crystal decisions™

Crystal Reports

Custom Functions

Overview

Custom functions are procedures you create to evaluate, make calculations on, or transform data in formulas. When using a custom function in a formula, all the operations in its definition are performed without having to specify them individually in the formula. Custom functions provide a way to share and reuse formula logic, which makes report creation easier and less time consuming.

Custom functions are possible in Crystal Reports 9 and later.

Contents

Introduction

This document gives step-by step instructions on creating custom functions using the Formula Editor, Formula Expert and Formula Extractor. It also explains how to categorize custom functions and how to add custom functions to the Repository.

Create Custom Functions Using the Formula Editor

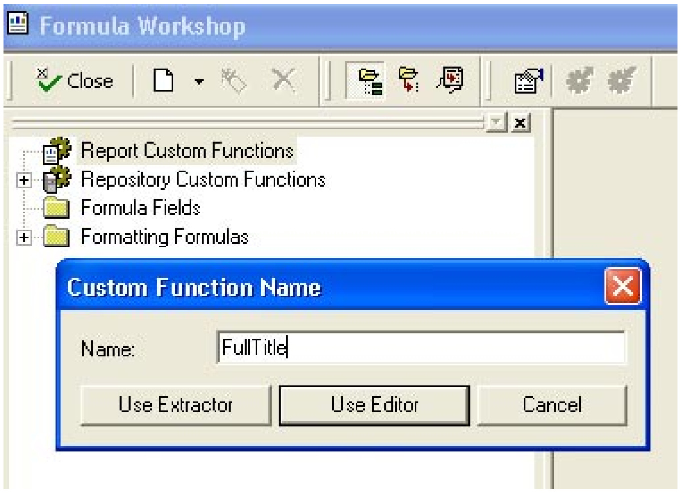
The Formula Editor is used to manually create a custom function.

Example:

You are asked to design a series of reports from a database that stores contact titles as abbreviations (for example; 'Mr.' Instead of the full 'Mister'). Instead of these abbreviations, you want the full title to appear on the report. You could create a formula that would do this but you would need to copy this formula for each report. Instead, extract the logic that converts the data and save it as a custom function.

Create a Custom Function

1. Create a new report based on the *Customer* table from the Xtreme.mdb sample database installed with Crystal Reports.
2. On the **Report** menu, click **Formula Workshop**.
3. Right-click **Report Custom Functions**, and then click **New**.
4. Name the custom function *FullTitle*.
5. Click the **Use Editor** button.



Name: FullTitle

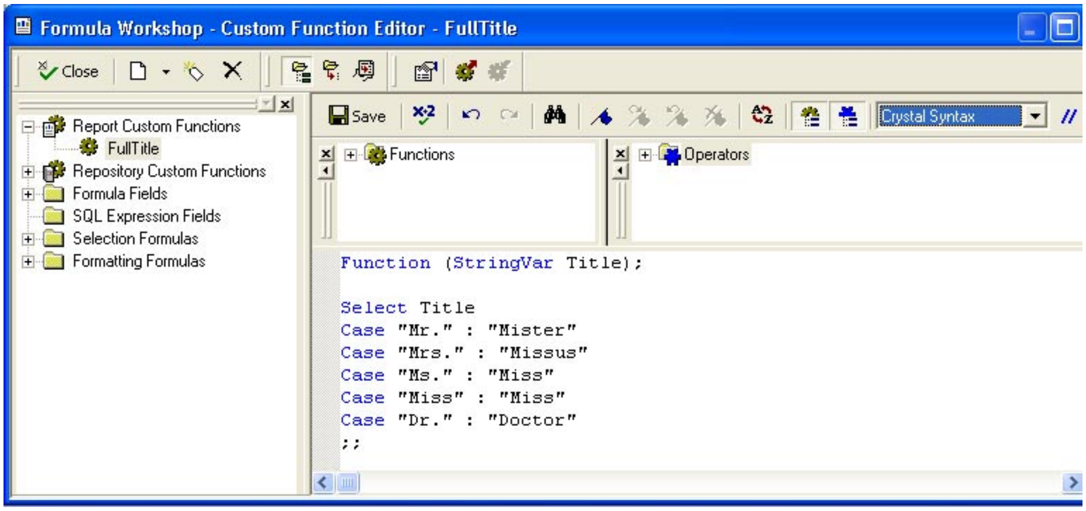
Figure 1.1

6. Click **Crystal Syntax** from the drop-down box.

7. For this example, we will be passing the function a field, which contains title abbreviations as text. This function's argument has a StringVar data type. This function will use a Select Case statement that will look at each value from the database field and change the title abbreviation to a full title.

