Running Totals and Advanced Formula

Rishi H. Heerasing
Lecturer

School of Innovative Technologies and Engineering
UTM

Learning Outcomes

- Understanding Running Totals
- Creating Running Totals for a List of Numbers
- Conditional Running Totals
- Working with Variables
- Variable Scope
- Control Structures
- Working with Arrays

Running Totals and Advanced Formula

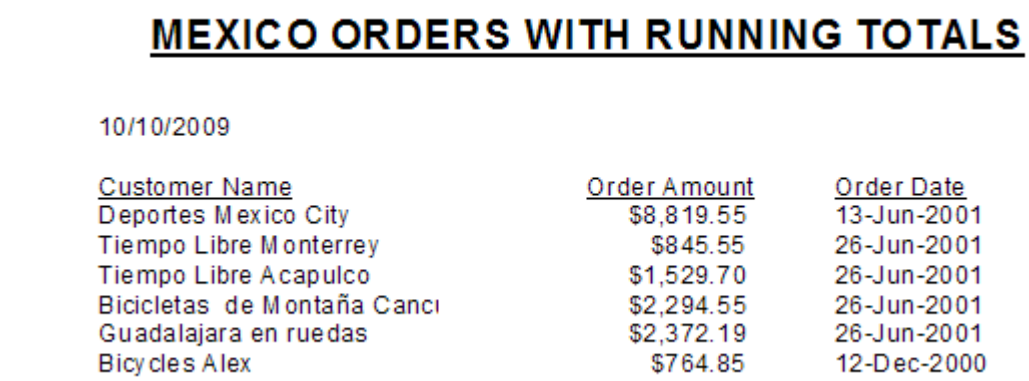
- Reports with summarized group totals that we have seen previously are valuable for many purposes, but they don't satisfy all needs.
- Sometimes it's helpful to see how the status of an item changes with time.
- This is where Running Totals steps-in.
- We will use an existing report to start up with.

Running Totals and Advanced Formula

- From the Start Page, select the Report Wizard and the Xtreme.mdb database.
- Select the Customer and Orders tables and make sure they're linked by the Customer ID field.
- Add the Customer Name field from the Customer table and the Order Amount and Order Date fields from the Orders table.
- For this report, skip Grouping view and move to Record Selection and select only records from Mexico, and then click Finish.

Running Totals and Advanced Formula

- Format the date and add a title to the report.



