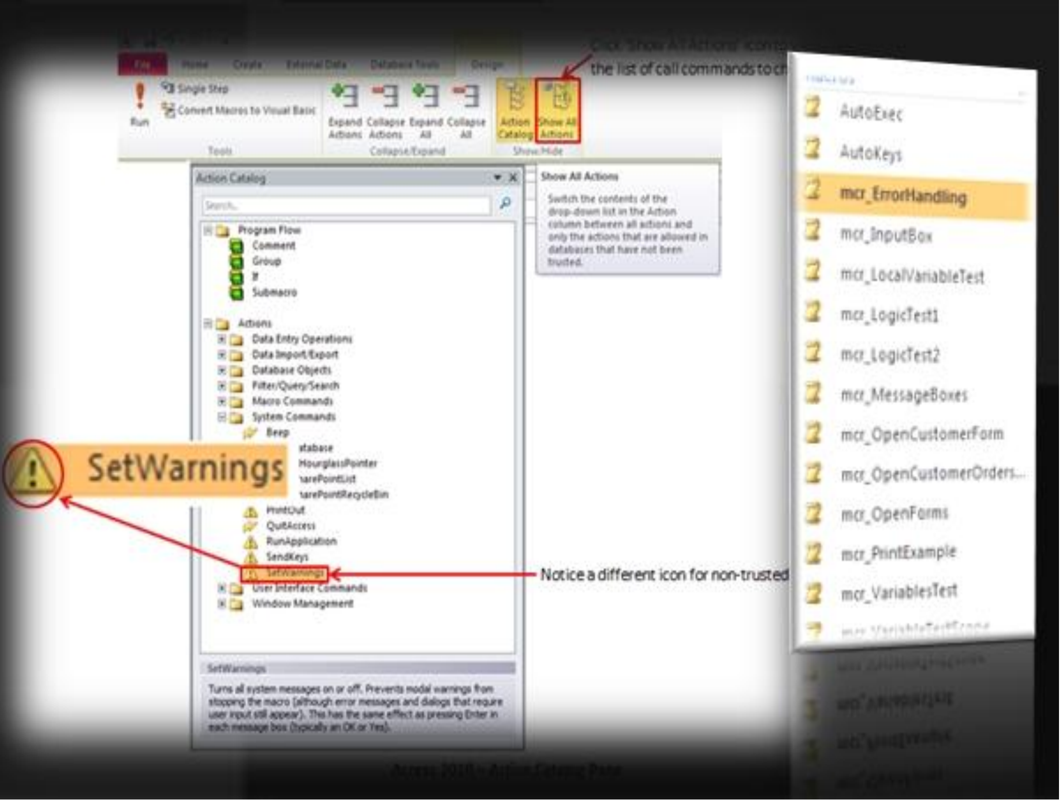
Understanding & Using

MICROSOFT ACCESS MACROS

Ben Beitler



Click 'Show All Actions' icon to the list of call commands to ch

Show All Actions

Switch the contents of the drop-down list in the Action column between all actions and only the actions that are allowed in databases that have not been trusted.

Notice a different icon for non-trusted

SetWarnings

Turns all system messages on or off. Prevents modal warnings from stopping the macro (although error messages and dialogs that require user input still appear). This has the same effect as pressing Enter in each message box (typically an OK or Yes).

Understanding and Using Microsoft Access Macros

Firstly, I would like to thank you for purchasing this Access database eBook guide; a useful reference guide on understanding and using Microsoft Access database macros.

Please keep this as a reference and a reminder when building macros in Microsoft Access, in which some of the references can be applied to versions as far back as Access 2000 though some of the screenshots will vary and therefore be different in places (*and where applicable*).

In this guide, I will discuss and show you, from the ground up, how to build macros and interact with the database application. If you're a new user to macros or other forms of programming with Access objects, then you're in the right place as I will give you the *heads up* and get you going in quick time.

An exclusive feature for Microsoft Access 2010 users (*known as Data Macros*) is also included in this guide to help extend the power of automating your Access database application.

FREE DOWNLOAD

If you require some sample data to test the techniques and illustrations in this document, you can go and download my sample Access database file consisting of six potentially related tables which these examples are all based on.

[DOWNLOAD LINKS NOT AVAILABLE IN PREVIEW]

CONTENTS

Overview of Microsoft Access Macros

Here's a quick overview of what Access macros are all about and when you should use them in your Access database application.

