

Proicere, Inc.

MS Access: Macros and Modules

Lesson Notes

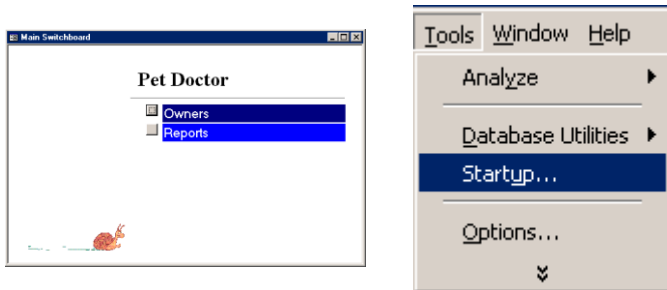
Author: Pamela Schmidt

Startup

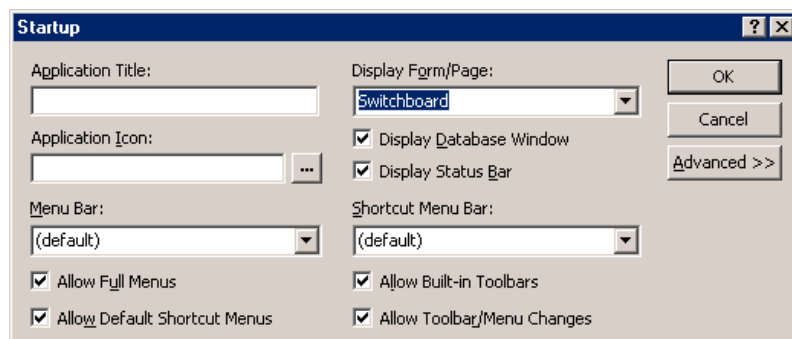
Menu

To bring up the switchboard or menu form on startup:

From the menu choose Tools, Startup



In the Startup dialog box, choose the desired form to open on startup.



Macros

Maximize a Form

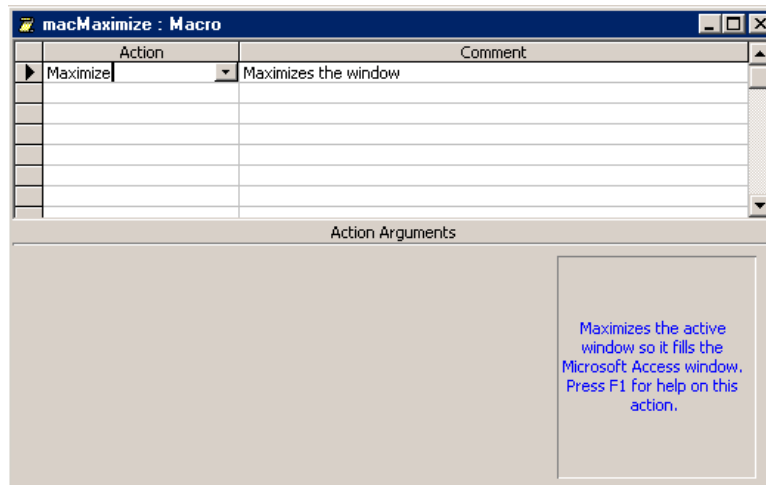
Create a Maximize a Form macro

Create a new macro

Under Action, choose Maximize.

The Comment field is optional.

Save the macro.



Use the Maximize macro as an event on a form

Open the form in the design view.

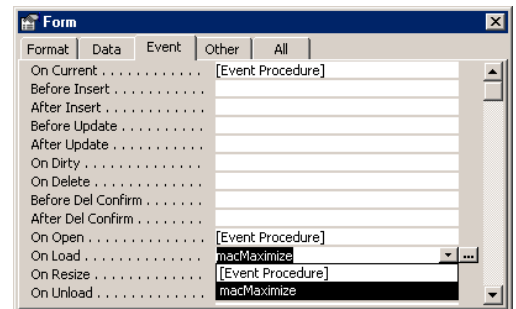
Open the properties of the form.

On the Events tab, click the “on load” event.

Choose the maximize macro off the drop down list.