   Enter the following in the Formula Editor:

```
Function (StringVar Title);
Select Title
Case "Mr." : "Mister"
Case "Mrs." : "Missus"
Case "Ms." : "Miss"
Case "Miss" : "Miss"
Case "Dr." : "Doctor"
;;
```



```
Function (StringVar Title);

Select Title
Case "Mr." : "Mister"
Case "Mrs." : "Missus"
Case "Ms." : "Miss"
Case "Miss" : "Miss"
Case "Dr." : "Doctor"
;;
```

Figure 1.2

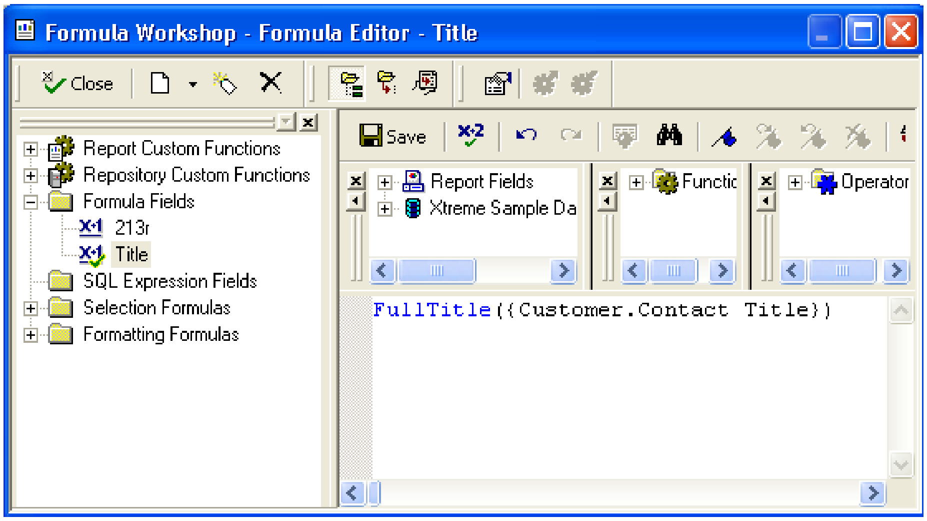
Use the Custom Function in a Formula

With the custom function created, you can now use it in a formula.

1. In the Formula Workshop, right click **Formula Fields**, and then click **New**.
2. Name the formula *Title*, and then click the **Use Editor** button.
3. Enter the following in the Formula Editor:

```
FullTitle({Customer.Contact Title})
```

This will use the *FullTitle* custom function to convert the *Customer.Contact Title* field from an abbreviation to the full title.



```
FullTitle({Customer.Contact Title})
```

Figure 1.3

4. Click the **Save** button, and then close the **Formula Editor**.
5. Insert the *Customer.Contact Title* field and the *Title* formula into the Details section of the report.

When the report is previewed, the *Title* formula will show the full titles next to the abbreviated titles from the *Customer.Contact Title* field.

