Working with Macros and VBA in Excel 2007

With the introduction of Excel 2007 Microsoft made a number of changes to the way macros and VBA are approached. This document outlines these special features of Excel 2007 for people wanting to create macros and work with VBA and for those migrating to Excel 2007 from an earlier version of Excel.

The Developer Tab

If you plan to work a lot with macros and VBA you might find it convenient to switch on the Developer tab. This is not included in the collection of tabs normally displayed in the Ribbon. It gives quick access to many useful tools so if you are planning to work with macros and VBA you should switch it on. Click the **Office Button** then click the **Excel Options** button at the bottom of the window to open the *Excel Options* dialog. In the **Popular** section check the box marked **Show Developer tab in the Ribbon** (*Fig. 1*) then click the **OK** button.
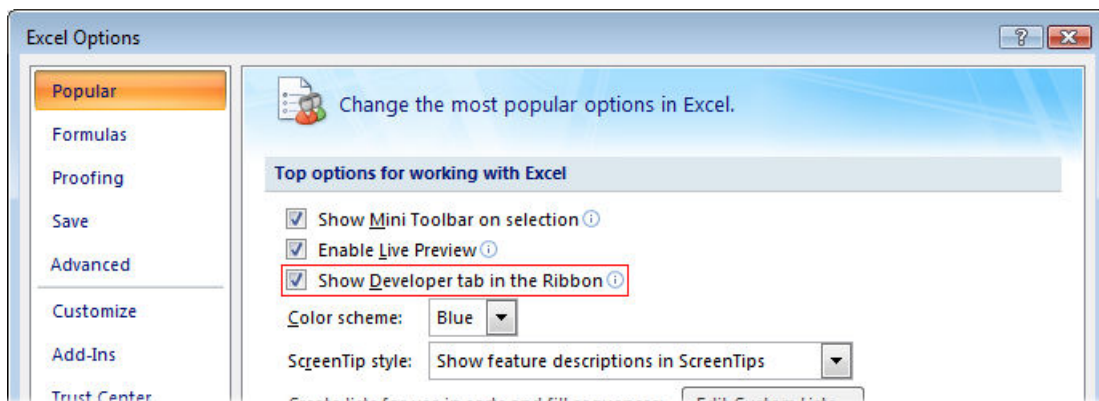

Fig. 1 Switch on the Developer Tab from Excel Options.

This displays the Developer tab in the ribbon. The tools on the Developer tab are arranged in four groups: *Code*, *Controls*, *XML* and *Modify* (*Fig. 2*).
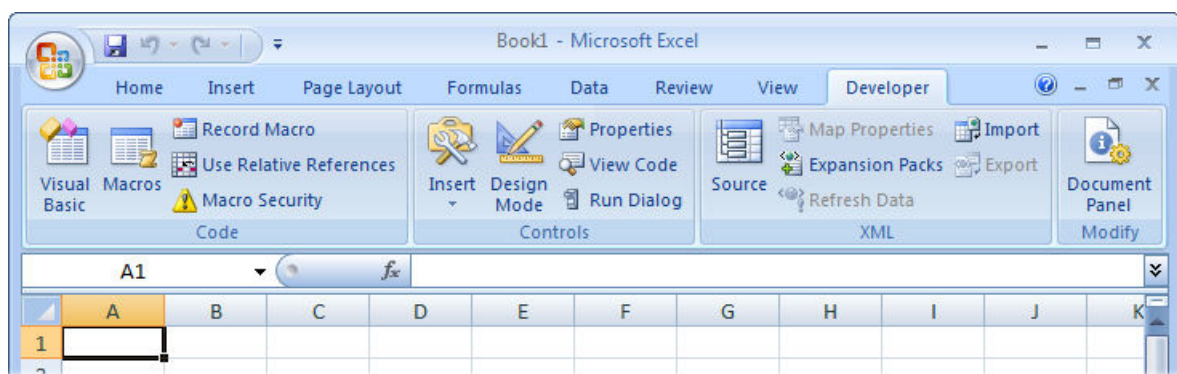

Fig. 2 The Developer Tab in Excel.

The *Code* group contains the tools most commonly used when recording and running macros and also gives quick access to the Visual Basic Editor. The *Controls* group contains tools for working with ActiveX and Form controls on a worksheet. The *XML* group contains all the functionality related to working with XML files in Excel. The *Modify* group allows you to specify the type of Document Information Panel the current workbook should display in other compatible programs. Each button on the ribbon displays a comprehensive tooltip describing its function.