Note: Open runs before Load.

Open events can be cancelled, load events can not.

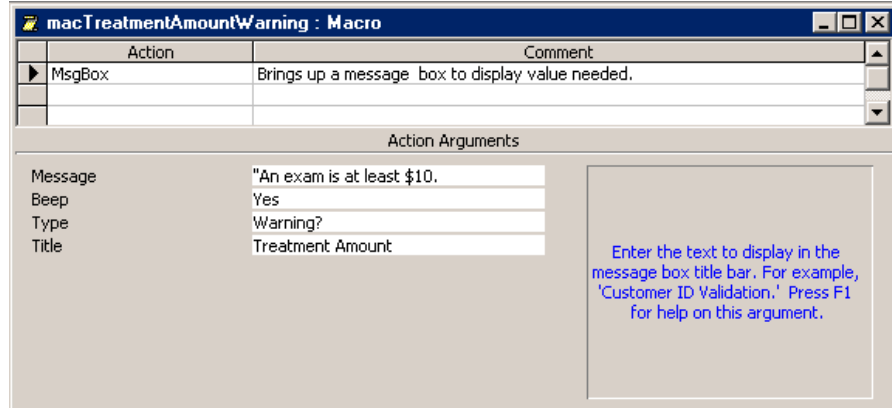


Prompt Macro

Create the Macro

Create a new macro

In the action field, choose MsgBox



In the Message field in lower half of screen, key in the message to displayed.

be

In the Type field, choose the symbol to be displayed in the pop up message box.

In the Title field, choose what should be displayed in the title bar of the Message Box.

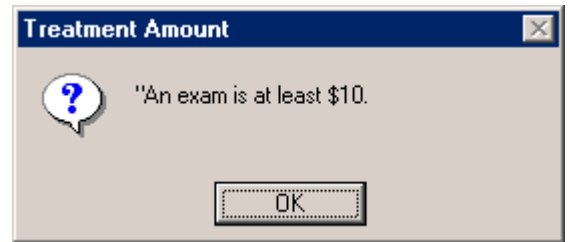
Use the Message Box macro as an event on a form

Open the form in the design view.

Open the properties of a field.

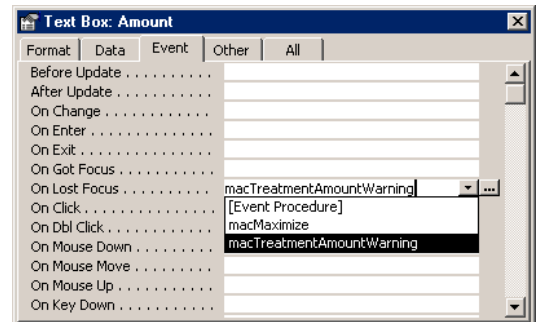
On the Events tab, click the appropriate event.

Choose the Message Box macro off the drop down list.



Note: On Got Focus runs the macro as soon as the field is selected before the user enters data.

On Lost Focus runs the macro when the field is left.

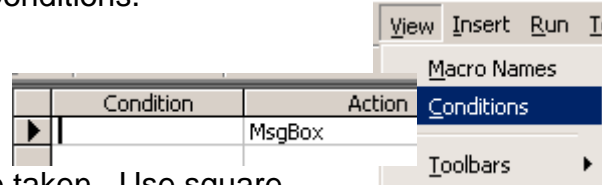




Conditional Macro

Create the Conditional Macro

- Open an existing macro, or create a new macro.
- On the menu in the macro, choose View, Conditions.
- A Condition column will open up.



Key in the condition next to the action to be taken. Use square brackets around field names.

	Condition	Action
▶	[Amount]<10	MsgBox

Finish the macro and save.

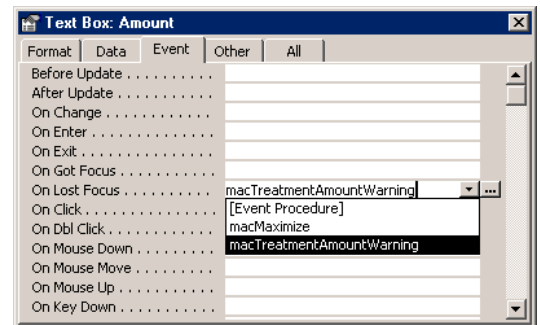
Example of a more complicated condition.

