

# **Access Database Macro Magic**

## **Part 1**



**By Paul Barnett**

Beginning Access Database Macros .....	3
What is an Access macro? .....	3
The macro designer .....	4
Creating our first macro .....	6
A macro to filter a form by a value .....	6
Create a macro to copy and back up an object.....	10

**Get the Full Version of This E-book**

<http://www.access-databases.com/ebooks/macros/>

## Beginning Access Database Macros

For beginners of Microsoft Access databases VBA programming can be a daunting task. Fortunately Access offers an alternative in the shape of macros.

### What is an Access macro?

A macro can best be thought of as an action or group of actions which can perform a database related operation. For example we can create a macro for opening a form, printing a report or automating many other tasks.

We can create a single macro containing a set of actions or a macro group. We can also add conditions to our macros in order to provide greater flexibility.

One complaint of MS Access macros has always been that they do not have any error checking. This is not now an issue with newer versions of Access – 2007 onwards.

Another issue is that if a macro fails it does not offer a good enough explanation. We get the standard Access macro error messages and cannot add our own.



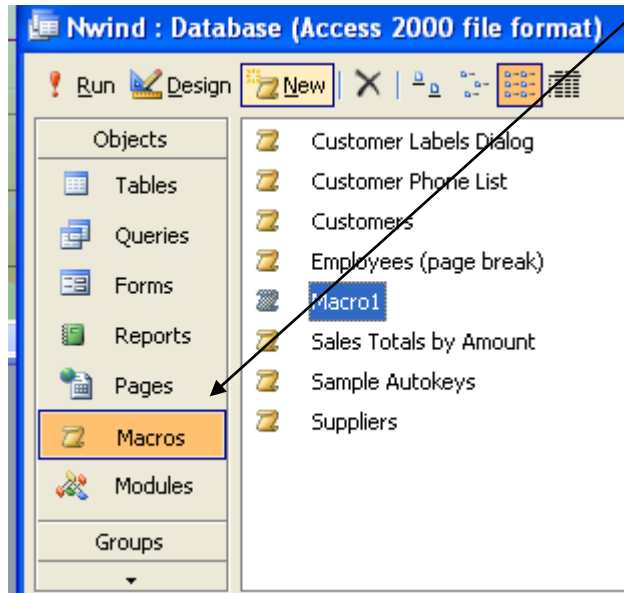
Apart from the issues mentioned, Access macros enable us to create some powerful functionality.

This guide will enable you to get started and experience the power of MS Access macros.

## The macro designer

Let's create a new macro.

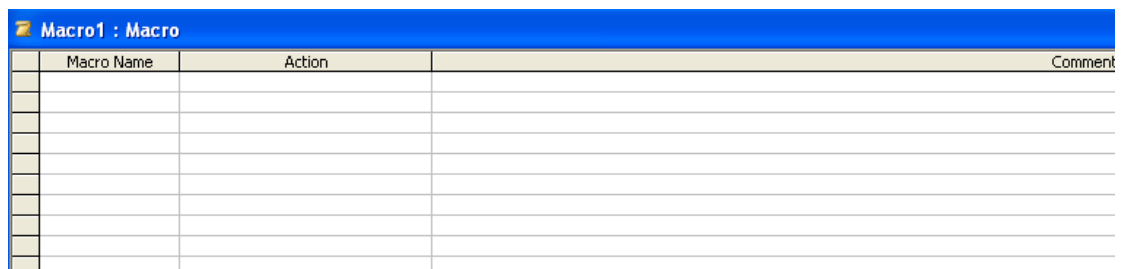
Go to the database window and select the 'Macros' option.



Now select the new option.



You will now see the macro designer window.



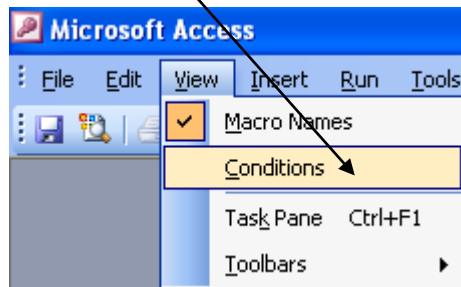
The screenshot shows the 'Macro1 : Macro' designer window. It contains a table with three columns: 'Macro Name', 'Action', and 'Comment'. The table is currently empty, with only the header row visible.

Macro Name	Action	Comment

There are 3 columns shown

- Macro Name
- Action
- Comment

You can also add another column by going up to the view menu and selecting conditions.



