Access Macros & Advanced Topics

Technical Support Services
Office of Information Technology, West Virginia University

OIT Help Desk – 293-4444 x 1

http://oit.wvu.edu/support/training/classmat/db/

Instructor: Kathy Fletcher
Last revised: July 14, 2005

Copyright 2004 West Virginia University
# Table of Contents

**Course Description** ............................................................................................................. 1  
**User Interface** ....................................................................................................................... 2  
**Automation: Changing the User Interface** ............................................................................. 3  
  - Changing the interface without programming ................................................................. 4  
  - Database Wizards ............................................................................................................. 5  
    - Activity 37: Using the Database Wizard ........................................................................ 5  
  - Command buttons ............................................................................................................ 6  
    - Activity 38: Command Buttons .................................................................................... 6  
  - Hyperlinks ....................................................................................................................... 7  
    - Activity 39: Inserting a Hyperlink ............................................................................... 7  
  - Switchboards .................................................................................................................. 8  
    - Activity 40: Switchboard Manager ............................................................................. 8  
**Controlling User Access** ....................................................................................................... 9  
  - Startup Properties ......................................................................................................... 9  
  - Password ....................................................................................................................... 10  
    - Activity 41: User Access ............................................................................................. 13  
**Objects, Events, and Properties** ............................................................................................ 14  
**Macros Environment** ......................................................................................................... 15  
**Macro Utilities** ................................................................................................................... 16  
**Running Macros** ................................................................................................................ 17  
**Macro Conditions** ............................................................................................................... 18  
**Error Handling** .................................................................................................................. 19  
**Working with Macros** ......................................................................................................... 20  
**Advanced Features** .......................................................................................................... 25  
  - Creating Indexes ............................................................................................................ 25  
  - Viewing or Editing Index Settings .................................................................................. 26  
  - Analysis tools ................................................................................................................ 26  
  - Sharing databases .......................................................................................................... 27  
  - Backing up a database .................................................................................................... 29  
  - Encrypting a database .................................................................................................... 30  
  - Repairing a database ...................................................................................................... 31  
  - Converting databases .................................................................................................... 31
Course Description

This is the final workshop in the Access XP series. In this workshop we will discuss macros. If time permits, we will also cover advanced topics that didn’t fit into the previous workshops.

Macros are generally used to automate tasks and functions within your database. They can be used to open or close objects, update data, or complete several tasks in sequence. Macros are not needed to have a well designed and functional database. They are generally created as part of the last stage of your database creation.

This workshop will familiarize you with the macro environment and how to create and use them within your database.

Our goal is to assist you to learn the software, understand some basic concepts and show you some tips and techniques so you can develop your database management/programming skills over time.

The five classes are:

- Introduction to Access
- Access Queries
- Access Reports
- Access Database Design
- Access Form Design
- Access Macros

Thank you,
The OIT Technical Support Services Trainers
West Virginia University
User Interface

Technically, the user interface is the view of the system that the user has to complete required system tasks. In real terms, that means that it is what the users see and have to work with when they are working on the system.

In Access, the user can work directly with the tables and queries. Data can be entered directly into the tables and queries can be created in the QBE Grid screen. However, if the user lacks an in-depth knowledge of the system and Access, then there is a high risk that the user will accidentally damage the system. In this scenario, the system designer has little or no automation responsibilities since the user can normally make routine changes or adjustments to the system in a reasonable time.

At the other extreme, the system can be fully automated so that the user only has the responsibility to start the system, select preprogrammed features, and enter data when prompted. The user needs little or no knowledge of the system or of Access. The system is nearly fail-safe since the high level of automation prevents the user from accidentally damaging the data files. However, the designer’s role has increased tremendously. The designer has the responsibility to fully automate the system and is then responsible for all system changes or adjustments. Such modifications will take the maximum amount of time since they also have to be fully automated and integrated into the system.

In reality, your system will probably fall somewhere between these two extremes. It will have a degree of automation that is acceptable to both the user and the designer. The system will minimize the efforts of the user in day-to-day operations, yet it will contain some routine protections. However, unusual actions may still result in system failure. The designer will be required to put forth a reasonable amount of effort to automate the more frequently used procedures in the system, but both the designer and the user will share responsibility for system changes and adjustments.

In most automated systems, the user's view of the system is normally limited to the database objects called forms. Some form objects may be used to provide data entry, modification, and browsing of the underlying tables. These objects may include some special features or commands that allow the execution of routine queries, the switching to other forms, and the production of reports.