When programming in Microsoft Access, you have essentially two ways of accomplishing this task. You can use VBA (*Visual Basic for Applications*) or Access macros.

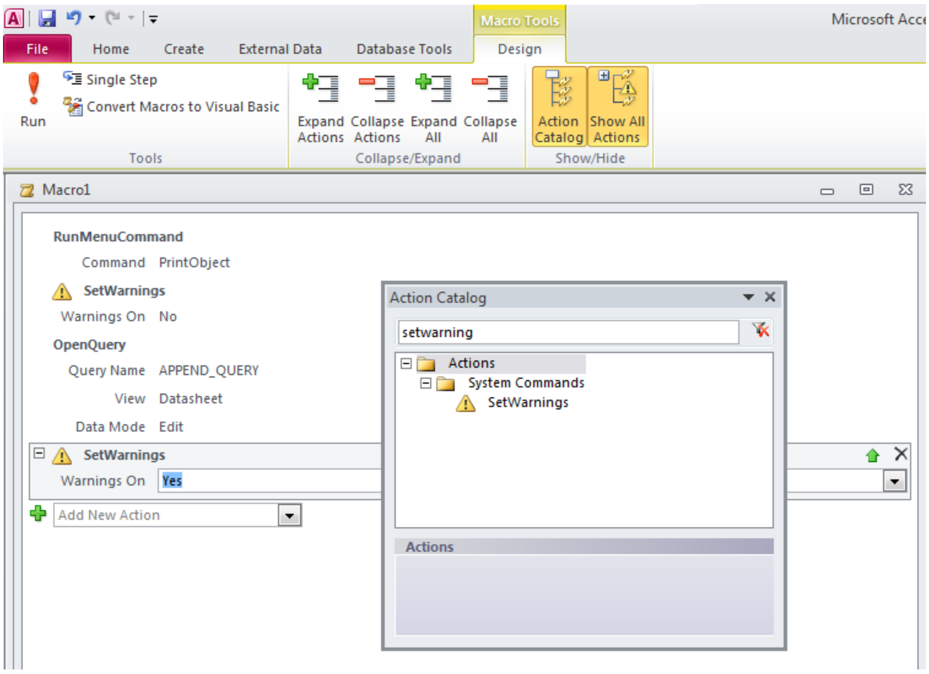
> ***Note:*** *The term 'macro' may be a little confusing when talking about other Microsoft Office applications including Excel, Word and PowerPoint as it refers to VBA (the code) which is one of the same. In Access however, they are two completely different processes and the term 'macro' here refers to a lower set of automating tasks (having its own language reference) leaving VBA to handle more complex programming functionality.*

Tasks that macros typically are used for may include:

- Opening a form or a report.
- Executing a query.
- Setting a value to a control (*like a Textbox*).
- Importing data from external spread sheets.
- Exporting data to a PDF file.
- Quitting the application.

Every time you click on a button, open a form or close a report you can call a macro to perform that task (*a procedure*) and start to automate your application. A macro is the simple programming alternative to using the more powerful VBA programming feature of Microsoft Access.

The level of programming knowledge therefore required is '*zero'* and the only skill needed is having the time to learn and having a logical mind (*which by the way, we all have!*).



RunMenuCommand
Command PrintObject
SetWarnings
Warnings On No
OpenQuery
Query Name APPEND_QUERY
View Datasheet
Data Mode Edit
SetWarnings
Warnings On Yes
Add New Action

setwarning

Continues/…

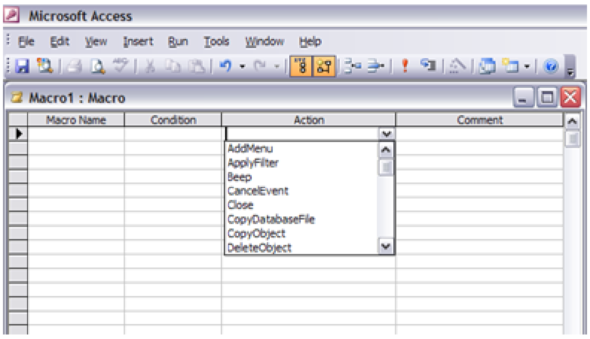
Originally, Access macros was the only way to program and automate your application (*which goes way, way back to the mid 1990's*) and has remained ever present since. During the different releases of Access over the years, Microsoft took a view and though it promoted macros as being the first choice to automating a database it soon leaned towards the more powerful approach of VBA. Now Access 2010 has included a newer set of macro commands with the addition of a '*Data Macros*'; it has swung the pendulum back to using macros as the first choice.

