CHAPTER 9: WORKING WITH REPORTS AND REPORT FIELDS

Training Objectives

Actively participating during this chapter helps you learn:

- How to add reports and report fields to your Visual Basic for Applications (VBA) Project.
- Common methods and properties used for reports and report fields.
- How to use VBA functions in the project.
- Common events used for reports.

Your use of this content is subject to your current services agreement

Adding Reports and Report Fields to Your Project

To use VBA with a Microsoft Dynamics™ GP report:

- The report must be added to your VBA project.
- Any fields on that window that you may want to reference must be added.

Adding Reports

To add a report object to your project, launch the Microsoft Dynamics GP Report Writer. (TOOLS→CUSTOMIZE→REPORT WRITER)

Multiple dictionaries are installed with Microsoft Dynamics GP. When Report Writer is launched from the menu, you are prompted to select a product, as illustrated below. Select the product to work with and create a new or modified version of the report you want to work with.

From the Toolbar, select the **Reports** button to open the Report Window. From this window, you can create a new report or modify an existing report. Select the report to work with and click **Insert** to add it to the Modified Reports List. Next click **Open** to display the Report Definition window.

Microsoft Official Training Materials for Microsoft Dynamics ™
Your use of this content is subject to your current services agreement

Select Layout to view the report. Once the report layout is open add the report to Visual Basic by clicking Add Report to Visual Basic from the Tools menu.

After the report has been added to your project, you can switch to the Visual Basic Editor and display the Project Explorer. It looks similar to the following illustration:

Removing Reports

To remove a report object from the VBA project, remove all references to the window within the VBA code. Next, switch to Microsoft Dynamics GP Report Writer and display the layout of the report you want to remove. From the Tools menu click Remove Report from Visual Basic. The report and all fields associated with the report will be removed from the VBA project.

Adding Report Fields

There are two types of report fields that can be added to your VBA project:

- **Table** fields – Fields that receive their value directly from a Microsoft Dynamics GP table.
- **Calculated** fields – Fields defined within the Report Writer using table fields, constants, and operators and typically are used to print data on a report only in certain situations.

To add table fields and calculated fields to your VBA project, first add the report that contains the fields. To do this, follow the instructions given. Then, switch to the Microsoft Dynamics GP Report Writer, and make sure the layout of the report is open. Within the report layout, use the arrow tool to select the field to add. From the Tools menu select Add Fields to Visual Basic. To add several fields at one time, hold down the SHIFT key while selecting each field. After all of the fields have been selected, click Add Fields to Visual Basic from the Tools menu.

NOTE: *If the field selected is not a table field or a calculated field, the Add Fields to Visual Basic menu item appears disabled.*

Report fields with a Field Type other than Data (such as: Count, Sum, etc.) have their Field Type and Band abbreviation appended to the field name when added to Visual Basic. For example, if a field appears in the Report Footer (RF) band with a Field Type of Count, it has a field name of fieldnameFREQRF. The following table lists field types and the suffixes appended to field names.

Field Type	Suffix
Average	AVG
Control Count	CNTLCOUNT
Count	FREQ
First Item In Group	FIRST
Last Occurrence	LAST
Maximum	MAX
Minimum	MIN
Previous Occurrence	PREV
Running Sum	RSUM
Sum	SUM

Microsoft Official Training Materials for Microsoft Dynamics ™
Your use of this content is subject to your current services agreement

Using Methods and Properties for Reports and Report Fields

Reports and report fields can be manipulated in different ways through the use of methods and properties. Methods are actions performed on a report object or report field object. Properties are characteristics retrieved or set for a report or a report field. The methods and properties available for reports and report fields are described in this section. In addition, corresponding examples are given. Further information about these methods and properties can be found in the "Appendix" of this manual or in the *Microsoft Dynamics GP VBA Developers Guide*.

Report Methods and Properties

Canceling a Report

A report can be canceled using the Cancel method. Depending on where the method is used, the report can be canceled at different places within its processing. The following example demonstrates how the cancel method can be used to cancel the Customer List report.

```
RMCustomerList.Cancel
```

Working with Legend Fields

A legend field contains data that is passed to a report at runtime, before the report actually starts printing. These fields usually include information such as the range of records used to print the report. The following illustration shows legend fields as they show in the layout of a report:

The Legend property can be used to set or return the value of any legend field that is on a report. The property's index parameter is used to refer to a specific legend field. The following example sets the first legend field on the Customer List report to "All Records:"

```
RMCustomerList.Legend(1) = "All Records"
```

NOTE: *You can only set or return the value of a legend field in the Start event for a report. The Start event is discussed in detail later in this chapter.*

Report Field Methods and Properties

Returning and Setting Report Field Values

Use the Value property to return the value of a **table** field, a Microsoft Dynamics GP **calculated** field, or a **calculated** field that you created. Also use the Value property to set the value of any calculated field you create.