Other form objects may be developed for the sole purpose of controlling the system and may function completely independently of the data tables. Therefore, the automation of a system is generally considered to be the development of a set of forms that will manage all of the desired functions of the system. Depending on the desired degree of automation, these forms will invoke special features of Access, macros, or Visual Basic modules to assist in the automation process.

These forms are, in fact, the main user interface. So in most cases, when we refer to the automation of a database system, we are actually talking about developing a set of forms that will both control the system and manage the data within the system. A secondary function of the automation process is the development of a set of standardized reports that are produced when a button is clicked on a form.
Automation: Changing the User Interface

An Access database consists of six objects; Tables, Queries, Forms, Reports, Macros, and Modules. If you are using only Tables, Queries, Forms, and Reports in your database system, then you are ignoring a third of the features in Access. And the features that you are ignoring probably contain half of the database power.

Without Macros and/or Modules, the database is interactive but not automated. The user:
  - Initiates actions by choosing general menu items or toolbar buttons.
  - Has to know which command to select.
  - Has to select the tasks in the proper order.
  - Has to know the function of each object.
  - Supplies the connection between objects.
  - Accepts responsibility for any damage or corruption to the database or the data.

With Macros and/or Modules, the database becomes an automated application. The designer creates lists of instructions and arranges for the computer to execute the entire list when the user chooses a custom command or button. The user doesn't require any knowledge of the system beyond normal operator functions.

Automation of an Access database can occur at three different levels:
  - Using some of the built-in features of Access to automate without programming.
  - Placing plain language commands in a list and designating the list as a macro.
  - Writing modules of Visual Basic like computer programming commands and triggering the modules to execute when an event occurs.
Changing the interface without programming

There are several very powerful features built into Access that allows you to improve the user’s view of the system without developing macros or modules. These features will reduce the complexity for the user without placing an extreme burden on the designer. Included in this list of features is the Database Wizard, the Command Button Wizard, Hyperlinks, and the Switchboard Manager.

The Database Wizard automatically creates database systems, sometimes called applications, from built-in templates. In some cases, the templates may not match exactly with the application that you need to develop. In these cases, select the closest template and customize the database later.

Depending on the application, the Wizard may create:

- Tables where the data will be stored
- Relationships between tables
- Any queries that are needed.
- Data entry forms for each table and it may create mainform/subform combinations as necessary.
- Switchboards that are used to navigate through the forms and reports. These switchboards are forms that contain command buttons programmed to perform some special action when they are clicked. That action might be to run a report, open another form, or close the application.
- Visual Basic code as necessary: the switchboard buttons are command objects. That is, they are controlled by blocks of Visual Basic for Applications (VBA) code contained in a module. The modules created by the Wizard can take two forms:
  - Standard modules that are listed under the module tab in the database window and are available to any object in the database
  - Individual procedures that are attached to the specific form or report they support
Database Wizards

A database wizard can be used to create a database based on templates that are included with Access. This wizard creates a database based on answer and choices that you make. Once the database is created, it can be modified in any way to better meet your needs.

Activity 37: Using the Database Wizard
1. Select File > New
2. Click on General Templates
3. Click on the Database tab
4. Click on Expenses
5. Click on OK
6. Navigate to the AccessXP folder
7. Enter a file name
8. Click on Create
9. Click Next on the first screen of the wizard
10. Choose which fields you want to include in the database by adding and/or removing check marks from boxes next to field names
11. Click on Next
12. Choose a style for the database screen displays
13. Click Next
14. Choose a style to be used in reports
15. Click on Next
16. Enter a title for the database
17. Click on Next
18. Make sure that Start the Database is checked
19. Click on Finish

Access will create the database and then display the main menu for you.
Command buttons
The buttons on the switchboard in the previous example are form objects commonly known as command buttons. When clicked, the command button executes a macro or a block of VBA code to perform the desired action. Command buttons are not limited to switchboards; they can be placed in any Access form either manually or through the use of the Command Button Wizard. The Command Button Wizard allows the designer to select from a set of pre-programmed actions without requiring that the designer be an experienced programmer. If, however, the designer is in fact an experienced programmer, then he/she is free to develop a unique set of commands (either macro or VBA) that can be assigned to buttons that are designed without the help of the wizard.

