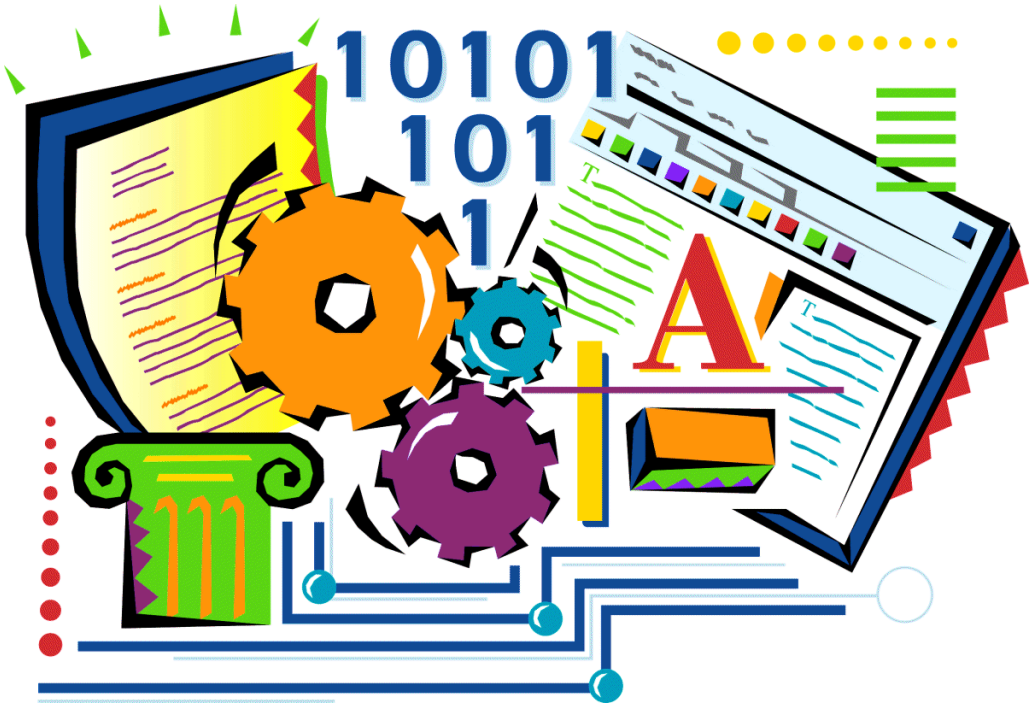
Learning to RAP



The End-User's Guide to Building Reports

INTRODUCTION TO RAP

OVERVIEW

CALISTHENICS FOR RAP-ERS

EXPLORING THE CALC WORKSPACE

RAP TUTORIALS

CONFIGURING REPORT COMPONENTS

Concatenation

Set Font Color and Style

Load Address Lines Into Memo

Learning To RAP

Introduction to RAP

RAP Tutorials

INTRODUCTION TO RAP

OVERVIEW

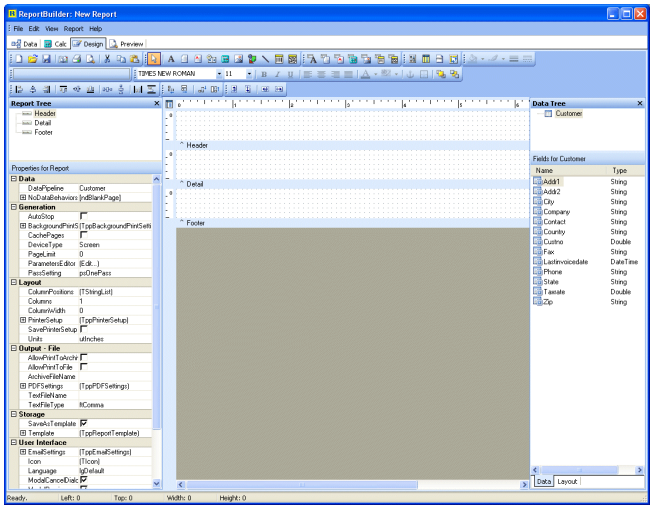
WELCOME

Welcome to Learning to RAP, a series of tutorials designed to teach you how to use the Report Designer's Calc Workspace to perform calculations and control visual aspects of the report while it is generating. For example you might want to calculate the number of stocks in a portfolio whose share price exceeds $50.00. In addition you might decide to conditionally display the share price of each stock in either black or red, depending upon whether its share price exceeds $50.00.

RAP stands for Report Application Pascal and is used to refer to the Report Designer's Calc Workspace. The Calc Workspace provides an event-based scripting environment in which you can implement blocks of code that are executed in response to specific events that occur during the report generation process.

RAP is also used to refer to the scripting language itself. RAP is powerful, yet easy to learn and read. The RAP scripting language is based upon Object Pascal, a modern version of the Pascal programming language that has historically been used to teach introductory programming courses.

The Report Designer, shown below, is the application you will use to code reports. Learning to RAP is written for the end user who is already familiar with ReportBuilder. If you are unfamiliar with the data, design, or preview tabs, then the Learning ReportBuilder guide is where to start.

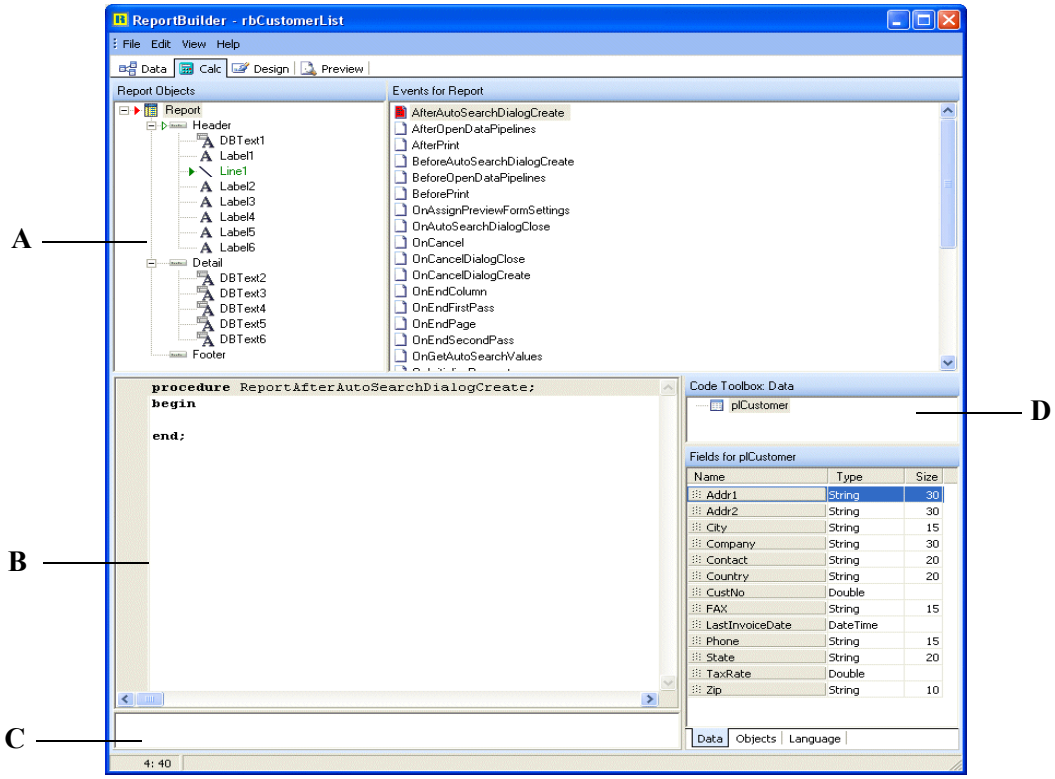


THE CALC WORKSPACE

The Calc workspace provides an event-based scripting environment that can be used to perform calculations and dynamically control visual aspects of the report layout while the report is generating. Scripts that respond to events are called event-handlers.

The Report Designer's Calc Workspace is divided into four main areas as shown in the diagram below. Each of these areas is covered in detail on the following pages.

The Calc Workspace



A Code Explorer: Used to organize and manage a report's scripts

B Code Editor: Used to edit scripts

C The Code Toolbox: Visual code repository, supports drag-and-drop script creation

D Message Window: Dislays messages from the script compiler that checks your code

Code Explorer

The Code Explorer is contained in the upper left and right panes of the Calc workspace.

The left pane contains a tree view - use this to navigate your report's code. The right pane contains a list view - it will display a variety of items depending on what is selected in the tree view. By right-clicking on the tree you can display a context menu that allows you to control the behavior of the Code Explorer.

Views

The Calc tab allows you to manage the report code using three different views. By selecting the View menu or by right-clicking over the white space of the left pane:

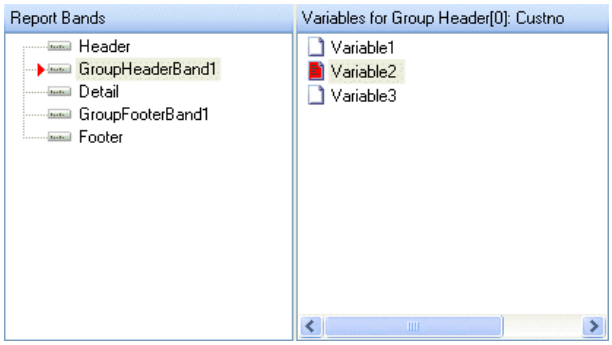
- Variables
- Events
- Module

The series of screen shots below illustrate how the same report looks with each view.

You can select a view by using the View menu or by positioning the mouse over the white space of the left pane and pressing the right mouse button to display a context menu.

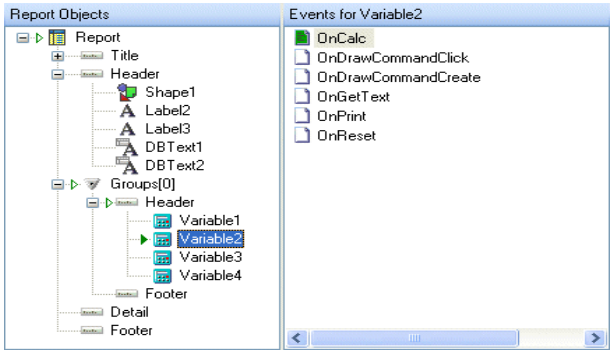
Variables View

This view allows you to see the Variables within a report. Variables allow you to perform calculations. Notice that only the bands are displayed in the left pane. The only component visible from this view is the Variable. All other components in the report are hidden in this view.



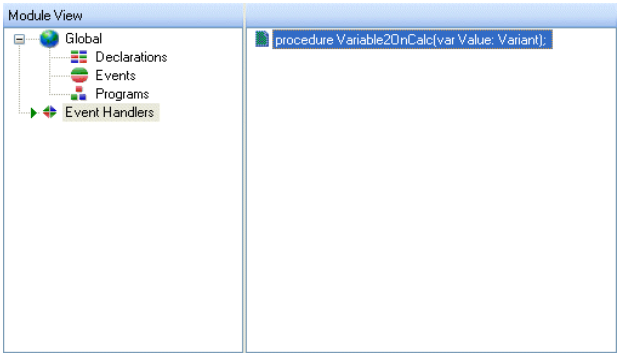
Events View

This view displays a tree view of all components contained within the report. The right pane displays any events associated with the currently selected component. Selecting an event will display the event script, if one exists.



