

MS Access LAB 7

Topic: Macros, Modules, Data Access Pages and Internet

Summary

- Macros: How to Create and Run
- Modules
- Macros vs Modules
- Jumping to Internet

Macros: What they are and how they work

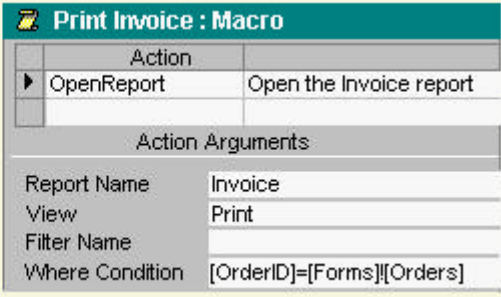
What is a macro?

A macro is a set of one or more actions that each perform a particular operation, such as opening a form or printing a report. Macros can help you to automate common tasks. For example, you can run a macro that prints a report when a user clicks a command button.

A macro can be one macro composed of a sequence of actions, or it can be a macro group. You can also use a conditional expression¹ to determine whether in some cases an action will be carried out when a macro runs.

When you create a macro, you enter the actions you want to carry out in this portion of the Macro window.

You can specify arguments for an action in this portion of the window. Arguments provide additional information on how to carry out the action, such as which object or data to use.



Print Invoice : Macro	
Action	
OpenReport	Open the Invoice report
Action Arguments	
Report Name	Invoice
View	Print
Filter Name	
Where Condition	[OrderID]=[Forms]![Orders]

Create a macro

- 1 In the Database window, click the Macros tab.
- 2 Click New.
- 3 In the Action column, click in the first cell and then click the arrow to display the action list.
- 4 Click the action you want to use.
- 5 Type a comment for the action. Comments are optional, but make your macro easier to understand and maintain.
- 6 In the lower part of the window, specify arguments² for the action, if any are required.

¹ conditional expression

An expression that Microsoft Access evaluates and compares to a specific value ³4 for example, If...Then and Select Case statements. If the condition is met, one or more operations are carried out. If the condition isn't met, Microsoft Access skips the operations associated with the expression and moves to the next expression. You can use conditional expressions in macros and Visual Basic code.

² argument

A constant, variable, or expression that supplies information to an action, event, method, property, or procedure.

- 7 To add more actions to the macro, move to another action row, and repeat steps 3 through 6. Microsoft Access carries out the actions in the order you list them.





Tip To quickly create a macro that carries out an action on a specific database object³, drag the object from the Database window to an action row in the Macro window. For example, you can create a macro that opens a form by dragging the form to an action row. To do this, click Tile Vertically on the Windows menu to position the Macro window and Database window so that they're both visible on your screen; then click the tab in the Database window for the type of object you want to drag, click the object, and drag it to an action row. Dragging a macro adds an action that runs the macro, while dragging other objects (tables, queries, forms, reports, or modules) adds an action that opens the object.

Run a macro

When you run a macro, Microsoft Access starts at the beginning of the macro and carries out all the actions in the macro until it reaches either another macro (if the macro is in a macro group) or the end of the macro.

You can run a macro directly, from another macro or an event procedure⁴, or in response to an event⁵ that occurs on a form, report, or control. For example, you can attach a macro to a command button on a form so that the macro runs when a user clicks the button. You can also create a custom menu command or toolbar button that runs a macro, assign a macro to a key combination, or run a macro automatically when you open a database.

Create a command button running a Macro

- 1 Open a form in Design view.
- 2 Click the Control Wizards tool  in the toolbox if it's pressed in. This turns off the wizard.
- 3 In the toolbox, click the Command Button tool .
- 4 On the form, click where you want to place the command button.
- 5 Make sure the command button is selected, and then click Properties  on the toolbar to open its property sheet.
- 6 In the OnClick property box, enter the name of the macro or event procedure that you want to run when the button is clicked, or click the Build button  to use the Macro Builder or Code Builder.
- 7 If you want to display text on the command button, type the text in the Caption property box. If you don't want text on the button, you can use a picture instead.