So now you get

A screenshot of the 'Macro1 : Macro' window in Microsoft Access. It displays a table with four columns: 'Macro Name', 'Condition', 'Action', and 'Comment'. The table is currently empty, with only the header row visible.

Macro Name	Condition	Action	Comment

Let's now go over what each column is used for.

### **Macro Name**

As the title suggests this is where we give the macro a descriptive name.

### **Condition**

This is where you specify what conditions the macro needs to meet when executed. For example a condition could be - does the order have an order number? If it does, then execute the macro actions.

### **Action**

This is the macro task to be performed. For example 'CopyObject' will copy an object such as a table or query to the current or another database.

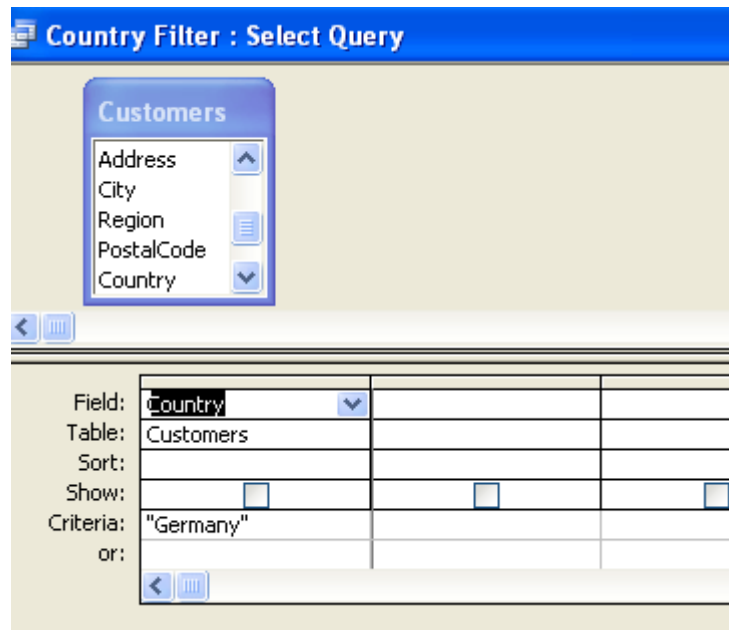
### **Comment**

A description of what the macro actually does. This is optional, but can be handy for a new developer working on an existing system.

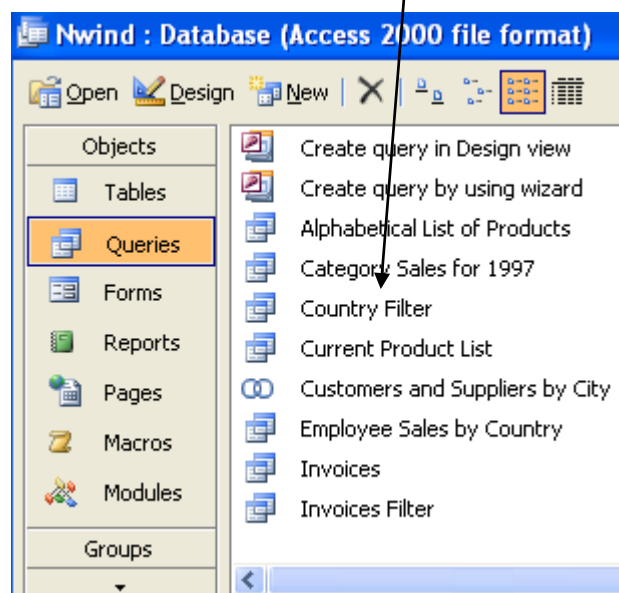
## Creating our first macro

### A macro to filter a form by a value

I have created and saved a query which filters records on my form by country. The country is set to Germany.

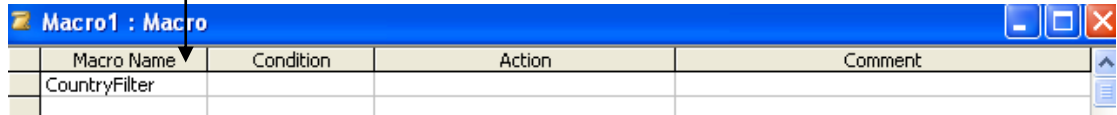


I have called the query 'Country Filter'.