MEXICO ORDERS WITH RUNNING TOTALS

10/10/2009

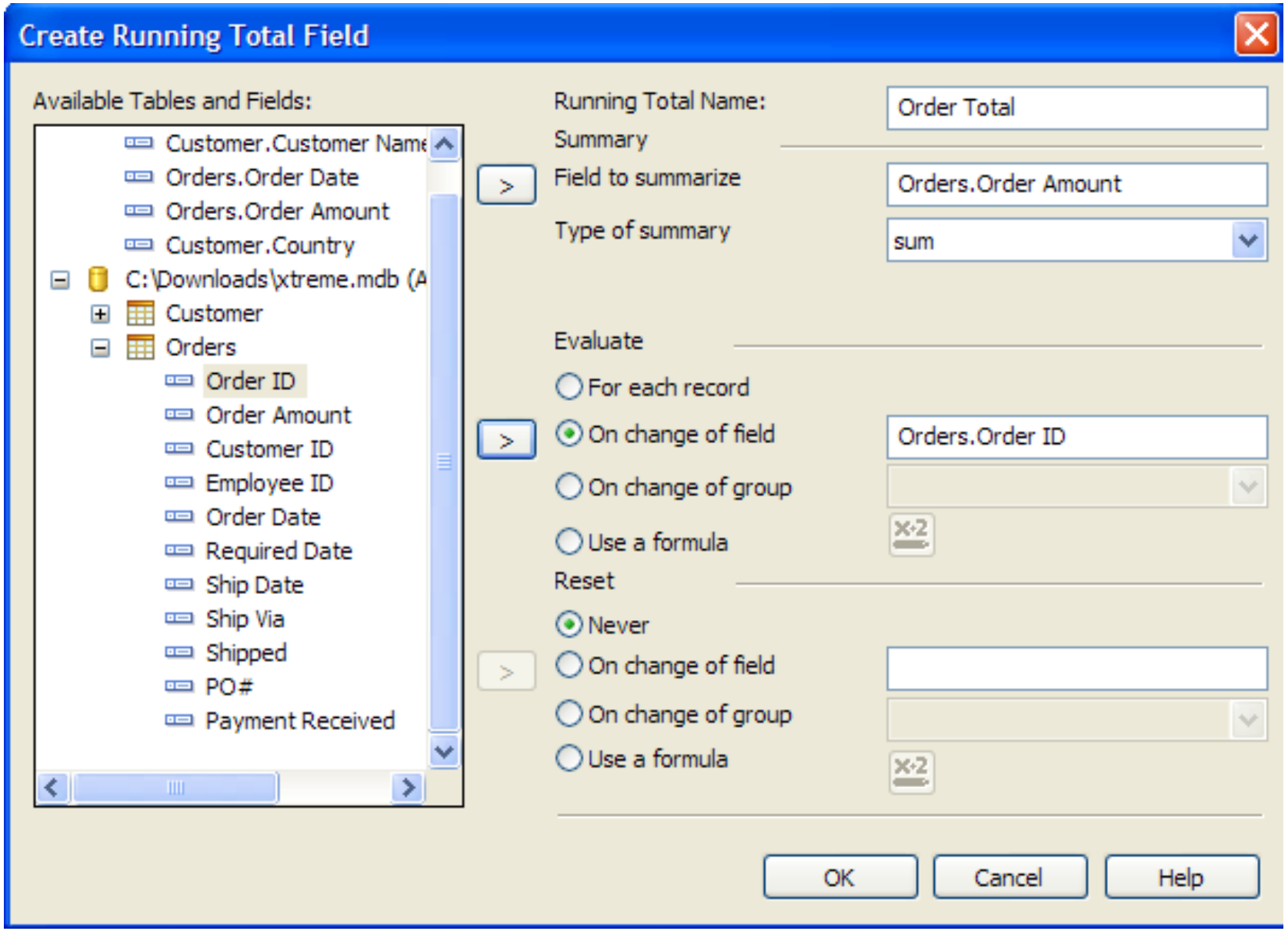
Customer Name	Order Amount	Order Date
Deportes Mexico City	$8,819.55	13-Jun-2001
Tiempo Libre Monterrey	$845.55	26-Jun-2001
Tiempo Libre Acapulco	$1,529.70	26-Jun-2001
Bicicletas de Montaña Canci	$2,294.55	26-Jun-2001
Guadalajara en ruedas	$2,372.19	26-Jun-2001
Bicycles Alex	$764.85	12-Dec-2000

- Now for the Running Total, we will add a fourth column next to Order date.
- We will now add a new Running Total Field. In Field Explorer, Right-click on Running Total Fields and select New.