Module View

This view displays a global view of the entire calc module. It lists all declarations, events, programs, and event handlers. This view is useful when you want to examine all of the report's code.



This view displays items, which are visible to all event handlers of the report:

Declarations – These are variables and constants that are globally visible throughout the report.

Events – These are, in essence, the report's events. In the case where the preview window is displayed, OnCreate and OnDestroy fire when the window is opened and closed, respectively. This is different from Report.BeforePrint and AfterPrint in that those methods will fire each time Report.Print is called. OnCreate and OnDestroy are good places for initialization and finalization code such as creating and freeing objects and initializing variables.

Programs – These are procedures and functions that are globally visible throughout the report and can therefore be called from any event handler.

Event Handlers – These are all event handlers that have been implemented in the report.

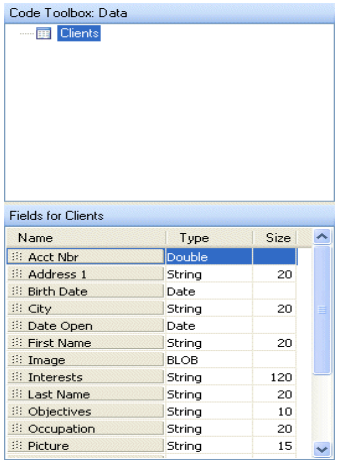
Code Toolbox

The Code Toolbox is a visual code repository. It contains most of the identifiers and code elements that the RAP compiler recognizes.

Notice that the Code Toolbox has three tabs: Data, Objects, and Language. Each tab consists of a tree-view and a list of identifiers.

Data Tab

The Data tab of the Code Toolbox displays data pipelines and fields, allowing you to drag and drop field references into the Code Editor.



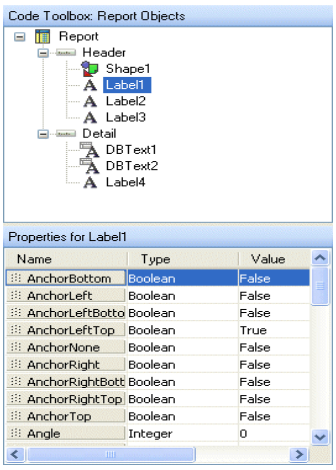
Selecting a pipeline from the list will display all the fields in that pipeline as well as data type and size information for the fields.

To insert a field value into the code editing window, select the field and drag it into the Code Editor. The code necessary to retrieve the field value from the pipeline will be generated. For example, dragging the 'City' field from the Code Toolbox pictured above would result in this code:

```
Value := Clients['City']
```

Objects Tab

The Objects tab of the Code Toolbox displays report objects and their properties, allowing you to drag and drop properties into the Code Editor.



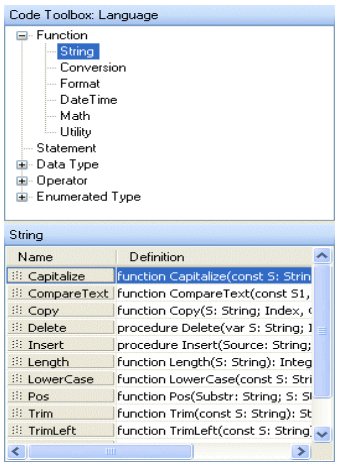
Selecting an object from the tree will display a list of that object's properties.

To insert a property into the Code Editor, select the property and drag it into the Code Editor. The code necessary to retrieve the value of the property or call the method will be generated. For example, dragging the 'AutoSize' property from the Code Toolbox pictured above would result in the following code:

```
Label1.AutoSize
```

Language Tab

The Language tab of the Code Toolbox displays RAP language elements, allowing you to drag and drop elements into the Code Editor.



Selecting a category from the tree will display a list of elements for that category.

To insert an element into the Code Editor, select the element and drag it to the Code Editor. The code necessary to reference or use the element will be generated. Note that when you drop an item such as a function call, the function's parameter list is provided. For instance, if you drag Copy into the Code Editor, it will expand as:

```
Copy(S, Index, Count);
```

Code Editor

Code appears in the Code Editor when you select an event in the list view of the Code Explorer or right-click and select New. You can create code by dragging and dropping code from the Code Toolbox, or you can write your own code. When you want to compile your code, just right-click over the white space of the Code Editor and select 'Compile'. The window below the Code Editor will indicate whether or not there were any errors in your code.

The Code Editor's context menu contains the following items:

New

New has the same effect as clicking in the Code Editor. It is only enabled if there is no implementation for the item currently selected in the Code Explorer.

Compile

Compile activates the RAP compiler to attempt to compile the current procedure and any procedures upon which the current one depends.

Save

The Calc workspace maintains an intermediate buffer for the Code Editor. Selecting Save will commit the current contents of the Code Editor to the buffer; it will not save the entire report. Selecting Save has the same effect as navigating away from, and then returning to the current procedure.

Revert

Use Revert to replace the contents of the Code Editor with what is currently contained in the code buffer. This has the effect of removing all changes since the last save.

Delete

Select Delete to remove the current procedure entirely.

Message Window

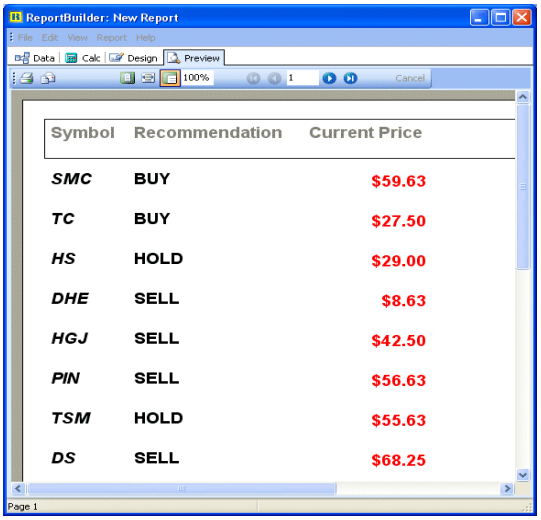
Messages from the compiler are displayed here. You can navigate to the location of the compiler error by double-clicking the error message.

CALISTHENICS FOR RAP-ERS

OVERVIEW

The best way to learn about the Calc tab is by exploring it. So let's build a simple report that will allow us to explore this new workspace.

In this section, you will implement a simple script that assigns the value of the 'Current Price' data field to a Variable component.



GETTING STARTED

1 Launch the Learning ReportBuilder application.

2 Create a new report by clicking the New Report  icon.

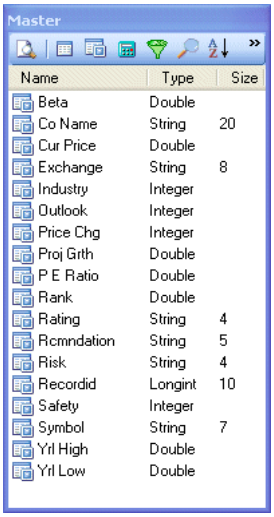
3 In the open Report Designer, click the Data tab.

4 Select File | New from the menu. The New Items dialog will appear.

5 Double-click the Query Wizard to begin creating a dataview. From the list of available tables, double click the Master table.

6 Select Finish.

7 Your dataview has now been created.



8 Return to the Design workspace by clicking the Design tab.

9 Select File | Save As.

10 Double-click the Completed Tutorials folder.

11 Click the New Folder icon and name the folder Rap.

12 Save the report as Quick Test RAP in Completed Tutorials\Rap.

BUILD THE REPORT LAYOUT

Task 1

Layout the Report

1 Select Report | Footer to remove the footer band from the report.

2 Place three labels in the Header band.

3 Assign the following captions to the labels:

Label1	Symbol
Label2	Recommendation
Label3	Current Price

4 Select the labels by shift-clicking, and select the Align Top icon on the Align or Space Toolbar.

5 Set the font of the Labels by shift-clicking:

Font	Arial
Size	14 pt.
Weight	Bold
Color	Gray

6 Right-click and select Position to set the following label positions:

Label 1	Left: 0.0729
Label 2	Left: 1.0208
Label 3	Left: 3.0104

7 Place a Shape in the Header band.

8 Right-click the shape and select ParentHeight and ParentWidth.

9 Now, right-click the shape and select Send to Back

10 Place two DBText components in the Detail band.

11 Connect DBText1 to the 'Symbol' data field .

12 Connect DBText2 to the 'Recommendation' data field.

13 Select the DBText components by shift-clicking, and select the Align Top icon on the Align or Space Toolbar.

14 Shift-click Label1 and DBText1 and select Align Left from the Align or Space Toolbar.

15 Now select Align Left for Label2 and DBText2.

16 Set the font for the DBText components:

Font	Arial
Size	14 pt.
Weight	Bold
Color	Black

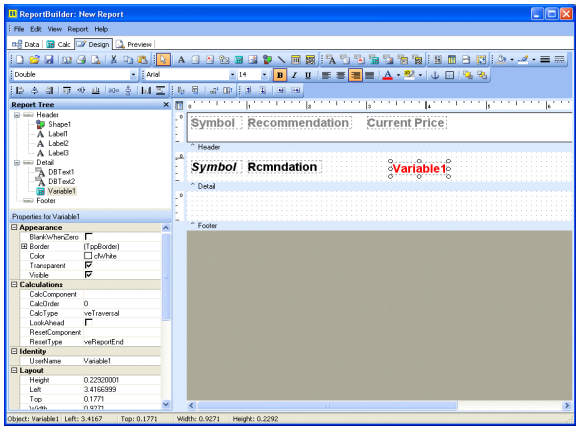
17 Set DBText1 to Italic

18 Right-click the DBText components and select AutoSize.

19 Place a Variable to the right of the DBText components in the Detail band.

20 Set the variable to right-justified.

21 Now select Align Right for Label3 and Variable1.



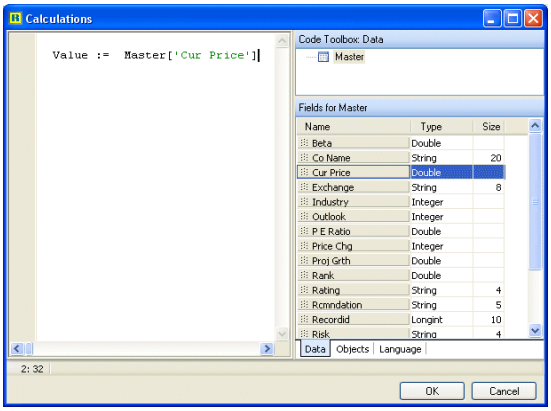
22 Set the font of Variable1 to Red.