Activity 38: Command Buttons
1. On the main switchboard on the Expenses database that you created in the previous activity, click on Enter/View Expense Reports by Employee
2. Switch to Design View
3. Make sure that the Control Wizards button is depressed (Wizards are on)
4. Click on the Command Button icon
5. In the form footer area, click and drag to draw a command button on the form
6. Choose Form Operations from the Category list and Close Form in the Action list
7. Click on Next
8. Click on the radio button next to Text and enter the text that you want to display on the button
9. Click on Next
10. Enter the name CmdClose for the button
11. Click on Finish
12. Save the form
13. Switch to Datasheet View and test the button
Hyperlinks

A Hyperlink is a piece of text, an image, or a command button that you click to move to another location. The target of the hyperlink could be:

- Another object in your database
- Another file on your computer
- Another file on a computer on your local area network (LAN) that allows file sharing
- A specified sub-location in a MS Office file, including a Word bookmark, an Excel spreadsheet, a PowerPoint slide, or an Access object or a web page
- A file on the Internet or on the company Intranet

The hyperlink, which is normally embedded in a form, must be in the correct format for the desired link. Hyperlinks can be stored in a data table using the hyperlink data type. This data type lets you store information in three parts, separated by # signs:

- Display text - The text that displays in the field
- Address - The hyperlink address (path or URL)
- Sub-address - The hyperlink sub-address

Each of the three parts can contain up to 2048 characters. The easy way to insert a hyperlink into a field or control is to position the cursor in the field and click on Hyperlink in the Insert menu.

Activity 39: Inserting a Hyperlink

1. Open the Enter/View Expense Reports by Employee form in Design View
2. Choose Insert > Hyperlink from the menus
3. In the Text to display: area, enter the text that you would like to appear on the form
4. In the Address: area, enter http://www.wvu.edu/~support/
5. Click on OK
6. Drag the hyperlink to the desired location on the form
7. Save the form and switch to Datasheet View

Note: This Activity inserts a hyperlink that will appear the same on all records. To insert a different hyperlink for each record, a field would need to be added with a data type of hyperlink to a table that the form is based on
Switchboards
When we created the database using the database wizard, it automatically created a switchboard that provides easy navigation between the forms and reports in the database. However, like command buttons, switchboards can also be created independent of the database wizard. If you prefer, you can create your own switchboard by using the Switchboard Manager or the Form Design Window.

Activity 40: Switchboard Manager
1. On the Tools menu, point to Database Utilities, and then click Switchboard Manager.
2. If Microsoft Access asks if you'd like to create a switchboard, click Yes.
3. In the Switchboard Manager dialog box, click the Edit button to edit the default Main switchboard.
4. In the Edit Switchboard Page dialog box, click on New to add a new item to the list of possible actions.
5. In the Edit Switchboard Item dialog box, type the text for the first switchboard button in the Text box, and then choose a command in the Command box.
6. Depending on which command you click, Microsoft Access displays another box below the Command box. Click an item in this box, if necessary. For example, if you chose Open Form In Edit Mode in the Command box in step 5, click the name of the form you want to open in the Form box, such as STUDENT Form, and then click the OK button.
7. Repeat steps 4 through 6 until you've added all the items to the switchboard. If you want to edit or delete an item, click the item in the Items On This Switchboard box, and then click Edit or Delete. If you want to rearrange items, click the item in the box, and then click Move Up or Move Down.
8. Click Close.
9. Look for your Switchboard under the Forms tab.
10. You can edit it in design view to change the text, colors, etc. Be careful: do not edit or delete the Switchboard table which controls your options on this form.
Controlling User Access

When you used some of the tools and features described in the previous section, you started automating your database application to make it easier for the user. But, you have not yet provided any protection to prevent a user from accidentally damaging the application or to prevent an unauthorized user from accessing the system. Three features that are available to provide this type of protection without programming are the Startup Properties, Passwords, and Custom Menus/Toolbars.

Startup Properties

The Startup Properties allows you to dictate how the application system will look on startup. In this dialog box you can specify if the Database Window is displayed on startup and you can disable the shortcut keys that will allow it to be displayed. You can disable the Status Bar, Tool Bars, Menus, and Shortcut Menus. You can also specify a form to open automatically on startup, which enables you to activate your independently developed switchboards.

To set Startup properties, click on Tools > Startup…

![Startup Properties Dialog Box]

To bypass these startup settings, hold down the Shift key while opening the database. You cannot disable the Shift key functionality without programming.
Password

You can prevent unauthorized access into the database by setting a Password. However, you must first close the database so you can specify Exclusive mode before you can set the password.

