CHAPTER 7: WORKING WITH GRIDS AND GRID FIELDS

Training Objectives

Actively participating during this chapter helps you learn:

- To add grids and grid fields to your Visual Basic for Applications (VBA) Project.
- Common methods and properties used for grids and grid fields.
- Common events used for grids and grid fields.

Overview

VBA refers to Microsoft Dynamics™ GP scrolling windows as "grids". Working with these differs from other types of windows in Microsoft Dynamics GP. Instead of only displaying a single record at a time, Microsoft Dynamics GP grids display multiple lines of information with each line representing one record in a corresponding Microsoft Dynamics GP table. Because the functionality provided with a grid is different than what is available with a window, VBA references each as separate objects.

NOTE: *You have the ability to expand a grid so that each line of the grid displays two or more rows of data for each record. You can expand and shrink a grid using the* ***Expand*** *and* ***Shrink*** *buttons usually found above the grid's scroll bar.*

Microsoft Dynamics GP supports three types of data grids:

- Browse-only – Allows users to view information but not change it. These are most often used in the application for lookup windows.
- Editable – Allows users to change the contents of a line, but does not allow the user to add a new line.
- Adds-allowed – Allows users to add lines of information to a grid along with the ability to edit existing records and are commonly used for transaction entry windows.

NOTE: *Microsoft Dynamics GP grids do not function like traditional grids found in VBA. VBA grids typically allow you to work with specified rows and cells within a grid. In comparison, with Microsoft Dynamics GP grids, you can only reference fields on the current line, and grid events occur only for that current line.*

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

Working with grids and grid fields in a VBA project involves each of the following processes:

- Adding grids and grid fields to your project.
- Using methods and properties to customize the way the grid displays or to change the business logic of the grid.
- Specifying the events used to execute the VBA code that you have written.

Adding Grids and Grid Fields to Your Project

To use grids and grid fields with VBA, first add the window that contains the grid to your Microsoft Dynamics GP project. When you add the window, it inserts both the window and the grid. Even though the two work together to display information in Microsoft Dynamics GP, within VBA they are considered two independent objects. After the grid has been added, add any fields from that grid that you may want to reference.

Adding Grids

To add a grid object to your project, switch to Microsoft Dynamics GP and display the window that contains the desired grid. From the Tools menu, select Customize, then select Add Current Window to Visual Basic. This automatically adds the currently active window and the window's grid to your Microsoft Dynamics GP project. When switching to the Visual Basic Editor, both the grid and window object display. It should look similar to the following illustration:

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

Removing Grids

To remove a window and grid object from your project, first remove all references to the window and grid within your VBA code. Next, switch to Microsoft Dynamics GP, and display the window to remove. On the Tools menu, point to Customize, and then select Remove Current Window from Visual Basic. The window, the grid, and all fields associated with the window and grid should then be removed from the project.

Adding Grid Fields

To add a grid field to your project, switch to Microsoft Dynamics GP, and display the window containing the grid. The grid must be active or available before adding a grid field to your VBA project. Use the **Browse** buttons to select a record to make the grid active. From the Tools menu, select Customize and then select Add Field to Visual Basic. This activates a field selection mode and changes the cursor to a selection pointer. Use your mouse to click on each grid field to use in the project. When finished adding fields, turn off field selection mode. This can be done by selecting Add Fields to Visual Basic from the Customize option on the Tools menu, by pressing the Esc key, or by switching to the Visual Basic Editor.

You only need to add fields from one line of a grid. References made to the grid field from your VBA project are for the line that currently has focus. Because of this it is not necessary to add the same field from multiple lines of the grid.

HINT: *Click the Expand button to add a field from the grid that is not currently displayed on the window. The expanded grid then shows multiple rows of information for each record.*

Using Methods and Properties

There are a number of different methods and properties available for grid fields. These are the same methods and properties available for window fields. The only difference is that VBA references the field as a field within a grid versus a field within a window. Some of these methods and properties are used later in this chapter. A complete listing can be found in "Appendix" of this manual or in the *Microsoft Dynamics GP VBA Developer's Guide*.

Using Events

There are a number of different events available for both Microsoft Dynamics GP grids and grid fields. For grid fields, the events are the same as available for window fields. For Microsoft Dynamics GP grid objects, there are some new events available.

Grid Events

Line Got Focus Events

Line Got Focus events occur when the user moves onto a line in a grid. These events are generally most helpful when checking the value of a field on the line that is gaining focus. The BeforeLineGotFocus event executes before any of the Microsoft Dynamics GP code that normally runs as focus moves onto the line. The AfterLineGotFocus event executes after the Microsoft Dynamics GP code. However, in contrast to Field Got Focus events, VBA does not allow you to cancel any of the Microsoft Dynamics GP code. Because of this, both the BeforeLineGotFocus and AfterLineGotFocus events perform in the same manner.

