

**Chapter
2**

**VISUAL BASIC
EDITOR**

*Get on the
Fast Track!*



TM

**SYS-ED/
COMPUTER
EDUCATION
TECHNIQUES, INC.**

Objectives

You will learn:

- C How to edit code in the Visual Basic Editor.
- C How to create, open, and access project(s).
- C How to edit scripts and use the code window.

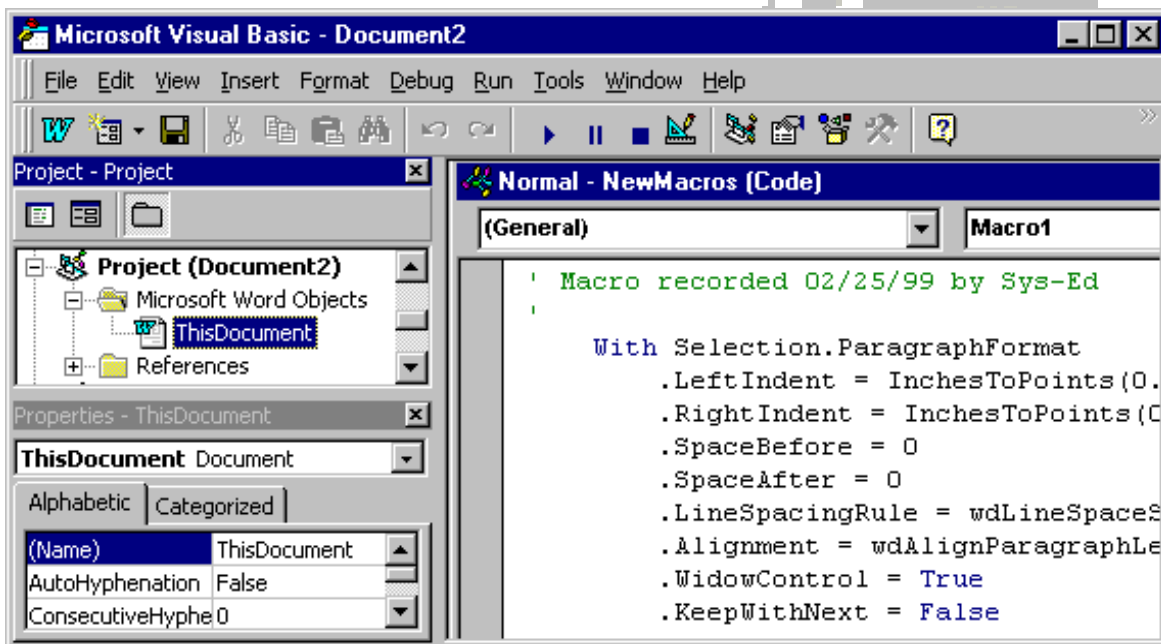


1 Getting Started

Microsoft Excel, Word, Access, and PowerPoint come equipped with the Visual Basic Editor.

This development environment can be used for creating, editing, debugging, and running code associated with Microsoft Office documents.

- C To open the Visual Basic Editor, click the Visual Basic Editor button on the Visual Basic toolbar.



C
Or
or
ati

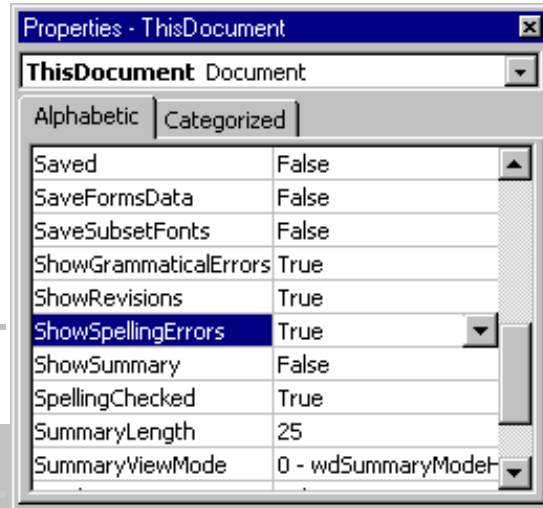
n about a particular window in the Visual Basic Editor, click in the window and then press F1 to open the appropriate Help topic.

F
inf
m
o

- C To see the Help topic for any other element of the Visual Basic Editor, such as a particular toolbar button, search Help for the name of the element.

1.1 The Properties Window

A property is a characteristic of an object, such as the object's color or caption. A property is set to specify a characteristic or behavior of an object.



The ShowSpellingErrors property of a Word document can be set to True to show spelling errors in the document.