23 Set the data type of the Variable to Double in the data type drop-down list.

24 Set Variable1 TextAlignment to right justify.

25 Select Label3, shift-click to select Variable1 and select Align Right from the Align or Space Toolbar.

26 Right-click Variable1 and select Calculations. The Calculations Dialog is displayed.



Note: The Calc Dialog provides a short-cut to implementing a Variable's OnCalc event. The Code Editor appears on the left and the Code Toolbox on the right.

27 In the Data Tab of the Code Toolbox, select the 'Cur_Price' field and drag the selection to the Code Editor. This will result in the following code:

```
Value := Master['Cur_Price'];
```

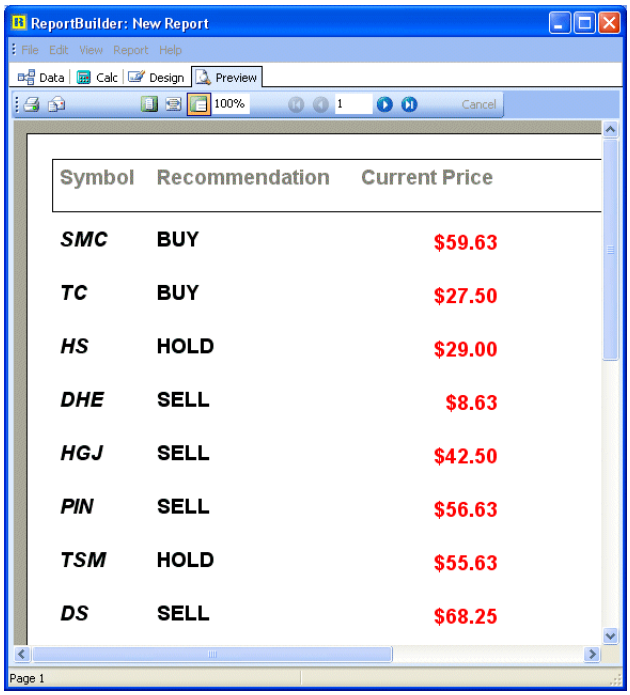
28 Press the right mouse button while positioned over the Code Editor and select Compile from the context menu.

29 Press Ok to close the Calc dialog.

30 Right-click the Variable and select Display Format.

31 Set the display format of the variable to $#,0.00;($#,0.00) so that the variable will be dispayed with a dollar sign and decimal point with two decimal places.

32 Preview the Report.



33 Save the report.

34 Close the report.

Congratulations! You have just completed your first report using RAP.

EXPLORING THE CALC WORKSPACE

VIEWING THE REPORT CODE

Now that we've created a report, let's see what it looks like in the Calc tab.

Task 1

Access the Variables View

1 Launch the Learning ReportBuilder application.

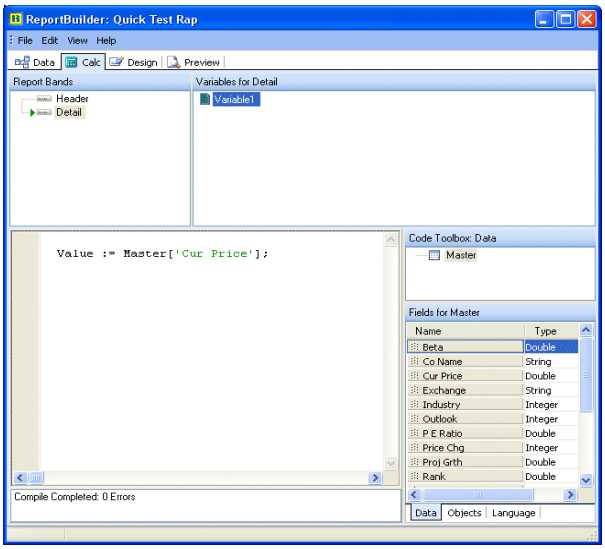
2 Select the Completed Tutorials folder and double-click Quick Test RAP to open the report.

3 Click the Calc tab to access the workspace.

4 Right-click over the white space of the tree view. You have a choice of three views of the report. Select the Variables view.

5 The header band is selected by default. Since it contains no variable components, the list view is empty.

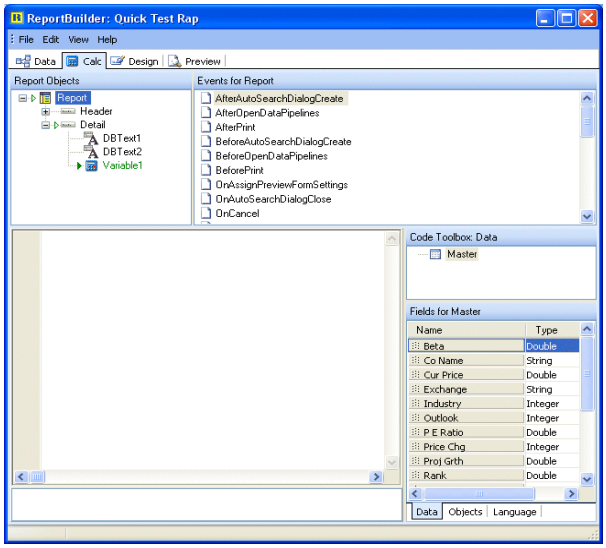
6 Select the detail band. The variable component we created for the detail band becomes visible in the list view.



Task 2

Access the Events View

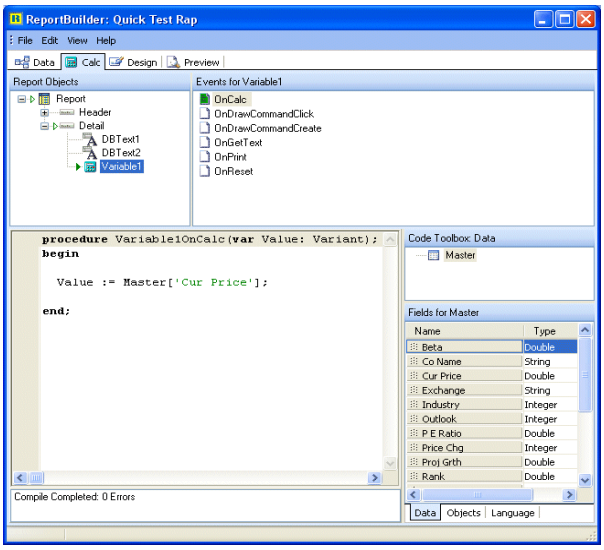
1 Right-click over the white space for the tree view and select Events. A view of the report appears as shown below:



Note: The list view contains events for each component and band.

2 Select the Variable1 component in the Code Explorer's Tree View. The events for Variable1 are displayed.

3 Select the OnCalc event from the Code Explorer's List View. The procedure for Variable1's OnCalc event is displayed.



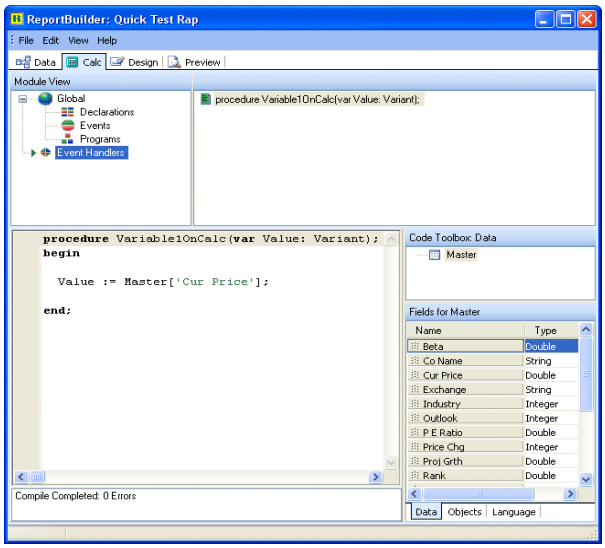
Task 3

Access the Module View

1 Right-click over the white space of the tree view.

2 Select Module. A list of declarations, events, and programs appears below the Global tree node, and Event Handlers appears as another tree node.

3 Select Event Handlers. A procedure for the variable in the detail band appears in the list view and code for the event handler appears in the code editor.



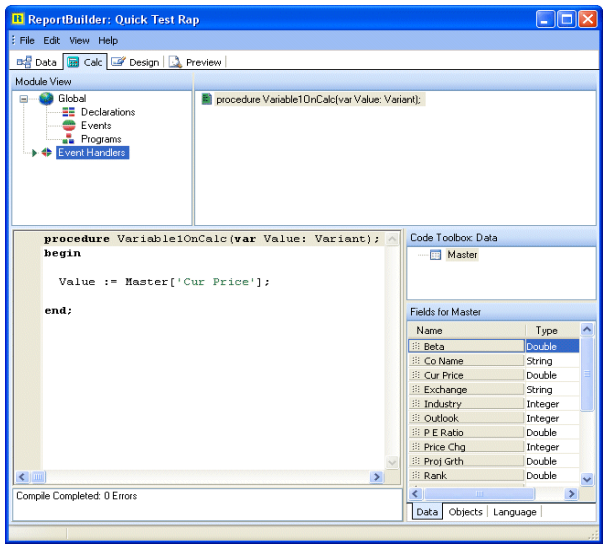
BROWSE THE CODE TOOLBOX

Task 1

Browse the Data Tab

1 Place your cursor over the line to the left of the code toolbox.

2 Drag the line to the left until you can see the 'Size' section of the field discription.



3 Make sure that all of the fields are visible. If they are not, you'll need to resize the designer by placing your cursor over the right corner of the designer. Once it turns into a double-sided arrow, you can increase the size of the designer by dragging down:

Note: The Data tab shows all of the fields in the report. Master is the name of the dataset from which these fields were taken.

TASK 2

Browse the Objects Tab

4 Click the Objects tab of the Code Toolbox.

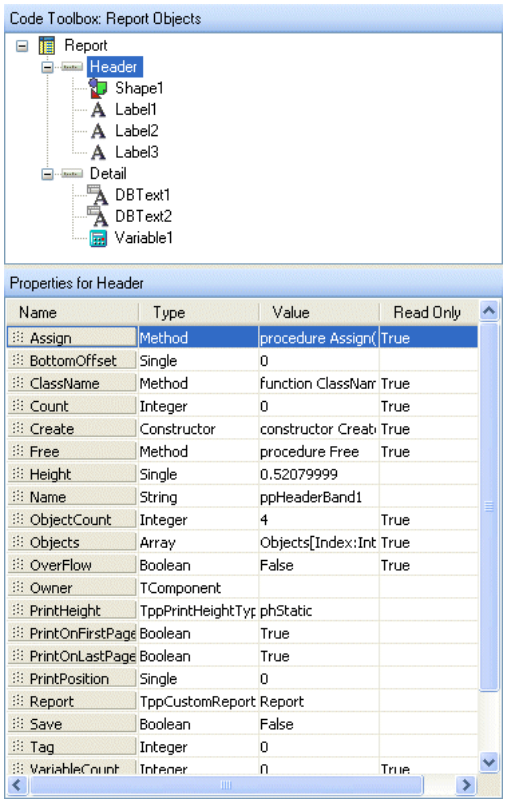
5 Select the line to the left of the Code Toolbox.

