CHAPTER 8: STORING ADDITIONAL DATA

Training Objectives

Actively participating during this chapter helps you learn:

- The DUOS Structure and Storage methods.
- Common methods and properties used for DUOS.
- Common events used for DUOS.
- To store, retrieve, save and update data in the DUOS table.

Dynamics User Object Store (DUOS) Overview

The Dynamics User Object Store (DUOS) allows you to use Visual Basic for Applications (VBA) to store and retrieve additional information used within Microsoft Dynamics™ GP. This can be data entered into existing Microsoft Dynamics GP fields and into new fields added using Modifier or into fields that exist on a new form created with VBA. Similarly, using the DUOS this information can be displayed on a Microsoft Dynamics GP window or on one of the new forms created with VBA.

NOTE: *There are other options for storing additional data rather than using DUOS. Use Open Database Connectivity (ODBC) drivers to accomplish this.*

Structure of the DUOS

If you are familiar with VBA technology, the Microsoft Dynamics GP VBA object model uses a combination of collections and objects to define DUOS data objects.

DUOSObjects collection contains one or more user-definable collections or sets of data created to store related information. This includes customer, vendor, or item collections. The DUOSObject object then represents a single object or record within a collection such as the customer object for "ADVANCED0001" within the customer's collection.

DUOSProperties collection contains all of the properties or fields that are available for each object within a collection. The DUOSProperty Object represents a single property for a single object within a collection. The following illustration shows the relationship of the collections and objects.

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

The following table shows an example of two user-defined collections that might exist within the DUOSObjects collection. The table includes a column for a Collection Name, Object ID, Property Name, and Property Value. The Collection Name specifies the user-defined collection that the data object belongs to. The Object ID is the unique value for the data object. The Property Name specifies all of the properties available for the data objects and the Property Value specifies the actual value for a given property.

"Customers".			Authorization Code	6995mailto:Jrossini@aol.com
A data object within the collection.		ADVANCED 0001	Email Address	mmonat@aol.com
			Authorization Code	9872
			Customer Pricing	RETAIL
	Employees	BANK0001	Date of Last Pay Increase	6/15/05
			Grade Level	6
		BURN0001	Grade Level	3

Storage of the DUOS

Microsoft Dynamics GP stores DUOS data objects in a table named SY_User_Object_Store (SY90000.*). The table is company-specific, so Microsoft Dynamics GP stores data objects created for different companies in separate tables. Each table is located in the COMPANY directory for the company it is associated with.

An example of the table is shown in the following illustration. The combination of the Collection Name, Object ID, Property Name, and Property Value make up a single record in this table with each component being a field within the record.

Collection Name	Object ID	Property Name	Property Value
Customers	ALTONMAN0001	Email Address	jrossini@aol.com
Customers	ALTONMAN0001	Authorization Code	6995
Customers	ADVANCED0001	Email Address	mmonat@aol.com
Customers	ADVANCED0001	Authorization Code	9872
Customers	ADVANCED0001	Customer Pricing	RETAIL

If two additional properties and property values are added to the ADVANCED0001 data object, Microsoft Dynamics GP adds two records to this table.

Collection Name	Object ID	Property Name	Property Value
Customers	ALTONMAN0001	Email Address	jrossini@aol.com
Customers	ALTONMAN0001	Authorization Code	6995
Customers	ADVANCED0001	Email Address	mmonat@aol.com
Customers	ADVANCED0001	Authorization Code	9872
Customers	ADVANCED0001	Customer Pricing	RETAIL
Customers	ADVANCED0001	Birth Date	08/08/61
Customers	ADVANCED0001	Receive Mailings	1

Using Methods and Properties for DUOS Objects

DUOS objects can be manipulated in different ways through the use of methods and properties. Methods are actions that can be performed on the DUOS object. Properties are characteristics that can be retrieved or set for a DUOS object. The methods and properties used most often are described in this section. Corresponding example code is also given. A complete listing of all methods and properties can be found in Appendix C of this manual or in the *Microsoft Dynamics GP VBA Developer's Guide*.

NOTE: *Methods and properties deal with how a DUOS object can be manipulated. Discussion about where the methods and properties can be used is covered later in this chapter in a section called "Using DUOS objects with Microsoft Dynamics GP."*

Declaring a Collection and Object

When working with DUOS, the Collection and Object must be declared before being recognized by methods and properties. The Collection is the container for the sets of data that is stored. The Object represents a single object or record within a collection.

