CHAPTER 6: WORKING WITH WINDOWS AND WINDOW FIELDS

Training Objectives

Actively participating during this chapter helps you learn:

- How to add windows and window fields to the Visual Basic for Applications (VBA) Project.
- How to use common methods and properties for windows and window fields.
- How to use common events for windows and window fields.

Adding Windows and Window Fields to Your Project

To use windows or window fields with VBA, you first must add the appropriate window to the Microsoft Dynamics™ GP project. Then add any fields on that window that you want to reference.

Adding Windows

To add a window object to your project, switch to Microsoft Dynamics GP and display the window to add. From the Tools menu select Customize, and select Add Current Window to Visual Basic. This adds the currently active or the topmost window to your project. The following illustration shows this menu option:

Some windows such as the Company Login window is referred to as a Modal Window. When a modal window is active, the menu bar items are not available. In situations where you need to work with the modal window and want to add it to your VBA project, you must use the keyboard equivalent to the menu item, CTRL+F11. To add fields you will use SHIFT+F11.

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

After a window has been added to the project, switch to the Visual Basic Editor and display the Project Explorer. It should look as follows:

Continue to switch between the Visual Basic Editor and Microsoft Dynamics GP to add windows as needed through the project's development. After adding a window to your Microsoft Dynamics GP project, it displays within Microsoft Dynamics GP with a period (.) after the window's title. This provides a visual cue that the window is part of the VBA project.

Removing Windows

To remove a window object from the VBA project, remove all references to the window within the Visual Basic Editor. Next switch to Microsoft Dynamics GP and display the window you want to remove. On the Tools menu, select Customize and select Remove Current Window from Visual Basic. The window and all fields associated with the window is removed from the project.

Adding Windows Fields

To add window fields to your VBA project, first add the window that contains the fields. To do this, follow the instructions given in the previous section of this chapter. After the window has been added, switch to Microsoft Dynamics GP and verify that the window is displayed. From the Tools menu select Customize, and select Add Field to Visual Basic. This activates a field selection mode and changes the cursor to a selection pointer.

After the selection mode has been activated, click on each data entry, push button, or list field to add to your project. Be sure to click on the field itself, not the static text prompt associated with the field. When finished adding fields, select Customize from the Tools menu, and select Add Fields to Visual Basic to deactivate field selection mode. Field selection mode can also be deactivated by pressing the Esc key. Otherwise it is automatically deactivated when you switch to the Visual Basic Editor

Window fields display within the Visual Basic Editor in a different manner than the window objects discussed previously. Instead of being listed in the Project Explorer, they appear in the Visual Basic Code window for their corresponding window. The following illustration shows window field objects that were added to a VBA project:

Spaces and special characters cannot be used in VBA field names and are automatically removed when the field is added to the project.

Some Microsoft Dynamics GP fields have names that are VBA reserved words, such as "Type." You can change the field name using the VBA properties window or fully qualify it when referencing it in your code.

Use the Visual Basic Code window to write VBA code for the window field or to reference the window field from other VBA procedures in your project.

NOTE: *You cannot remove individual window fields from your VBA project, only window objects. However, when removing a window object, all of the fields associated with the window are also automatically removed.*

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

Using Methods and Properties

Windows and window fields can be manipulated in different ways through the use of methods and properties. Methods refer to actions that can be performed on an object and properties refer to characteristics that can be retrieved or set. There are a number of different methods and properties available for both Microsoft Dynamics GP windows and window fields. Those used most often are described next, and a corresponding example is given for each.

NOTE: *Methods and properties deal with how a window or window field can be manipulated. Discussion about where the methods and properties can be used is covered later in this chapter.*

Window Methods and Properties

Opening and Closing Windows

Using VBA, open or close a Microsoft Dynamics GP window either directly or indirectly. To open and close it directly, use the window's Open and Close methods. For example, to open the Receivables Transaction Entry window, use the following code within one of your VBA procedures.

```
ReceivablesTransactionEntry.Open
```

To open and close it indirectly, select the field that is used to normally display the window in Microsoft Dynamics GP. For example, the following code opens the Customer and Prospects lookup window by executing the business logic behind the Lookup button named Lookup Buttons on the Receivables Transaction Entry window:

```
ReceivablesTransactionEntry.LookupButtons = 1
```

It is generally recommended to use the indirect method of opening and closing windows. This is because it allows all of the Microsoft Dynamics GP code to execute as it normally would when the window is opened or closed within the application.

Activating a Window

The VBA Activate method activates a window that is open and visible. By activating the window, it makes it the front most window and expands it if it is minimized. The following example activates the Customer Maintenance window.

```
CustomerMaintenance.Activate
```


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

Hiding and Showing Windows

Hiding a window can be useful within your development because even if a window is hidden, it is still open and the data on the window is still accessible to other objects within your VBA project. This makes hiding a window useful when referencing data contained in a window without displaying the entire window to the user.

There are two different ways to hide and show a Microsoft Dynamics GP window that you have opened:

- Use the Hide and Show methods for the window.
- Set the Visible property for the window object to False or True.