Create Custom Functions Using the Formula Expert

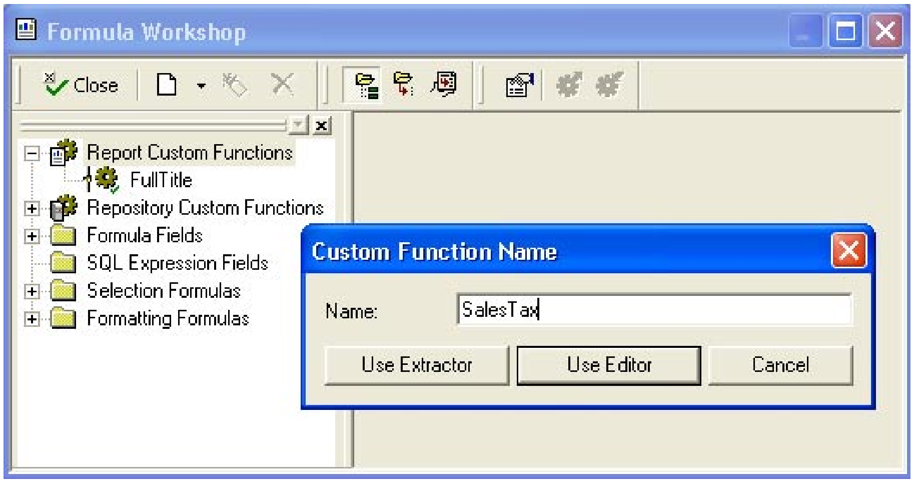
The Formula Expert can be used to create a formula quickly, based on a custom function.

Example:

You want to create a custom function that adds seven percent sales tax to a currency field. Then, you want to create a formula that uses it. In this example, follow these steps to create a formula off a custom function using the Formula Expert.

Create a Custom Function

1. Create a new report based on the *Customer* table from the Xtreme.mdb sample database installed with Crystal Reports (or use the same report from the previous example).
2. On the **Report** menu, click **Formula Workshop**.
3. Right-click **Report Custom Functions**, and then click **New**.
4. Name the custom function *ValueOutput*.
5. Click the **Use Editor** button.



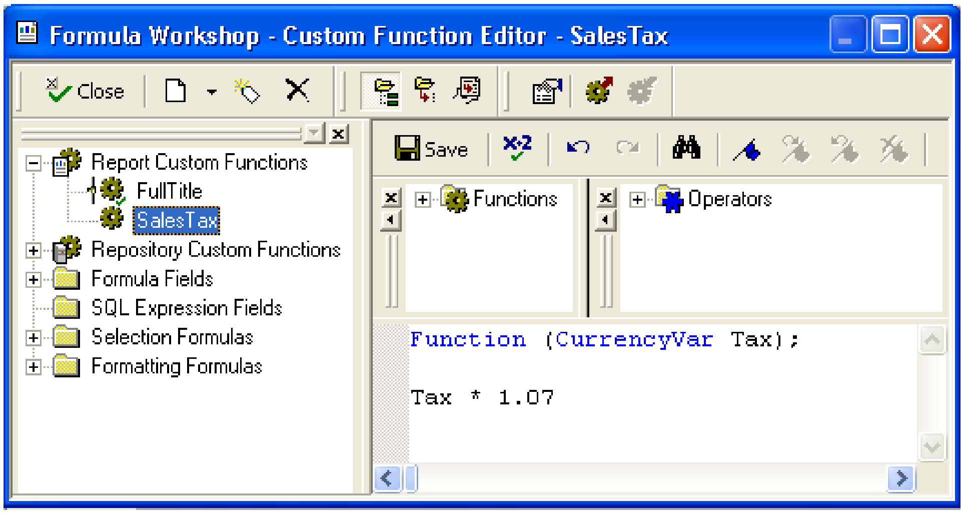
Name: SalesTax

Figure 2.1

6. Click *Crystal Syntax* from the drop-down box.

7. In the Custom Function Editor, enter the following:

```
Function (CurrencyVar Tax);
Tax * 1.07
```



```
Function (CurrencyVar Tax);

Tax * 1.07
```

