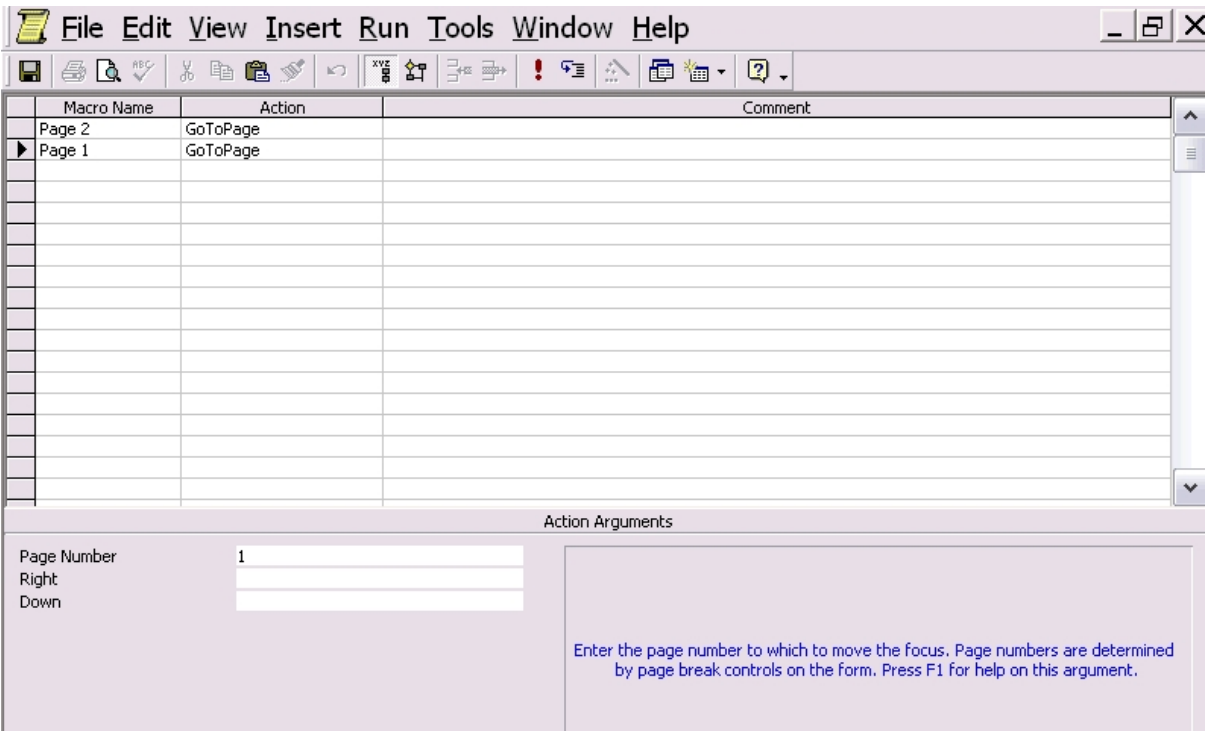


Macros.

Written by W.J.Pais

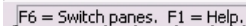
A macro is a set of one or more actions that each perform a particular operation, such as opening a form or printing a report. Macros can help you to automate common tasks. For example, you can run a macro that prints a report when a user clicks a command button.



On Data Base window, click Macro and click New. Let us make our first Macro to jump to page two and another to jump back to page 1.

It will be one MACRO named JUMP with two macro sub names, with different actions.

Written by W.J.Pais



and a cross-section of the population.

[illegible]

Working on this, I found out that the code was a little bit off. I specified the data like the starting water to

Macros.