	Condition	Action
▶	IIf([TreatmentCode]="Exam",[Amount]<10)	MsgBox

Use the Conditional Message Box macro as an event on a form

- Open the form in the design view.
- Open the properties of a field.
- On the Events tab, click the appropriate event.
- Choose the Conditional Message Box macro off the drop down list.

Note: Place the macro On Lost Focus so it will check the field after the user has entered data.



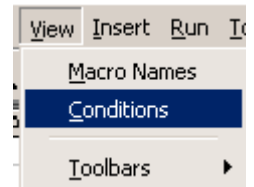


Conditional Macro comparing two dates

Create a conditional macro

On the menu in the macro, choose View, Conditions.
A Condition column will open up.

	Condition	Action
▶		MsgBox



Key in the condition next to the action to be taken. Use square brackets around the two field names.

	Condition	Action
▶	[EndDate] < [StartDate]	MsgBox

Finish the macro
and save

Use the Conditional Message Box macro as an On Lost Focus event on a form

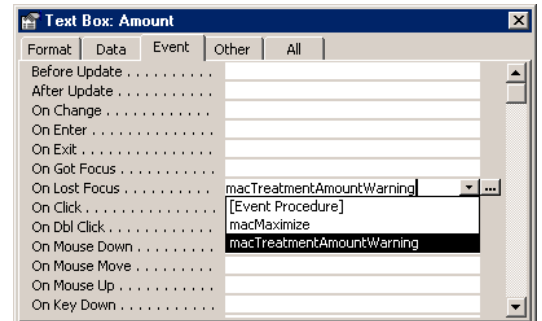
Open the form in the design view.

Open the properties of the end date field.

On the Events tab, click the On Lost Focus event.

Choose the Conditional Message Box macro off the drop down list.

Note: Place the macro On Lost Focus so it will check the field after the user has entered data.