Running Totals and Advanced Formula

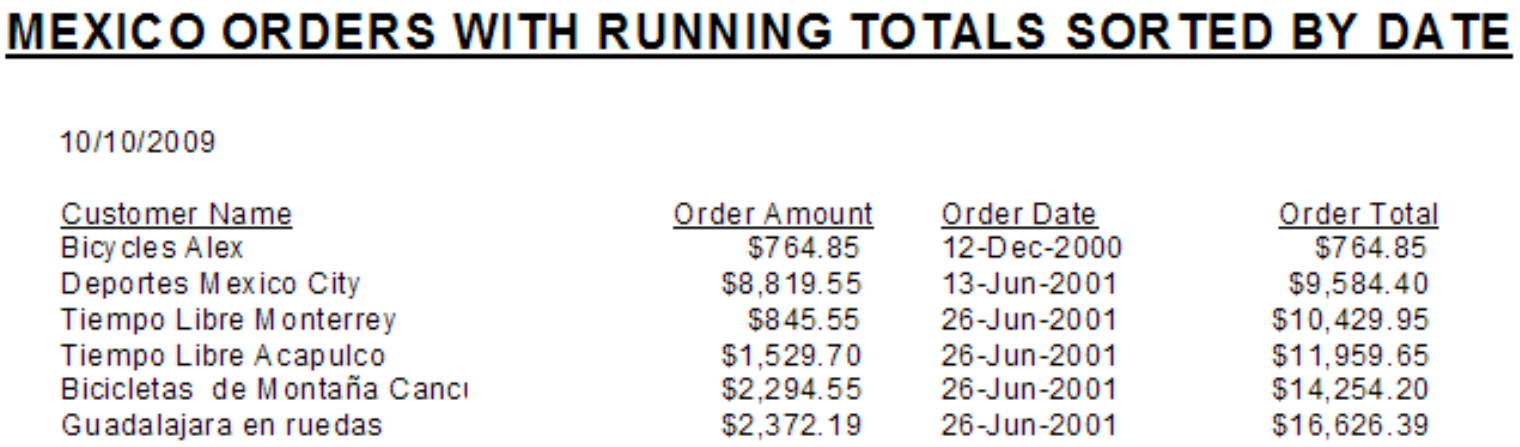
• In the Running Total Name box, replace the default name with Order Total.
• In the Available Tables and Fields pane, select Order.Order Amount from the Report Fields area. Then click the > button pointing to the Field to summarize box. The default *sum* appears.
• In the Available Tables and Fields pane, select Orders.Order ID from the Orders table. In the Evaluate area, click the On Change of Field option, and then click the > button.
•The report will display all the orders for Mexico, so you want the running total to be updated for each order (each time the Order ID Changes). We want the running total to be cumulative for the entire report, leave the Reset option set to Never. Then click OK.
•The Create Running Total Field dialog box disappears, once again showing Field Explorer.
•Drag the Order Total Running Total field from Field Explorer onto your report, just to the right of the Order Date field.

Running Totals and Advanced Formula



Running Totals and Advanced Formula

The report is not sorted in chronological order.
1. Click the Record Sort Expert icon on the Expert Toolbar.
2. Add Orders.Order Date to the Sort Fields pane.
3. Leave the Sort Direction at Ascending, and then click OK.
4. Save the report as Mexico Orders with Running Totals Sorted by Date as 10a.rpt



MEXICO ORDERS WITH RUNNING TOTALS SORTED BY DATE

10/10/2009

Customer Name	Order Amount	Order Date	Order Total
Bicycles Alex	$764.85	12-Dec-2000	$764.85
Deportes Mexico City	$8,819.55	13-Jun-2001	$9,584.40
Tiempo Libre Monterrey	$845.55	26-Jun-2001	$10,429.95
Tiempo Libre Acapulco	$1,529.70	26-Jun-2001	$11,959.65
Bicicletas de Montaña Canci	$2,294.55	26-Jun-2001	$14,254.20
Guadalajara en ruedas	$2,372.19	26-Jun-2001	$16,626.39

Running Totals and Advanced Formula

- We have seen earlier about Record Selection Formulas and how to create Formula Fields.
- We will now create Custom Report Functions. For example we will create a custom function called *concatenatewith1space* that will concatenate the First and Last names of a Contact Person separated by a space.
- We will reuse the previous report and display the Contact first and last names below that of the Customer Name.