For example, when a user is entering data on the Receivables Transaction entry window, you may want to access information that exists on the Customer Maintenance window without displaying a second window. Use either of the following examples to make the window invisible but still accessible through code:

```
CustomerMaintenance.Hide
CustomerMaintenance.Visible = False
```

Moving and Resizing a Window

Resize or reposition a window using Height, Left, Top, and Width properties. Resizing and repositioning a window is helpful when organizing workspace. For example, instead of opening the Distribution window over top of the entire Receivables Transaction Entry window, you may open the window to the right of the amount fields to see some of the information that was entered on the previous window. This can be implemented by using the following code:

```
SalesTransactionDistribution.Left =
ReceivablesTransactionEntry.Width
 / 2 + ReceivablesTransactionEntry.Left
```

Changing a Window Title

The Caption property allows you to set a window's caption or title. The following example changes the title of the Customers and Prospects lookup window from "Customers and Prospects" to "Patients."

```
CustomersandProspects.Caption = "Patients"
```

NOTE: *Microsoft Dynamics GP uses a "." at the beginning of a window caption on a modified form. A "." will also be placed at the end of the caption when VBA code is attached to the window. These periods become part of the caption string and may need to be included when referencing the caption within the VBA code.*

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

Verifying if a Window is Open

Use the IsLoaded property to verify if a window is loaded or opened before trying to reference fields on that window. For example, if you want to use a field on the Customer Options window from one of the procedures on the Customer Maintenance window, verify the window is open prior to referencing the field. This prevents VBA errors from occurring when referencing a field on a window that is not open.

```
If RMCustomerOptions.IsLoaded = True Then
```

Window Field Methods and Properties

Disabling and Locking Fields

The Enabled and Locked properties allow you to make a field read-only and unavailable to the user. To disable a field, set the Enabled property to False. It then appears dimmed, and any related prompts and lookup or expansion buttons are also disabled. To lock a field, set its Locked property to True. It is then inactive, but does not appear dimmed.

For example, the following code disables the **Post** button on the Receivables Transaction Entry window. In addition, it also locks the **Document Date** field so that it cannot be changed.

```
Post.Enabled = False
DocumentDate.Locked = True
```

Controlling the Focus

There are two methods that allow you to control the movement of focus from different fields on a window. These methods include the Focus method and the FocusSeg method. The Focus method moves focus to a specified field. The FocusSeg method moves focus between segments in a composite by referencing the corresponding index for the segment. For example, the following VBA code moves focus to the **Sales Amount** field:

```
SalesAmount.Focus
```

While the next line of code moves focus to the second segment of the **Account Number** field:

```
Account.FocusSeg (2)
```


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

Hiding and Showing Fields

Unlike windows, there is only one way of hiding or showing window fields within VBA. This is by setting the field's Visible property to False or True. If the Visible property is set to False, it is still on the window and accessible from other objects, but both the field and its corresponding caption are hidden from the user. In the following example the Trade Discount, Freight, and Miscellaneous Amounts are made invisible.

```
TradeDiscount.Visible = False
Freight.Visible = False
Miscellaneous.Visible = False
```

NOTE: *Once Microsoft Dynamics GP code has made a field invisible, you cannot set the value of the field. In addition, it is not possible to show a field once the field has been hidden by the Microsoft Dynamics GP code.*

Making Fields Required

The Required property for a field specifies whether the field must contain data before the user can save information displayed on a window. To make a field required, set its Required property to True. The next example sets both the **Freight Amount** and the **Shipping Method** fields to Required.

```
Freight.Required = True
ShippingMethod.Required = True
```

NOTE: *Selecting Microsoft Dynamics GP "**Show Required Fields**" also shows additional fields made required through the VBA code. New fields that have been added with Modifier must have a linked prompt to be affected by the **Show Required Fields** option.*

Returning and Setting Field Values

There are a couple of different ways to return or set the value of a field. Each method is described in the following section along with a simple example of each.

NOTE: *You can return or set a field value as long as the window containing the field is open. This is also true if you have disabled, locked, or hidden the field using VBA. On the other hand, if Microsoft Dynamics GP has disabled, locked, or hidden the field, you do not have access to set the field's value. The field value can be retrieved, however.*

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

The Value property can be used to return or set a window field value. Simply specify the field name without the Value property. VBA still assumes you want to return or set the fields value. The following are two examples of setting the **Salesperson ID** field.

```
'Using the Value property to set a field
SalespersonID.Value = "SEAN W."

'Setting a field without using the Value property
SalespersonID = "SEAN W."
```

VBA returns or accepts specific values for each type of field. Data entry fields such as strings, dates, and currency fields always use string values. While list fields, push buttons, check boxes, and option buttons use numeric values.

When working with date fields, it is usually best to set the field to an unformatted date that includes two digits for the month, two digits for the day and four digits for the year. For example, "07022000." This ensures that the year is handled correctly regardless of how the user's system is set up.

The ValueSeg property can be used to return or set the value of a given segment in a composite field. This is helpful when working with only a single segment of the field since the Value property can evaluate or affect the entire composite, not just a single segment. The following example returns the third segment of the **Account Number** field to determine if it is equal to "50."

```
If AccountNumber.ValueSeg(3) = "50" Then
```

The Empty property allows you to determine whether the value of a certain field is empty. Also use the Empty property to set a field to an empty or null value. The following is an example of clearing the **Trade Discount Amount** field or making it empty.

```
TradeDiscount.Empty = True
```

Controls as Fields in VBA

All window objects are added to VBA as "fields," whether they are data fields, buttons, drop-down lists, or any other type of control. Therefore, the properties available for a button are the same as those for a data field. To "push" a button, select a check box or select an item in a drop-down list the field property is used and is set to a numeric value. If you omit the value property from your code as seen below, VBA assumes you are returning or setting the value of a field.

```
Save.Value = 1
Save = 1
```


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

To set the value of a drop-down list, select the desired item by referring to its numeric position in the list.

```
CustomerSortBy = 3
```

Performing Numeric Calculations

Since all field values returned to VBA are strings values, numeric calculations, or comparisons may require you to convert the string value to the appropriate data type prior to performing calculations. This conversion can easily be done by using one of the supported VBA conversion functions. These supported functions are given in the following table:

Function	Description
CInt()	This converts a Microsoft Dynamics GP string value to an integer value.
CDate	This converts a Microsoft Dynamics GP string value to a date value.
CCur()	This converts a Microsoft Dynamics GP string value to a currency value.
CLng()	This converts a Microsoft Dynamics GP string value to a long integer value.