6 Drag the line to the left of until you can see the 'Read Only' section of the field description.

Note: The Objects tab displays all of the objects (bands, components) for the report in the top pane and all of the properties for the selected object in the bottom pane.

7 Drag the line that divides the two panes in the Code Toolbox up and down. You can choose how much of one pane you want to see by dragging this line.

8 Select the header in the code toolbox:



A different set of properties appears.

TASK 3

Browse the Language Tab

1 Click the Language tab. This tab contains code that you can drag into the Code Editor and use as part of your code.

LEARN ABOUT TYPES OF EVENTS

1 Select the report in the tree view of the Code Explorer.

Note: The right pane of the Code explorer is currently titled 'Events for Report' because the Report object is selected. This pane shows all of the possible events for the components of the Report object. The left pane is labeled 'Report Objects' when the events view is selected. This pane offers a tree view of all of the objects, or components, in the report.

2 Select the header band. The events for the header band are displayed in the right pane, which is now labeled Events for Header.

3 Now select the detail band. The same set of events appears in the right pane. All bands have the same events.

4 Select a label. Now the right pane reads Events for Label.

5 Select another label. The same events appear. Each component has a specific set of events.

The events for each of the components and bands are codeable, which means that if you inserted code for the OnPrint event of the label, that code would execute when the label printed. Let's try coding such an event right now!

EVENT CREATION

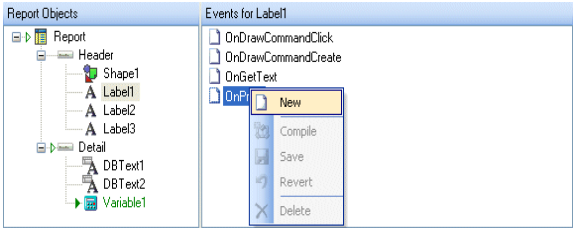
Scenario: You want the font of in the design workspace to be smaller so that you can fit all of the components in the report layout, but when you print, you want the font to be more readable. In order to accomplish this task, we'll create an OnPrint event that will change the font size.

Task 1

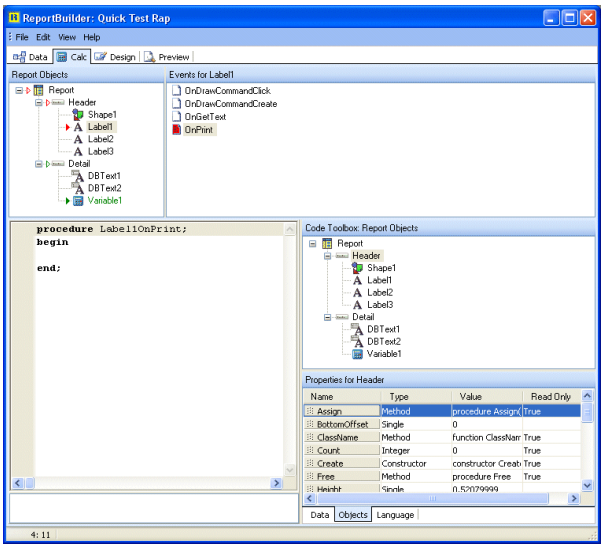
Create an OnPrint Event

1 Select Label1 in the tree view.

2 Right-click over the OnPrint event in the list view and select new:



A procedure will appear with the name of the label (Label1) and the name of the event (OnPrint). A shell, which is a begin and end statement, will also appear:



3 Type the following code beneath the begin statement:

```
procedure Label1OnPrint;
begin
  Label1.Font.Size := 14;
end;
```

You've just written your first line of code. Let's take a moment to break it down.

```
Label1.Font.Size := 14;
```

A B C D E

A Label1 is the name of the label. The '.' After the label is called a *member operator*, or a dot operator. It separates each of the properties that describe the label.

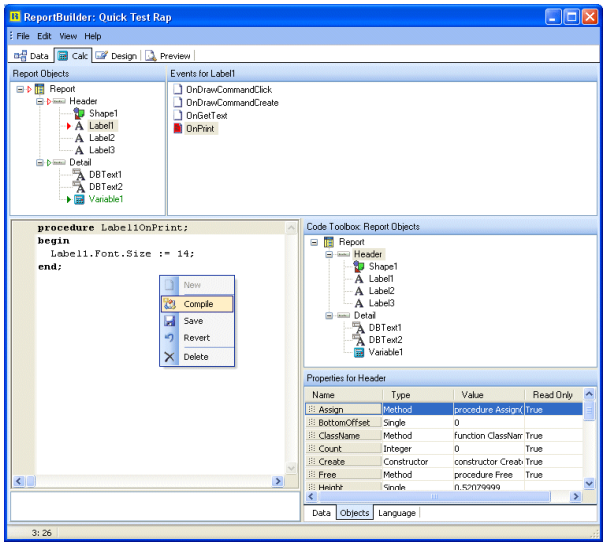
B Font refers to the typeface of the characters. The member operator serves the same function as the one after Label1.

C Size describes how big the font is going to be. In this case, we want it to be 14 pt.

D This is an *assignment operator*. It assigns the value 14 to the size of the font for the label.

E 14 is the value being assigned to the font size. The semicolon signifies the end of the statement.

4 Right-click over the code editor and select compile:



The triangles in the tree view and the paper icon in the list view should turn green:

The window below the code editor should read 'Compile Completed: 0 Errors':

5 Preview. The font for the Label1 component is now size 14 pt.

Task 2

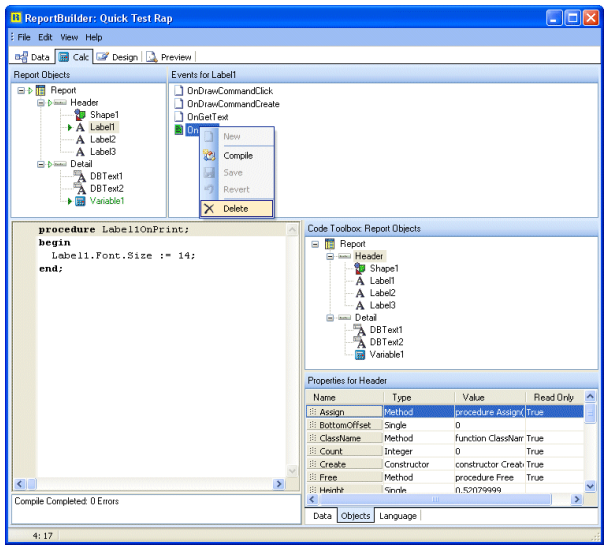
Set Font Size for all Labels

1 Access the Calc workspace

2 Select the code for the OnPrint event.

3 Copy the code into your clipboard (Ctrl + C).

4 Right-click over the OnPrint event in the list view an select Delete:



5 Select the Header band in the tree view.

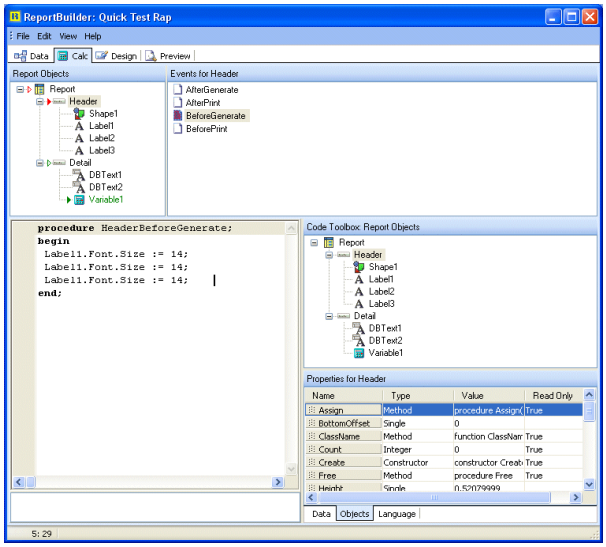
6 Right-click over the BeforeGenerate event and select new.

7 Paste the line of code you cut from the OnPrint event under the begin statement:

```
Label1.Font.Size := 14;
```

Note: We could have used the OnPrint event for Label1 to execute this task; however, because we are going to write code for each label in the header band, it's best if that band contains the code.

8 Press the Enter key and paste the line of code until you have a line for each of the labels. The code editor should look like this:



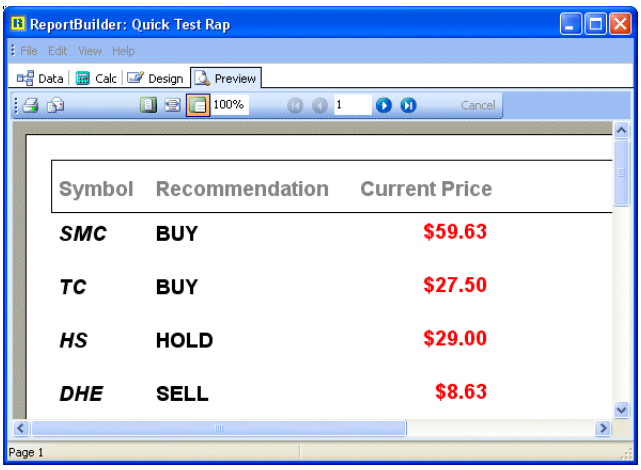
Task 3

Update and Preview

1 Change the code as shown below:

```
Label3.Font.Size := 14;
Label2.Font.Size := 14;
Label1.Font.Size := 14;
```

2 Compile and preview. The report should look like this:



Save the report as Calisthenics in Completed Tutorials\Rap.

RAP TUTORIALS

CONFIGURING REPORT COMPONENTS

CALCULATIONS

GLOBAL DECLARATIONS, OBJECTS, & PROGRAMS

CONFIGURING REPORT COMPONENTS

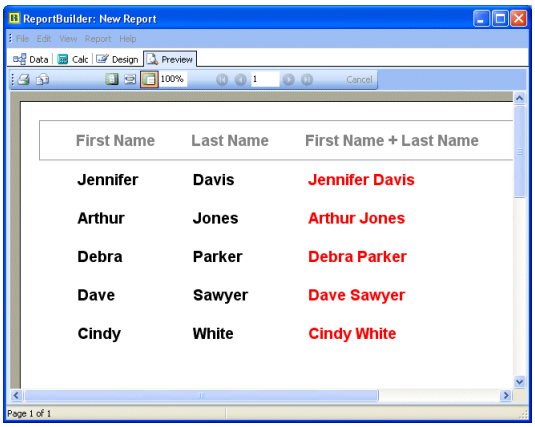
CONFIGURING REPORT COMPONENTS

CONCATENATION

OVERVIEW

This tutorial will show you how to do the following:

- Create a basic report at design-time
- Create a report that contains a RAP event handler attached to the BeforeGenerate event of the DetailBand. The event handler concatenates the first name and last name of a contact and assigns the result to the caption of a label.