It is not necessary to display the Developer tab in order to simply record and run macros. When the *View* tab is selected the ribbon displays a *Macros* group which also gives access to the required tools (*Fig. 3*).
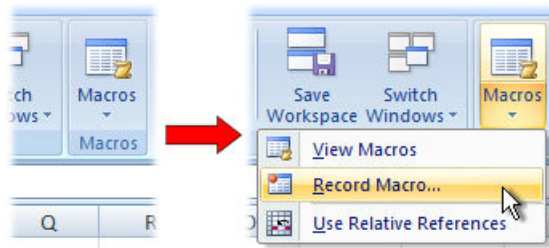
Fig. 3 Macro options on the View tab.

Macros and Excel File Types

The default file type for an Excel 2007 workbook has a **.xlsx* extension. If you record a macro or write VBA code and then save the workbook as the standard file type Excel displays a message warning that a VB project can not be saved in a *macro-free workbook* (**.xlsx*) (*Fig. 4*). If you continue and save in this format all your macro code will be lost.
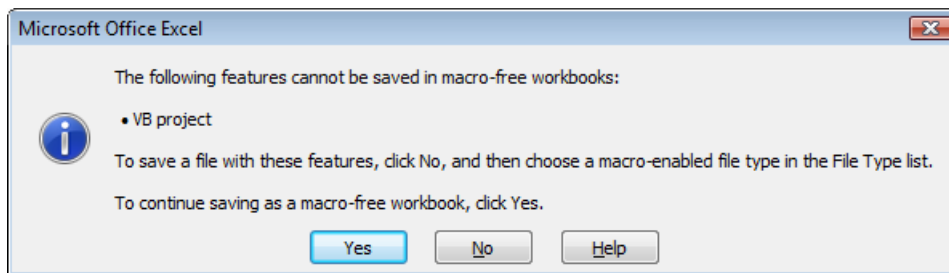

Fig. 4 A *.xlsx workbook can not contain macros.

To preserve your macro code you should choose to save the file as a *macro-enabled workbook* (**.xlsm*) (*Fig. 5*). Files saved in *97-2003 workbook* (**.xls*) format for compatibility with earlier versions of Excel do not distinguish at the point of saving between those which contain code and those which don't, so your code will not be lost if you save in this format.
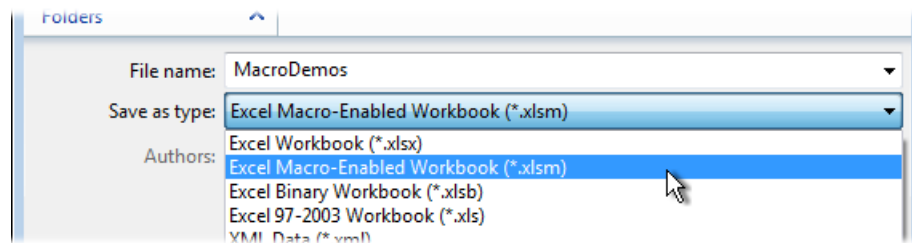

Fig. 5 Saving as an Excel Macro-Enabled Workbook

In addition to the different file extensions, Excel 2007 uses different icons to identify the various file types (*Fig. 6*, *Fig. 7*, *Fig. 8*). This helps to provide advanced warning that a workbook contains macros or other VBA code.
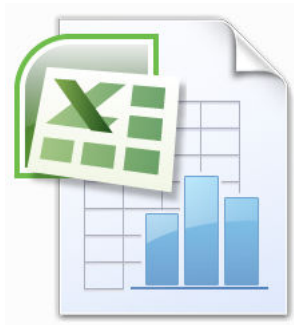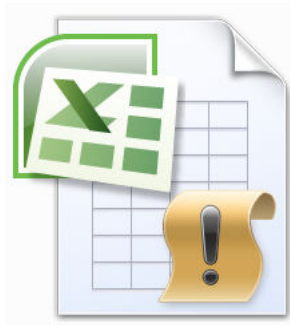

Fig. 6 Icon for an Excel Workbook (macro-free).

Fig. 7 Icon for a Macro-Enabled Workbook.

Fig. 8 Icon for an Excel 97-2003 Workbook.

Recording and Running Macros

The procedure for recording and running macros in Excel 2007 is the same as for earlier versions of Excel but the commands for initiating the processes are found in a different location.