To set a password on a database:

1. Close the database if open and choose File > Open
2. Select the database to be opened
3. Click on the arrow next to the open button to see a menu and choose Open Exclusive
4. Press Shift (to bypass the startup options) and click Open. If you know that the database does not have any Startup properties set, you do not have to do press Shift.
5. Select Tools | Security | Set Database Password from the menus
6. In the Password portion of the box, type the desired password. (it is case sensitive) In the Verify portion of the box, repeat the password to verify it accuracy.
7. Click on OK.
8. Close the database and reopen it
9. Enter the password and click on OK

Notes about passwords

- Hold down the shift key when reopening the database to bypass the startup properties, but the password is still required.
- When a password is set, you have to enter the password before you can import the database objects into another database or before you can compact the database.
- The password does not protect the data files from being viewed in a disk editor or utility program. To prevent this, encrypt the database.
- To remove or change the password, reopen the database as exclusive, and then select Tools - Security - Unset Database Password from the menus. You must know the password to remove it.
- The password can be cleared or changed by anyone who knows the current password and has access to the Unset Database Password command.
- Another way of securing your database would be to implement the User-Level Security feature of Access. With this feature, you can provide different levels of access to users or groups of users. Make sure that you make a backup of any database before doing this. You can lock yourself out.
Custom Toolbars

You can edit built-in menu bars, shortcut menus, and toolbars or you can create custom versions using the Customize Dialog Box. Access refers to all of these navigation devices as command bars, and the Customize Dialog Box provides a common way to manage them.

To display the Customize Dialog, right click on the toolbar and choose Customize from the shortcut menu.

The toolbar tab lists the built-in toolbars and any new ones that you create. It also includes the menu bar for the current window.

- You can display or hide any toolbar by clicking in the preceding check box.
- Click New to create a new toolbar
- Toolbar must be visible to add/remove command buttons
The **Commands** tab displays the command categories in the left pane and the commands within the category are on the right.

- Select a category name to view the commands
- Click and drag a command to the desired location on the toolbar
- To remove a command, click and drag the command off of the toolbar
- With a command chosen, click on Description to see what the command will do

The **Options** tab allows you to specify other command bar options.

- Check **Always show full menus** to be able to see all menu commands, whether you’ve used them recently or not.
- Check **Show ScreenTips on toolbars** to have screen tips appear when you hover the mouse over a toolbar command
**Activity 41: User Access**

1. Using the School5 database, choose *Tools | Startup*... from the menus
2. Select the *Switchboard* from the list under *Display Form/Page*:
3. Uncheck *Allow Full Menus*, *Allow built in toolbars*, and *Display Database Window*
4. Close the database and open it again.
5. Close the database (switchboard) and exit Access
6. Start Access
7. Choose *File | Open* from the menus
8. Select *School5.mdb* and choose *Open Exclusive* from the drop down menu on the *Open* button (hold the Shift key down while opening the database to override the startup settings)
9. Change the startup properties to allow menus, toolbars and the database window
10. Select *Tools | Security | Set Database Password* from the menus
11. Enter a password (make sure that you remember it!)
12. Click on OK
13. Close the database and open it again
14. Enter the password that you just set
15. Close the database and open it again in Exclusive mode
16. Remove the password
Objects, Events, and Properties

Objects

The common objects are the six objects in the Database Window. However, behind these objects is a much deeper structure of hidden objects and controls that Access uses to define and manage the six standard objects. Some of these background objects are obvious, like controls on a form or report, while others are pure system objects, such as TableDef and Document objects. Nearly all of these objects are accessible to you, the database designer. However, in order to access some of these objects, you must use the Visual Basic for Applications (VBA) programming language.

Properties

The characteristics of the objects within the database can all be described by a set of properties. At any instant, all database objects are uniquely defined by the current values of the assigned properties. The set of current values is called the state of the object and the collective set of current values for all objects is called the state of the database. Objects have two kinds of properties; those set when the object is created - called design properties, and those set when the database is running - called run-time properties.

Events

The run-time property values are dynamic. That is, they can be changed during the operation of the database. When a property is changed dynamically, the set of current values changes and therefore the state of the object and the database changes as well. The action that causes this change in state is called an event.

Objects can be manipulated by programs. Programs can be classified as either internal or external. Internal programs are the built-in functions and features of Access that you activate by selecting a menu item or clicking a button on a toolbar. These are the programs that make Access work. External programs are the ones you develop yourself or have someone develop them for you. These programs, called macros or procedures, respond to events and change the state of the Access application.

Events can be classed by type or function. The Access Macro Events table in the reference material handout defines the common events for macros. In most cases, the subset of events that is applicable to a particular object is available from the object's property list.
Macros Environment

A macro is one of the objects in the Access database window. It is really a small computer program used to automate repetitive tasks. Unlike macros in other Windows based applications, Access macros are not used to duplicate keystrokes or mouse movements. The command set for an Access macro is very small (only 56 commands) and the commands are expressed in plain language.