The one missing element to Access macros which some readers may have noticed is that unlike with Excel, Word or PowerPoint, you do not have a '*Macro Recorder*' feature (*which generated VBA for you*) and therefore means that you will start from a blank canvas and set your commands and parameters manually.

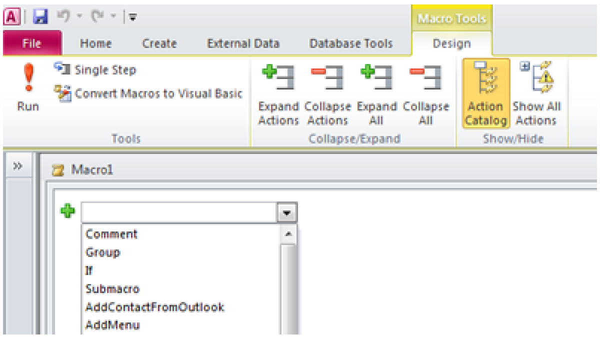
Today, macros still have an important role to play within your application especially for the latest release (*2010*) and can even be combined with VBA to bring a happy balance to controlling the simplest of tasks through to the more advanced routines that VBA is designed to do.

Finally, macro commands have changed (*and been increased*) from version to version and readers will need to be aware that I'm using Microsoft Access 2010 to show the latest features and the complete command list of which there are now 86 key commands. Earlier versions may have different keywords or no command available as illustrated in this guide.

The macro designer has changed in Access 2010 and the screen below shows an example of the new versus the older designer view.



Access 2003 (& earlier) – Older Macro Designer Tool



Access 2010 – New Macro Designer Tool

I will explain the designer interfaces further in the section '*Introducing the Macro Designer Window*' which will allow users to familiarise the tools and apply the examples in this guide which will be based in Access 2010.

Pros & Cons of Access Macros

In this section I'm going to list the advantages and disadvantages of using Access macros that is an exercise developers need to carry out to determine how they will programme their application going forward and avoid the dreaded result of having a '*pear shaped*' and inflexible database.

Benefits of using Access Macros

Here's the list:

1. ***Easier to write!*** You do not have to have a university degree in computer programming to understand and utilise macros. The command reference is intuitive and easy to apply. It just requires some investment of your time.

2. ***Disabled Mode*** – From version 2007 by default, any database opened that contained VBA code would not run as part of the security changes made to Microsoft Office applications and prevent unwanted macro virus threats that the VBA code could contain some malicious routines which some programmers seem to get a kick out of! Macros used within templates in Access are safe and run in normal mode.

3. ***Access Services*** – With the introduction of SharePoint server where you can now publish your Access database on the web in a secured environment, VBA code is not a web compatible procedural language and therefore will not run. Macros on the other hand are safe and will run via a web server.

4. ***Embedded Macros*** – As of Access 2007, you can now attach a macro inside a form or report as part of the host object and not have a dedicated separate macro object sitting in the navigation pane (*or database window for earlier versions*). This means when you copy a control like a command button which has an embedded macro the procedure copies across too, as it is part of the properties to the control.

5. ***Command Bars/Ribbon Bars*** – Macros have the ability to be attached to customised command buttons on a toolbar/menu bar (*pre Access 2007*) and ribbon bars (*from Access 2007*) which in turn can call VBA procedures should you need to.

6. ***Reserved Macro Names*** – There are two reserved macro names that automate the start-up and keyboard shortcut controls in your application without the need to code in VBA. A polished application will have reassigned keyboard shortcuts using the '*AutoKeys*' macro.

7. ***Variables are not reset*** – With VBA, public variables lose their values when a procedure ends or there is an error thrown. Macros can keep the values in place when an error occurs and have an advantage when handling errors in code.

Continues/…

Disadvantages of using Access Macros

1. ***Performance*** – VBA code is generally faster and more efficient that macros and is more noticeable for larger or longer procedures compared to a small piece of code where it is negligible. High end applications will use structured VBA more than macros.

2. ***Flexibility & Functionality*** – Macros allow you to do many things but VBA has the power to reach beyond and communicate with other applications and use features that macros simply cannot do, like play a sound, talk to web service or handle non Microsoft software.

