BONUS WORD CHAPTER

GETTING STARTED WITH VBA

In this chapter

by Bill Ray

www quehelp com

Special Edition Using

que

WHAT IS VISUAL BASIC FOR APPLICATIONS?

Visual Basic for Applications (VBA) is the main programming tool for automating the Microsoft Office family of applications. VBA is derived from one of the most widely used programming languages for personal computers: BASIC (short for Beginner's All-purpose Symbolic Instruction Code). Earlier versions of many Microsoft applications used variations of BASIC as their macro language, but until recently, many of these languages had significant differences from one another. Microsoft had been working for years to alleviate this problem and unify the macro language offerings available for Office applications.

Because Visual Basic is one of the most popular Windows application development languages in the world and it allows developers to create standalone, executable Windows applications, many VB programmers wanted Microsoft to create an automation language for its Office suite that was based on the language. It certainly didn't hurt that it was also a tool that they were already familiar with, and Microsoft listened. The result is Visual Basic for Applications (VBA), and it is the automation language used with Word, Excel, Outlook, Access, and PowerPoint. Besides the Office applications, other Microsoft programs, such as Project, use VBA. Microsoft has even licensed VBA for use in third-party applications. You can think of VBA as a subset of Visual Basic. Although VBA is a subset, it still offers awesome programming power and a feature-rich development environment.

VBA is a powerful programming environment, providing a full-featured programming language development environment. It is the latest in object-based programming, yet it remains easy to use. Microsoft has taken great effort to build online and context-sensitive help into the tools associated with VBA.

VBA's biggest advantage, perhaps, is its consistency from one application to another. After you learn to program in VBA in Word, you have a pretty good head start to programming in, for example, Excel or PowerPoint. In this scenario, all you have to do is learn the specific features for working with a spreadsheet or with presentation graphics.

Note

Is there any difference between a macro and a program? Not really. *Macro* has been the common name for the automation tools built into many programs ever when Lotus 1-2-3 and WordPerfect dominated the PC marketplace. By most definitions, any sequence of commands that give instructions to be carried out by a computer is a program. Give yourself credit: If you write macros, you're a programmer!

HOW VBA RELATES TO WORDBASIC

For all versions of Word for Windows through Word 95 (also known as Word 7.0), Word's macro language was WordBasic. VBA was introduced as Word's macro language beginning with Word 97. Because WordBasic is also a dialect of the BASIC programming language, it has some similarity to VBA. However, there are substantial differences between the two languages, especially in the way that Word commands are performed in the two languages.

Although versions of Word since Word 97 (2000 and 2002) no longer directly support WordBasic, you don't necessarily have to re-create your old macros from scratch when upgrading from an old version. When you open a Word 6 or Word 95 template in Word 2002, Word automatically translates the existing WordBasic macros into a format that VBA can use. You can even edit these converted macros if you need to make changes in their functionality.

In the long run, you'll probably want to convert your existing WordBasic macros to VBA to take fullest advantage of the newer features of Word 2002.

Caution

When you open an existing Word 6 or Word 95 template in Word 2002, Word saves the template in the new file format, which can't be read in the earlier Word versions. If you will be continuing to work in Word 6 or Word 95, make sure that you only work with copies of the original templates.

Keep in mind that the automatic conversion of WordBasic macro code into VBA doesn't always work perfectly, especially if your WordBasic macros are complex. Test your converted macros carefully to verify that they work the same way they did in WordBasic. If they don't, you might have to edit the macros, or re-create them in VBA.

WHEN TO USE VBA

Although recorded macros can do just about anything in Word, it's often useful to be able to edit macros or create new macros by writing your own VBA code. The following situations are just a few examples:

- You have recorded a macro that works well, but now you want to change the way it works without recording it again.
- You want to prompt the user to input information, such as his name or a filename.
- You want to display one of Word's dialog boxes so that a user can make selections from the dialog box.
- You want your macro to make logical decisions. For example, you can program a macro to function only if your current selection is in a table, or if the document is in Page Layout view.
- You want to repeat an action many times. For example, a macro can apply header or footer formatting to each section in a long document.
- You want to exchange data with other Office applications, or even perform actions using those applications.
- A long, complex macro runs slowly, and you want to improve its performance.

- You made a mistake while recording a macro, but you don't want to throw away the whole macro.
- The VBA macro recorder doesn't always produce the most efficient VBA code. You can often improve the performance of your macros by editing the recorded code.

Note

VBA is a *hosted language*. To use it, you must be running a Visual Basic Editor (explained n the following section) from within a hosting application, such as Word or another Office program.

READING AND EDITING YOUR RECORDED VBA CODE

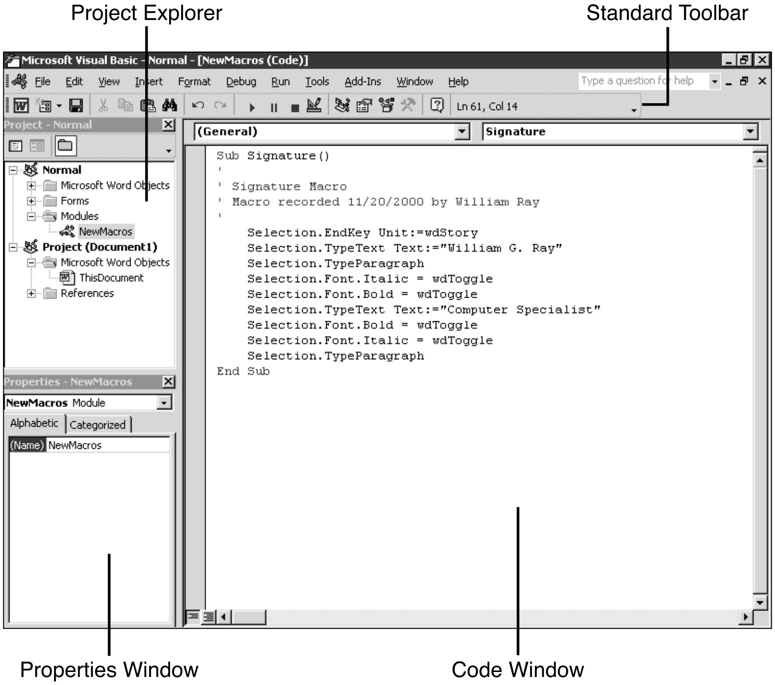
The best way to get started with the Visual Basic Editor is to record a macro and then to study the code you've recorded. For example, you can record a macro that adds a place for your signature to the end of any document. Follow these steps to record the macro:

