Microsoft Access
Introduction to Using Macros with Forms

Table of Contents

INTRODUCTION TO MACROS..................................................................................................................... 3
CREATING A MACRO.........................................................................................................................................3
ADDING AN ACTION .........................................................................................................................................4
REARRANGING ACTIONS....................................................................................................................................6
COPYING AN ACTION.........................................................................................................................................6
INSERTING AN ACTION......................................................................................................................................6
DELETING AN ACTION .......................................................................................................................................6
SAVING THE MACRO..........................................................................................................................................7
RUNNING THE MACRO FROM DESIGN VIEW ...............................................................................................7
RUNNING A MACRO FROM THE DATABASE WINDOW ....................................................................................7
RUNNING A MACRO FROM A MENU.................................................................................................................7
RUNNING A MACRO FROM THE WINDOWS DESKTOP ...................................................................................8

USING MACROS WITH FORMS................................................................................................................... 8
CREATING A SWITCHBOARD ................................................................................................................... 10
CREATING MACRO GROUPS .................................................................................................................. 12

USING CONDITIONS IN MACROS............................................................................................................. 13
SETTING CONDITIONS AND ACTIONS.......................................................................................................13

SYNCHRONIZING FORMS WITH OTHER OBJECTS ................................................................................... 14

WHEN A MACRO ISN'T NEEDED .............................................................................................................. 16
USING THE IMMEDIATE IF (IIF) STATEMENT ..............................................................................................16

COPYRIGHT

Copyright ©2002 by EZ-REF Courseware, Laguna Beach, CA http://www.ezref.com/ All rights reserved. This
publication, including the student manual, instructor's guide and exercise files, or any part thereof, may not be
reproduced or transmitted in any form or by any means, electronic or mechanical, without prior written permission of
EZ-REF Courseware.

TRADEMARKS

EZ-REF is a trademark of EZ-REF Courseware. All other products or brand names mentioned are trademarks or
registered trademarks of their respective holders.

DISCLAIMER

While EZ-REF Courseware has taken care to ensure the accuracy and quality of these materials, all material is provided
without any warranty whatsoever, including, but not limited to, the implied warranties of merchantability or fitness for a
particular purpose. Any persons or businesses mentioned in the text of this manual are strictly fictitious. Any
resemblances to existing or deceased persons, or existing or defunct businesses, is entirely coincidental.
INTRODUCTION TO MACROS

USAGE:
Macros allow you to automate routine or repetitive tasks, such as printing weekly reports. A macro is defined as a series of repetitive keystrokes or mouse actions combined to perform mundane or complicated tasks. By assigning a macro to a mundane task, you are able to save yourself considerable time. By assigning a macro to a complicated task, even a novice can run the macro - without having to understand the steps behind it. Within Access, a macro is considered to be an object - just like a report, a table or a form.

You can use macros within Access to perform the following tasks:

- Make your reports and forms work together
- Find and filter records automatically
- Ensure data accuracy
- Set properties for forms, reports and controls
- Create a customized environment

CREATING A MACRO

Follow these steps to create a macro:

1. Open the database window.

2. Click on the Macro icon (which is located on the left side of the database window).

3. Click on the New button.
The Macro Window appears, as illustrated below:

The top portion of the window is used to add the actions/commands to be performed. The lower portion of the window is used to include arguments/options for the action. An argument provides additional information on how Access should carry out the action. Each action will have a different set of arguments. For further information, select the argument and press F1 for help.

**ADDING AN ACTION**

1. Click in the first empty cell within the **Action** column. Either select the desired action from the pull-down “Action List” provided or type the action directly into the cell.

2. Click in the **Comment** column to add a comment explaining the purpose of the current action. These comments are only useful when writing or modifying the macro - they are not recognized by Access when running the macro.