3. ***Managing Code Procedures*** – Macros can be split into smaller units and now in Access 2010 you have a sub-macro tool; this can still be a restriction in handling control flows and other modular based calling procedures. Using external references like ADO is simply not possible and will be a deciding factor when planning your coded procedures.

4. ***Build Custom Functions*** – In VBA, you can build additional Access functions to sit alongside the standard functions and use them in queries, forms and reports like with any other function.

Spend the time analysing which set of programming tools you are going to apply. It may even be that a combination of the two will give you a fine balance and create a good working practice for end users. It could be a two tier structure whereby macros are used for basic processing that end users could manipulate and VBA is sealed in the background and critical workflows that no user is allowed to gain access to.

Types of Macros

With the introduction of Microsoft Access 2010, we now have three different types of macros to choose:

1. **Macro Objects**
2. **Embedded Macros**
3. **Data Macros**

Macro Objects (all versions)

These are physical objects which are stored globally in the Navigation pane (*Database Window for earlier versions*) and are stand-alone procedures that are typically called from one or more other objects including forms and reports.

This distinction and storage of such macros make them re-usable and is a good approach to exposing public based procedures that will typically call other objects like running a handful of queries in succession, exporting recordsets to other objects or applications and handling basic application commands (*i.e. quitting the application*).

This type of macro is available to all versions of Microsoft Access and is where your 'Auto' based macros are also stored (*AutoExec* and *AutoKeys*).

Embedded Macros (Access 2007 & 2010)

With the release of Microsoft Access 2007 (*& 2010*), you now have the opportunity (*and is now the default*) to create a procedure attached to a specific control's event. For example, the click event of a *Command Button*, the load event for a form or an update event for a *Combo Box*.

There is no physical or dedicated macro object found in the Navigation pane as it becomes a member of the host object (*i.e. a form or report*).

The scope changes from being public to private as it can only executed within the host when loaded and running.

The added advantage here is now when you copy a *Command Button* from one form to another and it has an event attached to it (*i.e. On Click*), the macro is also copied as it is part of the properties of the copied control.

This scales down (*and reduces memory*) of having to store too many macro objects and simple tasks like closing a form (*explicitly by name*) is an example why you bind an embedded macro instead.

When you now create a *Command Button* via the wizard control, it will generate an embedded macro and not VBA code as with the case for versions pre 2007. The reason for this action is down to the 'Disable mode' feature in Access that VBA code cannot run unless it has first been trusted, whereas macros are safe to run in this mode and therefore enabled.

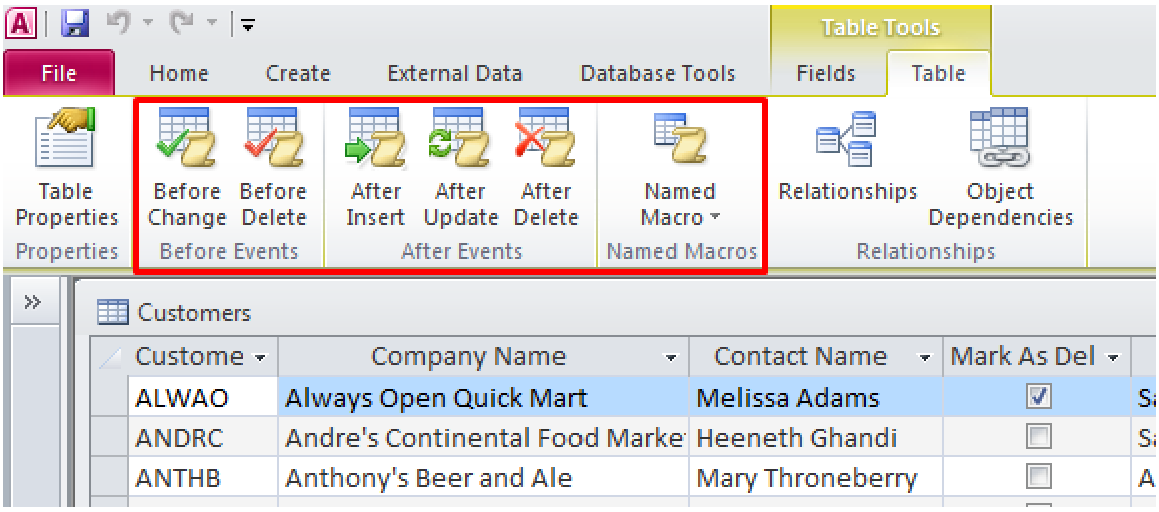
Continues/…

Data Macros (Access 2010 only)

This is new and exclusive to Microsoft Access 2010 and is compared to the more powerful database application, SQL Server's Triggers.