Written by W.J.Pais

Macro Name	Condition	Action	Comment
			Attached to the Customer Phone List form.
			Attached to AfterUpdate event of CompanyNameFilter option group.
Alpha Buttons	[CompanyNameFilter]	ApplyFilter	Filter for company names that start with A, Ä, Å, Å, or Ä.
	[CompanyNameFilter]	ApplyFilter	B
	[CompanyNameFilter]	ApplyFilter	C or Ç
	[CompanyNameFilter]	ApplyFilter	D
	[CompanyNameFilter]	ApplyFilter	E, È, É, Ê, or Ë
	[CompanyNameFilter]	ApplyFilter	F
	[CompanyNameFilter]	ApplyFilter	G
	[CompanyNameFilter]	ApplyFilter	H
	[CompanyNameFilter]	ApplyFilter	I, Î, Ï, or Ī
	[CompanyNameFilter]	ApplyFilter	J
	[CompanyNameFilter]	ApplyFilter	K
	[CompanyNameFilter]	ApplyFilter	L
	[CompanyNameFilter]	ApplyFilter	M
	[CompanyNameFilter]	ApplyFilter	N, or Ñ
	[CompanyNameFilter]	ApplyFilter	O, Ö, Ó, Ô, Õ, or Ö
	[CompanyNameFilter]	ApplyFilter	P
	[CompanyNameFilter]	ApplyFilter	Q
Action Arguments			
Enter a macro name in this column.			

Open Form Macro

Action: **Open Form.**

Action Arguments:

Form Name: as in the database

View: Form

Filter Name: Press F1 (CAN BE LEFT BLANK)

Where Condition is : Build from Expression Builder - click on 3 asterisk at the right (CAN BE LEFT BLANK)

Data Mode: Add, Edit or Read Only. (CAN BE LEFT BLANK)

Macros.

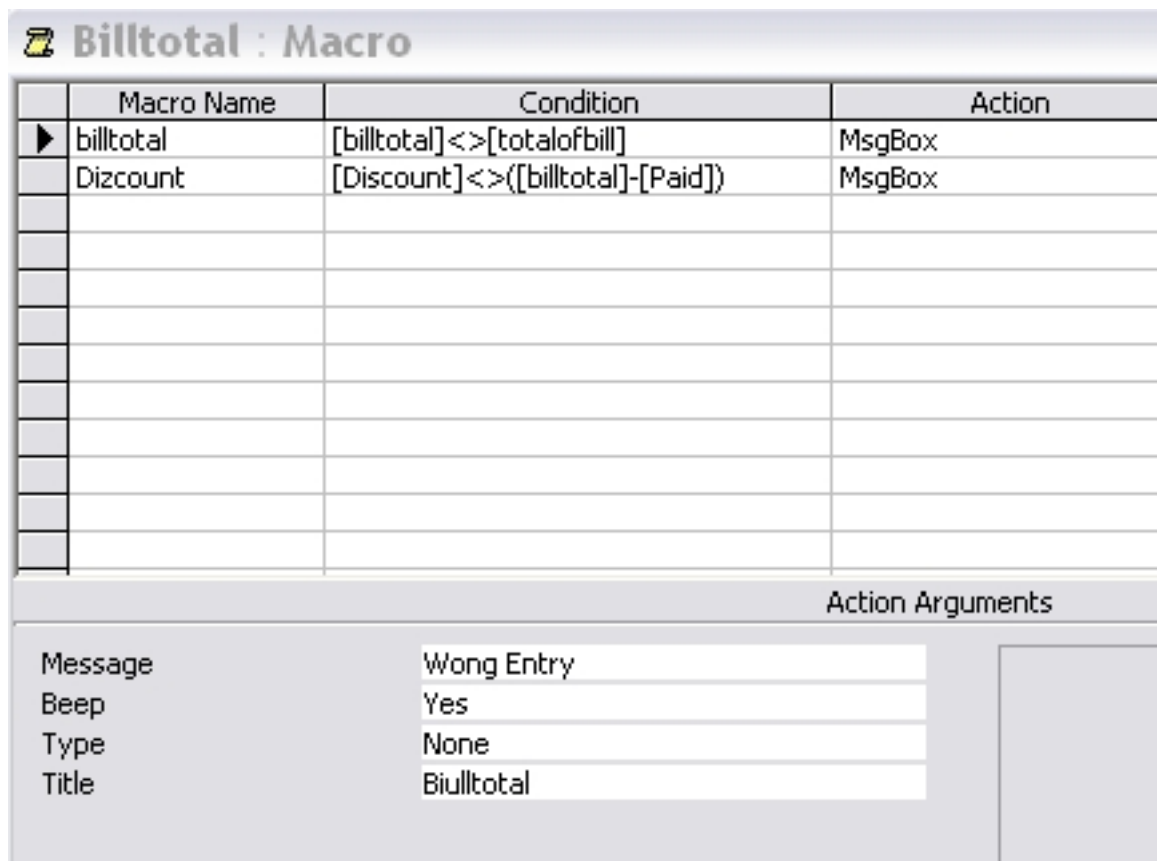
Written by W.J.Pais

Window Mode. NORMAL

Same applies to Open Table, Open Report etc. Always check by clicking F1 which brings up HELP

Conditional Macro

Some times you wish to warn the operator that he has input a wrong data, and a Message Box should apper and give that warning.