The next example compares two currency values where the first field (Cost) is equal to $100.00 and the second field (SalesAmount) is equal to $20.00. For each of the two fields the example uses the CCur() function to convert the string value to the appropriate currency amount. The conversion is critical in this situation, because without it the calculation is incorrect. The "2" within the $20.00 string with be considered larger than the "1" within the $100.00 string.

```
If CCur(Cost) > CCur(SalesAmount) then
```


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

Verifying Field Values

When setting a field's value within a Microsoft Dynamics GP project, VBA automatically executes the Microsoft Dynamics GP code written for that field. Then, Microsoft Dynamics GP can verify the field's value. However, depending on when the value is set, the verification may provide differing results. Many times as part of the verification performed by Microsoft Dynamics GP, focus is expected to be in the field being updated. If focus is not in the field, the following data verification error may display:

For many fields Microsoft Dynamics GP does little or no field verification, so setting the value before the field has focus does not generate errors. However, Microsoft Dynamics GP has a much more complex verification process for fields that affect Microsoft Dynamics GP business logic or for those fields that can be added on the fly. Therefore, if the field's value is set before focus moves into those types of fields, the VBA error occurs.

To avoid this problem, set a field's value when focus is on that field so that all of the Microsoft Dynamics GP code can execute as is expected when a user enters data into a field and moves out of the field. If it is necessary to set a field's value when focus is not on that field, try using the Focus method with an additional set value parameter. For example, the following code moves focus to the **Customer ID** field, and then sets its value to "CUST100."

```
CustomerID.Focus ("CUST100")
```

Using Events

This section discusses when the methods and properties available for windows and windows fields are executed. This is where window and window field events are used.

Window and window field events are VBA procedures written to correspond to processes or events that occur in Microsoft Dynamics GP. These processes include things such as the opening or closing of a window or the clicking of a push button. Window or window field events can execute either before or after the Microsoft Dynamics GP code that was written for that process or event. Because of this, window event procedures use names like "BeforeClose" and window field event procedures use names like "AfterUserChanged."

To view events available for each object, use the Code window in the Visual Basic Editor. After the object has been 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 window or window field. An example of the Code window is shown in the following illustration listing the events for the **Customer ID** field on the Microsoft Dynamics GP Receivables Transaction Entry form.

A number of different events are available for both Microsoft Dynamics GP windows and window fields. Each event is described next, and a corresponding example is given for those events that are used most often.

Window Events

BeforeOpen Event

Open Events occur when a user opens a specified window. The BeforeOpen event executes before any of the Microsoft Dynamics GP code that normally runs when a window is opened. This event is helpful when setting default field values for a window, while still allowing the Microsoft Dynamics GP code to override defaults as appropriate.

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

In the following example, the **Shipping Method** field on the Customer Maintenance form (CARDS→SALES→CUSTOMER) defaults to "PICKUP." However, because it is a BeforeOpen event, Microsoft Dynamics GP code runs after the event. The Microsoft Dynamics GP code can then overwrite the Shipping Method with a value from a data table when the window is opened with an existing record displayed such as when it is opened from the zoom field on the Receivables Transaction Entry window.

```
Microsoft_Dynamics_GP - CustomerMaintenance (Code)
Window                                  BeforeOpen

Option Explicit

Private Sub Window_BeforeOpen(OpenVisible As Boolean)
    'Set the Shipping Method field
    ShippingMethod = "PICKUP"
End Sub
```

AfterOpen Event

Open Events occur when the user opens a specified window. The AfterOpen event executes after any of the Microsoft Dynamics GP code that normally runs when opening a window. This event is helpful for completely overriding values defaulted by the Microsoft Dynamics GP code.

In the following example, the Default Site ID on the Sales Transaction Entry window (TRANSACTIONS→SALES→SALES TRX ENTRY) defaults to "NORTH" instead of the default value set by Microsoft Dynamics GP of "WAREHOUSE."

```
Microsoft_Dynamics_GP - SalesTransactionEntry (Code)
Window                                  AfterOpen

Option Explicit

Private Sub Window_AfterOpen()
    'Set the Default Site ID different than Microsoft Dynamics GP default
    DefaultSiteID = "NORTH"
End Sub
```

BeforeClose Event

Close Events occur when the user closes a specified window. The BeforeClose event executes before any of the Microsoft Dynamics GP code normally runs when a window is closed. This event is useful when cancelling the closing of the window if the contents of the window do not meet certain criteria. This is made possible through the use of a special parameter called AbortClose.

In the following example, if a "Substitute Item 1" is entered on the Item Maintenance Options window (CARDS→INVENTORY→ITEM → OPTIONS), the window cannot be closed unless a "Substitute Item 2" is also given. An appropriate message is given if the window cannot be closed.

```
Microsoft_Dynamics_GP - ItemMaintenanceOptions (Code)
Window                                   BeforeClose

Option Explicit

Private Sub Window_BeforeClose(AbortClose As Boolean)
    'Check to see if the 1st substitute item has been entered
    If SubstituteItem1.Empty = False And _
        SubstituteItem2.Empty = True Then
        'Stop the closing of the window
        AbortClose = True
        'Display a message to the user
        MsgBox "A second substitute item must be entered."
        SubstituteItem2.Focus
    End If
End Sub
```

AfterClose Event

Close Events occur when the user closes a specified window. The AfterClose event executes after any of the Microsoft Dynamics GP code that normally runs when closing a window. This event allows you to perform "clean-up" tasks, such as closing other windows, especially those opened invisibly by your VBA code.