Recording Macros

To record a macro click the **View** tab then open the **Macros** section of the ribbon and click the **Record Macro** button (*Fig. 9*) or click the **Developer** tab and click the **Record Macro** button in the **Code** section (*Fig. 10*).
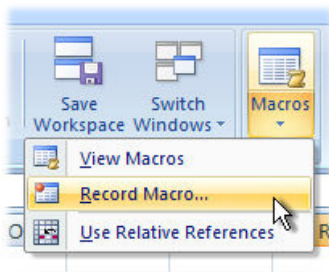

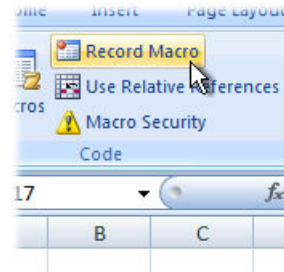|	
Fig. 9 Recording a macro from the View tab.	*Fig. 10 Recording a macro from the Developer tab.*

In the *Record Macro* dialog box enter a name for the macro, choose a location for it to be stored in, and optionally add a description and shortcut key (*Fig. 11*) then click the **OK** button to begin recording.
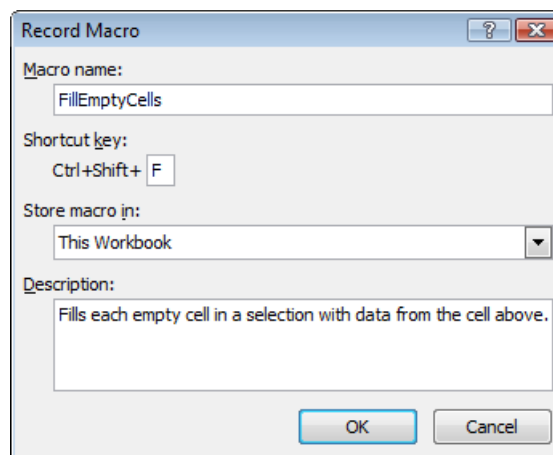

Fig. 11 The Record Macro dialog box.

Normally *absolute* cell references are recorded (i.e. the same cells that were used during the recording will be worked on when the macro is run). At any time whilst recording you can choose to record *relative* cell references (i.e. instead of recording the actual cell addresses Excel records their positions relative to each other so that the macro can be run on different ranges of cells). To do this visit the **View** or **Developer** tabs again and click the **Use Relative References** button. This button toggles the relative reference recording mode on and off.

Finally, return to the **View** tab (*Fig. 12*) or the **Developer** tab (*Fig. 13*) and click the **Stop Recording** button to save your macro.
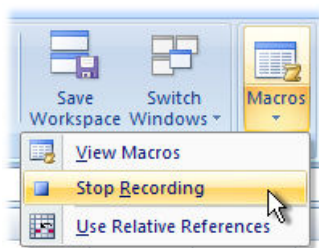

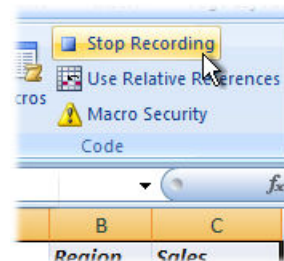|	
Fig. 12 The Stop Recording button on the View tab.	*Fig. 13 The Stop Recording button on the Developer tab.*

Running Macros

To run a macro click the **View** tab then open the **Macros** section of the ribbon and click the **Macros** button (*Fig. 14*) or click the **Developer** tab and click **Macros** button in the **Code** section. Alternatively use the keyboard shortcut **[Alt]+[F8]**. This opens the Macro dialog box (*Fig. 15*). Choose the macro you want to run and click the **Run** button.
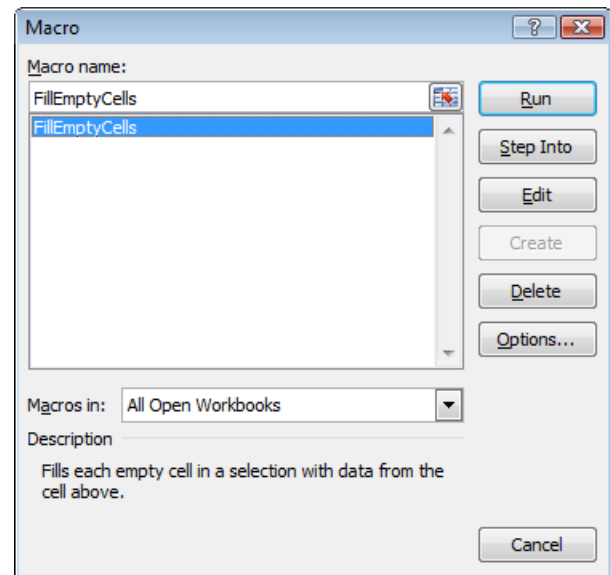

Fig. 14 The macros button

Fig. 15 The Macro dialog box.