Task 1

Create a New Application

1 Launch the Learning ReportBuilder application.

2 Create a new report by clicking the New Report  icon.

3 In the opened Report Designer, click the Data tab.

4 Select File | New from the menu. The New Items dialog will appear.

5 Double-click the Query Wizard to begin creating a dataview. From the list of available tables, double click the Clients table.

6 Select the Finish button.

BUILD THE REPORT

Task 2

Layout the Report

1 Click the Design tab and turn off the report's footer band by selecting Report | Footer from the Report menu.

2 Add three labels to the Header band.

3 Assign the following captions to the labels:

Label1	First Name
Label2	Last Name
Label3	First Name + Last Name

4 Select the labels by shift-clicking, and then use the format toolbar to set the following properties:

Font	Arial
Size	14
Weight	Bold
Color	Gray

5 Right-click and select Position to set the following label positions:

Label 1	Left: 0.5
Label 2	Left: 2.0
Label 3	Left: 3.5

6 Select the labels by shift-clicking, and select the Align Top icon on the Align or Space Toolbar.

7 Place a Shape in the Header band.

8 Right-click the shape and select ParentHeight and ParentWidth.

9 Now, right-click the shape and select Send to Back.

10 Add two DBText controls to the Detail band.

11 Set the font color of the DBText components to Black and the font to Arial 14 Bold.

12 Click on DBText1 and use the Data Field drop down list to select First Name.

layout check

13 Click on DBText2 and use the Data Field drop down list to select Last Name. If you preview the report now, you should see five detail records with first and last names displayed.

14 Place one Label component to the right of the DBText components in the detail band. The Label will display the concatenated fields.

15 Set the font color of the Label to Red and the font to Arial 14 Bold.

layout check

16 Select all of the components in the Detail band by shift-clicking, and select the Align Top icon on the Align or Space Toolbar.

17 Shift-click Label1 and DBText1.

18 Select Align Left from the Align or Space Toolbar. Do the same for Label2 & DBText 2, and Label3 & Label4.

Task 3

Navigate the Calc Workspace

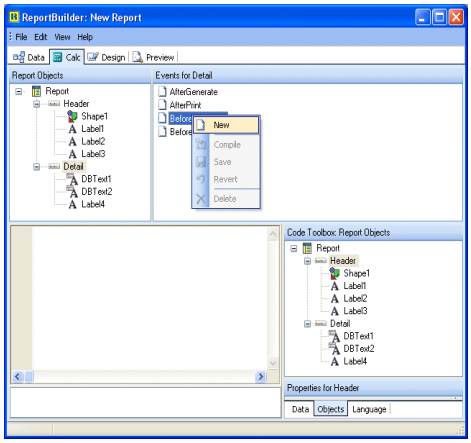
We will use the BeforeGenerate Event of the Detail band to concatenate the two fields.

1 Click the Calc tab to display the Calc Workspace.

2 Right-click the Code Explorer's tree view and select Events.

3 Click the Detail Band.

4 Right-click the BeforeGenerate event and select New.



ADD THE CODE

Task 4

Add the Concatenation Code

The Data tab of the Code Toolbox should be active, if it is not, select it. In the upper window of the Code Toolbox, you should see the pipeline we have added, Clients. Below that, you should see an entry for each field in the pipeline. These items are draggable.

We're going to enter the following line of code, but we are going to construct it via drag and drop. See the DetailBeforeGenerate Event code on the next page.

1 Place your cursor after the Begin line in the Code Editor and enter the following code:

```
Label4.Caption :=
```

2 Click on the First Name entry in the Toolbox and drag it to the Code Editor, just to the right of the "Label4.Caption := " line.

2 Drag the Last Name entry to the right of the line of code.

3 Type in the remaining characters of the line as shown below.

Code DetailBeforeGenerate Event

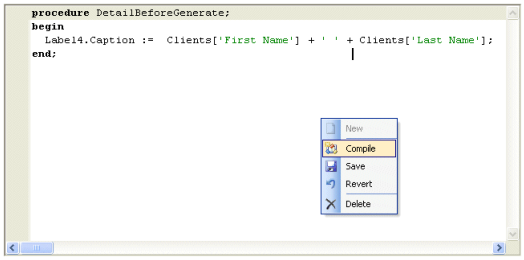
```
procedure DetailBeforeGenerate;
begin

  Label4.Caption := Clients['FIRST_NAME'] + ' ' + Clients['LAST_NAME'];
end;
```

PREVIEW AND FINISH

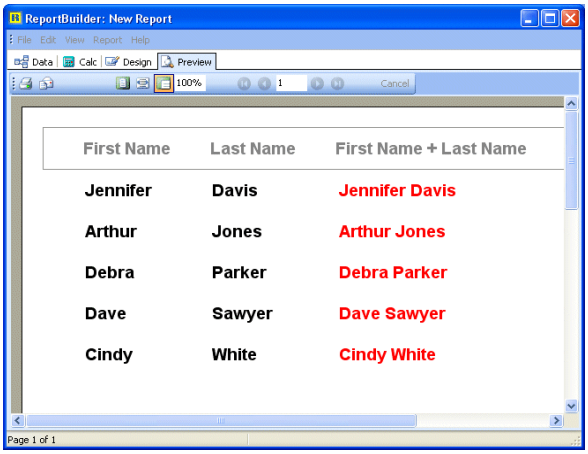
Task 5

Compile and Preview

1 To compile your code, right-click in white space of the Code Editor and select Compile.



2 To view the results, click the Preview tab.



Congratulations--you've successfully concatenated database fields using RAP!

Task 6

Save the Report

1 Click on the Design tab and select Save As from the File menu.

2 Navigate to the Completed Tutorials/Rap folder and save the report as Concatenation

Note that the RAP Code, the DataView, and the report layout have all been saved together.

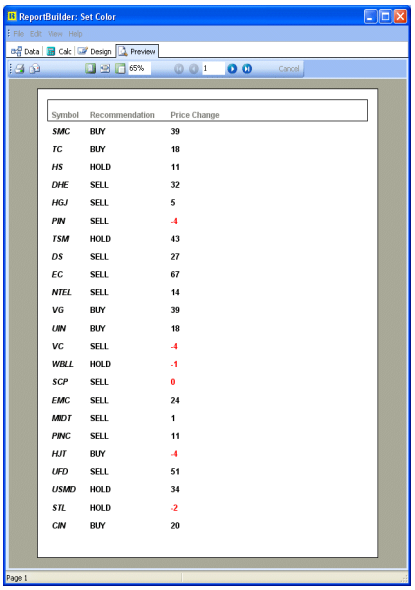
3 Select File | Close to close the report.

SET FONT COLOR AND STYLE

OVERVIEW

This report contains a RAP event handler attached to the BeforeGenerate event of the DetailBand.

The event handler color codes the font of a DBText component assigned to the Price Change field. When the Price Change is negative, the value is shown in red.



BUILD THE REPORT

Task 1

Create a new report

1 Launch the Learning ReportBuilder application.

2 Create a new report by clicking the New Report  icon.

3 In the open Report Designer, click the Data tab.

4 Select File | New from the menu. The New Items dialog will appear.

5 Double-click the Query Wizard to begin creating a dataview. From the list of available tables, double click the Master table.

6 Select Finish.

7 Return to the Report Designer by clicking the Design tab.

8 Select File | Save As and save the report as Set Color in the Completed Tutorials\Rap folder.

Task 2

Layout the report

1 Select Report | Footer to remove the footer from the report layout.

2 Right-click the Detail band and set the position:

Height	0.42

3 Place three labels in the Header band.

4 Assign the following captions to the labels:

Label1	Symbol
Label2	Recommendation
Label3	Price Change

5 Select the labels by shift-clicking and set the font for the labels:

Font	Arial
Size	14 pt.
Weight	Bold
Color	Gray

6 Place a Shape in the Header band.

7 Right-click and select Send to Back.

8 Right-click the shape and select ParentHeight.

9 Right-click and select Position to set the following label positions:

Label 1	Left: 0.1146
Label 2	Left: 1.0833
Label 3	Left: 3.0

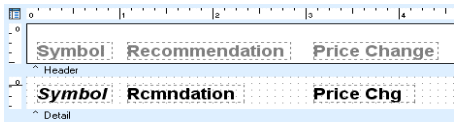
10 Select the labels by shift-clicking, and select the Align Top icon on the Align or Space Toolbar.

11 Place three DBText components in the Detail band.

12 Set the value of the DBText components:

DBText1	Symbol
DBText2	Rcmndation
DBText3	Price_Chg

layout check



13 Set the font color to Black.

14 Select Label1 and then shift-click to select the corresponding DBText.

15 Press the Align Left on the Align or Space Toolbar. Repeat for Label2 and DBText2.

16 Select Label3 and shift-click to select DBText3.

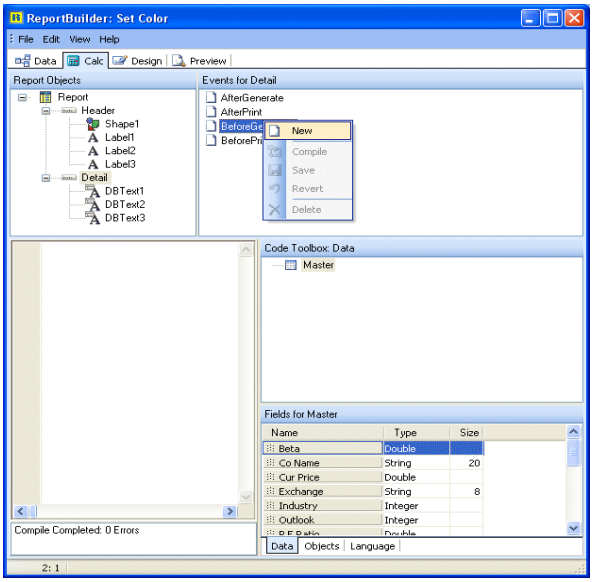
17 Press the Align Right on the Align or Space Toolbar.

18 Set the TextAlignment for DBText3 to Aligned Right.

19 Access the Calc Workspace by clicking on the Calc Tab.

20 Select the BeforeGenerate event of the Detail Band.

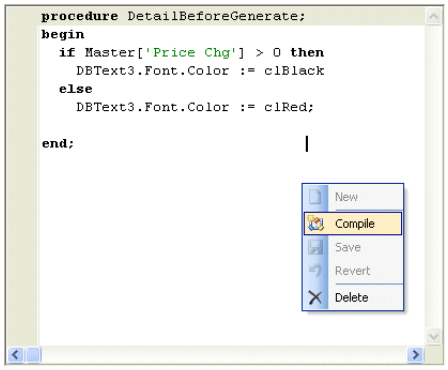
21 Right-click BeforeGenerate and select New.