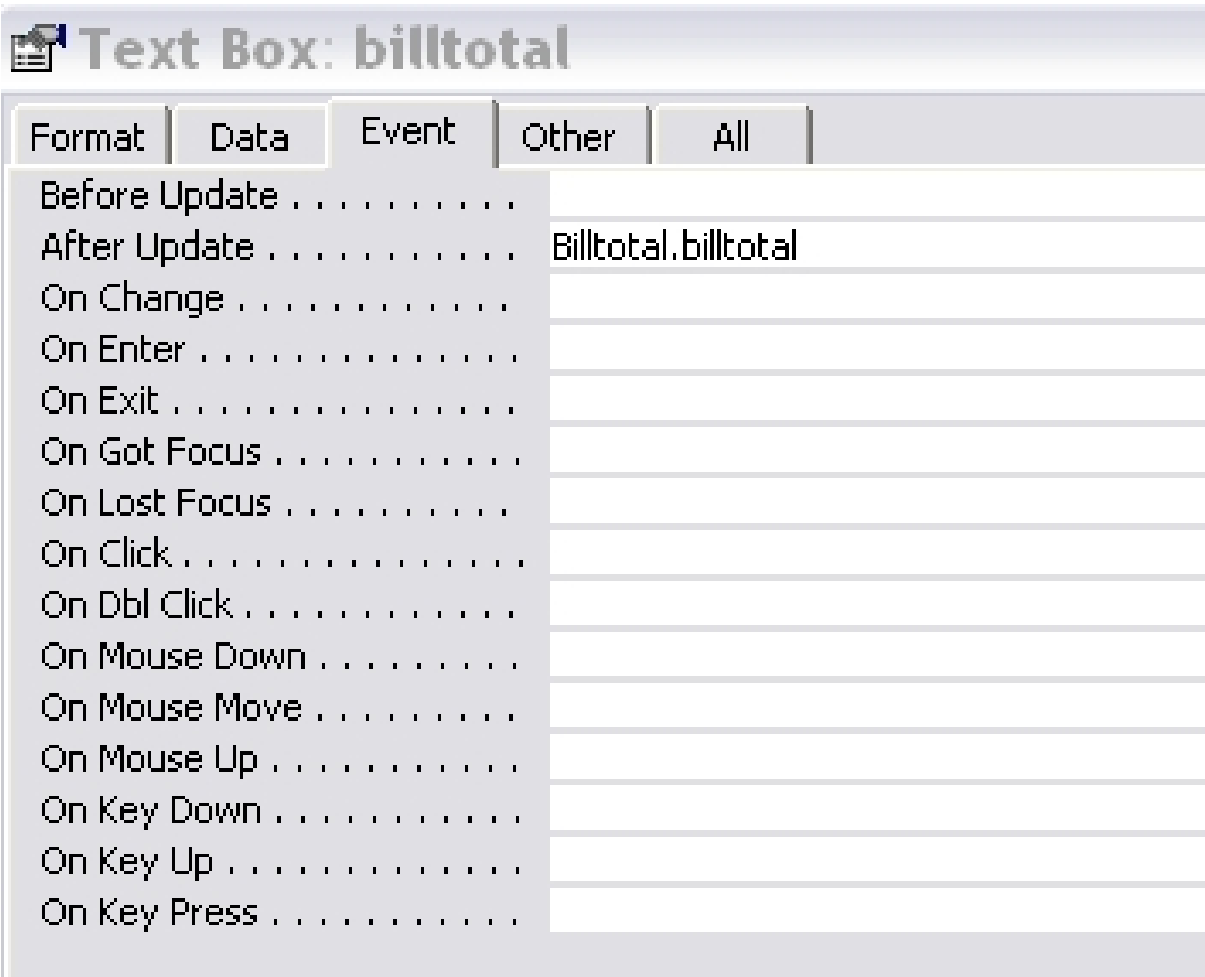


Macro Name	Condition	Action
billtotal	[billtotal]<>[totalofbill]	MsgBox
Dizcount	[Discount]<>([billtotal]-[Paid])	MsgBox