This type of macro is attached to a table and executes when an event like **insert**, **update** and **delete** is carried out to a record change.

The added benefit to this type of macro is now if you decide to upscale your database to the web using SharePoint server (Access services), they will also run.



Custome ▾	Company Name ▾	Contact Name ▾	Mark As Del ▾	
ALWAO	Always Open Quick Mart	Melissa Adams	☑	Sa
ANDRC	Andre's Continental Food Marke	Heeneth Ghandi	□	Sa
ANTHB	Anthony's Beer and Ale	Mary Throneberry	□	A

Access 2010 – Data Macro Ribbon Commands (Table Tab)

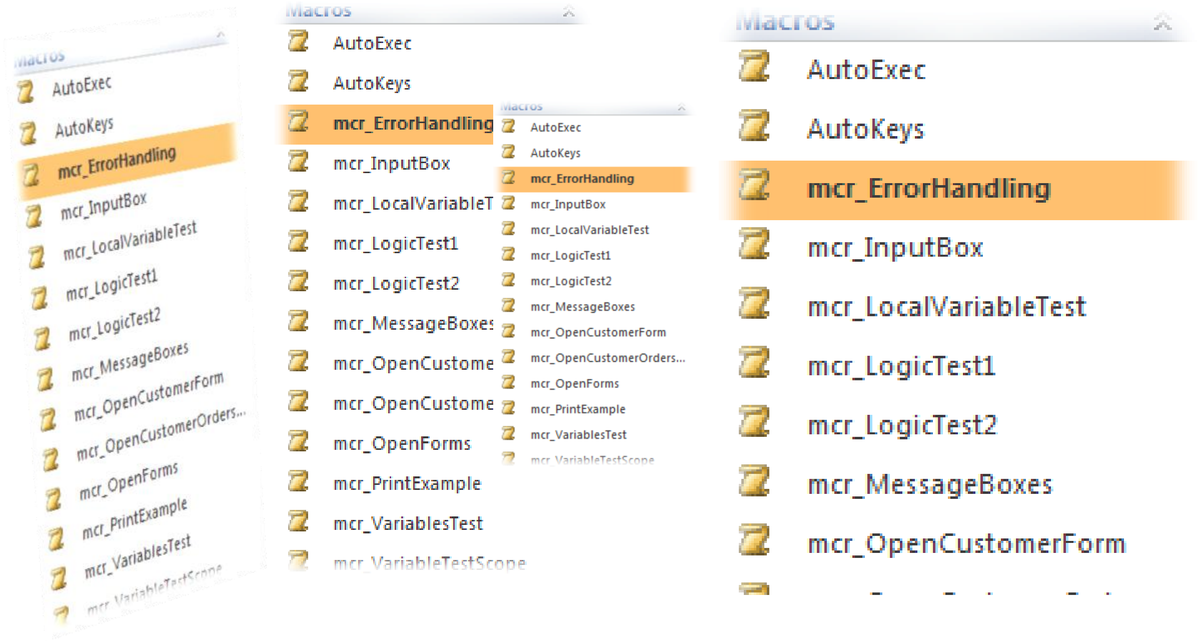
The scope for this type of macro is deemed private as it only applies to the calling table.

Note: *You must use .ACCDB (Access database 2007/2010 format) and not .MDB (pre Access 2007) in order to utilise Data Macros.*

End of preview…

To purchase this eBook, go to the link below

AccessDatabaseTutorial.com/eBooks/Macros-eBook



There you have it!

Spend the time learning this topic well and it will pay dividends in the long term.

DOWNLOAD EXAMPLE FILES FOR THIS USER GUIDE

If you would like a copy of some of the macro examples and other objects for this user guide please go to:

[DOWNLOAD LINKS NOT AVAILABLE IN PREVIEW]

You may want to review my other eBooks which cover the essential objects including Tables, Queries, Forms and Reports to fully extend your database knowledge.

http://AccessDatabaseTutorial.com/eBooks

Thank you

Ben Beitler – **"Your Access Database Expert"**

P.S. We'd love to hear your comments about this guide, so if you would rather email us rather than blogging, send your comments to ben@AccessDatabaseTutorial.com – and don't forget that my website has lots more articles and free videos to help you plan, build and implement an Access database.