Figure 2.2

8. Click the **Save** button, and then close the **Custom Function Editor**.

Use the Custom Function in a Formula

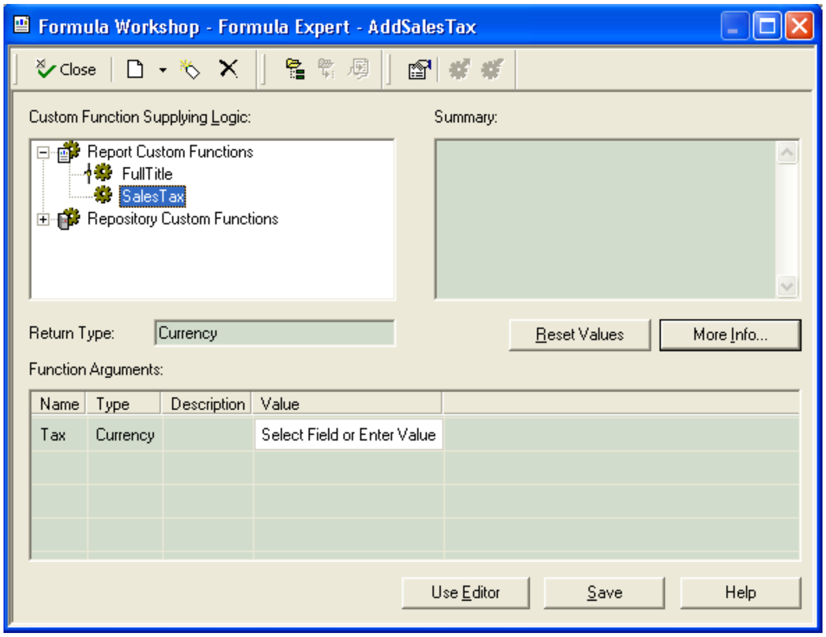
With the custom function created, we are ready to use this custom function in a formula.

1. In the Formula Workshop, right click **Formula Fields**, and then click **New**.

2. Name the Formula “AddSalesTax”, and then click the **Use Expert** button.

3. In the **Custom Function Supplying Logic** box, click the *SalesTax* Report Custom Function.

Click the **More Info** button to see the properties window for a custom function.



Custom Function Supplying Logic:

- Report Custom Functions
  - FullTitle
  - SalesTax
- Repository Custom Functions

Summary:

Return Type: Currency

Function Arguments:

Name	Type	Description	Value
Tax	Currency		Select Field or Enter Value

Figure 2.3

4. In the **Function Arguments** box, click *Customer.LastYears Sales* from the drop-down box under **Value**. This makes the *Customer.LastYears Sales* field the argument for the custom function.

5. Insert the **Save** button, and then close the **Formula Expert**.

6. Insert the *Customer.Last Year's Sales* field and the *AddSalesTax* formula into the details section of the report.

When the report is previewed, the *AddSalesTax* formula will show the *Customer.Last Year's Sales* field plus seven percent.

Create Custom Functions Using the Formula Extractor

The Formula Extractor is used to create a custom function based on an existing formula.

Example:

You have a formula that converts the *Customer.Contact Position* field to a corresponding number. You would like to create a custom function off that formula so that it can be used in other reports. In this example, follow these steps to create a custom function using the Formula Extractor.

Create a Formula

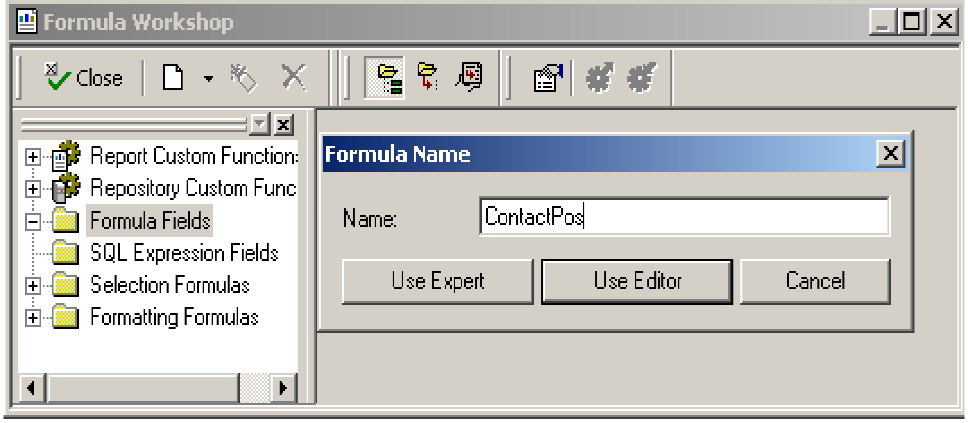
1. Create a new report based on the *Customer* table from the Xtreme.mdb sample database installed with Crystal Reports.