In the following example, if the **Item Number** field on the Invoice Entry window (TRANSACTIONS→SALES→INVOICE ENTRY) is not empty, the grid on the window automatically expands to show multiple lines of information for that item. This sample code could also be used for an AfterLineGotFocus event.

Line Lost Focus Events

Line Lost Focus events occur when the user moves off of a line in a grid. These events are generally most helpful when setting up the next line of the grid so it can accept new data. The BeforeLineLostFocus event executes before any of the Microsoft Dynamics GP code that normally runs as focus moves off of the line. The AfterLineLostFocus event executes after the Microsoft Dynamics GP code. However, in contrast to Field Lost Focus events, VBA does not allow you to cancel Microsoft Dynamics GP code that executes for a Line Lost Focus event. Because of this, both the BeforeLineLostFocus and AfterLineLostFocus events perform in the same manner.

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

In the following example, the grid on the Invoice Entry window (TRANSACTIONS→SALES→INVOICE ENTRY) automatically shrinks to show only one line of information for each record in the grid any time the user moves off of the current record.

```
Microsoft_Dynamics_GP - InvoiceEntryDetail (Code)
Grid                                   BeforeLineLostFocus

Private Sub Grid_BeforeLineLostFocus()
    'Shrink the grid when focus moves off of a line
    InvoiceEntry.ScrollingWindowExpandButton = 1
End Sub
```

BeforeLineChange Events

Line Change events occur when the user changes the contents of a line in an editable or adds-allowed grid and then moves focus off of that line. The BeforeLineChange event executes before any of the Microsoft Dynamics GP code that normally runs when the line is changed. This event is helpful to check the value of fields on a line and cancel the Microsoft Dynamics GP code from executing if certain criteria are not met. The Microsoft Dynamics GP processing is canceled by setting a parameter called KeepFocus to true. This also causes focus to remain on the current line of the grid.

In the following example, if the "Extended Price" of the item on the Invoice Entry window (TRANSACTIONS→SALES→INVOICE ENTRY) causes the total of the invoice to be greater than $1000, a message displays indicating that they are not authorized to enter a transaction of that amount. The Microsoft Dynamics GP processing then stops and focus remains on the current line.

```
Microsoft_Dynamics_GP - InvoiceEntryDetail (Code)
Grid                                   BeforeLineChange

Private Sub Grid_BeforeLineChange(KeepFocus As Boolean)
    'Check to see if the total will be over $1000
    If CCur(InvoiceEntry.Total) > 1000 Then
        'It is over $1000 give message and cancel processing
        KeepFocus = True
        MsgBox "You are not authorized to enter a transaction with a total over $1,000."
    End If
End Sub
```

AfterLineChange Event

Line Change events occur when the user changes the contents of a line in an editable or adds-allowed grid and then moves focus off of that line. The AfterLineChange event executes after any of the Microsoft Dynamics GP code that normally runs when the line is changed. This event can be used to perform other tasks in the window after the line item has been saved.

In the following example, if the subtotal of the invoice on the Invoice Entry window (TRANSACTIONS→SALES→INVOICE ENTRY) is greater than $100 after a line item has been saved, the Trade Discount is set to $10. If the Trade Discount is already greater than $10, the amount remains the same.

Microsoft_Dynamics_GP - InvoiceEntryDetail (Code)

Grid | AfterLineChange

```
Private Sub Grid_AfterLineChange()
'Check to see if the transaction is over $100 and if the trade discount is
'less than $100, set the discount to $10
  If CCur(InvoiceEntry.Subtotal) > 100 And _
    CCur(InvoiceEntry.TradeDiscount) < 10 Then
       InvoiceEntry.TradeDiscount = 10
    End If
End Sub
```

BeforeLinePopulate Event

The BeforeLinePopulate event occurs each time Microsoft Dynamics GP displays a new line in a grid. When Microsoft Dynamics GP initially displays a grid, the event occurs repeatedly until the grid is full. It also executes each time the user displays a new line in a grid by scrolling up or down in the window. This event is most helpful when filtering out certain lines of information so that they do not display within the grid. The record is filtered out of the grid by setting a parameter called RejectLine to true.

In the following example, the Items lookup window (TRANSACTIONS→SALES→INVOICE ENTRY→ITEM NUMBER LOOKUP) shows only items that have their Class ID set to "RETAIL." Since Lookup windows reside in the SmartList dictionary, this code is placed the SmartList project for the ItemsDetail code window.

SmartList - ItemsDetail (Code)

Grid | BeforeLinePopulate

```
Option Explicit

Private Sub Grid_BeforeLinePopulate(RejectLine As Boolean)
    'In the Items lookup window display only Retail items
    If ASICustomerItemNumber <> "RETAIL" Then
       RejectLine = True
    End If
End Sub
```

NOTE: *A more realistic version of this example allows the user to indicate which Class ID they want to use for the restriction. This can be done by either prompting the user for a Class ID or by adding an additional field to the Items lookup window. This additional field could be added using the Microsoft Dynamics GP Modifier.*

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

Conclusion

In this chapter we learned how to add grids and grid 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 grid.