The Properties window is used to set the properties of an object at design time. It is useful when working with custom dialog boxes and ActiveX controls.

For most objects, however, it's easier to set these properties at design time by using familiar commands in the user interface.

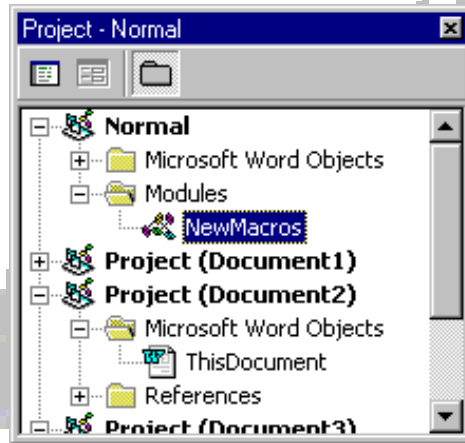
The ShowSpellingErrors property of a Word document can be set to True by selecting the Hide spelling errors in this document check box on the Spelling & Grammar tab in the Options dialog box (Tools menu).

If the Properties window right is not being used, it can be closed to simplify the work space a little. It can be opened at any time by clicking Properties Window on the View menu.

1.2 The Project Explorer

All the code associated with a workbook, document, template, or presentation is stored in a project which is automatically stored and saved with the workbook, document, template, or presentation.

In the Project Explorer of the Visual Basic Editor, it is possible to view, modify, and navigate the projects for every open or referenced workbook, document, template, or presentation. The Project Explorer can be resized and either docked or undocked from any of the sides of the Visual Basic Editor window.



In Word, because the Normal template is available from every Word document, there's always a project for Normal in the Project Explorer.

Within a project, there can be application objects that have events associated with them:

- C custom dialog boxes
(called forms in the Project Explorer)
- C standard modules
- C class modules
- C references

Folders in the Project Explorer divide project elements into categories. If no folders are visible, click the Toggle Folders button at the top of the Project Explorer.

In the Project Explorer, there's one project for each open or referenced workbook, document, template, or presentation.

In each project, there typically will be:

- C Objects such as Document objects, Workbook objects, and Worksheet objects which recognize events.
- C Forms, also known as UserForms, are custom dialog box interfaces which include code that control how the user interacts with a particular dialog box.
- C Standard modules, which contain code that isn't associated with a particular object or form.
- C Class modules, which contain information about a custom object type.
- C References to other projects.

To see the code in a module or the code associated with an object or form:


1. Click the element in the Project Explorer.
2. Click the View Code button  at the top of the Project Explorer.

To see the user interface for a particular object or form:

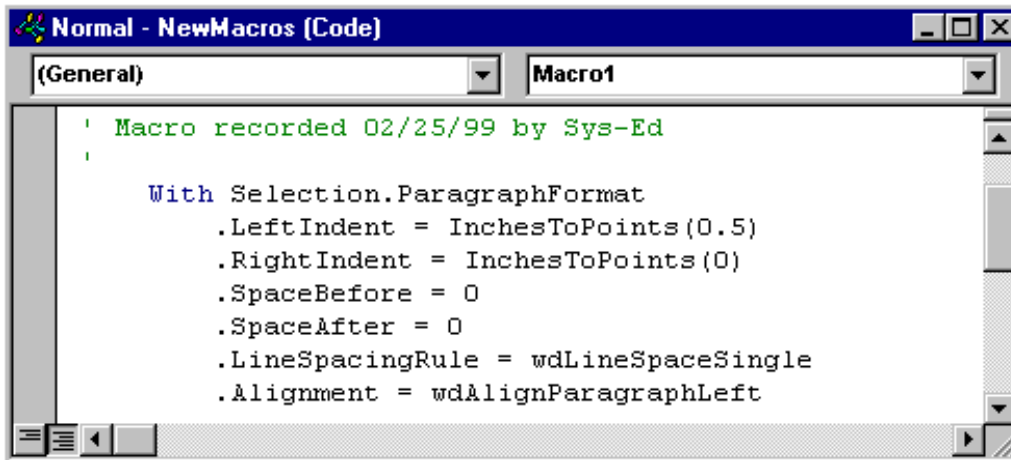
1. Click the object or form in the Project Explorer.
2. Click the View Object  button at the top of the Project Explorer.

1.3 The Code Window

To view the code in a project:

1. Go to the Project Explorer.
 2. Click the element that contains the code.
 3. Click the View Code button  at the top of the Project Explorer.
- C To be able to see more than one procedure in the code window at a time, select the Default to Full Module View check box on the Editor tab in the Options dialog box (Tools menu).
 - C To view just one procedure at a time, clear this check box.