If the Collection and Object are declared in the module's Declaration section, they will be available to all procedures within that module. An example of declaring the Collection and Object are listed below:

```
(General)                    (Declarations)
Option Explicit
Dim CustomerCollection As DUOSObjects
Dim CustomerObject As DUOSObject
```


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

DUOS Methods and Properties

Working with a Collection

When working with a DUOS data object, first specify the collection that contains the object. The DUOSObjectsGet method can be used to specify the collection. If the collection does not already exist, this method creates it.

The DUOSObjectGet method must be used prior to any of the other DUOS methods and properties and it must be declared before it can be used.

For example, use the following code within one of your VBA procedures to specify a collection named "Customers." This collection is used to store additional customer information data objects.

```
'Specify the collection that will contain the DUOS data
Set CustomerCollection = DUOSObjectsGet("Customers")
```

***NOTE**: Microsoft Dynamics GP provides its own Help files for VBA. For detailed information on DUOSObjectGet method highlight the word "DUOSObjectGet" and press the F1 key.*

Creating a Data Object

After a collection is specified for a data object, the data object can be assigned a unique ID or value using the collection's default property. If an object already exists with that ID, the collection uses the data object that already exists and updates it accordingly.

For example, the following code specifies a customer data object equal to the "Customer ID" on the current window. Depending on whether that value exists in the DUOS table, the property either creates a new object with that ID or retrieves an already existing object.

```
'Specify the collection that will contain the DUOS data object
Set CustomerCollection = DUOSObjectsGet("Customers")

'Specify the object
Set CustomerObject = CustomerCollection(CustomerID)
```

***NOTE**: Sample code found in the VBA Developer's Guide uses the .Item property to specify an object. For example, it can use the statement Set CustomerObject = CustomerCollection.Item(CustomerID) to specify the same object as shown in our example. The .Item property is the default property and is assumed if no other property is included in the statement. Because of this default functionality and to make the example code less complex, this training workbook does not include the additional .Item syntax.*

Setting an Objects Properties

Additional statements can then be used to create and assign the values of the properties or fields within the data object. The property's value can be set to a specific value directly in your VBA code or it can be set to the value of a window field.

The following example sets "Authorization Code" and "Email Address" properties to the value of two new fields called **Email Address** and **Authorization Code** that a user has added to an existing Microsoft Dynamics GP window.

```
'Specify the collection that will contain the DUOS data object
Set CustomerCollection = DUOSObjectsGet("Customers")

'Specify the object
Set CustomerObject = CustomerCollection(CustomerID)

'Set the objects properties
CustomerObject.Properties("Authorization Code") = AuthorizationCode
CustomerObject.Properties("Email Address") = EmailAddress
```

Retrieving a Data Object

Retrieving an object is similar to creating a data object. The collection and data object must be specified, and then the specific properties of the data object can be retrieved.

For example, in the following code, the customer data object equal to the Customer ID on the current window is retrieved, and two new fields on the window called **Email Address** and **Authorization Code** are set to their corresponding property values from the DUOS data object.

```
'Specify the collection
Set CustomerCollection = DUOSObjectsGet("Customers")

'Specify the object
Set CustomerObject = CustomerCollection(CustomerID)

'Set the window fields
AuthorizationCode = CustomerObject.Properties("Authorization Code ")
EmailAddress = CustomerObject.Properties("Email Address ")
```


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

Deleting a Data Object

A collection's Remove method can be used to remove the current data object from the DUOS table. For example, in the following code the customer data object equal to the Customer ID on the current window is deleted from the Customers' collection.

```
'Specify the collection
Set CustomerCollection = DUOSObjectsGet("Customers")

'Remove the data object
CustomerCollection.Remove (CustomerID)
```

Updating a Data Object

Update a DUOS data object using a procedure similar to one used for creating or retrieving a data object. Or globally update multiple data objects by looping through each object in a collection.