Running Totals and Advanced Formula

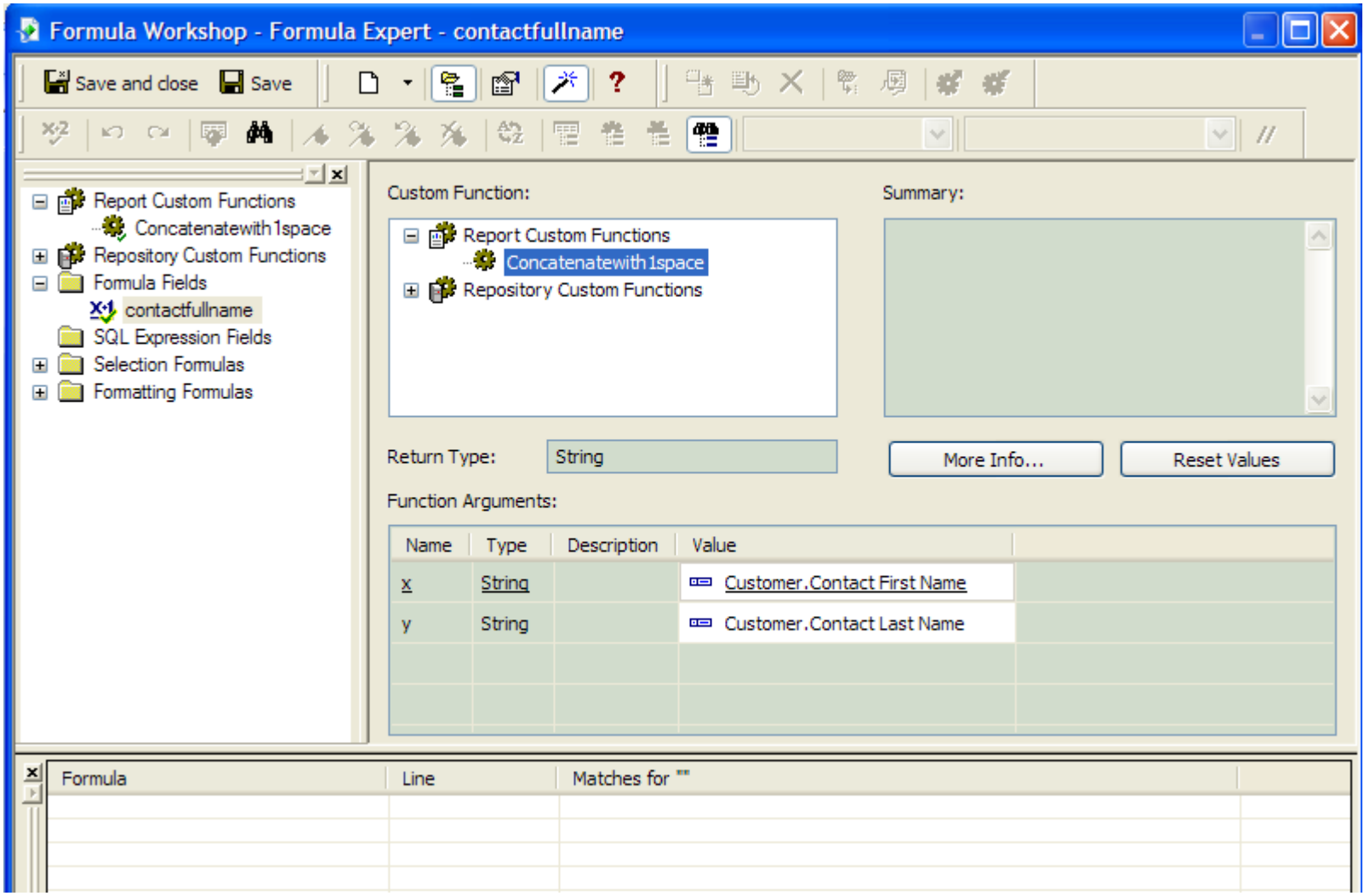
- On the Expert Toolbar, click the Formula Workshop icon to open Formula Workshop.
- Right-click the Report Custom Functions branch on the Workshop tree and choose *New* from the contextual menu that appears.
- The Custom Function Name dialog box appears, asking you to enter a name: Enter *concatenatewith1space*.
- Click the Use Editor button. Either drag the elements you need from the Functions and Operators panes or type your function directly.
- In the pane below the Functions and Operators panes, type the parameter declarations and the body of the function. Note that the word `Function ()` is already there.
- Type in `Function (stringVar x, stringVar y)  (x + ' ' + y);`
- Save and Close.