Macro Name
This allows you to store multiple macros within a single macro object in the database. This is commonly used to group related macros together. To display this column choose View | Macro Name

Condition
An expression can be entered into this column that will control when the macro executes. The actions will only execute, if the conditional statement has a true value. To display this column choose View | Macro Name

Action
An action is the desired task(s) in executing the macro. Actions are executed in order from top to bottom. This column is displayed by default

Comments
This column can be used to describe the action and what it will do. This is used to document the macro for future reference. This column is displayed by default
Macro Utilities

Moving Actions
1. Select the action by clicking on the box to the left of the row. Click and drag to select multiple actions at once
2. Click and drag the row to the desired location. A black line will show you where the action will be moved to

Deleting Actions
1. Select the action by clicking on the box to the left of the row. Click and drag to select multiple actions at once
2. Press the delete key on the keyboard or choose Edit | Delete from the menus

Copying Macros
1. Select the macro that you wish to copy
2. Right click on the macro and choose Copy or choose Edit | Copy from the menus, or press Ctrl C on the keyboard
3. Right click on a white area of the window and choose Paste or choose Edit | Paste from the menus, or press Ctrl V on the keyboard
4. Enter the name of the new macro in the Paste As dialog window

Renaming Macros
1. Select the macro that you wish to rename
2. Right click on the macro and choose Rename, choose Edit | Rename from the menu, or press F2 on the keyboard
Running Macros

Macros can be executed in several ways:

- Manually
- From within another macro
- From within an object such as a form
- Automatically on startup of the database

Manually

- In the database window, select the macro name and then click on the Run button
- From Design view of the macro, choose Run | Run from the menus or click on the run button on the toolbar
- Select Tools | Macro | Run Macro from the menu. Choose the Macro name and then click on OK

From within another Macro

1. Choose RunMacro as the action
2. Enter the macro name that you wish to run in the argument area for the action

From within an object

- Assign the macro name that you want to run to an event associated with when you would like to execute the macro. Ex: OnClick on a button

Automatically on startup of the database

- Create a macro called AutoExec. This macro is run automatically when the database starts
- Enter the RunMacro action and enter the macro name that you want to run on startup
- This macro will run every time that the database starts. To open the database without this macro executing hold down the Shift key while opening the database
Macro Conditions

The condition column in the action pane allows you to control the flow of data through the macro. The normal flow through the macro commands is sequential. That is, the actions are executed one after another in the sequence that they were placed in the action column of the macro design window. An alternative to sequential execution of the actions is conditional execution. A condition is a logical expression that evaluates to either True or False. If the expression is true, then the specified action is completed, otherwise it is ignored. The same condition can apply to more than one action, if they are on successive lines and the ellipsis (...) is entered in the condition column for the second and successive lines.

The combination of the macro action, its arguments, and the control functions of the condition allow the experienced designer to create macros with complex logic that emulates powerful computer programs developed with high level languages. These macros, which are beyond the scope of this course, can replicate the standardized building blocks of computer programs, known as the basic constructs. To learn more about these macros, the designer should refer to an introductory level programming manual for an event driven language such as Visual Basic.
Error Handling

If an Access macro fails to perform as expected, the designer can place it in "single step" mode while troubleshooting the problem. Single step allows you to move through the macro, one action at a time with pauses between actions so you can observe and analyze the results.

To single step a macro:

1. Edit the macro in the design window.
2. Select single-step from the toolbar or from the run menu.
3. Select run from the toolbar.

The single step dialog box shows the macro name, the action, and the arguments. The box contains three buttons:

- **Step**: Run this action. If no error occurs, display the next action.
- **Halt**: Stop the macro and close the dialog box.
- **Continue**: Turn off single-step and run the remainder of the macro.

If a macro causes an error in run or single-step modes, an Action Failed dialog box is displayed. It looks like the single-step dialog box except it only has the halt button available.
Working with Macros

This section contains a series of example macros that will help you navigate through and manage a database.