The Code window can be navigated by using the items listed in the Object and Procedure boxes at the top of the window.

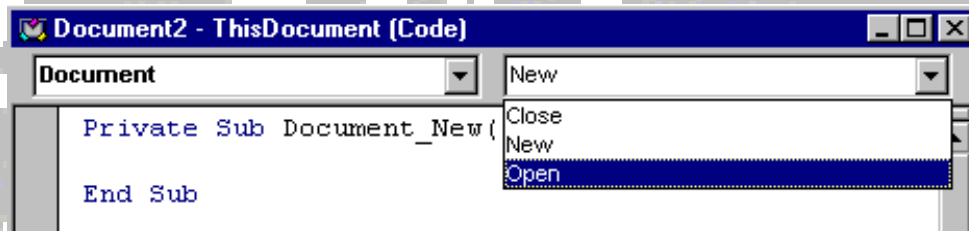


In the Object box:

- C Click (General).
- C Then click a procedure name in the Procedure box to see a procedure that isn't associated with a specific event.

In the Object box:

- C Click an object.



- C Then click an event in the Procedure box to see the code that runs when a specific event occurs.

2 Making Room in the Visual Basic Editor

When writing a simple procedure or editing a previously recorded macro, it usually will be a good practice to forego some of the advanced features of the Visual Basic Editor in the interest of a simpler workspace.

A few ways for simplifying a coding environment are to:

- C Close the Properties window.

Unless there is a need for working with custom dialog boxes or ActiveX controls, the Properties window probably will not be of much use.

To reopen Properties window, click Properties Window on the View menu.

- C Hide any toolbars not being currently used.

To redisplay the Debug, Edit, Standard, or UserForm toolbar, right-click the Visual Basic Editor menu bar, and then click the name of the toolbar to be displayed.

- C When working with code in a standard module and there is no need to navigate to other code in the project or to code in other projects, consider closing the Project Explorer.

To reopen the Project Explorer, click Project Explorer on the View menu.

3 Recording a Macro

The macro recorder can be used to translate user-interface actions into Visual Basic code. Recording a simple macro provides a jump start on creating a more complex macro, and also serve familiarize the programmer with the objects, properties, and methods of an application.

To record a macro:

1. To display the Visual Basic toolbar, point to Toolbars on the View menu in the application window (not in the Visual Basic Editor), and then click Visual Basic if it isn't already selected.
2. On the Visual Basic toolbar, click the Record Macro button.
3. In the Record Macro dialog box, replace the default macro name in the Macro name box if you want, and click OK.

The Store macro in box can be used to choose where the macro will be stored. For now, click This Workbook in Microsoft Excel, All Documents (Normal.dot) in Word, or the name of the active presentation in PowerPoint.

4. Perform the actions for which Visual Basic code is to be generated.
5. On the Stop Recording toolbar, click the Stop Recording button.

The macro has now been recorded.

To look at the macro code:

1. Point to Macro on the Tools menu.
2. Click Macros.
3. In the Macros dialog box, select the appropriate macro name, and then click Edit.

4 Getting Around in Projects

The Project Explorer is used to navigate to any procedure in any open project.

Start by finding the object that contains the macro. Most general procedures, including recorded macros, are stored in a standard module.

- C If there are folders displayed in the Project Explorer, standard modules are located in the Modules folder.
- C If folders are not visible in the Project Explorer, click the Toggle Folders button to display them.

After locating the object that contains the code, double-click the object to view the procedures it contains. This method can be used to get to procedures which have been either written from scratch or macros which have been recorded.

Where a recorded macro is stored depends on the location which has been specified in the Store macro in box in the Record Macro dialog box when the macro was recorded.

Application	Location
In Microsoft Excel	If This Workbook in the Store macro in box was clicked when the macro was being recorded, the macro will be stored in Module1 in the Modules folder of the project for the workbook in which the macro was recorded.
In Microsoft Word	If All documents (Normal.dot) in the Store macro in box was clicked when the macro was being recorded, the macro will be stored in the NewMacros module in the Modules folder of the Normal project.
In PowerPoint	If the name of the active presentation in the Store macro in box was clicked when the macro was recorded, the macro will be stored in Module1 in the Modules folder of the project for the presentation from which the macro was recorded.

To be able to see more than one procedure in the code window at a time, make sure that the Default to Full Module View check box is selected on the Editor tab in the Options dialog box (Tools menu).

Otherwise, the Procedure box in the code window can be used to move from one procedure to another.