In the following example, when the Invoice Entry window (TRANSACTIONS→SALES→INVOICE ENTRY) is closed, if the Invoices and Returns lookup window (TRANSACTIONS→SALES→INVOICE ENTRY→DOCUMENT NO. LOOKUP) is open, it automatically closes too.

```
Microsoft_Dynamics_GP - InvoiceEntry (Code)
Window                                   AfterClose

Private Sub Window_AfterClose()
    'Check to see if the Invoice and Returns Lookup is open, if it is close the window
    If SmartList.InvoicesandReturns.IsLoaded = True Then
        SmartList.InvoicesandReturns.Close
    End If
End Sub
```

Because the Invoices and Returns lookup window is a third-party form, it resides in the SmartList dictionary. For this exercise to work, you must mark a reference from the Microsoft Dynamics GP project to the SmartList project. You must also fully declare the object in code by including the name of the SmarLiist project.

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

BeforeActivate Event

Activate events occur when a user activates a window, either by opening it or by making it the front most window. The BeforeActivate event executes before any of the Microsoft Dynamics GP code that normally runs when activating a window. Microsoft Dynamics GP rarely executes application code during a window activate event, similarly you may not use this event very often.

One situation when you might use an Activate event is to set the default value of a Sort By field on a Microsoft Dynamics GP window. Because of the unique way that the Microsoft Dynamics GP code runs when a lookup window is open, it is likely that any values set through the use of a standard Open event will be overwritten when the Microsoft Dynamics GP code executes. Therefore, an Activate event may be more useful in handling that situation.

In the following example, the code will set the Additional Sorts on the Vendors lookup window (CARDS→ PURCHASING→VENDOR →VENDOR ID LOOKUP) to default to Payment Priority when the window is opened for the first time. Note that a Boolean variable is declared in this example, and that it is initialized in an Open event and evaluated in the Activate event.

Because the Vendor ID lookup window is a third-party form, it resides in the SmartList dictionary. This code was placed in the SmartList project for the Vendors window.

AfterActivate Event

Activate events occur when a user activates a window, either by opening it or by making it the front most window. The AfterActivate event executes after any of the Microsoft Dynamics GP code that normally runs when activating a window. Microsoft Dynamics GP rarely executes application code during a window activate event. Similarly, you may not use this event very often. However, the example given previously as a BeforeActivate event could also be used as an AfterActivate event.

BeforeModalDialog Event

Modal Dialog events occur when Microsoft Dynamics GP opens a modal dialog message. The BeforeModalDialog event executes before the Microsoft Dynamics GP modal dialog is displayed. This event allows you to dismiss the message automatically by writing code that "answers" the question for the user. In addition, this event allows you to change the contents of the message displayed. This is done through the use of special parameters available for the event. They include a PromptString parameter that represents the text on the message and an Answer parameter that represents the button selected.

Normally, on the Invoice Entry window (TRANSACTIONS→SALES→INVOICE ENTRY) if a user enters a Batch ID that does not exist, a message displays asking the user if they want to add this batch. In the following example, the code automatically dismisses the modal dialog and adds the new batch by opening the Invoice Batch Entry window.

Microsoft_Dynamics_GP - InvoiceEntry (Code)

Window | BeforeModalDialog

```
Option Explicit

Private Sub Window_BeforeModalDialog(ByVal DlgType As _
DialogType, PromptString As String, Control1String As _
String, Control2String As String, Control3String As _
String, Answer As DialogCtrl)
   'Verify which modal dialog is currently displayed
   If PromptString = "Do you want to add this batch?" Then
      'Add the batch by selecting the Add button
      Answer = dcButton1
   End If
End Sub
```

CAUTION: *The prompt string in the BeforeModalDialog Event may differ depending on what caused the message to be displayed. For example, on the Customer Maintenance window, if the user tries to close the window before saving changes, the message is as follows: "Do you want to save changes?" Comparatively, the message when you choose to display another record using the browse buttons is "Do you want to save changes to this customer?" Make sure your modal dialog event looks for all messages that deal with saving changes for the current record.*

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

AfterModalDialog Event

Modal Dialog events occur when Microsoft Dynamics GP opens a modal dialog message. The AfterModalDialog event executes after the Microsoft Dynamics GP modal dialog is displayed. This allows you to determine the way the user responded to the message and then executes additional code based on that response. This is done through the use of special parameters available for the event. They include a PromptString parameter that represents the text on the message and an Answer parameter that represents the button selected.

Normally, on the Invoice Entry window (TRANSACTIONS→SALES→INVOICE ENTRY) if a user enters a **Customer ID** that does not exist, a message displays asking the user to add this record. In the following example, if the user selects Cancel on the message so that the new customer is not added, focus returns to the **Customer ID** field and the field is also cleared.