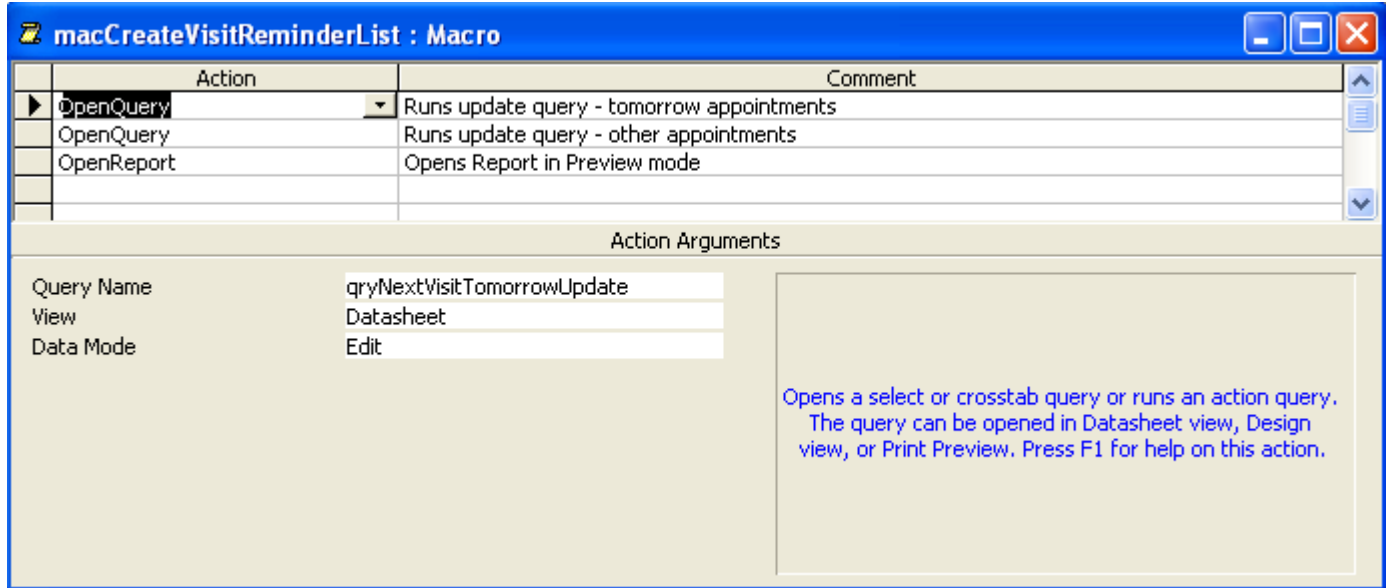


Macro to run/open several objects in sequence

Create a macro to run/open several objects in sequence

Create a new macro.

Enter each object to open as an action.



Fill in the Action Arguments for each Action.

Save the macro.

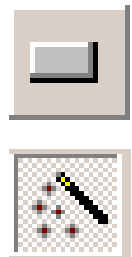
Attaching a Macro to a Command Button

Open the form in which the button is to be placed in the design mode.

Choose the command button tool off the toolbar.

Note: be sure to have the wizard button pressed down.

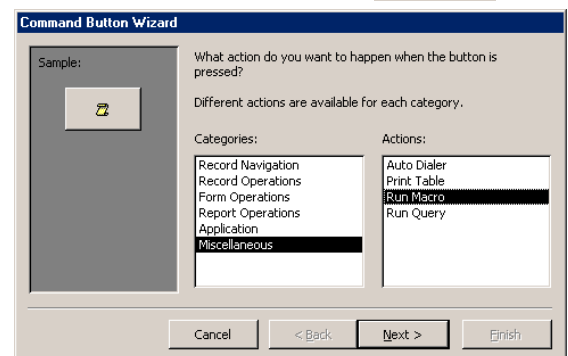
When the wizard appears, choose Miscellaneous, Run Query.



Click the next button then choose the desired query.

Continue through the wizard to choose a picture or key in text for the button face.

Use the “Next” button to go to the next step of the wizard. Do not hit the “Finish” button until after the “Assign a name for the Command Button”.



Macro to Refresh the Screen

Create a Refresh Macro

Create a new macro

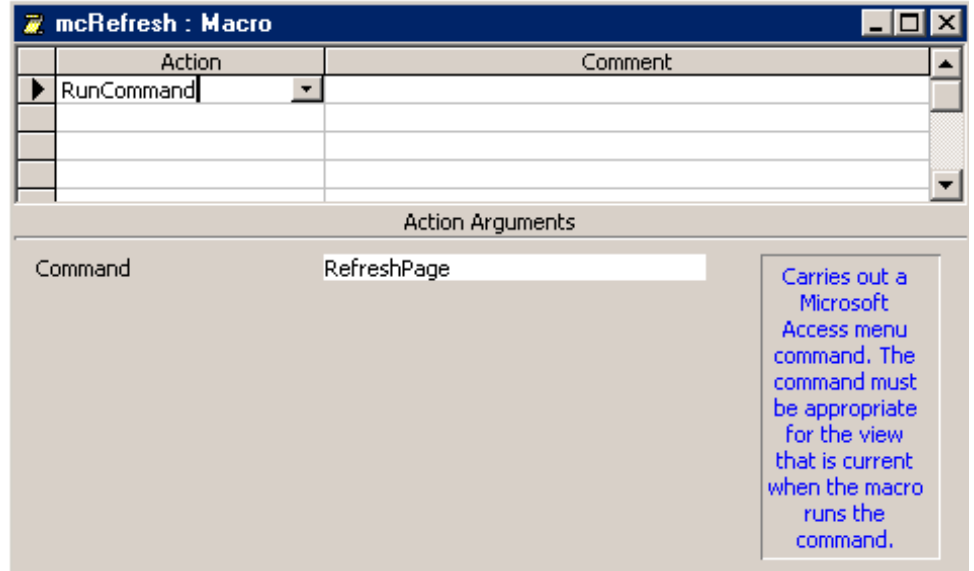
Under Action, choose

RunCommand.