³ database objects

Tables, queries, forms, reports, macros, and modules.

⁴ event procedure

A procedure automatically executed in response to an event initiated by the user or program code, or triggered by the system.

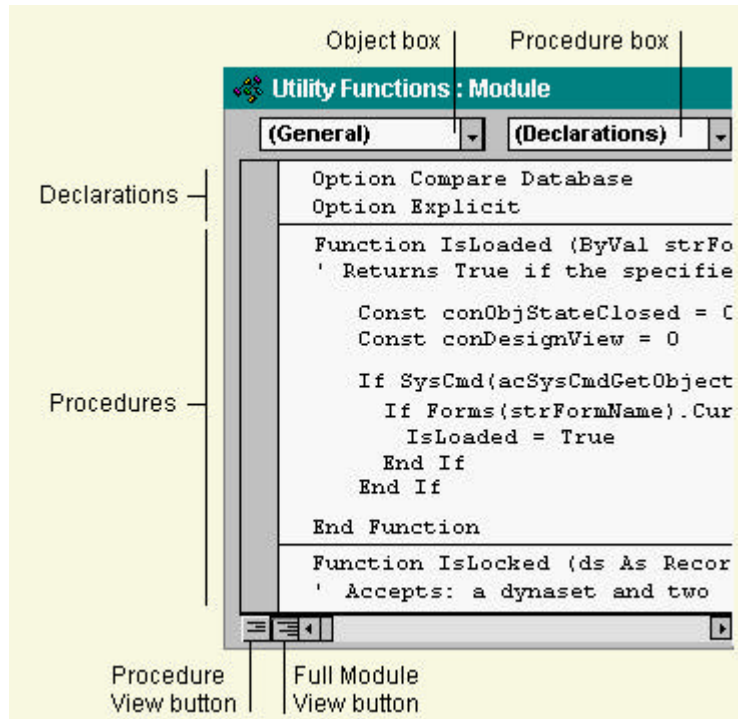
⁵ event

An action recognized by an object, such as a mouse click or key press, for which you can define a response. An event can be caused by a user action or a Visual Basic statement, or it can be triggered by the system. Using properties associated with events, you can tell Microsoft Access to run a macro, call a Visual Basic function, or run an event procedure in response to an event.

Modules: What they are and how they work

What is a module? A module is a collection of Visual Basic for Applications declarations and procedures that are stored together as a unit.

There are two basic types of modules: class modules⁶ and standard modules⁷. Each procedure in a module can be a Function procedure⁸ or a Sub procedure⁹.



Should I use a macro or Visual Basic?

⁶ class module

A module that can contain the definition for a new object. When you create a new instance of a class, you create the new object. Any procedures defined in the module become the properties and methods of the object. Class modules in Microsoft Access exist both independently and in association with forms and reports.

⁷ standard module

A module in which you can place Sub and Function procedures that you want to be available to other procedures throughout your database.

⁸ Function procedure

A procedure that returns a value and that can be used in an expression. You declare a function with the Function statement and end it with the End Function statement.

⁹ Sub procedure

A procedure that carries out an operation. Unlike a Function procedure, a Sub procedure doesn't return a value. You declare a Sub procedure with the Sub keyword and end it with an End Sub statement.

In Microsoft Access, you can accomplish many tasks with macros or through the user interface. In many other database programs, the same tasks require programming. Whether to use a macro or Visual Basic for Applications often depends on what you want to do.

When should I use a macro?

Macros are an easy way to take care of simple details such as opening and closing forms, showing and hiding toolbars, and running reports. You can quickly and easily tie together the database objects you've created because there's little syntax to remember; the arguments for each action are displayed in the lower part of the Macro window.

In addition to the ease of use macros provide, you must use macros to:

- Make global key assignments.
- Carry out an action or series of actions when a database first opens. However, you can use the Startup dialog box to cause certain things to occur when a database opens, such as open a form.

When should I use Visual Basic?