Macro Security Warnings

If Excel's macro security is set at the default level, when you open a macro-enabled workbook a warning is displayed in the *Message Bar* notifying you that macros have been disabled (*Fig. 16*).
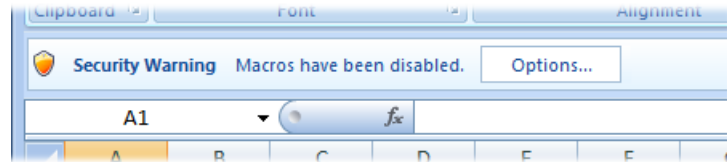

Fig. 16 The macro-enabled workbook triggers a security warning.

Click the **Options** button on the Message Bar to open the *Microsoft Office Security Options* dialog which explains why the message was shown and offers the option to enable the macros contained in the file (*Fig. 17*). If you trust the file select **Enable this content** then click the **OK** button to close the dialog and enable the macros.
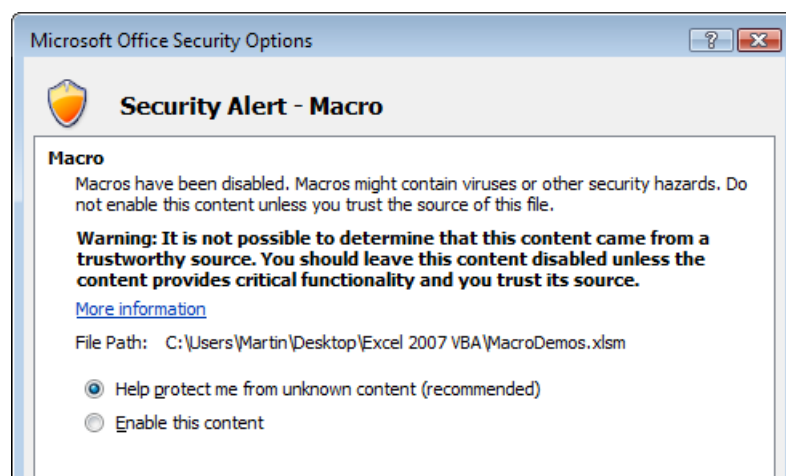

Fig. 17 The Security Options dialog box.

The Trust Center

Macro Settings

In Microsoft Office 2007 all security options are controlled from a single point called the *Trust Center*. The Security Options dialog includes a link to the Trust Center. You can also open it by clicking the **Office Button** then clicking **Excel Options** and choosing **Trust Center**. Finally, click the **Trust Center Settings** button to open the *Trust Center* dialog box.
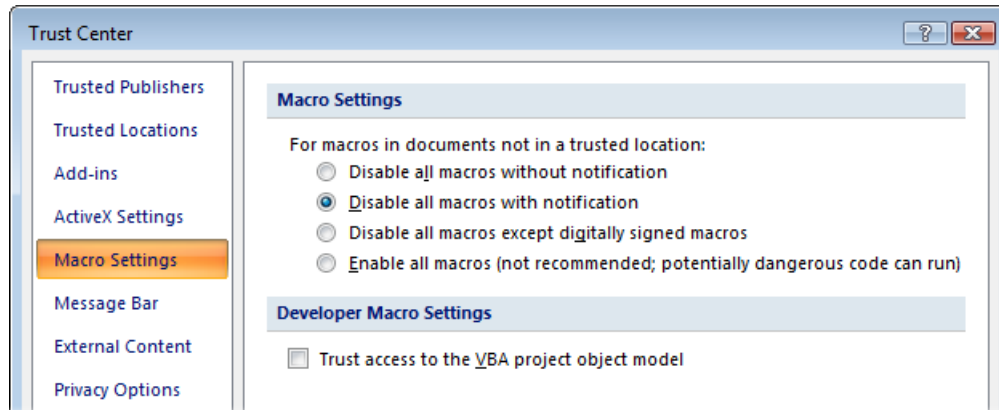

Fig. 18 The Macro Setting section of the Trust Center.