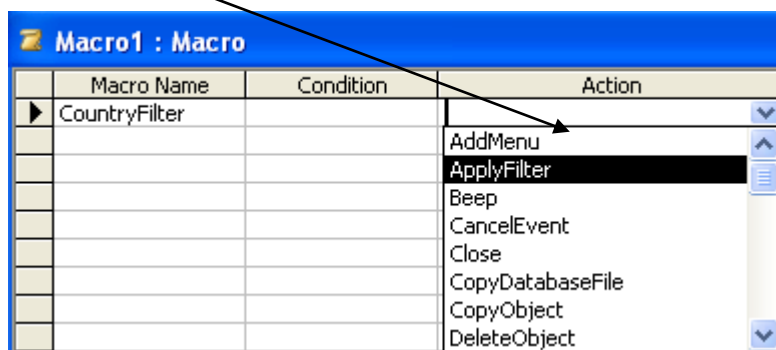
Back at the macro designer window.

In the macro name section I have typed in 'CountryFilter'.



I am not adding anything in the condition section.

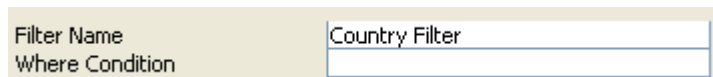
In the action section I select the 'ApplyFilter' action.



This brings up some properties at the bottom of the macro window.



In filter name I now enter the name of the saved query.

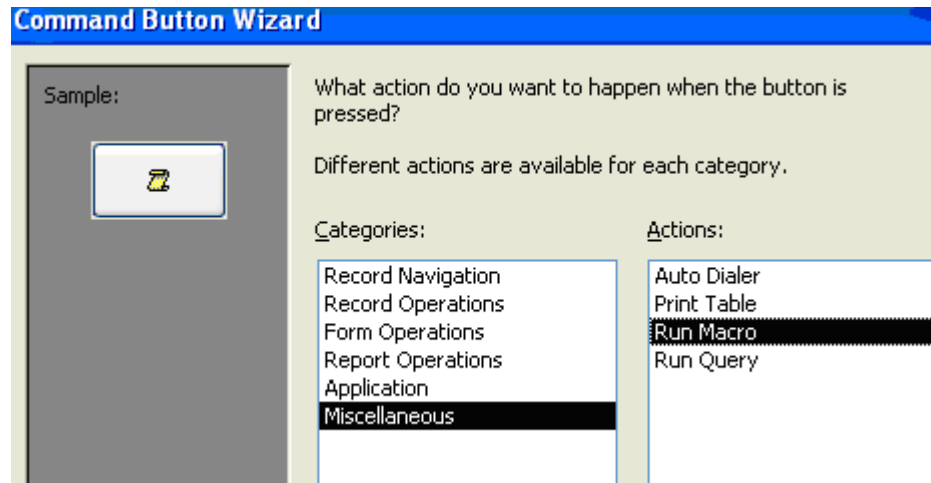


Save the macro as 'Macro1'.

Close the macro designer window.

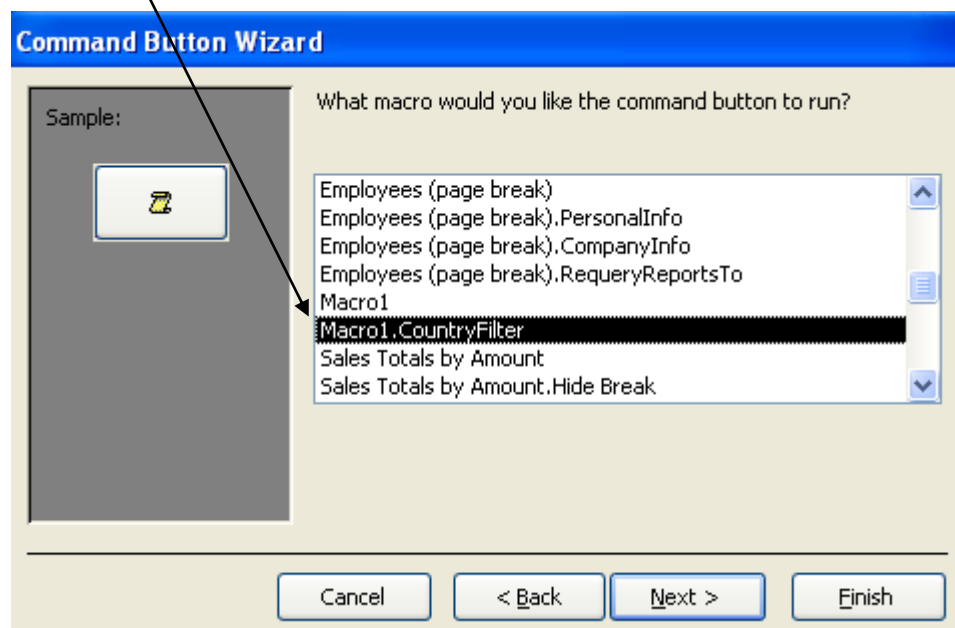
On my form I create a new command button using the wizard.