2. On the **Report** menu, click **Formula Workshop**.

3. Right-click **Formula Fields**, and then click **New**.

4. Name the Formula "ContactPos".

5. Click the **Use Editor** button.

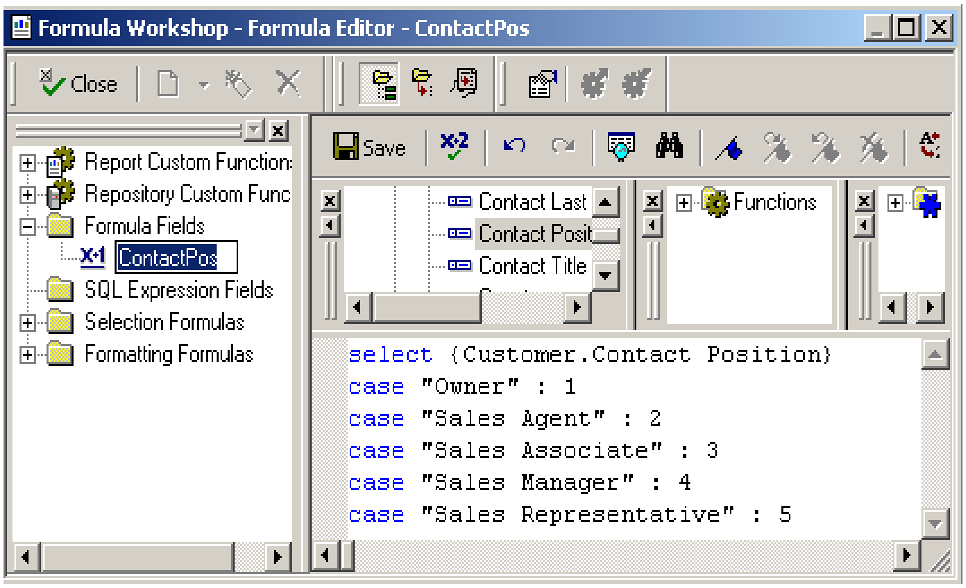


Name: ContactPos

Figure 3.1

6. Enter the following in the Formula Editor:

```
select {Customer.Contact Position}
case "Owner" : 1
case "Sales Agent" : 2
case "Sales Associate" : 3
case "Sales Manager" : 4
case "Sales Representative" : 5
```



```
select {Customer.Contact Position}
case "Owner" : 1
case "Sales Agent" : 2
case "Sales Associate" : 3
case "Sales Manager" : 4
case "Sales Representative" : 5
```