In the Command field
under Action Argument
choose RefreshPage.

The Comment field is
optional.

Save the macro.



Using the Refresh Macro

Open the form in the design view.

Open the properties of a field.

On the Events tab, click the appropriate event.

Choose the Refresh macro off the drop down list.

Creating the Print Only Current Information Macro

Prints the desired report with only the information from the current form.

Create Print Only Current Information Macro

Create a new macro containing two actions.

For the first Action, choose RunCommand.

In the Command field under Action Argument choose Save Record.

For the second Action, choose OpenReport

For Report Name – pick the desired report off the list

For View – choose Print Preview

Filter Name – not necessary

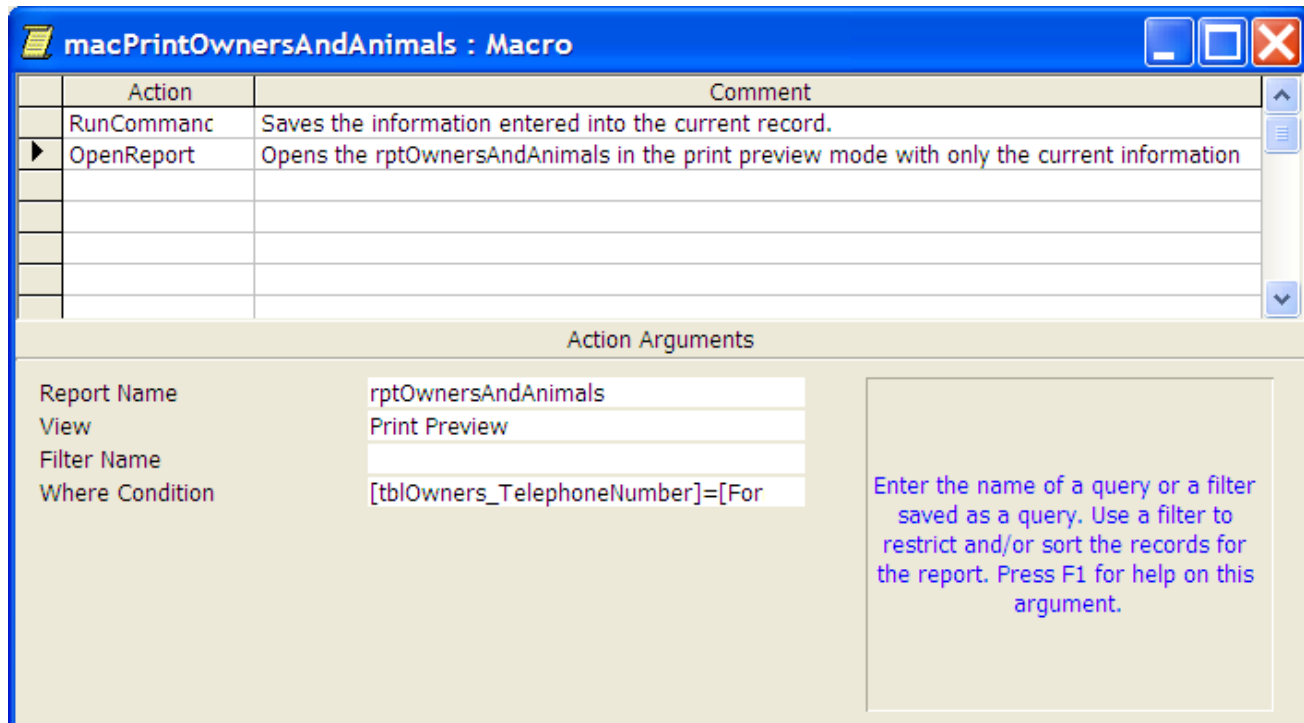
Where Condition – compare the field on the report to the matching field on the form. [*report field name*]=[Forms]![*form name*]![*form field name*]

i.e.

[tblOwners_TelephoneNumber]=[Forms]![frmOwners]![TelephoneNumber]

The Comment fields are optional.

Save the macro.



Using Creating the Print Only Current Information Macro

Add a command button to your form and attach the macro to the “on click” event.