For example, the following code sets the "Authorization Code" of each customer data object in the "Customers' collection to the **Customer ID** field on the current window. This can easily be done by taking advantage of the VBA For..Next statement.

```
'Specify the collection
Set CustomerCollection = DUOSObjectsGet("Customers")

'For each object, set the authorization code to the Customer ID
For Each CustomerObject In CustomerCollection
    CustomerObject.Properties("Authorization Code") = CustomerID
Next
```

Using DUOS Objects with Microsoft Dynamics GP

Using the DUOS with Microsoft Dynamics GP typically involves adding fields to a Microsoft Dynamics GP window using Modifier and writing VBA code that saves, retrieves, and deletes a DUOS data object when the corresponding Microsoft Dynamics GP record is saved, retrieved, and deleted.

The following illustration shows the Microsoft Dynamics GP Customer Maintenance window (CARDS→SALES→CUSTOMER) with two additional fields added to the window using Modifier. These fields are **Email Address** and **Authorization Code**.

The examples in the remainder of this section deal with this modified Customer Maintenance window and how window and window field events can be used to execute code as Microsoft Dynamics GP saves, retrieves, and deletes records using this particular window.

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

Adding Fields to Microsoft Dynamics GP

Use Microsoft Dynamics GP Modifier to add fields to Microsoft Dynamics GP windows. For detailed information about adding new fields, refer to the "Customizing Windows Layouts" chapter of this manual.

Retrieving DUOS Objects when Microsoft Dynamics GP Records are Displayed

After saving a DUOS data object, you must write VBA code to retrieve it. The object needs to be retrieved whenever the Microsoft Dynamics GP record associated with it displays. In our example, the Microsoft Dynamics GP record is displayed after a value is entered into the **Customer ID** field. This is when you need to retrieve the data from your DUOS object. Do this by using the **Changed** event attached to the control field. Most of the VBA code is equivalent to the example shown earlier for retrieving DUOS data objects.

If the Collection and Object are declared in the module's Declaration section, they are available to all procedures within that module. The Collection and Object must be declared before it is recognized by any of the code in any other events. An example of declaring the Collection and Object are listed below:

The following is a complete example of how this code is used as part of the **Changed** event for the **Customer ID** field.

Saving DUOS Objects when Microsoft Dynamics GP Records are Saved

There are two situations where you must save a DUOS data object:

- When clicking the **Save** button in a Microsoft Dynamics GP window.
- When attempting to close the window or display another record without saving changes in the window.

Using the Microsoft Dynamics GP Save Button

When clicking the **Save** button in a Microsoft Dynamics GP window, the Microsoft Dynamics GP code attached to that field executes and saves the information on the window to the corresponding Microsoft Dynamics GP data table. This is one situation where you also need to save your DUOS data object. This is done by using the BeforeUserChanged event for the Microsoft Dynamics GP **Save** button field. The methods and properties included in this event are similar to the example code used earlier for creating or retrieving DUOS data objects.

HINT: *You generally do not want to use the AfterUserChanged to save DUOS information. This is because in most cases the values of your window fields have been cleared by the Microsoft Dynamics GP code that already executed.*

The following is a complete example of how this code is used as part of the BeforeUserChanged event.

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

Using the Microsoft Dynamics GP Save Dialog

When you change a record and then attempt to close the window or move to another record, the Microsoft Dynamics GP save dialog automatically appears asking if you want to save the current record. An example of this dialog appears in the following illustration:

If you click the **Save** button, Microsoft Dynamics GP code executes and saves the information on the window to the Microsoft Dynamics GP data table. This is the other situation when you must save your DUOS data object. This can be accomplished by using the AfterModalDialog event attached to the Microsoft Dynamics GP save dialog. Most of the VBA code is similar to the example given previously for the Microsoft Dynamics GP **Save** button. However, some additional statements are required to verify that the Microsoft Dynamics GP save dialog message is currently displayed.

The prompt string that you look for in the AfterModalDialog Event can differ depending on what caused the message to be displayed. For example, on the Customer Maintenance window, if you try to close the window before saving the changes, the message you will look for is the following: "Do you want to save changes?" Comparatively, the message you will look for when you want to display another record using the browse buttons is "Do you want to save changes to this customer?" You want to make sure your modal dialog event looks for all messages that deal with saving changes for the current record.

The following is a complete example of how this code is used as part of the AfterModalDialog event.