Microsoft_Dynamics_GP - InvoiceEntry (Code)

Window | AfterModalDialog

```
Option Explicit

Private Sub Window_AfterModalDialog(ByVal DlgType As DialogType, _
PromptString As String, Control1String As String, Control2String As _
String, Control3String As String, Answer As DialogCtrl)
   'Determine which modal dialog is currently 'displayed
   If PromptString = "Do you want to add this customer record?" Then
      'Check to see if the Cancel button was selected
      If Answer = dcButton2 Then
         'Move focus back to Customer ID and make it empty
         CustomerID.Focus ("")
      End If
   End If
End Sub
```

Window Field Events

BeforeGotFocus Event

Got Focus events occur when initially entering into a field. The BeforeGotFocus event executes before any of the Microsoft Dynamics GP code that normally runs when focus moves into a field. This event allows you to set or evaluate the value of the field that is gaining focus. In addition, there is a special parameter available from this event that allows you to cancel the other got focus events for this field. The parameter is called CancelLogic and when set to true cancels the execution of all other code for the field.

In the following example, on the Payables Transaction Entry window (TRANSACTIONS→PURCHASING→TRANSACTION ENTRY) when focus moves to the **Freight** field, if the **Shipping Method** field is empty, a message is displayed asking them to also enter a shipping method.

Microsoft_Dynamics_GP - PayablesTransactionEntry (Code)

Freight | BeforeGotFocus

```
Option Explicit

Private Sub Freight_BeforeGotFocus(CancelLogic As Boolean)
    If ShippingMethod.Empty = True Then
        MsgBox "You will also need to enter a shipping method."
    End If
End Sub
```

AfterGotFocus Event

Got Focus events occur when the user initially enters into a field. The AfterGotFocus event executes after any of the Microsoft Dynamics GP code that normally runs when focus moves into a field. This event allows you to set or evaluate the value of the field that is gaining focus.

In the following example, if the **Payment Terms** field is empty on the Receivables Transaction Entry window (TRANSACTIONS→SALES→TRANSACTION ENTRY), the Payment Terms lookup window opens automatically when focus is moved into the field.

Microsoft_Dynamics_GP - ReceivablesTransactionEntry (Code)

PaymentTerms | AfterGotFocus

```
Private Sub PaymentTerms_AfterGotFocus()
    'Determine if a payment term has been entered
    If PaymentTerms.Empty = True Then
        'The field is empty so open the lookup window
        LookupButtons = 1
    End If
End Sub
```


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

BeforeLostFocus Event

Lost Focus events occur when the user leaves a field regardless of whether its contents have changed. The BeforeLostFocus event executes before any of the Microsoft Dynamics GP code that normally runs when focus moves out of a field. This event allows you to evaluate the value of the field that is losing focus. In addition, there are some special parameters available for this event. The parameters are called KeepFocus and CancelLogic and should be set to true to keep focus in the current field or to cancel all other lost focus events for the field.

In the following example, if the **Batch ID** field is empty when the user moves out of the field on the Sales Transaction Entry window (TRANSACTIONS→SALES→SALES TRX ENTRY), a message displays asking the user if they want to enter an ID. If they select yes, focus remains in the field and all other processing for the field is canceled.

Microsoft_Dynamics_GP - SalesTransactionEntry (Code)

BatchID | BeforeLostFocus

```
Option Explicit

Private Sub BatchID_BeforeLostFocus(KeepFocus As Boolean, _
CancelLogic As Boolean)
Dim Response As Integer

   'Determine if a Batch has been entered
   If BatchID.Empty = True Then
      'Ask the user if they would like to enter an ID
      Response = MsgBox("Do you want to enter a Batch ID?", vbYesNo)
      'If they choose to add an ID move focus into field
      If Response = vbYes Then
         CancelLogic = True
         KeepFocus = True
      End If
   End If
End Sub
```

AfterLostFocus Event

Lost Focus events occur when the user leaves a field regardless of whether its contents have changed. The AfterLostFocus event executes after any of the Microsoft Dynamics GP code that normally runs when focus moves out of a field. This event allows you to evaluate the value of the field that is losing focus and to perform any appropriate operations for the field.

In the following example, if the **Customer ID** field is empty and the user moves out of the field on the Sales Transaction Entry window (TRANSACTIONS→SALES→SALES TRX ENTRY), a message displays asking if the user want to select a **Customer ID**. If they select yes, the Customers and Prospects lookup window opens.

Microsoft_Dynamics_GP - SalesTransactionEntry (Code)

CustomerID | AfterLostFocus

```
Private Sub CustomerID_AfterLostFocus()
    Dim Response As Integer

    'Check to see if a customer ID has been entered
    If CustomerID.Empty = True Then
        'A customer hasn't been entered ask if they want to add one
        Response = MsgBox("Do you want to select a Customer ID?", vbYesNo)
        If Response = vbYes Then
            'Open the Customer Lookup window so an ID can be selected
            LookupButton5 = 1
        End If
    End If
End Sub
```

BeforeUserChanged Event

User Changed events occur whenever the field associated with the event has been changed by a user. These events occur at different times depending on the type of field you are working with. For fields that toggle, such as push buttons and drop-down lists, the event occurs when the user clicks on the field or selects an item from the list. For other types of fields, such as string fields, the event occurs when the user changes the value of the field and leaves the field.

The BeforeUserChanged event executes before any Microsoft Dynamics GP code that normally runs when the content of the field changes and focus leaves the field. This event allows you to evaluate the value of the field and to cancel the Microsoft Dynamics GP processing that normally executes for the field. The Microsoft Dynamics GP processing is canceled by setting a special parameter available for this event to true. This parameter is CancelLogic. In addition another parameter called KeepFocus can be set to true to keep focus in the current field.

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

In the following example, if the user enters a trade discount amount on the Receivables Transaction Entry window (TRANSACTIONS→SALES→TRANSACTION ENTRY) that is greater than $50, a message displays saying that they are not authorized to give a discount of that amount. Processing then stops and focus remains in the field so the amount can be reentered.