HINT: *Use the Value property to return the value of an* ***Accumulator*** *field. An* ***Accumulator*** *field is a table field or calculated field that has a field type that is not DATA. This includes field types such as COUNT or SUM, which is the number of occurrences of a field or the sum total of each occurrence of a field. In addition, the Value property can also be used to set the value of any* ***Accumulator*** *field that is based on a calculated field you create.*

The following example checks to see if the **Customer Number** field is equal to "ADVANCED0001:"

```
If CustomerNumber.Value = "ADVANCED0001" Then
```

Hiding and Showing Report Fields

The Visible property can be used to hide and show report fields. You can hide any table field or calculated field on a report. However, you can only show fields that you have been hidden using VBA. If a field is invisible, its value is still available through VBA, but it does appear on the report. The following example hides the **Phone Number** field:

```
Phone1.Visible = False
```


Microsoft Official Training Materials for Microsoft Dynamics ™
Your use of this content is subject to your current services agreement

Using VBA Functions in Your Project

There are many existing functions available in VBA. There are two particular functions that are helpful when working with Microsoft Dynamics GP reports. These functions can be used to format fields or can be used when exporting data to other applications.

Formatting Fields

Excluding composite fields, VBA returns report field values without any formatting. For example a **Phone Number** field is returned as 70128105505555 instead of (701) 281-0550 Ext. 5555, and an **Amount** field is returned as 15.50000 instead of $15.50. For the **Report** field value to appear with the proper format, a VBA function must be used. This function is the Format() function and can be used to define a format string for a specific field. In the following example, the **Phone Number** field is changed to include some formatting characters.

```
Format(Phone1, "(@@@) @@@-@@@@ Ext @@@@")
```

Exporting Data to Other Applications

Reports give users the ability to take information out of the Microsoft Dynamics GP accounting system and export it to other applications so that it may be analyzed in different ways. With VBA, you have the ability to export multiple records from Microsoft Dynamics GP to other VBA-compliant applications such as Microsoft® Excel, Microsoft® Word, or Microsoft®Outlook®. You can use VBA to look at each record in a report as the report is printing and then export specific field values to another program.

NOTE: *Before working with another application's objects, set a reference to that application's object library using the References window in the VB Editor. The References window can be opened by clicking References from the Tools menu. For information about the window, refer to the overview chapter of this workbook or the chapter called "Distributing your VBA Project."*

The Object Browser (VIEW→OBJECT BROWSER) is an excellent method of viewing an application's objects and methods, or "Classes" and "Members." If you do not see the application library listed, a reference has not yet been established.

The CreateObject() function gives the ability to create and return a reference to another VBA-compliant application. This establishes a type of connection to an object within that program. After that is complete, additional VBA code can be written to return information about the records on your report to the other application. The following example, uses the CreateObject() function to create and return a reference to Microsoft Outlook. A more complete example of exporting data to another application is provided later in this chapter:

```
Set DynOlApp = CreateObject("Outlook.Application")
```

NOTE: *This function is normally used within a report's Start event so it can be executed as soon as the report begins to generate. More information about the Start event is provided later in this chapter.*

Microsoft Official Training Materials for Microsoft Dynamics ™
Your use of this content is subject to your current services agreement

Using Report Events

Report Event Overview

Unlike fields on a window, report fields do not have field events associated with them. Therefore, you cannot write VBA code that executes for a specific report field. Instead you must reference the field through the code written as part of a report event. A report event corresponds to processes or events that take place when a report is generated. These processes include things such as the start or end of a report or the printing of the different sections on the report. Because of this, report event procedures use names such as "Start," "End," "BeforePH," and "BeforeBody."