Action Arguments	
Message	Wong Entry
Beep	Yes
Type	None
Title	Biulltotal

A Macro name is given which you enter in the Events Tab, under "After Update". You can choose to use the Command Button wizard and choose "Run Macro" and give the Macro Name. In this case the macro name is "billtotal" and the sub name is billtotal and discount. It

will appear as billtotal.billtotal or billtotal.discount



When the bill total which you want to enter into the table does not match the bill total generated by the computer, because the input data differs then the message will appear as

wrong entry.

The same appears when the discount figure is not equal to bill amount minus the amount paid.

Events: Making your database objects work together

What is an event?

An 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.) is a specific action that occurs on or with a certain **object**

Objects seen in a Microsoft Access database:

- tables, queries, forms, reports, macros, and modules
- objects that are defined by the system, such as the MSysObjects table

Objects seen in a Microsoft Access project:

- objects stored in the Access project file: forms, reports, macros, and modules
- objects stored in the Microsoft SQL Server database: tables, table properties (such as indexes, triggers, keys, constraints, defaults, rules, and user-defined data types), views, stored procedures, and database diagrams
- objects defined by the system, such as the sysindexes table

Objects seen in both Access databases and Access projects:

- data access pages, which are shortcuts (displayed in the Database window) to corresponding HTML files stored in the file system
- information from another application, such as a chart (graph) or a drawing

Microsoft Access can respond to a variety of events: mouse clicks, changes in data, forms opening or closing, and many others. Events are usually the result of user action.

By using either an

event procedure

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

or

a macro

An action or a set of actions you can use to automate tasks.

you can add your own custom response to an event that occurs on a form, report, or **control**

A graphical object, such as a text box, check box, command button, or rectangle, that you place on a form, report, or data access page in Design view. A control can display data, perform an action, or make the form, report, or data access page easier to read.

Suppose you want a Product Details form to open when you click a Product Details command button on an Orders form. The following examples show how to do this either by using an event procedure or by using a macro.

Responding to a Click event by using an event procedure

When you create an event procedure for an object, Access adds an event procedure template named for the event and the object to the **form module**

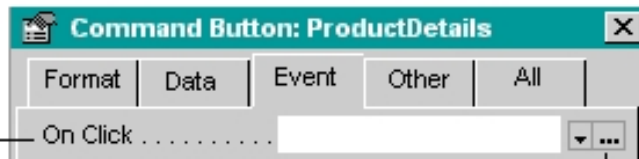
A module that includes code for all event procedures triggered by events occurring on a specific form or its controls. Additionally, a form module can contain other Sub or Function procedures that relate to the form. A form module is a type of class module.

or **report module**

A module that includes code for all event procedures triggered by events occurring on a specific report or its controls. Additionally, a report module can contain other Sub or Function procedures that relate to the report. A report module is a class module.

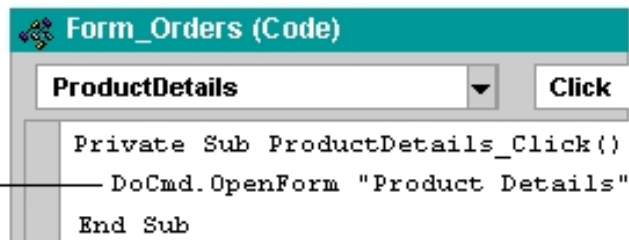
All you need to do is add code that responds in the way you want when the event occurs for the form or report.

To make an event procedure run when a user clicks the command button, click the OnClick property of the command button ...



... click the Build button, and then click Code Builder to add the Visual Basic code to the event procedure.

The ProductDetails_Click event procedure uses the OpenForm method to open the Product Details form.



Responding to a Click event by using a macro

To have a macro run in response to an event, open the

property sheet

In a Microsoft Access database, the window in which you can view or modify the properties of tables, queries, fields, forms, reports, data access pages, and controls.

In a Microsoft Access project, the window in which you can view or modify the properties of forms, reports, and data access pages. You use property pages to modify the properties of tables, queries, and other Microsoft SQL Server database objects.

