CHAPTER 10: ACCESSING DATA USING ODBC

Training Objectives

Actively participating during this chapter helps you learn:

- Setup and Installation of ODBC Drivers.
- Setup and Installation of the Data Access Methods.
- How to retrieve and update data in the Microsoft Dynamics™ GP tables using ADO

Overview

If a developer decides not to use the Microsoft Dynamics GP window interface to retrieve or save data or wants to access data that cannot be viewed in a Microsoft Dynamics GP window, they must use Open Database Connectivity (ODBC) drivers. ODBC drivers allow you to define ODBC data sources that applications, such as Crystal Reports, Microsoft® Access, Microsoft® Visual Basic®, or Microsoft Dynamics GP, to access data directly in ODBC – compatible tables. From Visual Basic for Applications (VBA), a data access object library is needed to write code that references an ODBC data source.

There are several types of data access object libraries available to the VBA developer for using ODBC drivers. They include:

- Microsoft® ActiveX® Data Objects (ADO). This is the most common method.
- Data Access Objects (DAO). Microsoft's DAO object libraries are typically installed as part of Microsoft® Office Professional. If you use DAO as your data access object, each user that needs to run this VBA code will need to have Microsoft Office Professional installed. There is no standalone installation of DAO.
- Remote Data Objects (RDO)Microsoft's RDO object library is automatically installed by some versions of Visual Basic. Although RDO can be freely used within standalone Visual Basic applications, it is not licensed for use in VBA applications that run on machine where full Visual Basic is not installed.

ODBC is an option that can be used instead of DUOS. There are specific benefits and drawbacks to both, be sure to research which option works best for you.

Since the most common method of accessing data through ODBC drivers is ADO, this chapter will discuss reading and writing SQL data using Microsoft's ADO and the Microsoft® SQL Server® ODBC drivers.

NOTE: *This document is intended as a starter document. More information can be found in the Visual Basic Help files and Microsoft SDK help files or any number of technical reference materials available from a computer bookseller.*

Setup and Installation of ODBC Drivers

When using Microsoft Dynamics GP with Microsoft SQL Server, the ODBC driver and data source are already set up during installation of the application.

Setup and Installation of Data Access Method

After the ODBC driver is installed and your ODBC data source for that driver is defined, you can access the data from your VBA application. There are different data access methods available but none of these are created by Microsoft Dynamics GP. Microsoft Dynamics GP recommends using the ADO object library method of accessing data. With the Microsoft Dynamics GP product, Microsoft's Data Access Controls (MDAC) is automatically installed, which contains the ADO object libraries.

NOTE: *Because these data access methods are Microsoft technologies, developers need to go to alternative sources of information such as the Visual Basic Help files and Microsoft SDK help files for more detailed information on how to use ADO.*

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

Adding a Reference

To use ADO methods and properties, the developer and the end users must have the Active Data Objects library installed on their computers. ADO is available from Microsoft's Web site as part of MDAC and can be distributed freely.

After the object library is installed, add a reference to it in your VBA project. To do this, select TOOLS→REFERENCES from the Visual Basic Editor. This opens the References dialog box.

Find your external application's object in the Available References list. Check the box to activate it, and make it accessible. If the object library you want to use is not listed, add the reference to the list using the Browse button.

NOTE: *The Available References list includes all object libraries installed on the computer. It is important to remember that not all libraries exist on all machines. Therefore, it is necessary to select the most recent version of a library that is common to all machines that will be running the software.*

ADO Definitions

Connection	An object variable that represents a connection to the data source. It allows you to execute commands (via the Execute method), such as a SQL command or a statement that returns data. If the command returns rows of data, a Recordset object is automatically created for you.
Recordset	An object variable that represents a set of rows returned from a database query.
Command	An object variable that represents a command processed by the data source.
Execute	A method used to execute the statements in the ADO Command object.
Transact-SQL	The standard language for communicating between applications and Microsoft SQL Server. The Transact-SQL language is an enhancement to Structured Query Language (SQL), the ANSI-standard relational database language.
Open	The command used to open a recordset.