Microsoft_Dynamics_GP - ReceivablesTransactionEntry (Code)

TradeDiscount | BeforeUserChanged

```
Private Sub TradeDiscount_BeforeUserChanged(KeepFocus As Boolean, _
CancelLogic As Boolean)
   Dim Response As Integer

   'Check to see the amount of the trade discount
   If CCur(TradeDiscount) > 50 Then
   'Give message indicating discount is too large, cancel processing and
   'move focus back into the discount field
      MsgBox ("You are not authorized to enter a trade discount greater than $50.")
      KeepFocus = True
      CancelLogic = True
   End If

End Sub
```

AfterUserChanged Event

The AfterUserChanged event executes after any Microsoft Dynamics GP code that normally runs when the contents of the field changes and focus leaves the field. This event allows you to evaluate the information entered into the field and to take appropriate actions based on that information.

In the following example, if the Document Type on the Receivables Transaction Entry window (TRANSACTIONS→SALES→TRANSACTION ENTRY) is set to "Service/Repairs," the **Freight Amount** field is disabled.

Microsoft_Dynamics_GP - ReceivablesTransactionEntry (Code)

DocumentType | AfterUserChanged

```
Private Sub DocumentType_AfterUserChanged()
   'Determine if document is a Service/Repair type
   If DocumentType = 4 Then
      'If Service/Repair don't allow a freight amount to be entered
         Freight.Enabled = False
      Else
         Freight.Enabled = True
      End If
End Sub
```

Changed Event

The Changed event occurs every time the contents of a field changes. This includes the following situations:

- The user changes the field
- The Microsoft Dynamics GP code changes a field
- VBA or DDE changes a field

This allows you to verify that the code executes in every possible situation. This is unlike the User Changed events, which only react to changes made by the user.

CAUTION: *In many cases, Microsoft Dynamics GP may cause the Changed event to occur numerous times for the same field event though the value of the field has not changed visibly. This executes your associated code multiple times, which may not be appropriate in all situations.*

In the following example, as the user browses through records on the Receivables Transaction Entry window (TRANSACTIONS→SALES→TRANSACTION ENTRY), if the "Total" is greater than $2500, a message is given saying that the transaction must be authorized.

HINT: *Type the first character of the field to reference and hit CTRL Spacebar. A drop-down list opens displaying all objects beginning with that letter. Arrow down to the appropriate object and press TAB to complete "typing" the object name. Type a "." after the name to display all of the properties and methods available for that object.*

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 Windows

The events written for the Microsoft Dynamics GP windows in your project can be assigned to the original Microsoft Dynamics GP windows, windows modified using Modifier, or can be set so that they never execute. This assignment can be made by using the EventMode property in the Visual Basic Properties window for each Microsoft Dynamics GP window in your project.

To have the events execute only for the original Microsoft Dynamics GP window, set the EventMode to 0 - emOriginal. If the events should execute only for the modified version of the window, set the EventMode to 1 - emModified. If you do not want the code to currently execute at all, set the EventMode to 2 - emNever.

Conclusion

In this chapter we learned how to add window and window fields to a VBA project. We talked about methods and properties and discussed how code can be added to events to change or control information in a Microsoft Dynamics window.

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

Lab 6.1 – Disabling Fields

Company: Fabrikam, Inc.

Scenario: This exercise shows you how to use VBA code to disable a push button on a Microsoft Dynamics GP window. When a transaction is entered on the Payables Transaction Entry window, it has been decided that it should always be included as part of a batch. To eliminate confusion for the user, the **Post** button should be disabled on the window.

Challenge Yourself!

- Use the following information to customize the Payables Transaction Entry window:

- Add the window and the appropriate field to the VBA project.
- Using VBA, select the window or window field event that could be used to disable the **Post** button.
- 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 runs in Microsoft Dynamics GP.
- Test the VBA code by viewing the Microsoft Dynamics GP window.

Need a Little Help?

- Use the following information to customize the Payables Transaction Entry window:
- Navigate to the Payables Transaction Entry window. (TRANSACTIONS→PURCHASING→TRANSACTION ENTRY) Add the following resources to the VBA project:
  - Payables Transaction Entry window
  - **Post** field

- In the code window, select the appropriate resource and event. Using methods and properties, complete this task by writing code within the Microsoft Dynamics GP VBA project.
- Compile the code to verify it runs in Microsoft Dynamics GP. (DEBUG→COMPILE MICROSOFT_DYNAMICS_GP)
- Test the VBA code by viewing the Microsoft Dynamics GP window.

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

Lab 6.2 – Locking Fields

Company: Fabrikam, Inc.

Scenario: This exercise shows you how to use VBA code to lock a field on a Microsoft Dynamics GP window. This exercise works with the Receivables Transaction Entry window. When the user selects a Debit or Credit Memo, the **Document Date** field populates with the current date. This date must be used for the transaction. Therefore you do not want to allow the user to enter the **Document Date** field to change it.

Challenge Yourself!

- Use the following information to customize the Receivables Transaction Entry window:

- Add the window and the appropriate fields to the VBA project.
- Using VBA, select the window or window field event that could be used to lock the **Document Date** field.
- 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 runs in Microsoft Dynamics GP.
- Test the VBA code by viewing the Microsoft Dynamics GP window.

Need a Little Help?