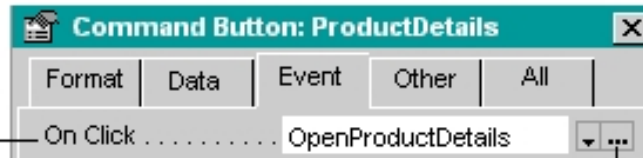
On a data access page in an Access database or Access project, the Property Toolbox is the window in which you set properties for Microsoft Office Web Components. You can set these same properties in the property sheet instead; the property sheet includes additional properties not displayed in the Property Toolbox.

for the form, report, or control. Find the property that corresponds to the event, and set it to the macro you want to run.

Macros.

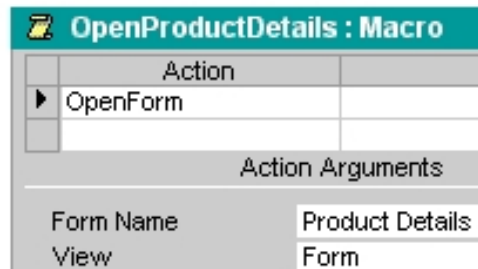
Written by W.J.Pais

To make a macro run when a user clicks the command button, click the OnClick property of the command button ...



... click the Build button, and then click Macro Builder.

The OpenProductDetails macro uses the OpenForm action to open the Product Details form.



Run a macro or event procedure for an event on a form, report, or control

Microsoft Access responds to many types of [events](#) on forms, reports, and **controls**, including mouse clicks, changes in data, and forms or reports being opened or closed.

1. Open the form or report in **Design view**.
2. Create a macro or an **event procedure**. For example, you can create a macro or event procedure that displays a message when you click a command button.
3. Set the appropriate event property of the form, report, or control to the name of the macro, or to **[Event Procedure]** if you're using an event procedure.

For example, to use a macro to display a message when you click a command button, set the command button's **OnClick** property to the name of a macro that displays the message. To use an event procedure, create a Click event procedure for the command button, and then set its **OnClick** property to **[Event Procedure]**.

Notes

- **Create a macro**

Macros.

Written by W.J.Pais

1. In the [Database window](#) , click **Macros**

Displays all macros in the current Microsoft Access database or Microsoft Access project. Use the buttons on the Database window toolbar to run the selected macro, edit the selected macro, or create a new macro.

under **Objects**.

2. Click the **New** button on the Database window toolbar.

3. In the **Action** column, click in the first cell and then click the arrow to display the action list

The list appears when you click the arrow in the Action column of the Macro window.

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 arguments

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

for the action, if any are required. For tips on setting action arguments, [click](#) .

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 object**, 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 side by side on your screen; in the Database window under

Objects

, click the object type you want, click the object, and then drag it to an action row. Dragging a macro or stored procedure adds an action that runs the macro or procedure, while dragging other database objects adds an action that opens the object.

database objects

In a Microsoft Access database, the types of database objects available are tables, queries, forms, reports, macros, and modules.

For a Microsoft Access project connected to a SQL Server database, the database objects are tables (including indexes, triggers, keys, constraints, defaults, rules, and user-defined data types), views, stored procedures, and database diagrams. The Access project objects are forms, reports, macros, and modules.

From both Access databases and Access projects, you can also create data access page objects, which are stored as separate files outside the Access database or Access project file.

8. For information on creating an event procedure, [click](#)

```
Sub HelpPopup(sFile,sID) L_SecurityT1_ErrorMessage="Help can't show you this
procedure because the security setting in your browser is set too high"
L_SecurityT2_ErrorMessage="or the ActiveX control Ouactrl.ocx didn't install correctly."
L_SecurityE1_ErrorMessage="- Select a lower security setting in your browser"
L_SecurityE2_ErrorMessage="- If you receive this message after selecting a lower security
setting, please see your system" L_SecurityE3_ErrorMessage=" administrator for help
troubleshooting the installation of the ActiveX control Ouactrl.ocx"
L_SecurityE4_ErrorMessage=" located in the folder you installed Microsoft Office to."
sSecurityMSG=L_SecurityT1_ErrorMessage & chr(13) & L_SecurityT2_ErrorMessage & chr(13)
& chr(13) & L_SecurityE1_ErrorMessage & chr(13) & L_SecurityE2_ErrorMessage & chr(13) &
L_SecurityE3_ErrorMessage & chr(13) & L_SecurityE4_ErrorMessage
```

```
L_App_DialogTitle="Microsoft Office Help" On Error Resume Next r=oua.HelpPopup(sFile,sID)  
If Err0 Then MsgBox sSecurityMSG,48,L_App_DialogTitle End Sub
```