3. Specify the **Arguments** for the action, if any are required.
Macro to Select California Clients and Sort Them by City

<table>
<thead>
<tr>
<th>Action</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>OpenTable</td>
<td>This macro displays a list of clients from California and the cities in which they live; this step opens the table.</td>
</tr>
<tr>
<td>GoToControl</td>
<td>Moves to the AreaCode field</td>
</tr>
<tr>
<td>RunCommand</td>
<td>Hides the AreaCode field</td>
</tr>
<tr>
<td>GoToControl</td>
<td>Moves to the Phone field</td>
</tr>
<tr>
<td>RunCommand</td>
<td>Hides the Phone field</td>
</tr>
<tr>
<td>GoToControl</td>
<td>Moves to the ActiveCust field</td>
</tr>
<tr>
<td>RunCommand</td>
<td>Hides the ActiveCust field</td>
</tr>
<tr>
<td>ApplyFilter</td>
<td>Filter the data for clients living in California</td>
</tr>
<tr>
<td>GoToControl</td>
<td>Moves to city field</td>
</tr>
<tr>
<td>RunCommand</td>
<td>Sorts city field in ascending order</td>
</tr>
</tbody>
</table>

### Action Arguments

| Control Name | ActiveCust |

Moves the focus to a specified field or control on the active datasheet or form. Press F1 for help on this action.

Each of the actions in this macro is controlled by its arguments in the Action Arguments portion of the screen.

**NOTE:** If you add more than one action to a macro, they are executed in sequential order from the top down. Be sure to add your actions/commands in the order in which you want them executed.
REARRANGING ACTIONS

If you have multiple actions within a macro and want to change the order in which the actions are executed, you can easily move them around by following the steps outlined below:

1. Click on the row selector (the gray bar) to the left of the action name. The row will become highlighted.

2. Click the row selector again and begin dragging the action to a new location. A dark horizontal line will appear as you drag - indicating where the action will be placed when you release the mouse button.

COPYING AN ACTION

1. Click on the row selector (gray bar) to the left of the action.

2. Click on the COPY tool. The action is temporarily copied to the clipboard.

3. Click on the row to copy the action to.

4. Click on the PASTE tool.

INSERTING AN ACTION

1. Click on the row selector (the gray bar) to the left of the action name where the new action should be inserted.

2. Click on this tool to insert a row.

DELETING AN ACTION

1. Click on the row selector to the left of the action name.

2. Click on this tool to delete the selected row.
SAVING THE MACRO

Once you have added the necessary actions, comments, and arguments for the macro, you will want to save it to disk.

Click on this button to save the macro.

Enter a name for the macro (up to 64 characters, including spaces). When done, select OK.

RUNNING THE MACRO FROM DESIGN VIEW

Before closing the macro window, you may want to test it out to be sure it works by running it.

Click on the Macro Run button.

OTHER WAYS TO RUN MACROS

Once a macro has been created and saved, it can be run from locations other than the Macro design window. You may run a macro from the Database window, from any menu bar, or from the Windows desktop.

You may also run a macro by attaching it to a command button on a report or form, or by attaching it to an event, for example opening a form or moving to a control on a form. These methods are discussed under “Using Macros with Forms and Reports”.

RUNNING A MACRO FROM THE DATABASE WINDOW

1. Select the Macro objects tab from the Database window.
2. Double-click on the macro you wish to run.

RUNNING A MACRO FROM A MENU

From any object window,
1. Select TOOLS, MACRO from the menu.
2. Select the macro you wish to run from the drop-down list.
3. Click the OK button or press the Enter key.

**RUNNING A MACRO FROM THE WINDOWS DESKTOP**

You may wish to run a macro directly from the Windows Desktop that opens a form or runs a report. This eliminates having to start up Access before running the macro.

1. After creating and saving the macro, select Macro objects tab of the Database window.
2. Reduce the size of the Access application window so that a portion of the Desktop is visible.
3. Press and hold the Control key on your keyboard and drag the icon for the macro to the desktop. This creates a Desktop shortcut for the macro.
4. You may now close Access and run this macro by double-clicking its icon on the Desktop.