To view the available events for a report, use the Code window within the Visual Basic Editor. After the report is added to the Microsoft Dynamics GP project, select it from the first drop-down list. Click on the second drop-down list to view the events available for the report. An example of the Code window is shown in the following illustration listing the events for the RM Salesperson Report:

There are two kinds of events you work with when using Report events:

- Start/end events
- Band events

Start/End Events

Start Event

The Start event occurs just before the report starts to print. It executes before any data actually prints. The primary use of the Start event is to set report legends. Also use this event to initialize any variables you might want to use for the report.

The following example declares a local integer variable called Count and uses the Start event to initialize its value. This variable can then be used to print only the first ten customers on the Customer List report. The remaining code, used to implement the restriction, is shown later during the discussion of band events.

```
Microsoft_Dynamics_GP - RMCustomerList (Code)
Report                                   Start

Private Sub Report_Start()
    'Initialize the count
    Count = 0
End Sub
```

End Event

The End event occurs after a report prints. This event can be used to launch other applications, to open and close other windows, or to perform clean-up activities for the report.

In the following example, the Sales Setup Reports window is closed after the RM Customer Report has printed:

```
Microsoft_Dynamics_GP - RMCustomerList (Code)
Report                                   End

Private Sub Report_End()
    SalesSetupReports.Close
End Sub
```

Band Events

A report is made up of several different sections called bands. In VBA, each band has a corresponding band event. As a report prints, each band event occurs just before the corresponding data within the band is printed. The following illustration shows the different bands that compose a report:

Microsoft Official Training Materials for Microsoft Dynamics ™
Your use of this content is subject to your current services agreement

The primary use of band events is to return or set the value of a field in a corresponding band. This is done using the Value property. Use a band event to return a value for any table field or calculated field on the report. You can set a value only for calculated fields added using the Report Writer. To manipulate the value of an existing report field, make the original field invisible, and "mask" it with a new calculated field.

Each band event also includes a parameter called SuppressBand. When set to True, it stops the current instance of the corresponding band from printing. If a band prints multiple times in a report, you can conditionally suppress a given instance of the band using this parameter. For example, you could suppress all Customers that are from a given state. Sample code for this is shown later as a BeforeBody procedure.

NOTE: *If a report uses an accumulator field to calculate report totals or to count the number of bodies in a report, that field is still sum or count data from fields that were suppressed using the SuppressBand parameter.*

BeforePH Event

The BeforePH event occurs before items in the page header are printed. Items in the page header are placed at the top of every report page. **Page number**, **date** and **time** fields are commonly placed in the header.

The following example uses an integer variable called Count that was declared and initialized in the previous example for the Start event. The BeforePH event uses this variable so that it can count the number of pages that have printed for the report. After printing the first two pages of the report, the event cancels the report. The example code shows the BeforePH event.

Microsoft_Dynamics_GP - RMCustomerList (Code)

Report | BeforePH

```
Option Explicit
Private Count As Integer

Private Sub Report_BeforePH(SuppressBand As Boolean)
   'Increment the count for the number of pages
   Count = Count + 1
   If Count = 3 Then
      RMCustomerList.Cancel
   End If
End Sub
```

BeforeRH Event

The BeforeRH event occurs before items in the report header are printed. Items in the report header appear on the first page of a report. The title of the report and introductory information are often included in this section. If a page header is also included on the first page, the report header appears after the page header.

BeforeAH Event

The BeforeAH event occurs before the report's additional header is printed. A report may have no, one, or several additional headers, each indicated on the report layout by H1, H2, etc. Each prints when the data changes for the field that the header is based on. If a report uses multiple additional headers, the Level parameter can be used in the BeforeAH event specifying an index that corresponds to the number of the additional header the event occurs for.

HINT: *If the report only has one additional header, the BeforeAH event only occurs for that one section without having to specify a value for the Level parameter.*

On the RM Customer Report, a majority of the information about each customer is found in an additional header. The following example uses the **State** field included in that header to determine if the record should be suppressed from the report. Only the customers from "ND" and "MN" show with complete information. Do this by using the BeforeAH event that contains the **State** field. The SuppressBand parameter discussed previously is used to suppress the records that do not meet your criteria.

NOTE: *Report Header bands fire in sequence from first to last; report Footer bands fire in sequence from last to first.*

BeforeBody Event

The BeforeBody event occurs before each instance of the report's body prints. Microsoft Dynamics GP uses the report body for the bulk of a report and is typically made up of table fields. A report prints the body repeatedly until all records on the report are printed.