Running Totals and Advanced Formula

- We can not however use a custom function directly in a report, we must wrap the function in a formula.
- So now create a New Formula field in Formula Workshop itself.
- Create one called *contactfullname*.
- In the Workshop tree on the left, expand the Report Custom Functions node. You will notice that *ConcatenateWith1Space* is listed under it. Note also that *contactfullfame* is listed under Formula Fields even though you haven't added functionality to it yet. The formula exists; it just does not do anything yet.
- Click on the *contactfullname* function and invoke the Formula Expert. This is the icon in the Menu bar with the magic wand on it.
- Click on the *ConcatenateWith1Space* function and on the drop-down value select Other fields... and browse Contact first and last names for x and y respectively.

Running Totals and Advanced Formula

- You should have something similar to this.



Running Totals and Advanced Formula

- Save and Close when done.
- Now return in Design View and drag the *contactfullname* formula field just below the Customer Name. Delete the associated header. You should now have something as below. Save it as 10b.rpt

MEXICO ORDERS WITH RUNNING TOTALS SORTED BY DATE

10/11/2009

Customer Name	Order Amount	Order Date	Order Total
Bicycles Alex **Alex Alvarez**	$764.85	12-Dec-2000	$764.85
Deportes Mexico City **Julio Estes**	$8,819.55	13-Jun-2001	$9,584.40
Guadalajara en ruedas **Laura Torres**	$2,372.19	26-Jun-2001	$11,956.59
Bicicletas de Montaña Canci **Ana Maria Fernandez**	$2,294.55	26-Jun-2001	$14,251.14
Tiempo Libre Acapulco **Manuel Hernandez**	$1,529.70	26-Jun-2001	$15,780.84
Tiempo Libre Monterrey **Jose Garcia**	$845.55	26-Jun-2001	$16,626.39

Running Totals and Advanced Formula

- We will have a look at Formatting Formula.
- Suppose we want to change the Background colour of the Page Header of the previous report to Yellow.
- Open Formula Workshop once again and expand the Formatting Formulas node in the Workshop tree.
- Several subnodes appear, including the Page Header node. Expand the Page Header node.
- Right-click on the Page Header and Select New Formatting Formula and Select Background Colour and Click on Use Editor.

Running Totals and Advanced Formula

- In the formula window, simply type a colour constants, scroll down after the last comment and write on an empty line: crYellow
- Do the same with *contactfullname* in Details and add a Silver background. Close the Formula Workshop and preview your report. It should look similar to this.

MEXICO ORDERS WITH RUNNING TOTALS SORTED BY DATE

10/11/2009

Customer Name	Order Amount	Order Date	Order Total
Bicycles Alex	$764.85	12-Dec-2000	$764.85
Alex Alvarez			
Deportes Mexico City	$8,819.55	13-Jun-2001	$9,584.40
Julio Estes			
Guadalajara en ruedas	$2,372.19	26-Jun-2001	$11,956.59
Laura Torres			
Bicicletas de Montaña Canc	$2,294.55	26-Jun-2001	$14,251.14
Ana Maria Fernandez			
Tiempo Libre Acapulco	$1,529.70	26-Jun-2001	$15,780.84
Manuel Hernandez			
Tiempo Libre Monterrey	$845.55	26-Jun-2001	$16,626.39
Jose Garcia			

Running Totals and Advanced Formula

- Whenever the formula parser encounters a variable, it looks for the value represented by the variable, and then plugs the value into the formula.
- Before you can use a variable, you must declare it, i.e. you must specify three things: its name, its scope, and its data type.
- When you declare a variable's data type, stick `Var` on the end of the type, as in `stringVar` or `numberVar`.
- Scope may be *local*, *global*, or *shared*. A *local* variable is valid only in the formula in which it is declared. A *global* variable is available to all the formulas in a report (except subreports). A *shared variable* is available to all formulas in a report that declare it, including subreports.