You should use Visual Basic instead of macros if you want to:

- Make your database easier to maintain. Because macros are separate objects from the forms and reports that use them, a database containing many macros that respond to events on forms and reports can be difficult to maintain. In contrast, Visual Basic event procedures are built into the form's or report's definition. If you move a form or report from one database to another, the event procedures built into the form or report move with it.
- Create your own functions. Microsoft Access includes many built-in functions, such as the IPmt function, which calculates an interest payment. You can use these functions to perform calculations without having to create complicated expressions. Using Visual Basic, you can also create your own functions either to perform calculations that exceed the capability of an expression or to replace complex expressions. In addition, you can use the functions you create in expressions to apply a common operation to more than one object.
- Mask error messages. When something unexpected happens while a user is working with your database, and Microsoft Access displays an error message, the message can be quite mysterious to the user, especially if the user isn't familiar with Microsoft Access. Using Visual Basic, you can detect the error when it occurs and either display your own message or take some action.
- Create or manipulate objects. In most cases, you'll find that it's easiest to create and modify an object in that object's Design view. In some situations, however, you may want to manipulate the definition of an object in code. Using Visual Basic, you can manipulate all the objects in a database, as well as the database itself.
- Perform system-level actions. You can carry out the RunApp action in a macro to run another Windows-based or MS-DOS-based application from your application, but you can't use a macro to do much else outside Microsoft Access. Using Visual Basic, you can check to see if a file exists on the system, use Automation or dynamic data exchange (DDE) to communicate with other Windows-based applications such as Microsoft Excel, and call functions in Windows dynamic-link libraries (DLLs).
- Manipulate records one at a time. You can use Visual Basic to step through a set of records one record at a time and perform an operation on each record. In contrast, macros work with entire sets of records at once.
- Pass arguments to your Visual Basic procedures. You can set arguments for macro actions in the lower part of the Macro window when you create the macro, but you can't change them

when the macro is running. With Visual Basic, however, you can pass arguments to your code at the time it is run or you can use variables for arguments — something you can't do in macros. This gives you a great deal of flexibility in how your Visual Basic procedures run.

Jumping to the Internet

There are three types of Web pages Microsoft Access creates:

- Data access pages
- Server-generated HTML
- Static HTML

Data access pages

You create a data access pages as a database object that contains a shortcut to the location of the page's corresponding HTML file. Learn about creating data access pages.

Use pages to view, edit, update, delete, filter, group, and sort live data from either a Microsoft Access database or a Microsoft SQL Server database, in Microsoft Internet Explorer 5 or later. A page can also contain additional controls called including a **spreadsheet**, a **PivotTable** list, and a **chart**.

To make your pages available on the World Wide Web, you publish the pages to Web Folders or a Web server, and make the Access database or SQL Server database available to users of the page. Internet Explorer needs to download the page only once from the Web server to let you view and interact with the data on the page. Because a page uses Dynamic HTML, access to the database is generally very efficient in a client/server environment.

Server-generated HTML

You can create server-generated HTML files, either ASP or IDC/HTX, from tables, queries, and forms. Server-generated HTML files are displayed in a table format in a Web browser. Use server-generated HTML files when you want to use any Web browser, your data changes frequently, you need to see live data in a table connected to an ODBC data source, but you only need to see read-only data.

Once you output ASP or IDC/HTX files, you must publish these files to be processed on a supported Web server product and platform.

Each time a user opens or refreshes an ASP or HTX file from a Web browser, the Web server dynamically creates an HTML file, and then sends that HTML file to the Web browser.

Static HTML

You can create static HTML files from tables, queries, forms, and reports. In a Web browser, reports display in a report format, and tables, queries, and forms display in a datasheet format. Use static HTML files when you want to use any Web browser that supports HTML version 3.2 or later and your data does not change frequently.

To make your static HTML files available on the World Wide Web, you publish the files to Web Folders or a Web server.

When you access the data through a Web browser, the browser only needs to download the static HTML file once from the Web server to let you view the data. However, the resulting HTML files are a snapshot of the data at the time you published your files. There is no ODBC data source connected to the static HTML file, and if your data changes, you must export your files again to be able to view new data in a Web browser.

Note To learn more about these 3 different web pages, go to Help and type in “Publish Web Pages”