Activity 42: Closing a form with a macro

Part 1: Create a macro.
1. Open the database School5.mdb
2. Click on the **Macro** tab
3. Click on **New** to create a new macro
4. Select the **Close** action in the first cell of the **Action** column
5. Enter **Close the Student Form** in the corresponding Comment cell
6. In the argument pane, set **Object Type = Form**
7. In the argument pane, set **Object Name = Student Form**
8. In the argument pane, set **Save = Prompt**
9. Close the Macro Design Window by clicking the **X** in the upper right corner
10. Save the macro with the name **CloseStudentForm**

Part 2: Assign the macro to a command button.
1. Open the form **Student Form** in **Design View**
2. Make sure the Command Wizard button in the Toolbox is not depressed
3. Click on the **Command Button** in the Toolbox
4. Move the mouse pointer (Do Not Drag) to the bottom of the form and click to create a command button
5. Right click on the button and select **Properties** from the shortcut menu that appears
6. In the Properties list, click on the **All** tab to see all properties
7. Change the **Caption** property to **Close Form**
8. Scroll down to the **OnClick** property
9. Select **mCloseStudentForm** from the drop down list
10. Close the form and save changes
11. Open the form and observe the new button on all forms
12. Click the button to close the form and return to the Database Window.
Activity 43: Multiple Action Macro.

1. Open the macro mCloseStudentForm in Design View.
2. Click in the first row to position the cursor and then right click.
3. Select Insert Rows from the shortcut menu.
4. In the first action cell, select the Hourglass action and accept the default argument of Hourglass On = Yes.
5. Leave the Close action in the second cell.
6. In the third cell, select the MsgBox action and set the arguments:
   - Message = "Are you sure?"
   - Beep = Yes
   - Type = Warning?
   - Title = Close Student Form
7. In the fourth cell, select the action Beep (No arguments)
8. Close the Design Window and save changes.
9. Open the Student form and click on the button
The message box in the preceding example was used to transmit a message to the user. The message box can also be used as a function to create a condition within a macro. The message box function takes the form:

```
MsgBox("message", type, "title")
```

- message is a string that contains the message that is to be displayed,
- title is a string that contains the title to be displayed in the box, and
- type is a number that controls the buttons and icons.

Type is the sum of three parts: Button Type + Icon Style + Default Button

<table>
<thead>
<tr>
<th>Button Type</th>
<th>Icon Style</th>
<th>Default Button</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>None</td>
<td>0 First</td>
</tr>
<tr>
<td>1</td>
<td>Critical (X)</td>
<td>256 Second</td>
</tr>
<tr>
<td>2</td>
<td>Warning (?)</td>
<td>512 Third</td>
</tr>
<tr>
<td>3</td>
<td>Warning (!)</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>Information (i)</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

When using the MsgBox() function as a condition, we frequently need to evaluate its return code.

The value returned in the code indicates which button was selected:

<table>
<thead>
<tr>
<th>Button Clicked</th>
<th>Value Returned</th>
</tr>
</thead>
<tbody>
<tr>
<td>OK</td>
<td>1</td>
</tr>
<tr>
<td>Cancel</td>
<td>2</td>
</tr>
<tr>
<td>Abort</td>
<td>3</td>
</tr>
<tr>
<td>Retry</td>
<td>4</td>
</tr>
<tr>
<td>Ignore</td>
<td>5</td>
</tr>
<tr>
<td>Yes</td>
<td>6</td>
</tr>
<tr>
<td>No</td>
<td>7</td>
</tr>
</tbody>
</table>

Given this function, let's modify our mCloseStudentForm macro to include a MsgBox () function used as a condition.
Activity 44: Conditional Multiple Action Macro

1. Open the macro mCloseStudentForm in Design
2. Delete the actions for MsgBox and Beep
3. Insert a new line between Hourglass and Close
4. Choose View | Condition from the menus to display the Condition column
5. In the new line, under the condition column, enter the message box function as MsgBox("Are you sure?", 289, "Close Student Form") <> 1
6. The action for the new line would be StopMacro. If the condition evaluates as true, the macro will stop. Otherwise, it will continue.

7. Close the macro design window, save the changes, and test your new macro.

Notes:
- The value 289 was calculated from a Button Type = 1 plus Icon Style = 32 plus Default Button = 256.
- The condition <> 1 (not equal to 1) is used to evaluate the return code. Since the Button Type provides an OK and a Cancel button we want to proceed with the close if the OK is clicked and abort the close if the Cancel is clicked.
The final examples of macros involve opening more than one form at a time and then synchronizing the data in them.

**Activity 51: Opening multiple forms**

1. Create a new macro
2. In the first Action cell, select OpenForm
3. In the argument pane, enter a Form Name of **Teacher Records**
4. Close the macro and save it under the name **OpenTeacher**
5. In the Database Window, click on the Form tab and open the **Class Form** in Design
6. Display the Properties window for the Form and click on the Event tab
7. In the On Open property, select the macro **OpenTeacher**
8. Close the Form saving changes

Note that when you open the Class Form, both forms open on the screen at the same time. However, they probably do not indicate synchronized data. The common field of Teacher ID is not necessarily the same in both forms. We can modify the macro to accomplish this task for us. The common fields in these forms are C-Teach in the Class Form and T-ID in the Teacher Record.