Microsoft_Dynamics_GP - CustomerMaintenance (Code)

Window | AfterModalDialog

```
Private Sub Window_AfterModalDialog(ByVal DlgType As DialogType, _
PromptString As String, Control1String As String, Control2String As String, _
Control3String As String, Answer As DialogCtrl)
   If PromptString = "Do you want to save changes to this customer?" Then
      If Answer = dcButton1 Then
         'Set the name and value for the object's properties
         CustomerObject.Properties("Authorization Code ") = AuthorizationCode
         CustomerObject.Properties("Email Address") = EmailAddress
      End If
   End If
End Sub
```

Deleting DUOS Objects When Microsoft Dynamics GP Records are Deleted

You should delete a DUOS data object when deleting its corresponding record in Microsoft Dynamics GP. This ensures that you will not have an "orphaned" DUOS data object after its related record is deleted in Microsoft Dynamics GP. However, instead of simply writing VBA code that deletes the DUOS object whenever you click the Microsoft Dynamics GP **Delete** button, you want to delete a DUOS data object based on the response to the Microsoft Dynamics GP messages displayed. For example, a Microsoft Dynamics GP delete dialog message is displayed when the **Delete** button is made available. This message is shown in the following illustration:

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

You can delete the Microsoft Dynamics GP record by clicking the **Delete** button, or cancel the operation by clicking the **Cancel** button. Delete your DUOS records only if **Delete** is selected on the dialog message. Similar to saving DUOS objects, you accomplish this by using the AfterModalDialog event. The code added to this event then tracks whether you selected the **Delete** button.

In some situations, there is also another issue to deal with when deleting a DUOS object. It pertains to an additional dialog that can appear if Microsoft Dynamics GP was unable to delete the record. For example, an additional message is displayed if a customer record has outstanding transactions corresponding to it. This dialog appears in the following illustration:

However, this second message must be handled differently than the delete dialog mentioned earlier. This is because there is no AfterModalDialog event available for the secondary message. It instead needs to be handled as a separate Microsoft Dynamics GP window and must be added to the VBA project. To add this window to the VBA project, make the dialog appear in Microsoft Dynamics GP and then use the keyboard equivalent (CTRL+F11) to "Add Current Window to Visual Basic." Then you attach an AfterOpen event to the window. This event includes code that determines whether this message prevented the Microsoft Dynamics GP record from being deleted.

Use a combination of AfterModalDialog and AfterOpen events to determine whether certain messages stopped the Microsoft Dynamics GP delete operation from completing. With the use of an AfterUserChanged event on the **Delete** button, you can execute code to delete your DUOS data objects only if the Microsoft Dynamics GP delete operation was not interrupted.

The following is a complete example of VBA code that could be used to delete DUOS objects that correspond to a Microsoft Dynamics GP customer record. This code is behind window and field events in the Customer Maintenance Code Window. There are three different variables used throughout the example code. These are declared as global variables and then initialized and set in the appropriate events.

Microsoft_Dynamics_GP - CustomerMaintenance (Code)

(General) (Declarations)

```
Option Explicit
Dim CustomerCollection As DUOSObjects
Dim CustomerObject As DUOSObject

Public OktoDelete As Boolean
Dim WanttoDelete As Boolean
Dim CustomerIDtoDelete As String
```

Microsoft_Dynamics_GP - CustomerMaintenance (Code)

CustomerID Changed

```
Private Sub CustomerID_Changed()
    Set CustomerCollection = DUOSObjectsGet("Customer Information ")
    If CustomerID.Empty = False Then
        'There is a new value field. Retrieve the DUOS object
        Set CustomerObject = CustomerCollection(CustomerID)

        'Update the window fields
        AuthorizationCode = CustomerObject.Properties("Authorization Code ")
        EmailAddress = CustomerObject.Properties("Email Address ")
    End If

    If CustomerID.Empty = False Then
        'Initialize variables
        CustomerIDtoDelete = CustomerID
        OktoDelete = True
        WanttoDelete = False
    End If
End Sub
```


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

Microsoft_Dynamics_GP - CustomerMaintenance (Code)