Visual Basic for Applications

Visual Basic for Applications is included with MS Access. Visual Basic code can be written as a module that can be used globally by all forms. Visual Basic can also be written within a form where only that form has access to the code.

Code as Module

This code is a function that checks to see if a form is open. This code can be copied directly into a module without any changes. As a module, all forms will be able to pull this function.

```
Public Function IsOpen(ByVal strFormName As String) As Boolean
    ' Returns True if the specified form is open in Form view.

    Const conDesignView = 0
    Const conObjStateClosed = 0

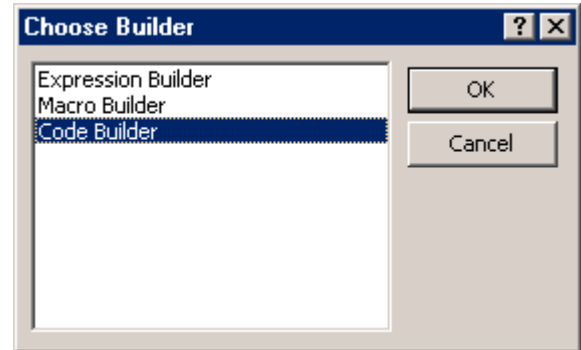
    IsOpen = False
    If SysCmd(acSysCmdGetObjectState, acForm, strFormName) <> _
        conObjStateClosed Then

        If Forms(strFormName).CurrentView <> conDesignView Then
            IsOpen = True
        End If
    End If
End Function
```

Code within a form

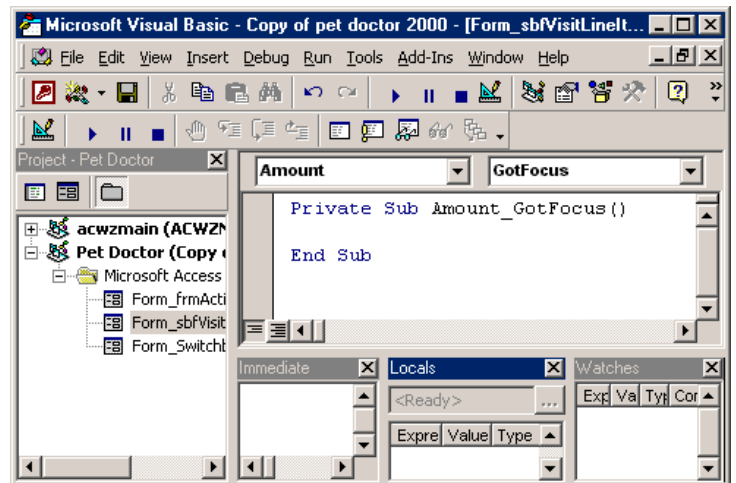
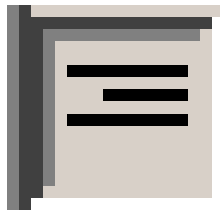
To enter Visual Basic for Applications code into a form, open the form in the design mode. Choose the event on which to add the code. Click on the ellipse button.

In the Choose Builder dialog box, choose Code Builder.



The Microsoft Visual Basic window will open. The Private Sub for the event will be open with the name of the sub and the “End Sub” already entered.

Use the button in the bottom of the sub window to change the view to show just the one sub.



Visual Basic Code Samples

Sample code can be copied into an existing database. The word in italic need to be changed to match your database objects.

Copies values from one field into another at time of data entry.

In order for this to work, it must be placed on the **form events** "before update". A refresh macro must also be used on the lost focus of the first field entered.

```
Private Sub Form_BeforeUpdate(Cancel As Integer)
If IsNull(field2) Then
field2 = field1
Else
End If
End Sub
```

Inserts key from previous form into newly opened form.

Upon opening formB with a call from formA, formB will pull the key from formA into formB. To use this coding, a module must exist with the IsOpen function created.

```
Private Sub Form_Open(Cancel As Integer)
If IsOpen(" formA") Then
Dim fnum As String
key.DefaultValue = "Forms!formA!key"
Else
End If
End Sub
```