Running Totals and Advanced Formula

- After you declare a variable, you can assign it a value. Here's an example:

```
//Declare topic1 as global, String type that specifies a book topic.
Global StringVar topic1;
topic1 := "Crystal Reports";
```

- You can also declare a variable and assign it a value in a single statement, as follows:

```
Local StringVar topic2 := "SQL";
```

Running Totals and Advanced Formula

- Control structures enable you to alter the flow of execution from a strict sequential order to something else.
- For example, you can branch one way or another with an *If-Then-Else* control structure. You can branch multiple ways with a *Select Case* structure. You can loop through an expression or a set of expressions multiple times with a *For* or *While Do* structure.
- Say you want to give a 5% discount to customers who order more than $10,000 of products in a single order.

```
If {Orders.Order Amount} > 10000.

Then {Orders.Order Amount} * 0.95

Else {Orders.Order Amount};
```

Running Totals and Advanced Formula

- Use the *Select Case* control structure when you have more than two alternatives to choose from and you want to do a different thing in each case.
- Suppose you want to give discounts based on the volume of orders:

```
Select {Orders.Order Amount}
   Case 15000. To 1000000.: {Orders.Order Amount} * 0.93
  Case 12000. To 14999.99:  {Orders.Order Amount} * 0.94
  Case 10000. To 11999.99:  {Orders.Order Amount} * 0.95
  Default:
   {Orders.Order Amount};
```

Running Totals and Advanced Formula

- A *For* loop causes execution to pass through a single piece of code multiple times.
- Say you have a text field named *Size* in a table Product, and you want to know how many instances of the letter *x* it contains. You can find out using a *For* loop:

```
Local NumberVar Index; Local NumberVar Xcount := 0;

Local NumberVar StringLength := Length ({Product.Size});

For Index := 1 to StringLength Step 1 Do

   (If ({Product.Size} [Index] = “x”) Then (Xcount := Xcount + 1;)

Else (Xcount := Xcount;)); //specified for clarity

Xcount   //returns the value Xcount
```

Running Totals and Advanced Formula

- Suppose that in the preceding example, you wanted to know the character position of the first x rather than the total number of instances of x in the string. Because you don't know how far into the string the first x occurs (if at all), using a *While Do* loop is appropriate:

```
Local NumberVar Index := 1; Local NumberVar Xpos := 0;

Local NumberVar StringLength := Length ({Product.Size});

Do (If ({Product.Size} [Index] = "x") Then

        (Xpos := Index;) Else (Xpos := Xpos;)

Index := Index + 1;

While Index <= StringLength And Xpos = 0);

Xpos
```

Running Totals and Advanced Formula

- *Arrays* are ordered lists of values that are all the same type. In Crystal Reports, an array can contain data of a simple type or of a range type. Array elements are enclosed in square brackets, as in these examples:

  ["Mercury", "Venus", "Earth", "Mars"]

  [2, 3, 5, 7, 11, 13] [3] specifies 5, the 3rd element in the array.

  ["Mercury", "Venus", "Earth", "Mars"] [3 To 4]

  this creates a new array, ["Earth", "Mars"]

  70 To 100 includes values between, and including 70 and 100.

  70_To_100 includes values between, but excluding 70 and 100.

  70_To 100 includes values between, including 100 but excluding 70.

  UpTo 100 includes all numbers up to and including 100.