I select the miscellaneous category and the run macro action.



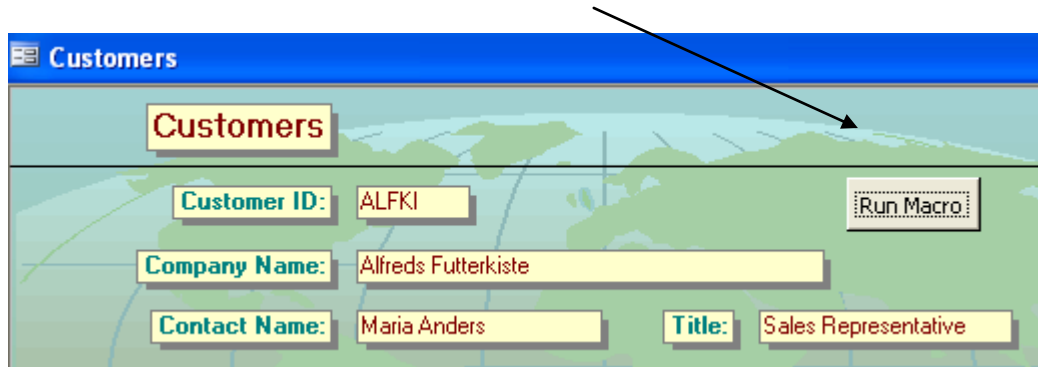
I now select the macro I want to use behind the button.

Notice it gives the macro name and the action within it.

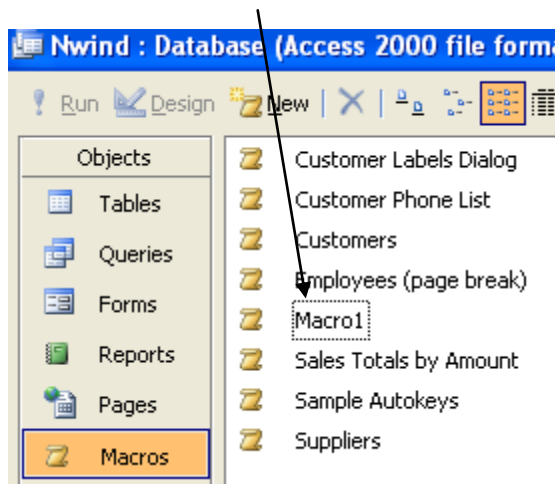


So now I have wired up my macro to a button on a form.

When the button is clicked my macro will fire and will apply a filter to the form for all records in Germany.



We have our Macro1 saved which applies the filter.



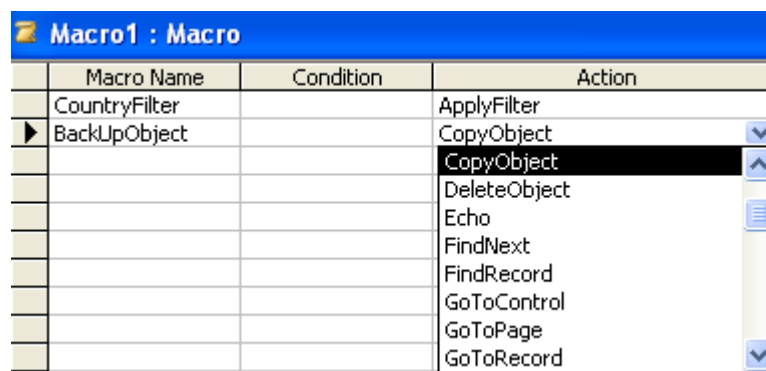
We could set up more macros and call them Macro2,3,4,5 etc. However, it is better to set up a macro group to hold more than one macro. Each one is identified and called by name.

## Create a macro to copy and back up an object

I am now going to use the 'CopyObject' macro action.

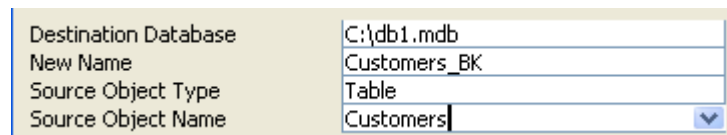
This action will copy objects such as tables and queries to the same or another database.