To set the default security settings select **Macro Settings** (*Fig. 18*). This section of the Trust Center lets you define how files containing macros are handled. The *Macro Settings* section of the dialog offers four levels of macro security:

- **Disable all macros without notification** – Use this setting if you do not trust macros at all. All macros will be disabled and you will not see any security alerts. It does not, however, prevent you from benefiting from macros you trust because you can put these files in a *Trusted Location*. Files in trusted locations are allowed to run without being checked by the Trust Center security system.

- **Disable all macros with notification** – This is the default security setting. Macros are automatically disabled but you are shown a warning when you open a file containing macros and you have the option to enable them on a file by file basis.

- **Disable all macros except digitally signed macros** – This is the same as the default setting with the exception that digitally signed macros are enabled providing they are from a *Trusted Publisher*. If the publisher has not yet been included in the list of trusted publishers, or if the digital signature has expired, you are notified.

- **Enable all macros (not recommended; potentially dangerous code can run)** – All macros are automatically enabled and no warnings are displayed. This option leaves the computer vulnerable to malicious code and is NOT RECOMMENDED!

The *Developer Macro Settings* section of the dialog has a single item allowing you to **Trust access to the VBA project object model**. The default setting is not to do so. It is possible to write code that manipulates the Visual Basic Editor, in other words code that itself writes code. Whilst this has its uses the security setting should only be changed to allow it when trusted code is being run. To allow it on a permanent basis would pose a serious security risk.

Trusted Publishers

If you want to trust macros that are digitally signed you must first add them to the list of Trusted Publishers. When you open a workbook that contains a digital signature that is not already on the list of Trusted Publishers Excel displays a security alert giving brief details of the digital signature (*Fig. 19*).

This gives you the options to *Enable this content*, treating the file in the same way you would treat any document containing macros, or to *Trust all documents from this publisher.* Choosing the second option enables the macros in the current workbook and adds the details of the digital signature to the Trusted Publishers list so that next time you open this workbook or any other bearing the same digital signature it is automatically enabled. Remember that you have to set your macro security level to *Disable all macros except digitally signed macros* to take advantage of this.
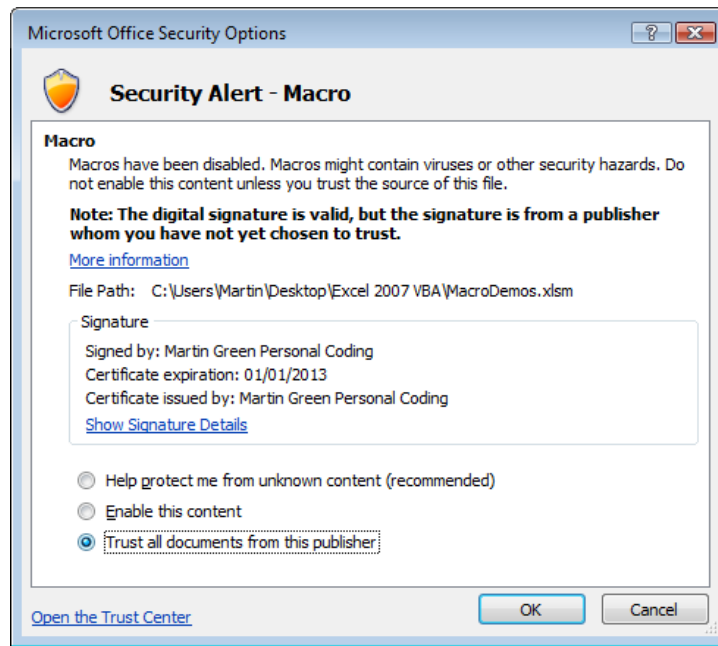
Fig. 19 The workbook includes an new Digital Signature.