**USING MACROS WITH FORMS**

Once you have created your macros, you may wish to add them to your forms/reports to run based on certain events that occur. For example, you might have a button on a form that would run a macro whenever it is selected. Another example could be a report that prints a control based on values from another control (i.e., if the state is California, print out tax information).

When using a form or report, Access responds to certain occurrences as **Events**. For example, moving to the next record is an event, as is clicking on a control, as is updating the value in a field. Every control on a form has a set of events that can occur. These events are displayed in the Property window.

You specify how the event will be responded to by assigning a macro to that event. For example, a button control might have a macro assigned to the event called **On Click** that would open another form or print a report.

1. Open the form in design mode.
2. Select the control that you wish to assign the macro to or if the macro is to be associated with the entire form, select the form itself.

3. Open the Properties window by either clicking on the [RIGHT] mouse button and choosing Properties from the pop-up menu.

4. Select the Event tab to display the available events.

5. Click on the cell to the right of the event you wish to customize. You will see both arrow and ellipsis buttons.
   - If the macro has already been created, click on the list box arrow and select the macro from the pull-down list provided.

If the macro has not yet been created, you must create the macro now:

Click on which will display the following box:

```
Expression Builder
Macro Builder
Code Builder
```

Select Macro Builder and choose

```
Macro Name: Macro1
```

©2002 EZ-REF Courseware
You will now be asked to name the macro. Type a name (up to 64 characters including spaces) and choose OK.

Add your actions to the macro and then save it.

CREATING A SWITCHBOARD

A switchboard is a form that contains a series of command buttons. The buttons are attached to macros that perform tasks such as opening a table, query, report or form object. You can control the way that an object will open, for example, opening a table that is displayed through a certain filter, or opening a form in data-entry-only mode. Switchboards can be used to create a main menu for your database that will allow users to select an option with the click of a mouse.

There are two methods for creating a switchboard with macros:

**Method One:**

1. Create the form you will use for your switchboard. The form will not have a data source, since it will contain only command buttons.
2. Add the command buttons, creating the macro for each button as it is created.
3. Create labels for the buttons as needed

**Method Two:**
1. Create the macros you will use for the switchboard. It is often most efficient to store them as a macro group.

2. Create the form you will use for the switchboard.

3. Add command buttons to the form, attaching each to the appropriate macro as you create them.

4. Create labels for the buttons as needed

If you wish to start your switchboard from the desktop without having to open Access first, you can drag the icon for the saved switchboard form to the desktop.

A Sample Switchboard

<table>
<thead>
<tr>
<th>Acme Novelties Main Menu</th>
</tr>
</thead>
<tbody>
<tr>
<td>What Would You Like to Do?</td>
</tr>
<tr>
<td>![Icon] View Client Records</td>
</tr>
<tr>
<td>![Icon] Edit Client Records</td>
</tr>
<tr>
<td>![Icon] View Client Phone List</td>
</tr>
</tbody>
</table>
CREATING MACRO GROUPS

If you have several short macros that will be used together in one form, you can store them all in one macro window. Instead of having to keep track of several different macros, it is more convenient to group several related macros together.

Each macro within the group must be individually named and would consist of one or more actions. For example, assuming you have a form providing a user with five different reports, you could have a button for each report. When the user clicks on one of the five buttons, Access executes a macro to run that particular report. Instead of cluttering the list of macros within the database window with five small macros, you could create one macro containing the five smaller ones.

Be sure you are in Design view for the macro before you continue.

Click on the Macro Names button.