22 Enter the following code for the event handler:

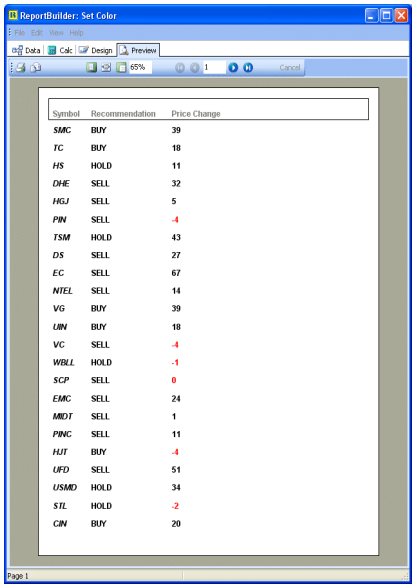
```
procedure DetailBeforeGenerate;
begin
  if Master['PRICE_CHG'] > 0 then
    DBText3.Font.Color := clBlack
  else
    DBText3.Font.Color := clRed;
end;
```

23 Right-click and select Compile.



24 Save the report.

25 Preview the report. Your report should look like this:



26 Close the report.

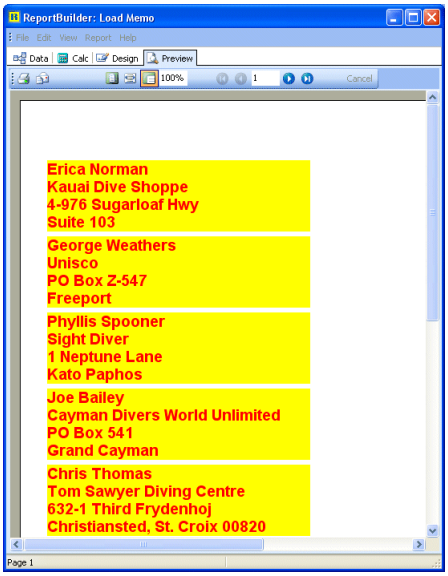
Congratulations! You now know how to set the font color and style of your reports using RAP.

LOAD ADDRESS LINES INTO MEMO

OVERVIEW

This tutorial will show you how to create a report that contains an event-handler in the OnPrint event of a Memo object.

The event-handler loads address data from the datapipeline into the Lines of the memo. Formatting is performed to suppress address lines and add commas where needed.



Task 1

Create a new report

1 Launch the Learning ReportBuilder application.

2 Create a new report by clicking the New Report  icon.

3 In the open Report Designer, click the Data tab.

4 Select File | New from the menu. The New Items dialog will appear.

5 Double-click the Query Wizard to begin creating a dataview. From the list of available tables, double click the Customer table.

6 Select Finish.

7 Return to the Report Designer by clicking the Design tab.

8 Select File | Save As and save the report as Load Memo in Completed Tutorials\Rap.

BUILD THE REPORT
Task 2

Layout the report

1 Place a Memo in the Detail band.

2 Set the Position of the Memo by right-clicking the component:

Left 0.1042

Top 0.0147

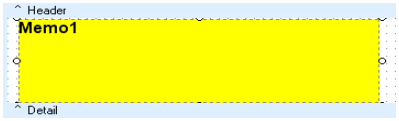
Width 3.4167

Height 0.9271

3 Drag the Detail band to the bottom of the Memo.

4 Set the Highlight Color of the memo to Yellow.

layout check



5 Set the Font properties of the memo:

Font Arial

Size 14 pt.

Weight Bold

Color Red

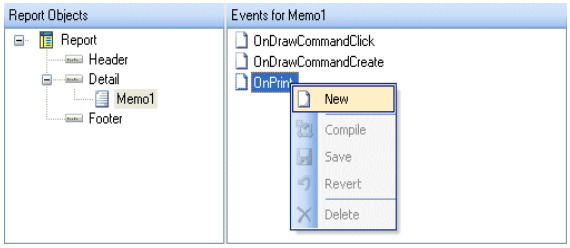
ADD THE CODE
Task 2

Enter the code for the event handler

1 Access the Calc Workspace by clicking on the Calc Tab.

2 Select Memo1.

3 Right-click OnPrint and select New.

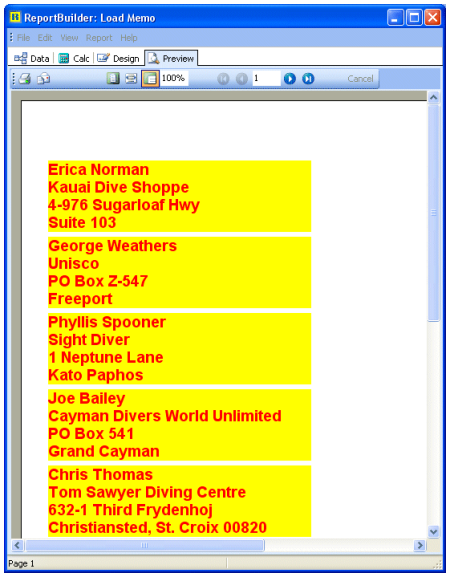


4 Enter the following code for the event handler:

```
var
  lsLine: String;
  lsState: String;
  lsZIP: String;
begin
  {clear memo}
  Memo1.Lines.Clear;

  {add contact}
  lsLine := Customer['Contact'];

  Memo1.Lines.Add(lsLine);

  {add company}
  lsLine := Customer['Company'];

  Memo1.Lines.Add(lsLine);

  {add address line1}
  lsLine := Customer['Addr1'];

  if lsLine <> '' then
    Memo1.Lines.Add(lsLine);

  {add address line2}
  lsLine := Customer['Addr2'];

  if lsLine <> '' then
    Memo1.Lines.Add(lsLine);

  {add city, state zip}
  lsLine := Customer['City'];
  lsState := Customer['State'];

  if lsState <> '' then
    lsLine := lsLine + ', ' + lsState;

  lsZIP := Customer['ZIP'];

  if lsZIP <> '' then
    lsLine := lsLine + ' ' + lsZIP;

  Memo1.Lines.Add(lsLine);

  {add country}
  lsLine := Customer['Country'];

  Memo1.Lines.Add(lsLine);

end;
```

5 Right-click and select Compile.

6 Preview the report. Your report should look like this:



7 Save the report.

8 Close the report.

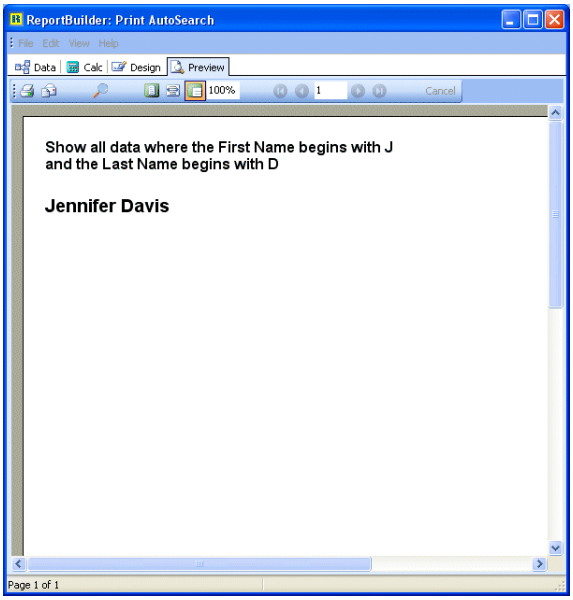
Congratulations! You have created a report that uses RAP to load address lines into a Memo.

PRINT DESCRIPTION OF AUTOSEARCH CRITERIA

OVERVIEW

This tutorial will walk you through the following

- Gain access to the AutoSearch field descriptions via RAP
- Print a description of the search values specified for the report.



Task 1

Create a New Report

1 Launch the Learning ReportBuilder application.

2 Create a new report by clicking the New Report  icon.

3 In the open Report Designer, click the Data tab.

4 Select File | New.

5 Select Query Designer from the New Items dialog and click OK.

6 On the Tables tab, double-click Clients.

7 On the Fields tab, check the All Fields checkbox.

Next we will create search criteria...

Task 2

Create Search Criteria

1 On the Search tab, double-click First Name to add a criteria.

2 For the new criteria, set Operator to "Like" and set Value to "J."

3 Check the AutoSearch checkbox.

4 Double-click Last Name to add a criteria.

5 For this criteria, set Operator to "Like" and set Value to "D."

6 Check the AutoSearch checkbox.

7 Click the OK button in the Query Designer. You should now see the DataView.

8 Click the Preview button on the DataView. You should see one record displayed: Jennifer Davis.

BUILD THE REPORT

Task 3

Layout the Report

1 Click the Design tab.

2 Select File | Save As and save the report as Print AutoSearch in the Completed Tutorials\Rap.

3 Turn off the Footer band by selecting Footer from the Report menu.

4 Drop a Variable  in the Detail band.

5 Set the properties of Variable1:

Font	Arial
Size	14 pt.
Weight	Bold
Color	Black

6 Set the position of Variable1:

Left	0
Top	0.25
Width	0.8229
Height	0.2292

7 Drop a Memo  in the Header band.

8 Right-click the memo and set the position:

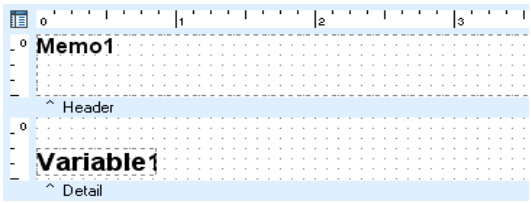
Left	0
Top	0
Width	4.0208
Height	0.5104

9 Set the font:

Font	Arial
Size	12 pt
Weight	Bold
Color	Black

10 Right-click and set the Memo to Stretch – this will hold the AutoSearch field descriptions.

layout check



Task 4

Write the Code

1 Click the Calc tab.

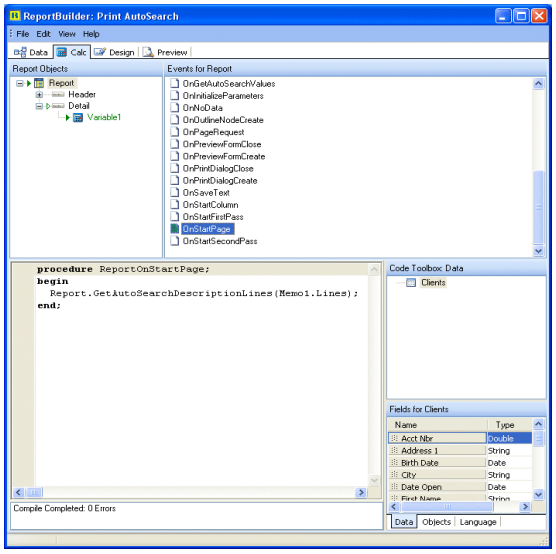
2 Right-click the Code Explorer's treeview and select Events.