The following example uses an object created previously in a Start Event to return information from the Microsoft Dynamics GP RM Salesperson Report to Microsoft Outlook. Each salesperson on the report is added as a contact in Microsoft Outlook. Only the salesperson's name and address information is exported. Since that information is printed in the body of the report, use the BeforeBody event to transfer the data to Outlook. The example shows the declaration of the objects, the Start event where CreateObject() function is used to create the object, and the BeforeBody event where the processing is performed.

Microsoft Official Training Materials for Microsoft Dynamics ™
Your use of this content is subject to your current services agreement

NOTE: *Before working with another application's objects, set a reference to that application's object library using the References window in the VB Editor. The References window can be opened by clicking References from the Tools menu. For information about the window refer to the chapter in the workbook called Distributing your VBA Project.*

```
Option Explicit
'Declaration of variables
Dim GPItem, GreatPlainsApp As Object

Private Sub Report_BeforeBody(SuppressBand As Boolean)
   'Add salesperson information to Microsoft Outlook
   Set GPItem = GreatPlainsApp.CreateItem(olContactItem)

   'Set the properties for the contact object to
   'the field values on the report
   GPItem.FullName = SalespersonName
   GPItem.CompanyName = SalesTerritory
   GPItem.BusinessAddressStreet = Address1
   GPItem.BusinessAddressCity = City
   GPItem.BusinessAddressState = State
   GPItem.BusinessAddressPostalCode = Zip

   'Save the contact within Outlook
   GPItem.Save
End Sub

Private Sub Report_Start()
   'Create the object within Outlook
   Set GreatPlainsApp = CreateObject("Outlook.Application")
End Sub
```

BeforeAF Event

The BeforeAF event occurs before the report's additional footer prints. A report may have no, one, or several additional footers, each indicated on the report layout by F1, F2, etc. Each prints when the data changes for the field that the report is based on. Microsoft Dynamics GP uses additional footers to display summary data such as a total of all records in the report's body. If a report uses multiple additional footers, the Level parameter can be used in the BeforeAF event to specify an index that corresponds to the number of the additional footer.

HINT: *If the report only has one additional footer, the BeforeAF event only occurs for that one section without having to specify a value for the Level parameter.*

In the following example, the BeforeAF event is used to print a blank line on the Short Invoice Form if no line item comments are to be printed. For the example to be possible, it is necessary to add a calculated field to the report. This calculated field should be set to a blank string. Place it inside of the additional footer that contains the **Comment[1]** field and its Field Type should be set to data. Also, change the section options for that additional footer so it is not suppressed if one of its calculated fields is empty. After the report is modified, the following code places a blank line in the additional footer if there is no line item comment to be printed for that section:

```
Private Sub Report_BeforeAF(ByVal Level As Integer, SuppressBand _
As Boolean)
   If Level = 5 Then
      If Comment1LASTF5 = "" Then
         BlankComment = "_________________________"
      End If
   End If
End Sub
```

BeforeRF Event

The BeforeRF event occurs before the report footer is printed. Items in the report footer appear only on the last page of a report. Summary information is often included in the report footer. If a page footer is also included on the last page, the report footer prints before the page footer.

In the following example, the BeforeRF event on the Short Invoice Form is used to hide the Trade Discount prompt when the Trade Discount Amount is $0.00. Because the Visible property can only be used to hide table fields or calculated fields and not text prompts, it is also necessary to modify the report using the Report Writer before executing this event. Replace the prompt on the report with a calculated field set to the string "Trade Discount." After the calculated field is added to the report, its Field Type should be set to Data.

```
Private Sub Report_BeforeRF(SuppressBand As Boolean)
   If TradeDiscountAmountLASTRF = 0 Then
      TradeDiscountAmountLASTRF.Visible = False
      TradeDiscountPrompt.Visible = False
   Else
      TradeDiscountAmountLASTRF.Visible = True
      TradeDiscountPrompt.Visible = True
   End If
End Sub
```


Microsoft Official Training Materials for Microsoft Dynamics ™
Your use of this content is subject to your current services agreement

BeforePF Event

The BeforePF event occurs before the report's page footer is printed. Items in the page footer are placed at the bottom of every report page. The page footer often includes administrative information such as the name of the person running the report.

Returning Field Values from Multiple Band Events

Because a field value is only available as the band that contains it is printing, you might need to use module-type variables to store the report information so it can be used by other bands.