Window | AfterModalDialog

```
Private Sub Window_AfterModalDialog(ByVal DlgType As DialogType, _
PromptString As String, Control1String As String, Control2String As String, _
Control3String As String, Answer As DialogCtrl)
   If PromptString = "Do you want to save changes to this customer?" Then
      If Answer = dcButton1 Then
         'Set the name and value for the object's properties
         CustomerObject.Properties("Authorization Code ") = AuthorizationCode
         CustomerObject.Properties("Email Address") = EmailAddress
      End If
   End If

   If PromptString = "Are you sure you want to delete this customer record?" Then
      If Answer = dcButton1 Then
         'The user chose Delete on the delete dialog message
         WanttoDelete = True
      End If
   End If
End Sub
```

Microsoft_Dynamics_GP - CustomerMaintenance (Code)

Delete | AfterUserChanged

```
Private Sub Delete_AfterUserChanged()
   'They choose delete on dialog message and no transactions exist
   If WanttoDelete And OktoDelete Then
      'Remove the object
      CustomerCollection.Remove (CustomerIDtoDelete)
   End If
End Sub
```

This code deals with the Microsoft Dynamics GP Window. This window is a modal dialog window that appears to display a message indicating that transactions exist for this customer. When this window opens, its code runs BEFORE the "Delete_AfterUserChanged()" event.

Microsoft_Dynamics_GP - MicrosoftDynamicsGP (Code)

Window | AfterOpen

```
Option Explicit

Private Sub Window_AfterOpen()
   If CustomerMaintenance.IsLoaded Then
      CustomerMaintenance.OktoDelete = False
   End If
End Sub
```

Conclusion

In this chapter we learned how to store and retrieve additional information using DUOS tables. You learned about specific methods and properties that define how to work with collections and objects within the tables. We also reviewed which VBA events are used to save, retrieve and delete information from these tables when processes occur in a Microsoft Dynamics window. Keep in mind, DUOS tables are not the only way to store additional data. ODBC connections can be used such as ADO to store data in SQL tables. The events would be the same whether using DUOS or another ODBC connection.

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

Lab 8.1 – Adding Fields to a Microsoft Dynamics GP Window Using Modifier

Company: Fabrikam, Inc.

Scenario: In this exercise, two new local fields will be added to a Microsoft Dynamics GP window using Modifier. Once the fields have been added, VBA code can be used in the following exercises to store, retrieve, delete, and update information in the DUOS table. This exercise works with the Vendor Maintenance window.

Challenge Yourself!

- Navigate to the Vendor Maintenance window, and add two new local fields to the window so the window looks like the following illustration:

- Validate your changes by viewing the window within Modifier.
- Allow the appropriate users to access this customized window.

Need a Little Help?

- Navigate to the Vendor Maintenance window, and add two new local fields to the window. To access this window navigate to the Vendor Maintenance window within the application, and select Modify Current Window on the Tools menu. You can also open this window within Modifier using the Technical name, which is PM_Vendor_Maintenance.
- Use the Toolbox to create two new local string fields. Add the fields to the window layout.
- Use the static text tool to add text prompts for the new fields added.
- Validate your changes by viewing the window within Modifier.
- Return to the Microsoft Dynamics GP application, and grant security to the Modified window using the Security Setup window.

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

Lab 8.2 – Retrieving From the DUOS Table

Company: Fabrikam, Inc.

Scenario: In this exercise, the two new local fields that were added to the Microsoft Dynamics GP window are retrieved from the DUOS table whenever the corresponding Microsoft Dynamics GP record is displayed. This exercise works with the Vendor Maintenance window.

Challenge Yourself!

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

- Add the window and the appropriate fields to the VBA project.
- Using VBA, select the field event that could be used to retrieve the URL Address and the General Satisfaction information from the DUOS table when you enter a Vendor ID.
- 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.

- Please complete all exercises regarding DUOS and test each exercise when they are all complete.

NOTE: *The retrieve is done before the save to determine if the record exists. If no record is found, it is treated as a new blank record.*

Need a Little Help?

- Use the following information to customize the Vendor Maintenance window:
- Navigate to the Vendor Maintenance window. (CARDS→PURCHASING→VENDOR) Add the following resources to the VBA project:
  - Vendor Maintenance window
  - **URL Address** field
  - **General Satisfaction** field
  - **Vendor ID** field
- Using VBA, select the field event that could be used to retrieve the URL Address and the General Satisfaction information from the DUOS table when you enter a Vendor ID.
- Compile the code to verify it runs in Microsoft Dynamics GP. (DEBUG→COMPILE MICROSOFT_DYNAMICS_GP)
- Please complete all exercises regarding DUOS, and test each exercise when they are all complete.