3 Click the Report node.

4 Right-click the OnStartPage event in the list-view and select New.

5 In the Code Editor, enter the following code (note that you can either type this in or drag and drop the code from the Code Toolbox):

```
Report.GetAutoSearchDescription-
Lines(Memo1.Lines);
```



6 Right-click the OnCalc event for Variable1 and select New.

7 Enter the following code into the Code Editor:

```
procedure Variable1OnCalc (var Value:
Variant);
begin
  Value :=  Clients['First Name'] + ' ' +
Clients['Last Name'];
end;
```

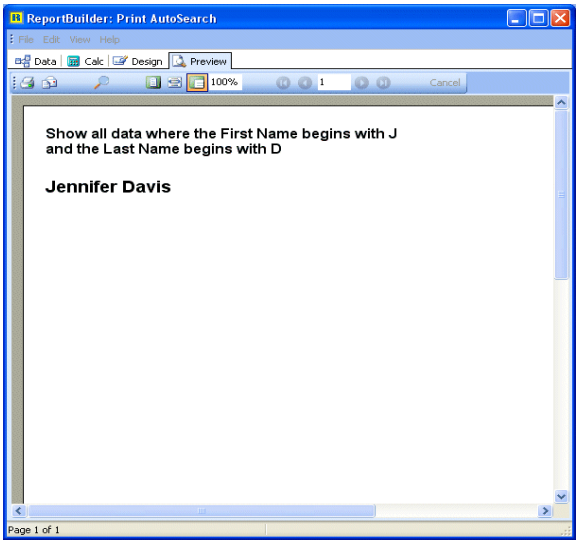
PREVIEW AND FINISH

Task 5

Compile and Preview

1 To compile your code, right-click on the Code Editor and select Compile.

2 To view the results, click the Preview tab.



3 Save the report as Print AutoSearch in Completed Tutorials\Rap.

4 Close the report.

Congratulations! You should see a description of the AutoSearch fields in the Memo.

CALCULATIONS

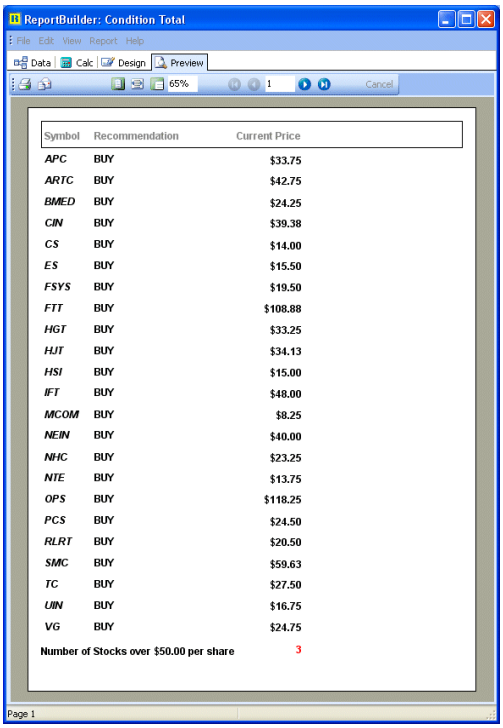
CALCULATIONS

CONDITIONAL GROUP TOTALS

OVERVIEW

This tutorial will walk you through the following

- Create an OnCalc event handler using RAP to count all of the Current Price values over $50.00.



BUILD THE REPORT

Task 1

Layout the Report

1 Launch Learning ReportBuilder.

2 Open the 'Quick Test RAP' report we created in the Calisthenics for RAP-ers section.

3 Save the report as Conditional Total in Completed Tutorials\Rap.

Task 2

Prepare the Data for Grouping

1 Click the Data tab to access the Data workspace.

2 Click the Sort icon on the DataView tool window.

3 Double-click Rcmndation and Symbol.

4 Select Ok to exit the dialog.

5 Return to the Design workspace.

6 Right-click the Detail band and set the position:

Height 0.4

7 Select Report | Groups from the Report Designer menu.

8 Select Master.RCMNDATION from the drop-down list and press the Add button.

9 Click OK.

10 Place a Label component in the Group Footer(0): RCMNDATION.

11 Set the position of Label4:

Left	0
Top	0.1458

12 Configure the label:

Font	Arial
Size	14
Weight	Bold

13 Set the Caption of the label to 'Number of Stocks over $50.00 per share:'

14 Move Variable1 down to the Group Footer(0): RCMNDATION to the right of Label you just created.

layout check

15 Set the text color of Variable1 to Red.

16 Set the data type to Integer.

17 Right-click and clear the Display Format.

18 Add a DBText component to the Detail Band and connect to Curr_Price field.

19 Right-click and set the Display Format: $#,0.00;($#,0.00).

20 Set the font of DBText3:

Font	Arial
Size	14
Color	Black
Weight	Bold

21 Shift-click Label3, DBText 3, Variable1 and select Align Right from the Align or Space Toolbar.

22 Set DBText3 TextAlignment to Right Justify.

23 Set DBText3 Display Format to $#,0.00;($#,0.00).

ADD THE CODE

Task 2

Enter the code for the event handler

1 Right-click Variable1 and select Calculations.

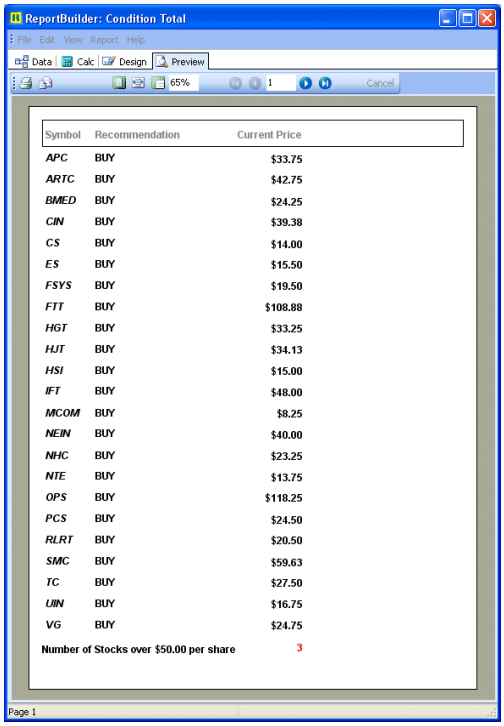
2 Change the code to the following:

```
if (Master['Cur_Price'] > 50) then
Value := Value + 1;
```

3 Right-click and select Compile.

4 Click OK to exit the Calculations Dialog.

5 Click Preview to view the report. Your report should appear as follows:



Symbol	Recommendation	Current Price
APC	BUY	$33.75
ARTC	BUY	$42.75
BMED	BUY	$24.25
CIN	BUY	$39.38
CS	BUY	$14.00
ES	BUY	$15.50
FSYS	BUY	$19.50
FTT	BUY	$108.88
HGT	BUY	$33.25
HJT	BUY	$34.13
HSI	BUY	$15.00
IFT	BUY	$48.00
MCOM	BUY	$8.25
NEIN	BUY	$40.00
NHC	BUY	$23.25
NTE	BUY	$13.75
OPS	BUY	$118.25
PCS	BUY	$24.50
RLRT	BUY	$20.50
SMC	BUY	$59.63
TC	BUY	$27.50
UIN	BUY	$16.75
VG	BUY	$24.75
Number of Stocks over $50.00 per share		3

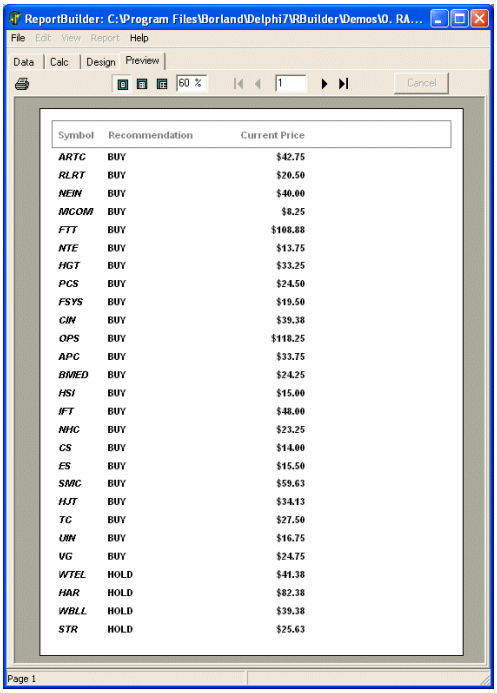
6 Save and close the report.

Congratulations! You have created a report that uses RAP to calculate a conditional group total.

CONDITIONAL GRAND TOTALS

OVERVIEW

In this tutorial, you will use RAP to create an OnCalc event handler to count all of the Current Price values over $50.00.



Symbol	Recommendation	Current Price
ARTC	BUY	$42.75
RLRT	BUY	$20.50
NEIN	BUY	$40.00
MCOM	BUY	$8.25
FTT	BUY	$108.88
NTE	BUY	$13.75
HGT	BUY	$33.25
PCS	BUY	$24.50
FSYS	BUY	$19.50
CIN	BUY	$39.38
OPS	BUY	$118.25
APC	BUY	$33.75
BMED	BUY	$24.25
HSI	BUY	$15.00
IFT	BUY	$48.00
NHC	BUY	$23.25
CS	BUY	$14.00
ES	BUY	$15.50
SMC	BUY	$59.63
HJT	BUY	$34.13
TC	BUY	$27.50
UIN	BUY	$16.75
VG	BUY	$24.75
WTEL	HOLD	$41.38
HAR	HOLD	$82.38
WBLL	HOLD	$39.38
STR	HOLD	$25.63

BUILD THE REPORT

Task 1

Layout the Report

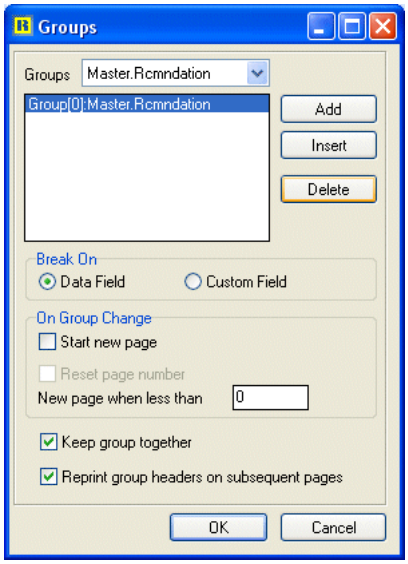
1 Open Conditional Total.

2 Select File | Save As and save the report as Conditional Grand Total in Completed Tutorials\Rap.

3 Select Report | Summary to create a Summary band for the report.

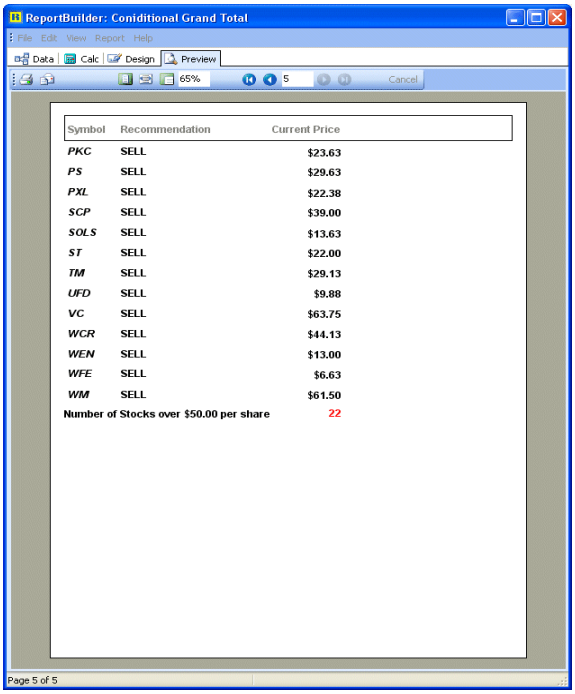
4 Move the Label and Variable from the Group(0): RCMNDATION to the Summary band by shift-clicking the components and dragging.