Lab 7.1 – Setting Field Values in a Grid

Company: Fabrikam, Inc.

Scenario: This exercise shows you how to use VBA code to set a default value for a field in a Grid. This exercise works with the GL Transaction Entry window. When the user enters the Grid, the **Distribution Reference** field in the Grid should default to a value that is the combination of the Journal Entry Number and the transaction's Reference.

Challenge Yourself!

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

- Add the window and the appropriate fields to the VBA project.
- Using VBA, select the grid event that could be used to default the value of the **Distribution Reference (Description)** field as the user enter the grid. The new value will be a combination of the Journal Entry Number and the Reference.

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 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 GL Transaction Entry window:
- Navigate to the GL Transaction Entry window. (TRANSACTIONS→FINANCIAL→GENERAL) Add the following resources to the VBA project:
  - GL Transaction Entry window
  - **Journal Entry** field
  - **Reference** field
  - **Distribution Reference** (When you click on this field, the name that appears in the VBA code window is Description. The picture below illustrates the scrolling window opened with Modifier so you can see the actual field names.)

- Using VBA, select the grid event that could be used to set the value of a field in the Grid as the Users enters the grid. The value of the **Distribution Reference** field or **Description** field should be set to a combination of the Journal Entry Number and the Reference.
- 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.

Lab 7.2 – Restricting Data in Grid Fields

Company: Fabrikam, Inc.

Scenario: This exercise shows you how to use VBA code to restrict users from entering specific information into a field in a Grid for a Microsoft Dynamics GP window. This exercise works with the Invoice Entry window. If the line item's U of M is "Case," a message should be given indicating that they are not authorized to specify a unit of measure of that type. Focus should be set back to that field so another value can be entered.

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 grid field event that could be used to stop the user from entering Case in the **U of M** field. Display a message to the user and keep them in the same field so another value can be entered.
- 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.

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 Invoice Entry window:
- If you did not add the Invoice Entry window to the VBA exercise in a previous lesson, navigate to the Invoice Entry window. (TRANSACTIONS→SALES→INVOICE ENTRY) Add the following resources to the VBA project:
  - Invoice Entry window
  - **U of M** field in the Grid

- Using VBA, select the grid field event that could be used to stop the user from entering Case in the **U of M** field. Display a message to the user, and keep them in the same field so another value can be entered.
- 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.

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

Lab 7.3 – Setting Field Values on Different Lines in a Grid

Company: Fabrikam, Inc.

Scenario: This exercise shows you how to use VBA code to set a default value for a field in a Grid from a different line item. This exercise works with the GL Transaction Entry window. When the user enters the Grid, the account number should default to the account number that was entered for the previous line of the scrolling window. For example, as focus moves to the second line in the grid, the account number should automatically be set to the account number that was entered for the first line item.

Challenge Yourself!

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

- Add the window and the appropriate fields to the VBA project.
- Using VBA, select the grid event that could be used to default the value of the Account.
- Use this event along with the appropriate methods and properties to complete this task within your Microsoft Dynamics GP VBA project.

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

- For this exercise, you most likely need to use two different types of events to implement this enhancement in your VBA project. In addition, because you are using two different events, you need to declare a new variable in VBA in order to pass the default account number from one event to another. The variable can be declared using the DIM or Private statement – for example, DIM DefaultAcct As String.
- New Variables can be declared using the following syntax:
- Dim or Private located in the General section of the code window. The variable is then available to all Sub routines on the code window.
- Dim located within a Sub routine. The variable is available within that routine only and is initialized each time the event is run.
- Public located in the General section of a code window. The variable will be available to all routines in the project.
- Public located in the Sub routine. The variable will be available to the entire module.
- Static for a variable that will retain its value between procedures.
- 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 GL Transaction Entry window:
- Navigate to the GL Transaction Entry window. (TRANSACTIONS→FINANCIAL→GENERAL) If the window was not added in a previous project, add the following window and the other resources to the VBA project:
  - GL Transaction Entry window
  - **Account Number** field

- Using VBA, select the grid event that could be used to set the value of a field in the Grid as the Users enters the grid. The value of the **Account Number** field should default to the account number entered for the previous line item.

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

- For this exercise, you most likely need to use two different types of events to implement this enhancement in your VBA project. In addition, because you are using two different events, you need to declare a new variable in VBA in order to pass the default account number from one event to another. The variable can be declared using the DIM or Private statement – for example, DIM DefaultAcct As String.

  New Variables can be declared using the following syntax:
- Dim or Private located in the General section of the code window. The variable is then available to all Sub routines on the code window.
- Dim located within a Sub routine. The variable is available within that routine only and is initialized each time the event is run.
- Public located in the General section of a code window. The variable will be available to all routines in the project.
- Public located in the Sub routine. The variable will be available to the entire module.
- Static for a variable that will retain its value between procedures.
- Compile the code to verify it runs in Microsoft Dynamics GP. (DEBUG→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

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