I select the original macro I used and go into design mode again. Under the macro name column I enter a new macro name 'BackUpObject'. I also select the 'CopyObject' macro action.



Macro Name	Condition	Action
CountryFilter		ApplyFilter
BackUpObject		CopyObject

At the bottom of the window I set the following:



Destination Database	C:\db1.mdb
New Name	Customers_BK
Source Object Type	Table
Source Object Name	Customers

Destination database is where I want the object saved. I enter a path here to the database.

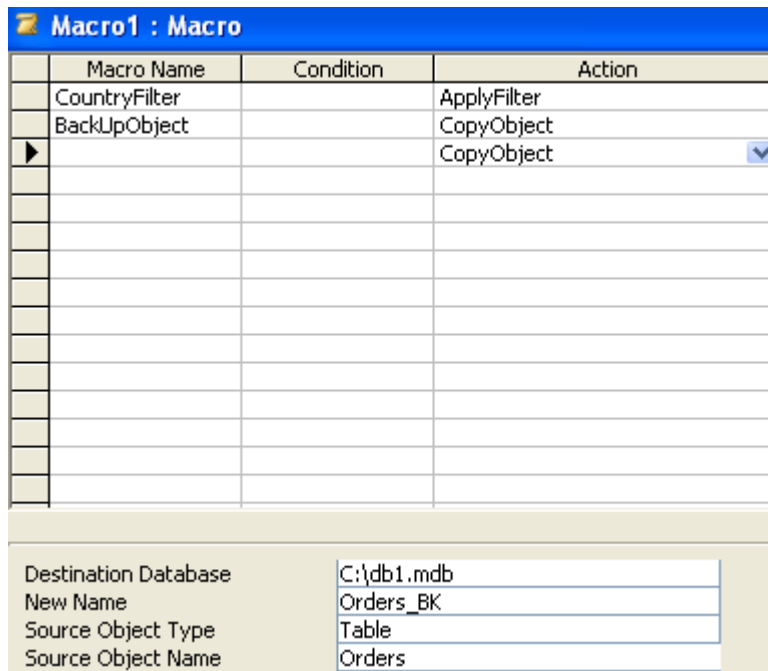
New name is any new name I want to give to this object. As I am making a backup in another database I called it 'Customers\_BK'.

In the source object type tell it the type of object you are selecting. In this case my type is a table.

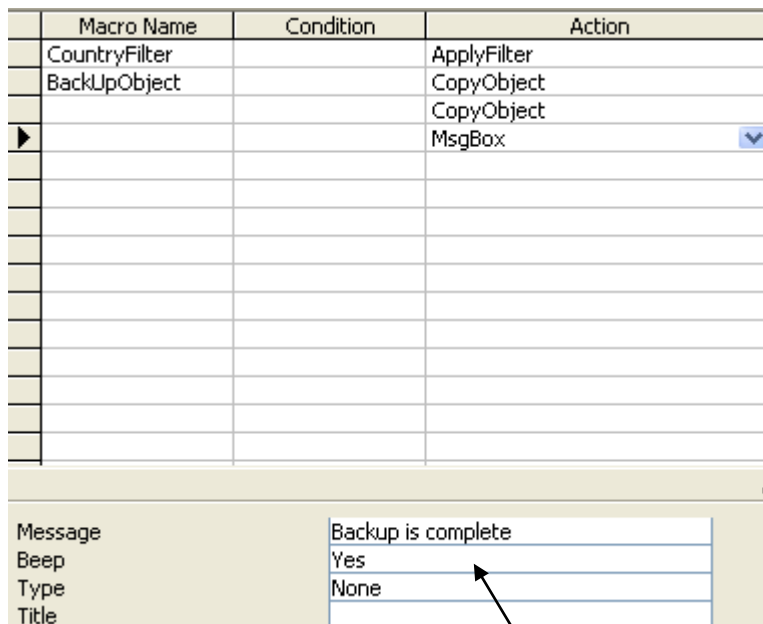
The source object name will be the current name of the object in the database. In this case it is my customers table.

You may also want to back up additional tables. The procedure would be the same. Just add more 'CopyObject' actions to this group.

I've also added my orders table in, so now I have two 'CopyObject' actions.