You can view the list of Trusted Publishers any time by visiting the **Trusted Publishers** section of the **Trust Center** (*Fig. 20*). Here you can view a publishers' security certificate and, if you wish, remove a publisher from the Trusted Publishers list.
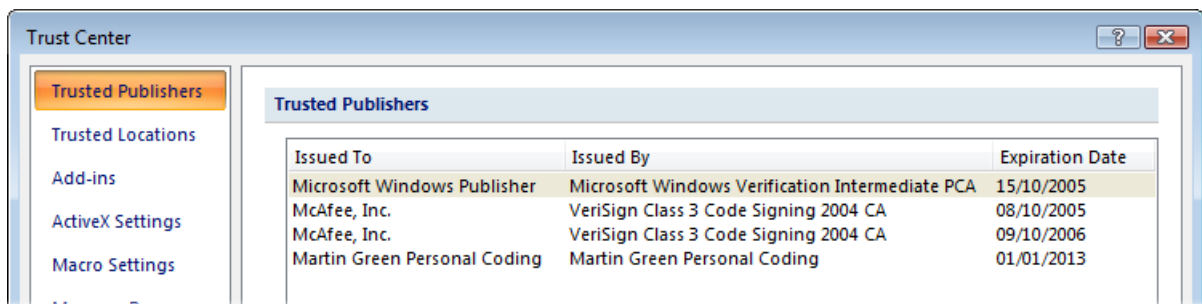

Fig. 20 The Trust Center displays a list of Trusted Publishers.

Trusted Locations

If you have numerous files that contain macros and that you know are safe an alternative to using the digital signature system is to store those files in a *Trusted Location*. Visit the **Trusted Locations** section of the **Trust Center** to view a list of existing locations. To add a new location to the list click the **Add new location** button to open the *Microsoft Office Trusted Location* dialog box then click the **Browse** button to locate and select the folder that you want to designate as a Trusted Location (*Fig. 21*). You can use an existing folder or create a new one. Click the **OK** button to complete the process.
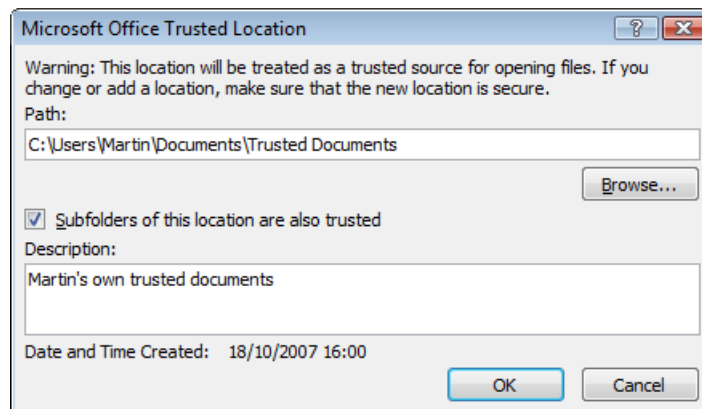

Fig. 21 Creating a new Trusted Location

Your new Trusted Location will now appear on the list of Trusted Locations in the Trust Center (*Fig. 22*). Here you can modify existing locations, remove them from the list or temporarily disable them. It is not recommended that you use a network location that other users have access to as a Trusted Location.
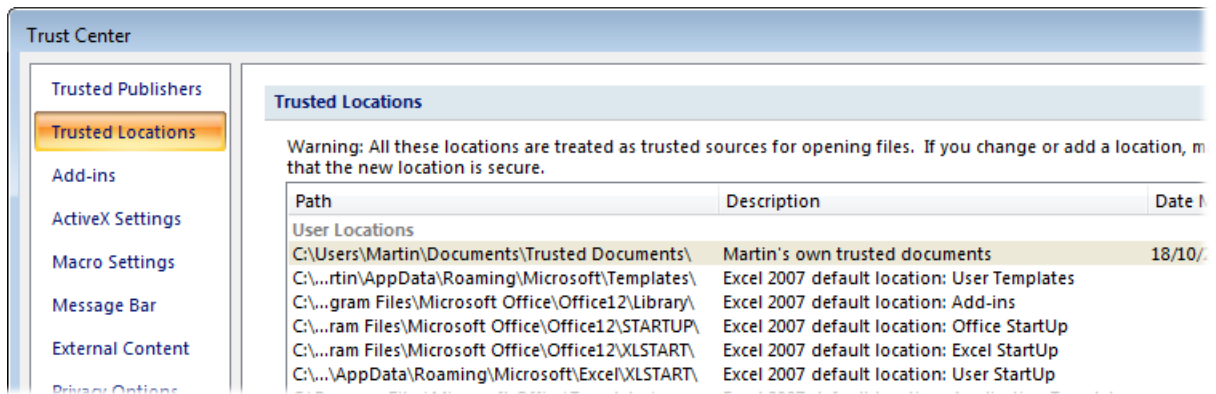

Fig. 22 The Trust Center displays a list of Trusted Locations

Now, any workbooks containing macros that you store in a Trusted Location will be trusted by Excel and their macros automatically enabled when you open them.