**Activity 52: Synchronizing data in forms**

1. In the Database Window, click on the Macro tab and open the macro **OpenTeacher** in Design.
2. In the argument pane, enter the following expression in the Where setting box: 
   \[T-ID] = Forms![Class Form]![C-Teach]\]
3. Close the macro saving changes.
4. Reopen the **Class Form** and the **Teacher Record** opens that is synchronized with the data in the Class Form. However, as you scroll through the class records, the teacher records do not re-synchronize themselves.
5. To correct this problem, change the **OpenTeacher** macro from the On Open property to the On Current property in the **Class Form**.

Note that in the Where expression, the left side needed only the field name since it is automatically tied to the form described in the macro. However, the right side had to have a complete identifier.
Advanced Features

Creating Indexes
A Microsoft Access table is normally stored in primary key order. If you frequently need to search or sort it based on a field other than the primary key, you can speed up the operation by building an index. An index is a separate file that Access uses to cross reference the data in a table. It works like the index in a book and lets you find things quickly. You can have up to 32 indexes for each table (Access uses the first one to set an index on the primary key). Five of the indexes can be multiple field indexes and each of them can contain up to 10 fields.

Index fields are automatically updated during data entry. Therefore, while an index increases the speed of a search, it decreases the speed of data entry. You should consider indexing if:

- The field data type is text, numeric, currency, or date. Memo, Hyperlink and OLE fields cannot be indexed.
- You plan to use the field for repetitive search or sort operations.
- The values in the field vary considerably. If most of the values in the field are the same, no increase in speed will be noticed.
- The table has a large number of records in it. Small tables may run slower with indexing.
- The data in the table is fairly static. Dynamic tables with data that changes frequently reduce the speed of data entry thereby lowering overall table performance.

To create an index on a single field, you simply set the Indexed Property of the field in the Table Design mode.

1. Open the table in Design View.
2. Select the desired field in the top portion of the dialog box and the field properties will be displayed in the lower portion.

When selected, the Indexed property offers three choices for indexing:

- No
  - Do not index (default for all fields except the primary key).
- Yes (Duplicates ok)
  - Create an index and allow duplicate values in the field.
- Yes (No Duplicates)
  - Create a unique index on the field (Default value for the primary key).
Viewing or Editing Index Settings
You can view the index settings by opening the table in design view and selecting Indexes from the View menu. This displays the Index dialog box.
You can change indexes or properties in the dialog box.
- To delete an index, delete its row from the window. This does not delete the field itself from the table. It simply deletes the use of this field as an index.
- You can save changes by choosing the Save command from the File menu.

Analysis tools
The Performance Analyzer will examine your database and offer suggestions of ways to improve the performance of the objects. It does not analyze Access itself or the underlying operating system.
To use the performance analyzer:
1. Open the database that is to be analyzed.
   Note: The database needs a reasonable amount of data in the tables before the analyzer can make realistic projections.
2. Select the Analyze command from the Tools menu and click the Performance entry.
3. Select the type of object and then the name of the object that is to be optimized.
   - Repeat the until all desired objects are selected or use the All Object Types tab to view a list of all objects in the database. A Select All button is provided to simply the selection process.
4. Click the OK button to run the analysis.
   - The Analyzer examines each of the objects and offers suggestions, recommendations, and ideas.
The Table Analyzer examines your tables and looks for design flaws. It tests your tables to see if they conform to the commonly accepted rules of data normalization and then offers suggestions for improving the tables.

To use the table analyzer:

1. With the database open, select the Analyze command from the Tools menu and click the Table entry.

2. When the Table Analyzer Wizard starts, move to the table selection screen and indicate the table that you want to analyze.
   - When you let the wizard automatically analyze your table, it will advise you of changes that need to be made (like splitting the table into multiple tables) and then it will make them for you.

Sharing databases
If your computer is connected to a network, you and others can work with one Access database at the same time.

There are several ways you can share data in a multi-user environment.

Share the entire Access database
You can put the entire Access database on a network server or in a shared folder. This is the easiest method to implement. Everyone shares the data and uses the same forms, reports, queries, macros, and modules. Use this strategy if you want everyone to use the Access database the same way or if you can't support users creating their own objects.
Share only the tables in the Access database

You can put only the tables on a network server, and keep other database objects on users' computers. In this case, the Access database's performance is faster because only data is sent across the network.

When you split a database into a back end (tables) and a front end, users can customize forms, reports, and other objects in their front-end databases without affecting other users.