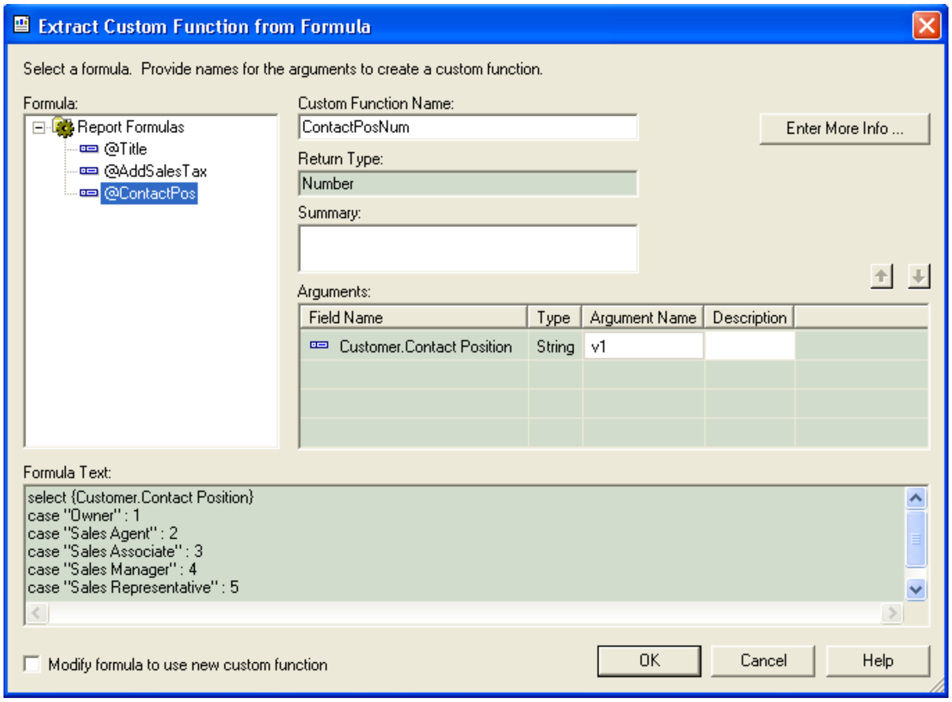
Figure 3.2

7. Click the **Save** button, and then close the Formula Editor.

Create a Custom Function Based on a Formula

With the formula created, you can now create a custom function from it.

1. In the Formula Workshop, right-click **Report Custom Functions**, and then choose **New**.
2. Name the custom function “ContactPosNum”.
3. Click the **Use Extractor** button.
4. In the **Extract Custom Function from Formula** dialog box, click the *ContactPos* formula.
5. Click **OK** and you will now see the *ContactPosNum* custom function in the Formula Workshop.



Select a formula. Provide names for the arguments to create a custom function.

Formula:
- Report Formulas
  - @Title
  - @AddSalesTax
  - @ContactPos

Custom Function Name: ContactPosNum

Return Type: Number

Summary:

Arguments:

Field Name	Type	Argument Name	Description
Customer.Contact Position	String	v1	

Formula Text:

```
select {Customer.Contact Position}
case "Owner" : 1
case "Sales Agent" : 2
case "Sales Associate" : 3
case "Sales Manager" : 4
case "Sales Representative" : 5
```

☐ Modify formula to use new custom function

Figure 3.3

NOTE	The Formula Extractor fills in all the necessary arguments. In this case the *Customer.Contact Position* field is used as a string argument.

Categorize Custom Functions

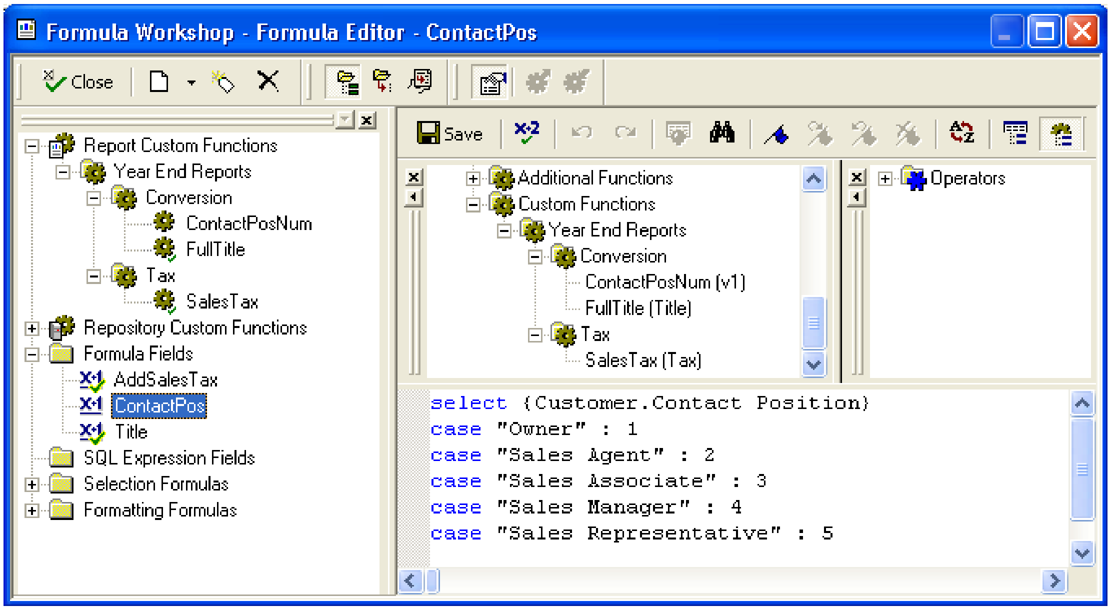
Categorize custom functions so they can be grouped together.