A column will be added to the left of the Action column, as illustrated in the example below:

<table>
<thead>
<tr>
<th>Macro Name</th>
<th>Action</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>OpenSales</td>
<td>MsgBox</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Beep</td>
<td></td>
</tr>
<tr>
<td></td>
<td>OpenForm</td>
<td></td>
</tr>
<tr>
<td></td>
<td>GoToRecord</td>
<td></td>
</tr>
<tr>
<td>OpenInvoice</td>
<td>OpenReport</td>
<td></td>
</tr>
<tr>
<td>EditData</td>
<td>OpenForm</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>ApplyFilter</td>
</tr>
</tbody>
</table>

Add your macro names and actions as needed. As shown in the example above, macro names can consist of more than one action. Each macro name (located in the far left column) is used to identify a separate macro.
USING CONDITIONS IN MACROS

SETTING CONDITIONS AND ACTIONS
There may be times when you would like your macro to be run only if certain conditions have been met. For example, you may only want to add tax to an invoice if the client is located within your state. The macro condition would check the State field and if it matches the state you are looking for, the tax field would be calculated. If the condition is not met, the field would be ignored.

Condition statements use logical operators (such as > < and =) and field names (enclosed in square brackets), as illustrated in the example below:

\[ \text{State} = \text{"CA"} \]

The example shown above would check the State control to see if it matched "CA". If it did, the condition would be met and the action listed on the same row as the condition statement would occur. If the condition is not met, the action would not occur.

To add a condition to a macro, display the Conditions column.

![Click on the Conditions button.](image)

<table>
<thead>
<tr>
<th>Macro Name</th>
<th>Condition</th>
<th>Action</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>OpenInvoice</td>
<td></td>
<td>OpenForm</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>GoToRecord</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>RunMacro</td>
<td></td>
</tr>
<tr>
<td>CalcTax</td>
<td>Forms][sales][State]=&quot;CA&quot;</td>
<td>SetValue</td>
<td></td>
</tr>
</tbody>
</table>

A new column appears between the “Macro Name” and “Action” columns labeled Condition. Enter the appropriate condition in the column. If the condition entered is met, Access carries out the action on that row. If the condition is not met, the action is ignored.

If more than one action is to be performed for a condition, each subsequent action must have an ellipsis (…) in the condition column to indicate that it is part of the same condition actions (see sample below).
Macro To Calculate Sales Tax By State

NOTE: The ellipsis (…) is used to indicate multiple actions for one condition. Place the ellipsis in the condition column for each additional action to be performed for a condition.

SYNCHRONIZING FORMS WITH OTHER OBJECTS

You can use a macro to synchronize actions between two forms, or between a form and a report. The macro is attached to the first form; the data displayed on the second form or on the report is determined by the data displayed on the first form when the macro is executed. This is accomplished by adding a WHERE argument to the macro action that opens the second form. For example, while looking at an order for a customer, you can open a form that displays the corresponding invoice for the order. The macro that opens the invoice contains a WHERE argument that synchronizes the two sets of data on the order number from the first form. See the sample macro below.
The WHERE condition sets the Client Invoice Report to display using the OrderNum field from an order form.

The WHERE condition is added as an argument to the Open action in the macro, and uses a common field (usually a key field) to synchronize the two objects.
WHEN A MACRO ISN'T NEEDED

USING THE IMMEDIATE IF (IIF) STATEMENT

Sometimes you may need a control on a form that tests for a simple condition and then performs one of two tasks depending on that condition. For simple conditions such as these, you can use an Access function called the IIf statement instead of writing a conditional macro. The IIf statement is entered into a calculated control on a form or report (note that it can also be used to create a calculated field in a query) and executes automatically without having to attach a separate macro to a field or command button.

The format for the IIf statement is:

IIF(expr, truepart, falsepart)

Expr is the expression that you want to evaluate. The expression can check the contents of a field with =, >, <, is true, is false, is null, and similar operators.
Truepart Expression or task to be performed if the above expression is true
Falsepart Expression or task to be performed if the above expression is false

Example:
IIf([ST])="NJ", [sellprice]*1.06,[sellprice]

This statement checks to see if the state field [ST] on a record is equal to New Jersey "NJ". If so, the selling price [sellprice] is multiplied by 1.06 to add 6% sales tax. If the state is not New Jersey, the sales price remains unchanged.