Tips on setting action arguments in a macro

After you add an action to a macro, you set the arguments for the action in the lower portion of the Macro window. These arguments give Microsoft Access additional information on how to carry out the action.

Here are some tips on setting action arguments:

- You can type a value in an argument box, or in many cases you can select a setting from a list.
- In general, it's a good idea to set action arguments in the order they're listed, because choices for one argument may determine those for arguments that follow.
- If you add an action to your macro by dragging a database object from the Database window, Microsoft Access automatically sets appropriate arguments for that action.
- If an action has an argument that calls for the name of a database object, you can set the argument and the corresponding object type argument automatically by dragging the object from the Database window to the argument box.
- You can use an expression preceded by an equal sign (=) to set many action arguments.
- For more information on setting a specific action argument, click the action argument and press F1.

expression

Any combination of operators, constants, literal values, functions, and names of fields (columns), controls, and properties that evaluates to a single value. You can use expressions as settings for many properties and action arguments; to define calculated controls in forms, reports, and data access pages; to set criteria (search conditions) or define calculated fields in queries; and to set conditions in macros. In a Microsoft Access project, you can use expressions to define check constraints.

You also use expressions in Microsoft Visual Basic.

```
Sub HelpPopup(sFile,sID) L_SecurityT1_ErrorMessage="Help can't show you this
procedure because the security setting in your browser is set too high"
L_SecurityT2_ErrorMessage="or the ActiveX control Ouactrl.ocx didn't install correctly."
L_SecurityE1_ErrorMessage="- Select a lower security setting in your browser"
L_SecurityE2_ErrorMessage="- If you receive this message after selecting a lower security
setting, please see your system" L_SecurityE3_ErrorMessage=" administrator for help
troubleshooting the installation of the ActiveX control Ouactrl.ocx"
L_SecurityE4_ErrorMessage=" located in the folder you installed Microsoft Office to."
sSecurityMSG=L_SecurityT1_ErrorMessage & chr(13) & L_SecurityT2_ErrorMessage & chr(13)
& chr(13) & L_SecurityE1_ErrorMessage & chr(13) & L_SecurityE2_ErrorMessage & chr(13) &
L_SecurityE3_ErrorMessage & chr(13) & L_SecurityE4_ErrorMessage
L_App_DialogTitle="Microsoft Office Help" On Error Resume Next r=oua.HelpPopup(sFile,sID)
If Err0 Then MsgBox sSecurityMSG,48,L_App_DialogTitle End Sub
```

Create an event procedure

Microsoft Access makes it easy to run code in response to an action performed by the user. When you set an event property for a form, report, or control to [Event Procedure], Microsoft Access creates the event procedure template for you. All you have to do is add the code you want to run in response to a particular event on the form, report, or control. For example, you can have code run when a user clicks a command button or changes data.

1. Open a form or report in form or report Design view.
2. Display the property sheet for the form, report, or a section or control on the form or report.
3. Click the **Event** tab.

4. Click the event property for the event that you want to trigger the procedure. For example, to display the event procedure for the Change event, click the

OnChange

property.

5. Click **Build** next to the property box to display the **Choose Builder** dialog box.

6. Double-click **Code Builder** to display the event procedure **Sub** and **End Sub** statements in the form

module

or

report module

. These statements define, or declare, the event procedure.

Microsoft Access automatically declares event procedures for each object and event in a form or report module by using the **Private** keyword to indicate that the procedure can be accessed only by other procedures in that module.

7. Add the code to the event procedure that you want to run when the event occurs. For example, to produce a sound through the computer's speaker when data in the CompanyName text box changes, add a **Beep** statement to the CompanyName_Change event procedure, as follows: Private Sub CompanyName_Change()

Beep

End Sub

The event procedure runs each time the Change event occurs for the object.