5 Remove the Report Group by selecting Report | Groups | Delete.



6 Preview the report.

7 The last page of your report should look like this:



8 Save and close the report.

Congratulations! You have created a report that uses RAP to calculate a conditional grand total.

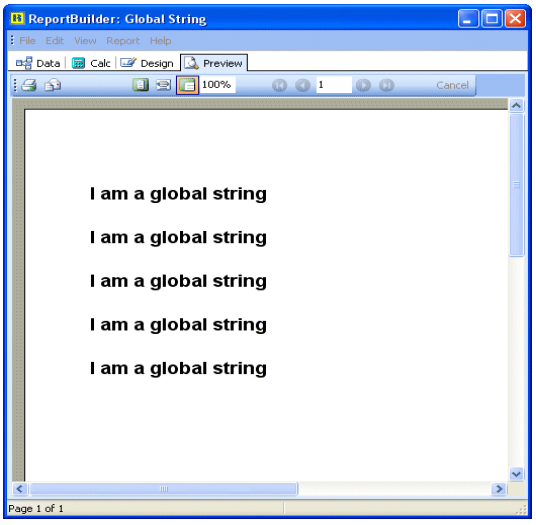
GLOBAL DECLARATIONS, OBJECTS, & PROGRAMS

GLOBAL DECLARATIONS, OBJECTS, & PROGRAMS

GLOBAL STRING

OVERVIEW

In this tutorial, you will learn how to use RAP to print the value of a global string constant.



BUILD THE REPORT

Task 1

Create a new report

1 Launch the Learning ReportBuilder application.

2 Create a new report by clicking the New Report  icon.

3 In the open Report Designer, click the Data tab.

4 Select File | New from the menu. The New Items dialog will appear.

5 Double-click the Query Wizard to begin creating a dataview. From the list of available tables, double click the Clients table.

6 Select Finish.

7 Return to the Design workspace.

8 Select File | Save As and save the report as Global String in Completed Tutorials\Rap.

Task 2

Layout the report & enter the code for the global constant

1 Place a Label  in the Detail band.

2 Select the Calc tab to enter your event handler code.

3 Right-click the Code Explorer and select Module.

4 Click on Declarations and declare a new constant by right-clicking Constants and selecting new.

5 Insert the following code for the constant:

```
const
gcCaption = 'I am a global string';
```

ADD THE CODE

Task 3

Enter the code for the event handler

1 Right-click the Code Explorer and select Events.

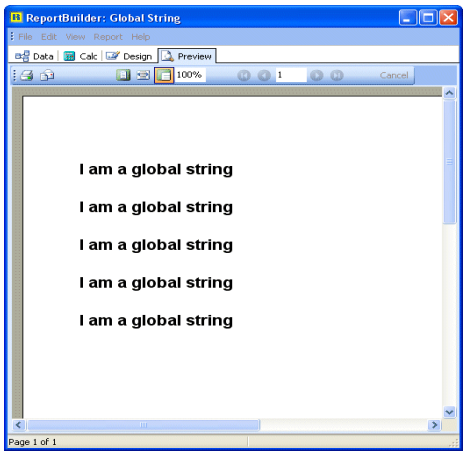
2 Create an OnPrint event handler for Label1 by selecting Label1, right-clicking OnPrint, and selecting New.

3 Insert the following code for the event handler:

```
procedure Label1OnPrint;
begin
  Label1.Caption := gcCaption;
end;
```

4 Right-click and select Compile.

5 Preview the report. Your report should look like this:



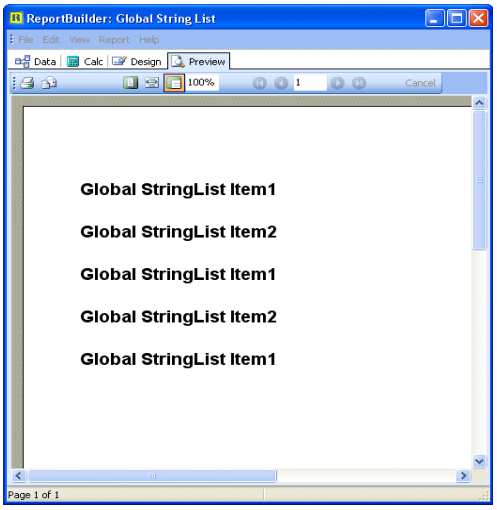
6 Save and close the report.

Congratulations! You now know how to use RAP to print the value of a global string constant.

GLOBAL STRINGLIST

OVERVIEW

In this tutorial, you will learn how to use RAP to create a program which will create, use, and then free a global stringlist.



BUILD THE REPORT

Task 1

Create a new report

1 Launch the Learning ReportBuilder application.

2 Create a new report by clicking the New Report  icon.

3 In the open Report Designer, click the Data tab.

4 Select File | New from the menu. The New Items dialog will appear.

5 Double-click the Query Wizard to begin creating a dataview. From the list of available tables, double click the Clients table.

6 Select Finish.

7 Select the Design tab.

8 Select File | Save As and save the report as Global String List in Completed Tutorials\Rap.

Task 2

Layout the report and enter the code for the global string list

1 Place a Label in the Detail band.

2 Select the Calc tab to enter your event handler code.

3 Right-click the Code Explorer and select Module.

4 Click Declarations, right-click Variables and select New to create a new variable

5 Insert the following code for the variable:

```
var
gStringList: TStringList;
```

6 Click on Events, right-click OnCreate and select New.

7 Insert the following code:

```
procedure GlobalOnCreate;
begin
  gStringList := TStringList.Create;
  gStringList.Add('Global StringList
Item1');
  gStringList.Add('Global StringList
Item2');

  end;
```

8 Now create an OnDestroy GlobalOnCreate event by right-clicking OnDestroy and selecting New.

9 Insert the following code for the OnDestroy event:

```
procedure GlobalOnDestroy;
begin
  gStringList.Free;
end;
```

ADD THE CODE

Task 3

Insert the following code for the Label OnPrint event

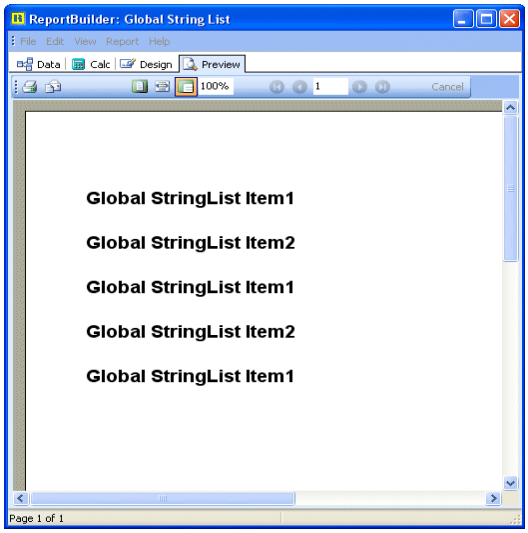
1 Right-click the Code Explorer and select Events.

2 Create an OnPrint event handler for Label1 by right-clicking OnPrint and selecting New.

```
procedure Label1OnPrint;
var
  liItem: Integer;
begin
  liItem := (Detail.Count-1) mod 2;
  Label1.Caption := gStringList[liItem];

end;
```

3 Right-click and select Compile.

4 Preview the report. Your report should look like this:



5 Click the Design tab.

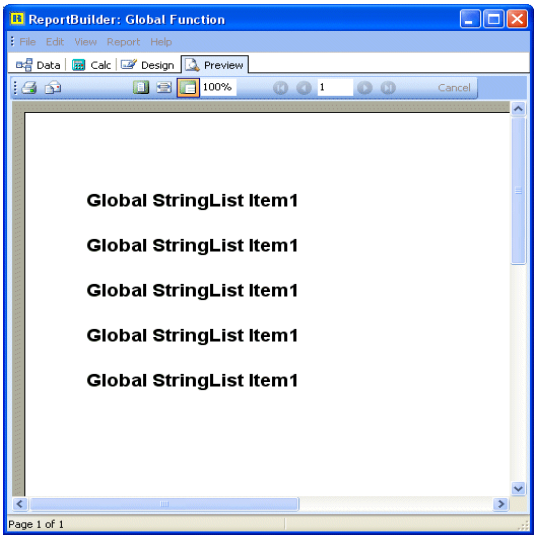
6 Save and close your report.

Congratulations! You now know how to use RAP to print the value of a global stringlist.

GLOBAL FUNCTION

OVERVIEW

This tutorial will show you how to create a program that creates, uses and then frees a global string list. A global function is used to retrieve the values from the global string list.



BUILD THE REPORT

Task 1

Create a new report

1 Launch the Learning ReportBuilder application.

2 Open the Global StringList report created in the previous section.

3 Select File | Save As and save the report as Global Function in Completed Tutorials\Rap.

ADD THE CODE

Task 2

Insert the event handler code

1 Access the Calc workspace.

2 Right-click the Code Explorer and select Module view.

3 Click on Programs, right-click in the Code Editor, and select New Function.

4 Insert the following code:

```
function gfGetString(aIndex: Integer):
String;
Result := gStringList[aIndex];
end;
```

PREVIEW AND FINISH

Task 3

1 Right-click the Code Explorer and select Events.

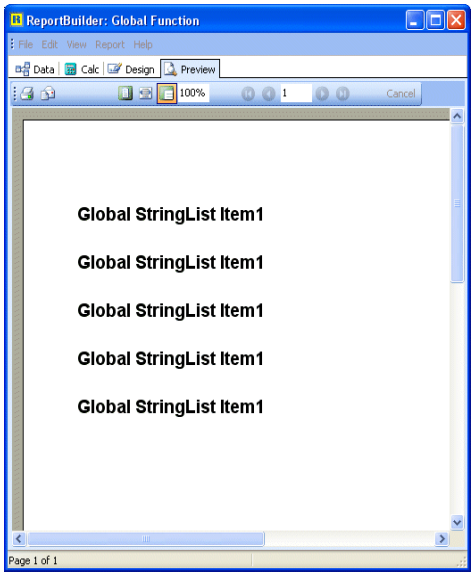
2 Modify the OnPrint event handler for Label1.

3 Insert the following code for the event handler:

```
procedure Label1OnPrint;
begin
  Label1.Caption := gfGetString(0);
end;
```

4 Right-click and select Compile.

5 Preview the report. Your report should look like this:



6 Save and close your report.

Congratulations! You now know how to use RAP to print the value of a global function.

INDEX