- Use the following information to customize the Receivables Transaction Entry window:
- Navigate to the Receivables Transaction Entry window. (TRANSACTIONS→SALES→TRANSACTION ENTRY) Add the following resources to the VBA project:
  - Receivables Transaction Entry window
  - **Document Type** field
  - **Document Date** field
- In the code window, select the appropriate resource and event. Using methods and properties, complete this task by writing code within your Microsoft Dynamics GP VBA project.
- Compile the code to verify it runs in Microsoft Dynamics GP. (DEBUG→COMPILE MICROSOFT_DYNAMICS_GP)
- Test the VBA code by viewing the Microsoft Dynamics GP window.

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

Lab 6.3 – Setting Required Fields

Company: Fabrikam, Inc.

Scenario: This exercise shows you how to set a field on a Microsoft Dynamics GP window to be a required field using VBA code. This exercise works with the Invoice Entry window. The user must enter a value into the **Freight** field before the record can be saved or the window can be closed.

Challenge Yourself!

- Use the following information to customize the Invoice Entry window:

- Add the window and the appropriate fields to the VBA project.
- Using VBA, select the window or window field event that could be used to make the **Freight** field a required field.
- 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 runs in Microsoft Dynamics GP.
- Test the VBA code by viewing the Microsoft Dynamics GP window.

Need a Little Help?

- Use the following information to customize the Invoice Entry window:
- Navigate to the Invoice Entry window. (TRANSACTIONS→SALES→INVOICE ENTRY) Add the following resources to the VBA project:
  - Invoice Entry window
  - **Freight** field
- In the code window, select the appropriate resource and event. Using methods and properties complete this task by writing code within your Microsoft Dynamics GP VBA project.
- Compile the code to verify it runs in Microsoft Dynamics GP. (DEBUG→COMPILE MICROSOFT_DYNAMICS_GP)
- Test the VBA code by viewing the Microsoft Dynamics GP window.

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

Lab 6.4 – Conditionally Disabling Fields

Company: Fabrikam, Inc.

Scenario: This exercise shows you how to use VBA code to disable a button on a Microsoft Dynamics GP window if a specific condition is met. This exercise works with the Invoice Entry window. If the total of a transaction is greater than $10,000, the **Post** button should be disabled. This ensures that the transaction is approved as part of a batch prior to its posting.

Challenge Yourself!

- Use the following information to customize the Invoice Entry window:

- Add the window and the appropriate fields to the VBA project.
- Using VBA, select the window or window field event that could be used to disable the **Post** button if the total transaction is greater than $10,000.
- 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 runs in Microsoft Dynamics GP.
- Test the VBA code by viewing the Microsoft Dynamics GP window.

Need a Little Help?

- Use the following information to customize the Invoice Entry window:
- Navigate to the Invoice Entry window. (TRANSACTIONS→SALES→INVOICE ENTRY) Add the following resources to the VBA project:
  - Invoice Entry window
  - **Post** button
  - **Total** field
- In the code window, select the appropriate resource and event. Using methods and properties, complete this task by writing code within your Microsoft Dynamics GP VBA project.
- Compile the code to verify it runs in Microsoft Dynamics GP. (DEBUG→COMPILE MICROSOFT_DYNAMICS_GP)
- Test the VBA code by viewing the Microsoft Dynamics GP window.

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

Lab 6.5 – Opening Lookup Windows

Company: Fabrikam, Inc.

Scenario: This exercise shows you how to use VBA code to open a Lookup window from a Microsoft Dynamics GP window if a specific field is empty. This exercise works with the Purchase Order Entry window. When a user moves to the **Vendor ID** field and it is empty, the Vendor Lookup window should automatically open so that a vendor can be selected from the existing list.

Challenge Yourself!

- Use the following information to customize the Purchase Order Entry window:

- Add the window and the appropriate fields to the VBA project.
- Using VBA, select the window or window field event that could be used to open the Vendor Lookup window if the **Vendor ID** field is empty.
- 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 runs in Microsoft Dynamics GP.
- Test the VBA code by viewing the Microsoft Dynamics GP window.

Need a Little Help?

- Use the following information to customize the Purchase Order Entry window:
- Navigate to the Purchase Order Entry window. (TRANSACTIONS→PURCHASING→PURCH. ORDER ENTRY) Add the following resources to the VBA project:
  - Purchase Order Entry window
  - Lookup Button Next to the **Vendor ID** field
  - **Vendor ID** field

- In the code window, select the appropriate resource and event. Using methods and properties, complete this task by writing code within your Microsoft Dynamics GP VBA project.
- Compile the code to verify it runs in Microsoft Dynamics GP. (DEBUG→COMPILE MICROSOFT_DYNAMICS_GP)
- Test the VBA code by viewing the Microsoft Dynamics GP window.

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

Lab 6.6 – Changing Dialog Messages

Company: Fabrikam, Inc.

Scenario: This exercise shows you how to use VBA code to change a Dialog Message on a Microsoft Dynamics GP window. This exercise works with the Customer Maintenance window. If the user tries to delete a record from the Customer Maintenance window and the title of that window has been changed to "Patient Maintenance," the dialog message that is displayed should refer to deleting patient records instead of customer records.

Challenge Yourself!

- Use the following information to customize the Customer Maintenance window:

- Add the window and the appropriate fields to the VBA project.
- Using VBA, select the window or window field event that could be used to change the text on a Dialog Message.
- 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 runs in Microsoft Dynamics GP.
- Test the VBA code by viewing the Microsoft Dynamics GP window.

Need a Little Help?

- Use the following information to customize the Customer Maintenance window:
- Navigate to the Customer Maintenance window. (CARDS→SALES→CUSTOMER) Add the Customer Maintenance window to the VBA project.
- In the code window, select the appropriate resource and event. Using methods and properties, complete this task by writing code within your Microsoft Dynamics GP VBA project.