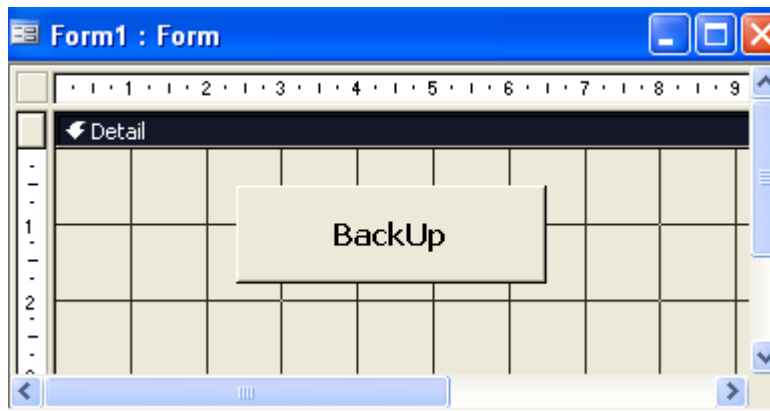
I've also added a message box action to display a message when the operation is complete.



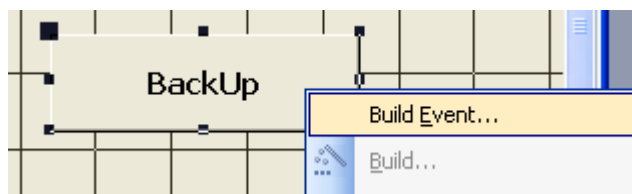
The message I want displayed is 'Backup is complete'.

I now save and run the macro.

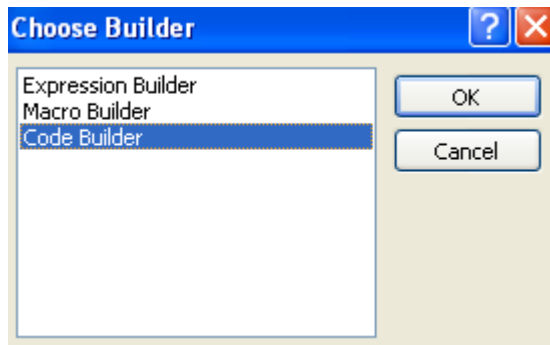
I create a simple form with a command button.



Right click on the button and select 'Build Event'.



Now select code builder and click ok.



The code window will appear.

My procedure looks like the following.

```
Private Sub Command0_Click()  
    DoCmd.RunMacro ("Macro1.BackupObject")  
End Sub
```

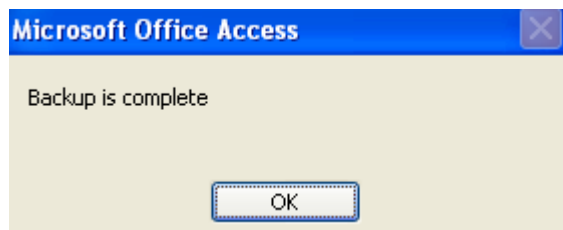
The line that calls the macro is

```
DoCmd.RunMacro ("Macro1.BackupObject")
```

Macro1 is the name of the saved macro shown in the database window. BackupObject is the name of the group inside the macro1 that contains the actions.

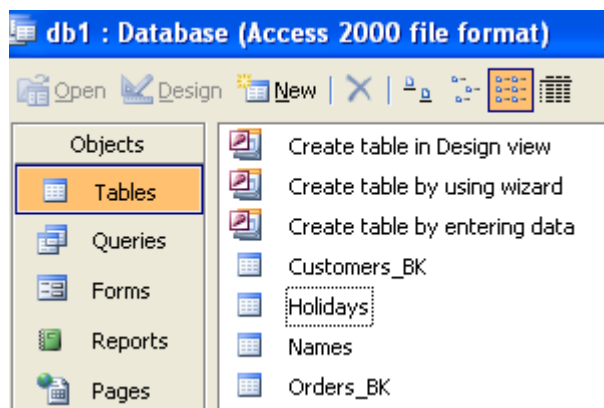
The other groups will not be run, only the one referred to in the code above.

When the macro has finished running the message specified in the macro is displayed.



And in my external database the tables have been saved.

- Customers\_BK
- Orders\_BK



Objects such as forms and reports can also be copied by using the 'CopyObject' action.

**Get the Full Version of This E-book**

<http://www.access-databases.com/ebooks/macros/>

© 2010 Paul Barnett. All Rights Reserved.  
No part of this publication may be reprinted  
or reproduced without permission.