The following example exports data from the Microsoft Dynamics GP Short Invoice Form to a text file. Multiple module-type variables are used to store the data to export as each band is printed. This is so all fields can be exported together at one time from one band event. The fields exported in this example include the **Customer PO Number** and **Customer Number** fields from the RH section of the report and the **Document Amount** field from the RF section of the report.

```
Option Explicit
Private CustNum, CustPONum, TtlAmt As String

Private Sub Report_BeforeRH(SuppressBand As Boolean)
   'Set the module-level variables for the fields to
   'export from the Report Header
   CustPONum = CustomerPONumber
   CustNum = CustomerNumber
End Sub
```

```
Private Sub Report_BeforeRF(SuppressBand As Boolean)
   'Set the module-level variable for the field to
   'export from the Report Footer
   TtlAmt = Str(DocumentAmountLASTRF)

   'Create a text file
   Open "InvAmts.TXT" For Append As #1
   Print #1, "Customer PO Number: " + CustPONum
   Print #1, "Customer Number: " + CustNum
   Print #1, "Total Amount: " + TtlAmt
   Print #1, ""

   'Close the text file
   Close #1
End Sub
```

Using Events to Print Data From a DUOS object

After using the Dynamics User Object Store to save additional information, use VBA to print the DUOS data object detail on a report. Printing data from the DUOS involves:

1. Adding calculated fields to the report.
2. Adding the report and the new calculated fields to your project.
3. Then using VBA to set the calculated fields to the values stored in the DUOS objects.

The VBA code that you write can be executed from the Report event that corresponds to the section of the report where the calculated fields have been added.

In the following example, two new calculated fields (AuthorizationCode and EmailAddress) are added to the RM Customer Report to represent the "Authorization Code" and "Email Address" from the "Customers" DUOS data collection. The fields were added to the Additional Header section of the report. The following code uses the BeforeAH Event to print information from the DUOS object. It is similar to the example code used for retrieving a DUOS object. However, note that the **Customer Number** field on the RM Customer Report represents the Customer ID field that is on the Customer Maintenance window.

```
Private Sub Report_BeforeAH(ByVal Level As Integer, SuppressBand As Boolean)

    Dim CustomerCollection As DUOSObjects
    Dim CustomerObject As DUOSObject

    'Return the customer collection
    Set CustomerCollection = DUOSObjectsGet("Customers")
    Set CustomerObject = CustomerCollection(CustomerNumber)

    'Update the report fields
    AuthorizationCode = CustomerObject.Properties("Authorization Code")
    EmailAddress = CustomerObject.Properties("Email Address ")
End Sub
```


Microsoft Official Training Materials for Microsoft Dynamics ™
Your use of this content is subject to your current services agreement

Using Events with Original or Modified Reports

In your project, events written for a Microsoft Dynamics GP report can either be assigned to the original Microsoft Dynamics GP report, a report modified using the Report Writer, or set so they never execute. This assignment can be made by using the EventMode property in the Visual Basic Properties window for each Microsoft Dynamics GP report in your project.

For the events to execute only for the original Microsoft Dynamics GP report, set the EventMode to emOriginal. If the events should execute only for the modified version of the report, set the EventMode to emModified. If you do not want the code to currently execute at all, set the EventMode to emNever.

NOTE: *You only need to use Report Writer to add report and report field objects to VBA. If you do not modify the report directly using the Report Writer such as by adding new fields to the report, you can use the VBA events written with the original Microsoft Dynamics GP report.*

Lab 9.1 – Adding Fields to Existing Microsoft Dynamics GP Reports

Company: Fabrikam, Inc.

Scenario: This exercise shows you how to add fields to existing Microsoft Dynamics GP reports using Report Writer. This exercise works with the Vendor List report. Since two fields were added to the Microsoft Dynamics GP Vendor Maintenance window in the previous chapter, the Vendor List report should also be modified to display those two new fields.

Challenge Yourself!

- Use the following information to customize the Vendor List report:

- Add two calculated fields to the Vendor List report to represent the additional fields on the maintenance window.
- After creating the new fields, add them to the report.
- Save the report layout.

Microsoft Official Training Materials for Microsoft Dynamics ™
Your use of this content is subject to your current services agreement

Need a Little Help?