HINT: *To test this event, use Modifier or VBA to change the title or caption of the window. In your VBA code, include a "." in the title of the window for the event to run properly. Depending on how the window is modified, the "." may need to be before the title, after the title, or both. For example, ".Patient Maintenance."*

- Compile the code to verify it runs in Microsoft Dynamics GP. (DEBUG→COMPILE MICROSOFT_DYNAMICS_GP)
- Test the VBA code by viewing the Microsoft Dynamics GP window.

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

Lab 6.7 – Restricting Data in Fields

Company: Fabrikam, Inc.

Scenario: This exercise shows you how to use VBA code to restrict the user from entering a specific value into a field on a Microsoft Dynamics GP window. This exercise works with the Receivables Transaction Entry window. This exercise restricts the user from entering C.O.D. in the **Payment Terms** field.

Challenge Yourself!

- Use the following information to customize the Receivables Transaction Entry window:

- Add the window and the appropriate fields to the VBA project.
- Using VBA, select the window or window field event that could be used to restrict the user from entering C.O.D. into the **Payment Terms** field. You will give them a warning message as well as keep the user in the **Payment Terms** field and stop any other Microsoft Dynamics GP code from running.
- 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 runs in Microsoft Dynamics GP.
- Test the VBA code by viewing the Microsoft Dynamics GP window.

Need a Little Help?

- Use the following information to customize the Receivables Transaction Entry window:
- Navigate to the Receivables Transaction Entry window. (TRANSACTIONS→SALES→TRANSACTION ENTRY) If you completed the earlier exercise (Locking Fields Using VBA Code), the Receivables Transaction Entry window already exists in the VBA project. If you did not complete the earlier project, you need to add the following resources to the VBA project:
  - Receivables Transaction Entry window
  - **Payment Terms** field
- Using VBA, select the window or window field event that could be used to restrict the user from entering C.O.D. into the **Payment Terms** field. You will give them a warning message as well as keep the user in the **Payment Terms** field and stop any other Microsoft Dynamics GP code from running.
- Compile the code to verify it runs in Microsoft Dynamics GP. (DEBUG→COMPILE MICROSOFT_DYNAMICS_GP)
- Test the VBA code by viewing the Microsoft Dynamics GP window.

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

Lab 6.8 – Setting Drop-Down Fields

Company: Fabrikam, Inc.

Scenario: This exercise shows you how to use VBA code to set the value of a drop-down field on a Microsoft Dynamics GP window. This exercise works with the Purchase Order Entry window. Set the default of the **Type** field to display Drop Ship.

Challenge Yourself!

- Use the following information to customize the Purchase Order Entry window:

- Add the window and the appropriate field to the VBA project.
- Using VBA, select the window or window field event that could be used to set the **Type** field to Drop Ship.
- 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 runs in Microsoft Dynamics GP.
- Test the VBA code by viewing the Microsoft Dynamics GP window.

Need a Little Help?

- Use the following information to customize the Purchase Order Entry window:
- Navigate to the Purchase Order Entry window. (TRANSACTIONS→PURCHASING→PURCH. ORDER ENTRY) If you completed the earlier exercise (Opening Lookup Windows using VBA), the Purchase Order Entry window already exists in the VBA project. If you did not complete the earlier project, you need to add the following resources to the VBA project:
  - Purchase Order Entry window
  - **Type** field
- Using VBA, select the window or window field event that could be used to set the default value of the **Type** field to display Drop Ship.
- Compile the code to verify it runs in Microsoft Dynamics GP. (DEBUG COMPILE MICROSOFT_DYNAMICS_GP)
- Test the VBA code by viewing the Microsoft Dynamics GP window.

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

Lab 6.9 – Calculating and Setting Field Values

Company: Fabrikam, Inc.

Scenario: This exercise shows you how to use VBA code to calculate a percentage and set the value of a field on a Microsoft Dynamics GP window. This exercise works with the Receivables Transaction Entry window. If the Sales Amount is less than $50, the Freight should automatically calculate as 10% of the Sales Amount. If it is $50 or greater, the Freight should simply default as $5.

Challenge Yourself!

- Use the following information to customize the Receivables Transaction Entry window:

- Add the window and the appropriate field to the VBA project.
- Using VBA, select the window or window field event that could be used to calculate a percent and set the value of the **Freight** field when the Sales Amount is changed.
- 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 runs in Microsoft Dynamics GP.
- Test the VBA code by viewing the Microsoft Dynamics GP window.

Need a Little Help?

- Use the following information to customize the Receivables Transaction Entry window:
- Navigate to the Receivables Transaction Entry window. (TRANSACTIONS→SALES→TRANSACTION ENTRY) If you completed one of the earlier exercises working with the Receivables Transaction Entry window, it should already exist in the VBA project. If you did not complete the earlier project, add the following resources to the VBA project:
  - Receivables Transaction Entry window
  - **Sales Amount** field
  - **Freight** field
- Using VBA, select the window or window field event that could be used to perform the following calculation when the value of the **Sales Amount** field changes. When the value of the **Sales Amount** field is less than $50, calculate 10 percent of the Sales Amount, and set the value of the **Freight** field to that calculated value. If the Sales Amount is more than $50, set the value of the **Freight** field to $5.
- Compile the code to verify it runs in Microsoft Dynamics GP. (DEBUG→COMPILE MICROSOFT_DYNAMICS_GP
- Test the VBA code by viewing the Microsoft Dynamics GP window in run mode. Switch to Microsoft Dynamics GP by clicking the **View Microsoft Dynamics GP** button from the Toolbar. You can also select Microsoft Dynamics GP from the View menu.

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