort Service Lines appears below. In the classes and Objects pane, the Resort Class has been expanded, and the Resort object has been double clicked so that the function @Where(Resort/Resort) appears in the Where box.



7. Click OK in each of the dialog boxes and save the universe.

The Where clause for the child object (Resort Service Lines) is now automatically updated with any modifications made to the Where clause of the parent object (Resort).