For this example, all three custom functions that you have created will be under the *Year End Reports* category folder. The *FullTitle* and *ContactPosNum* custom functions will be under the *Conversion* category subfolder while the *SalesTax* custom function will be under the *Tax* category subfolder.

1. In the Formula Workshop, double-click **Report Custom Functions**.
2. Click the *FullTitle* custom function. In the right-hand window, enter "Year End Reports/Conversion" in the **Category** box. Click the **Save** button.
3. Click the *ContactPosNum* custom function. In the right-hand window, enter "Year End Reports/Conversion" in the **Category** box. Click the **Save** button.
4. Click the *SalesTax* custom function. In the right-hand window, enter "Year End Reports/Tax" in the **Category** box. Click the **Save** button.

NOTES	• If you don't see the **Category** box in the right-hand window, right-click the custom function, and then click **Toggle Property Display**. • If the **Category** box is not available, the custom function has been added to the Repository. Right-click the custom function, and then click **Disconnect from Repository**. After adding a category, you can add the custom function back to the Repository.

The categories appear as subfolders under **Report Custom Functions** in the Formula Workshop. They also appear as subfolders under **Custom Functions** in the Functions window of the Formula Editor dialog box.



```
select {Customer.Contact Position}
case "Owner" : 1
case "Sales Agent" : 2
case "Sales Associate" : 3
case "Sales Manager" : 4
case "Sales Representative" : 5
```

Figure 4.1

Add Custom Functions to the Repository

With the custom functions created, you can now add them to the repository so they can be used in other reports. This is not a requirement for custom functions, but if you want to reuse them in other reports it is.

1. In the Formula Workshop, double-click **Report Custom Functions**, and then right-click the *FullTitle* custom function.
2. Click **Add to Repository**.
3. Click **OK** to add the custom function to the Crystal Repository.
4. Add the SalesTax and ContactPosNum custom functions to the repository as well.

Now, when you double-click **Repository Custom Functions** you will see the custom functions added to the repository. If the custom functions have been categorized, these categories will also appear under Repository Custom Functions.

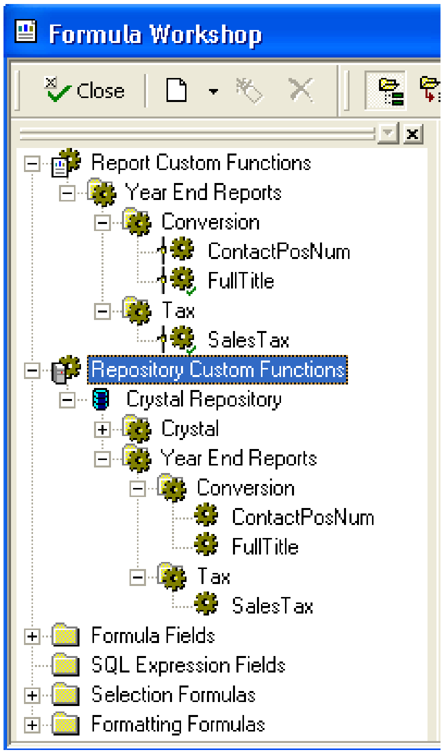


Figure 5.1

Finding More Information

For more information on custom functions, refer to the Crystal Reports Online Help by pressing the F1 key.

Contacting Crystal Decisions for Technical Support

We recommend that you refer to the product documentation and that you visit our Technical Support web site for more resources.

Self-serve Support:

http://support.crystaldecisions.com/

Email Support:

http://support.crystaldecisions.com/support/answers.asp

Telephone Support:

http://www.crystaldecisions.com/contact/support.asp