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

Conclusion

In this chapter you learned that ODBC drivers can be used to retrieve and update data in Microsoft Dynamics GP tables and that ADO is the preferred method of data access.

Lab 10.1 – Adding Local Fields to a Microsoft Dynamics GP Window Using Modifier

Company: Fabrikam, Inc.

Scenario: The next two exercises demonstrate how to add more fields to a Microsoft Dynamics GP window and populate them with information from another table within Microsoft Dynamics GP using ADO methods. Add data from the Item Master table (IV00101) to the Item Transaction Inquiry window.

Since SQL statements and ODBC will be used, log in as "sa" for all exercises in this section. For the first exercise, two additional fields and a push button are added to the Microsoft Dynamics GP Item Transaction Inquiry window. This is done using Modifier.

Challenge Yourself!

- Use the following information to customize the Item Transaction Inquiry:

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

- Using Modifier, add two new fields under the **Description** field, and a push button to the status area of the window.
- Add a text prompt for the fields and link the text prompt with the associated field.
- Make the **Short Name** field non-editable.

Need a Little Help?

- Use the following information to customize the Item Transaction Inquiry. INQUIRY→INVENTORY→ITEM TRANSACTION.
- Create three new local fields and add them to the window layout . Two new fields are added under the **Description** field, and a push button is added in the status area of the window. The window should look like the following illustration:
  - **Short Name**
  - **Standard Cost**
  - **PB Update**
- Add static text for the two new fields, and change the properties so the underline appears on the static text.
- Link the Text Prompts with their associated fields.
- Change the property for the **Short Name** field to be editable = false.
- Preview the window, if satisfied with the changes set security to this modified window.

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

Lab 10.2 – Populating Local Fields with Data Using ADO and VBA Code

Company: Fabrikam, Inc.

Scenario: This exercise demonstrates how to populate the local fields added in the last exercise with information from another table within Microsoft Dynamics GP using ADO methods. Specifically, you must add data from the Item Master table (IV00101) to the Item Transaction Inquiry window.

Since SQL statements and ODBC will be used, log in as "sa" for all exercises in this section.

In this exercise, an example of ADO methods is used in the IV Item Transaction Inquiry window. The form is changed with Modifier to add the Short Name and Standard Cost for the item. VBA code is written to retrieve this data and display it on the Inquiry window. There is also a button to allow you to update the Standard Cost information from the Inquiry Screen. Typically, you do not have an Update button on an Inquiry Window. We are demonstrating this only for the purpose of showing the update functionality.

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

Challenge Yourself!

- Add the window and the appropriate fields to the VBA project. Add code so you can change the standard cost and update the data in the table using the **Update Standard Cost** button.

- Mark a reference to the ADO library.
- Using VBA, declare variables used for objects from external applications.
- 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?

- Add the window and the appropriate fields to the VBA project.
  - **Item Number** field
  - **Short Name** field
  - **Standard Cost** field
  - **Update Standard Cost** button

- Mark a reference to the ADO library. TOOLS→REFERENCES
- Using VBA, declare variables used for objects from external applications. Create the objects you need using the Set statements inside the VBA procedure. Declare variables to use ADO methods in the IV Item Transaction Inquiry window. These variables are declared in the General Declaration section of the IV Item Transaction Inquiry window.
- Write code to establish a connection to the database through the ODBC data source. Only execute this once within a session, if possible. Use the BeforeOpen event of the Window.
- Write code to close the ODBC connection. This connection is closed in the AfterClose event of the Window.
- After defining the database connection and closing the connection, write code to retrieve the data on the screen. This code is placed in the Changed event of the Item Number.
- Add code to update the Standard Cost information. This code is placed in the BeforeUserChanged event of the Update **Standard Cost** button.
- 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

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