Record Locking

When you open an Access database file (.mdb) in shared mode, Microsoft Access also creates a locking information file (.ldb) with the same file name (for example, Northwind.ldb) and in the same folder as the database file. This locking information file stores the computer name (such as mypc) and security name (such as Admin) of each shared user of the database. Microsoft Access uses this information to control concurrency. In most cases, Microsoft Access automatically deletes the locking information file when the last user closes the database file.

In order to setup a shared database you need to instruct Access how to handle a shared database.

This is done in the Advanced tab of the Options dialog box.

1. Display the **Options** dialog by selecting the **Options** command from the **Tools** menu.

   ![Options Dialog Box](image)

2. Click the **Advanced** tab in the dialog box.
3. Click the **Shared** radio button in the **Default open mode** of the dialog box.
4. Click the **Edited record** radio button in the **Default record locking** section of the dialog box.
5. Click the **OK** button to close the dialog box.
Backing up a database

Access database objects are all stored in one file. Therefore it is extremely important that you protect your data by creating backups at regular intervals. **Access does not have any internal back up procedure.** To back up your database, you must exit Access and use a commercial back-up package, the DOS copy command, or the copy facilities of Windows Explorer.

**NOTE:** To back up your database you need to copy the database file (*.mdb) and, if it exists, the workgroup file (*.mdw). The workgroup file contains information on the custom settings for multiple users and on file security. If this file is lost or damaged and you do not have a back-up of it, then you will be unable to start Access. Further, if you cannot restore it from a backup, then you will have to completely re-install Access and set up all of your shared options and security again.
Encrypting a database

Encrypting ensures that a database can be opened and viewed only in Access. It does not prevent the database from being opened by another copy of Access, but it cannot be opened by a word processor, spreadsheet, etc.

6. If the database is open, you must close it before Encrypting or Decrypting.

7. Select the Security command from the Tools menu and click the Encrypt /Decrypt Database… entry.
   - This displays a standard file dialog box.

8. Select the desired database and click the OK button.
   - A second standard file dialog box is displayed prompting you to rename the new encrypted database file.

9. Specify a name and location for the file and click the OK button.

Note: Access contains a full range of security settings and procedures that allow you to password protect any or all Access objects. For detailed security settings, see 'Administering and Securing an Access File' under the Contents tab of the Help window.
Repairing a database

If a database becomes damaged, Access will automatically attempt to repair it as it tries to open it. Occasionally, however, Access will be unable to determine the problem and you, the user, will have to intervene.

10. If a database is open, it must be closed before using the repair utility.

11. Make a backup copy of the database before attempting to repair it.

12. Select the Database Utilities command from the Tools menu and click the Compact and Repair Database… entry.
   - This displays a standard file dialog box.

13. Select the desired database and click the OK button.
   - A second standard file dialog box is displayed prompting you to rename the new encrypted database file.

14. Specify a name and location for the file and click the OK button
   - You should then open the new database file and check its contents to insure that the data appears intact.

Converting databases

Access XP provides several conversion options. The option you choose depends upon what versions of Access you and your fellow users have upgraded to.

When to convert from Access 97 format to Access 2000 or 2002 (XP)

In most cases, you'll want to convert a previous-version Microsoft Access database to Access 2000 file format if all users have upgraded to Access 2000 or later, and you'll want to convert to Access 2002 file format if all users have upgraded to Access 2002.

- Although you can open or enable an Access 97 or earlier database in Access 2002 without converting it, you can only change the design of objects in the previous version (97).
- Once you’ve converted an Access database to Microsoft Access 2000 or Access 2002 file format, you can’t open it in the previous version of Access.
- You can convert an Access 2000 database back to Access 97. However, you can't convert an Access 2000 database back to Access version 2.0 or 95.
If your Access database is a multi-user (shared) database, and all users can't upgrade to Microsoft Access 2000 or later at the same time, you can split the database so that it is a front-end/back-end application. You can then have different versions of the front end connected to the back end, which remains unaltered. Users of Access 2000 or later can use a converted version of the front end, where they can take advantage of new features.

When to convert an Access 2000 format to Access 2002 (XP) format


Convert to Access 2002 file format only if all users have upgraded to Access 2002, and you have been successfully using the file in Access 2000 file format. If all users have upgraded to Access 2002, you can develop an application in the Access 2002 file format without having to make sure that any object, method, property or function you use is also available in Access 2000.

Additional information about converting files is available using the following pathway in the Contents tab in the Help window:

**Microsoft Access Help ➔ Getting Started ➔ Converting Access Databases**