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

Lab 8.3 – Saving From the DUOS Table

Company: Fabrikam, Inc.

Scenario: In this exercise, the two new local fields that were added to the Microsoft Dynamics GP window are saved to the DUOS table whenever the corresponding record is saved. This exercise works with the Vendor Maintenance window.

Challenge Yourself!

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

- Add the window and the appropriate fields to the VBA project.
- Using VBA, select the field event that could be used to save the URL Address and the General Satisfaction information to the DUOS table when you save a Vendor record.
- 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.
- Please complete all exercises regarding DUOS, and test each exercise when they are all complete.

Need a Little Help?

- Use the following information to customize the Vendor Maintenance window.
- Navigate to the Vendor Maintenance window. (CARDS→PURCHASING→VENDOR) Add the **Save** button to the VBA project.
- Using VBA, select the field event that could be used to save the URL Address and the General Satisfaction information to the DUOS table when you save a Vendor record. Remember to deal with all possible situations such as when a modal dialog asks you if they want to save the record.
- Compile the code to verify it runs in Microsoft Dynamics GP. (DEBUG→COMPILE MICROSOFT_DYNAMICS_GP)
- Please complete all exercises regarding DUOS, and test each exercise when they are all complete.

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

Lab 8.4 – Deleting From the DUOS Table

Company: Fabrikam, Inc.

Scenario: In this exercise, the two new local fields that were added to the Microsoft Dynamics GP window are removed from the DUOS table whenever the corresponding Microsoft Dynamics GP record is deleted. This exercise works with the Vendor Maintenance window.

Challenge Yourself!

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

- Add the window and the appropriate fields to the VBA project.
- Using VBA, select the field event that could be used to delete the URL Address and the General Satisfaction information to the DUOS table when you delete a Vendor record.
- 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.

- Please complete all exercises regarding DUOS, and test each exercise when they are all complete.

Need a Little Help?

- Use the following information to customize the Vendor Maintenance window.
- Navigate to the Vendor Maintenance window. (CARDS→PURCHASING→VENDOR) Add the **Delete** button to the VBA project.
- Using VBA, select the field and window event that could be used to delete the URL Address and the General Satisfaction information from the DUOS table when you delete a Vendor record. Remember to deal with the modal dialog window, which asks if you want to delete the record. You must add code behind this event rather than using an event for the **Delete** button. Only remove the record if you select the Delete option from the modal dialog window.
- Compile the code to verify it runs in Microsoft Dynamics GP. (DEBUG→COMPILE MICROSOFT_DYNAMICS_GP)
- Please complete all exercises regarding DUOS, and test each exercise when they are all complete.

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

Lab 8.5 – Selectively Deleting Records

Company: Fabrikam, Inc.

Scenario: In this exercise, if you select the option to delete the Vendor from the Vendor Maintenance window, the record will only be deleted if the General Satisfaction level is not Excellent. Write a warning message stating the record cannot be deleted if the General Satisfaction level is excellent, and stop Microsoft Dynamics GP code from running and deleting the record from the Microsoft Dynamics GP table. This exercise works with the Vendor Maintenance window.

Challenge Yourself!

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

- Add the window and the appropriate fields to the VBA project.
- Using VBA, select the field event that could be used to stop you from deleting a Vendor if the General Satisfaction level is Excellent. Write a warning 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.
- Since all exercises are now complete in the DUOS section, test each exercise.

Need a Little Help?

- Use the following information to customize the Vendor Maintenance window.
- Navigate to the Vendor Maintenance window. (CARDS→PURCHASING→VENDOR) If you have not added the **Delete** button in a previous project, add it to the VBA project.
- Using VBA, select the field event that could be used to stop anyone from deleting a Vendor if the General Satisfaction level is Excellent. Write a warning message. Use the event that has the Cancel Logic and Keep Focus parameters to stop Microsoft Dynamics GP code from running and deleting the Vendor.
- Compile the code to verify it runs in Microsoft Dynamics GP. (DEBUG→COMPILE MICROSOFT_DYNAMICS_GP)
- Since all exercises are now complete in the DUOS section, test each exercise.

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