- Use the following information to customize the Vendor List report.
- From the Microsoft Dynamics GP application, navigate Report Writer. (TOOLS→CUSTOMIZE→REPORT WRITER)
- Select the Microsoft Dynamics GP product and click **OK**. Click the Reports button from the Toolbar to list the Microsoft Dynamics GP reports.
- Navigate to the Vendor List report, select it, and click the Insert button to add it to the Modified Reports List.
- Add two calculated fields to the Vendor List report to represent the additional fields on the maintenance window. Name the fields **URL Address** and **General Satisfaction**.

- After creating the new fields, add them to the body section of the report.
- Save the report layout by closing the layout window.

Lab 9.2 – Populating Calculated Fields in Microsoft Dynamics GP Reports

Company: Fabrikam, Inc.

Scenario: In this exercise, you need set the new calculated fields on the report to the values that were saved to the DUOS table from the modified Vendor Maintenance window. This exercise works with the Vendor List report.

Challenge Yourself!

- Use the following information to retrieve the data from the DUOS table to the Vendor List report:

```
System:      12/27/2005  5:08:00 PM                    Fabrikam, Inc.
User Date:   4/12/2007                             VENDOR SUMMARY LIST
                                                   Payables Management

Ranges:
  Vendor ID:
  Vendor Name:
  Vendor Class:

  Vendor Status:
  1099 Type:

Sorted By:

Vendor ID       Name                            Class ID          Phone
-----------------------------------------------------------------------
  Address
-----------------------------------------------------------------------
    City                          State                          ZIP C
-----------------------------------------------------------------------
ACETRAVE0001    A Travel Company                AUS-NSW-M         (298)
  123 Riley St.
    Sydney                        NSW                            2086
      URL Address for  Ace            Excellent
```

- Add the report and the appropriate fields to the VBA project.
- Using VBA, select the event that could be used to retrieve information from the DUOS table and populate the two new calculated fields created in the previous project.

HINT: *Drop-down lists are always stored in the Microsoft Dynamics GP tables as integer values. Since the* ***General Satisfaction*** *field is a drop-down list, it is stored in the DUOS table as 1,2,3,4, and 5 instead of Excellent, Good, Average, Fair, and Poor. When the DUOS information is placed on a report, by default it appears as stored in the table. If you would like the corresponding text values to be shown, include additional code in your VBA event to translate those values.*

Microsoft Official Training Materials for Microsoft Dynamics ™
Your use of this content is subject to your current services agreement

- Use this event along with the appropriate methods and properties to complete this task within your Microsoft Dynamics GP VBA project.
- Compile the code to verify it will run in Microsoft Dynamics GP.
- Print the report to view the new information.

Need a Little Help?

- Use the following information to retrieve the data from the DUOS table to the Vendor List report:
- Within Report Writer, open the layout of the Vendor List report. Add the following resources to the VBA project:
  - Vendor List report
  - **Vendor ID** field
  - **URL Address** field
  - **General Satisfaction** field

- Using VBA, select the event that could be used to retrieve information from the DUOS table and populate the two new calculated fields created in the previous project.

HINT: *Drop-down lists are always stored in the Microsoft Dynamics GP tables as integer values. Since the* ***General Satisfaction*** *field is a drop-down list, it is stored in the DUOS table as 1,2,3,4, and 5 instead of Excellent, Good, Average, Fair, and Poor. When the DUOS information is placed on a report, by default it appears as it is stored in the table. If you would like the corresponding text values to be shown, include additional code in your VBA event to translate those values.*

- Use this event along with the appropriate methods and properties to complete this task within your Microsoft Dynamics GP VBA project.
- Compile the code to verify it will run in Microsoft Dynamics GP. (DEBUG→COMPILE MICROSOFT_DYNAMICS_GP)
- From Report Writer, print the report to view the new information.

Your use of this content is subject to your current services agreement

Conclusion

In this chapter you learned that data stored in tables can be added to reports using calculated fields. To accomplish this task, both Modifier and VBA are used together. We discussed the different methods, properties and events for Reports and that VBA code exist as part of a Report event, such as start, end or band events. Remember a band event refers to a section of a report such as the body or header and its primary use is to return or set the value of a field within that band.

Microsoft Official Training Materials for Microsoft Dynamics ™
Your use of this content is subject to your current services agreement

Quick Interaction: Lessons Learned

Take a moment and write down three Key Points you have learned from this chapter:

1.

2.

3.

Microsoft Official Training Materials for Microsoft Dynamics ™
Your use of this content is subject to your current services agreement