1. Open a document to use as a test document while you are recording the macro.
2. Select Tools, Macro, Record New Macro (or double-click the REC button on the status bar) to open the Record Macro dialog box.
3. Type **Signature** as the macro name and click OK.
4. Press Ctrl+End to move to the end of the document.
5. Press Enter twice to add a paragraph and a blank line.
6. Type **Sincerely** and then press Enter four times.
7. Type your name and press Enter.
8. Press Ctrl+I to turn on italic.
9. Type your title; then press Ctrl+I to turn off italic.
10. Press Enter to add another paragraph.
11. Click the Stop button or double-click the REC button on the status bar to turn off the macro recorder.

➔ To find more information about recording macros with Word 2002, **see** Chapter 28, "Recording and Running Visual Basic Macros," **p. 871.**

Now that you have recorded a macro, you can open the Visual Basic Editor to view the VBA code you just created.

1. Choose Tools, Macro, Macros to show the Macros dialog box.
2. Select the name of the macro you just recorded from the Macro Name list box.
3. Choose Edit to open the Visual Basic Editor, displaying the selected macro, as shown in Figure 1.



```
Sub Signature()
'
' Signature Macro
' Macro recorded 11/20/2000 by William Ray
'
    Selection.EndKey Unit:=wdStory
    Selection.TypeText Text:="William G. Ray"
    Selection.TypeParagraph
    Selection.Font.Italic = wdToggle
    Selection.Font.Bold = wdToggle
    Selection.TypeText Text:="Computer Specialist"
    Selection.Font.Bold = wdToggle
    Selection.Font.Italic = wdToggle
    Selection.TypeParagraph
End Sub
```

Figure 1
You can use the Visual Basic Editor to create, edit, debug, and manage your VBA macros.

Tip from
Bill Ray

If you don't need to work on a particular existing macro, you can quickly activate the Visual Basic Editor by pressing Alt+F11 or by choosing Tools, Macros, Visual Basic Editor.

The Visual Basic Editor (VBE) contains the common set of macro programming tools for editing VBA code in Office XP. All of Office XP's applications use the VBE to create and edit VBA code. The VBE provides a wide variety of features to assist you in creating VBA macros:

- A code editing window for viewing and editing VBA code
- Form design tools for creating custom dialog boxes
- Tools that enable you to organize your code and work with several projects and modules at once
- Debugging tools to help you create and test complex macros

Throughout the remainder of this chapter, you'll learn about each of the tools available in the VBE.

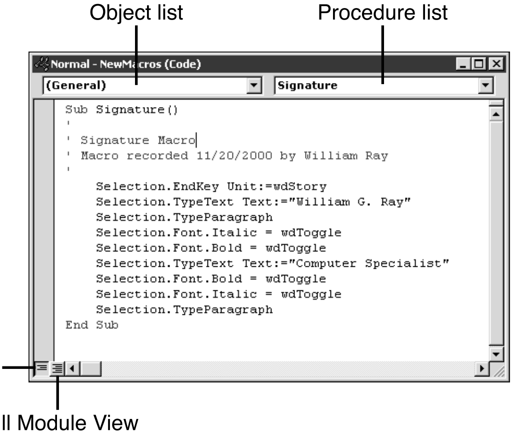
WORKING WITH THE CODE WINDOW

The code window, which usually appears as a large editing window on the right side of the VBE window, is where you can view and edit VBA code. Each code window contains all the VBA code for a single module. A *module* is a collection of macros and other VBA code. Each module might contain any number of macros. When you record a macro, the VBA code is added to the NewMacros module. Figure2 shows the code window.

Note

There is no particular significance to the name NewMacros. It's simply the name Word gives to the module that contains newly recorded macros. You can change the name of any module by using the Properties Window in the VBE.

Figure 2
Use the code window to edit the code in your VBA macros. In Procedure view, you can see only one macro or procedure at a time.



```
Sub Signature()
'
' Signature Macro
' Macro recorded 11/20/2000 by William Ray
'
    Selection.EndKey Unit:=wdStory
    Selection.TypeText Text:="William G. Ray"
    Selection.TypeParagraph
    Selection.Font.Italic = wdToggle
    Selection.Font.Bold = wdToggle
    Selection.TypeText Text:="Computer Specialist"
    Selection.Font.Bold = wdToggle
    Selection.Font.Italic = wdToggle
    Selection.TypeParagraph
End Sub
```

➔ For more information about modules and how Word incorporates macros into them, **see** "Understanding Projects and Modules," **p. 19**.

The code window has two viewing options. In Procedure view, you can see the code for only one macro or procedure at a time, as shown previously in Figure 2. In Full Module view, shown in Figure 3, the code window displays all the procedures in a module as a continuous document. You can switch between these two views by clicking the two view buttons in the lower-left corner of the code window.

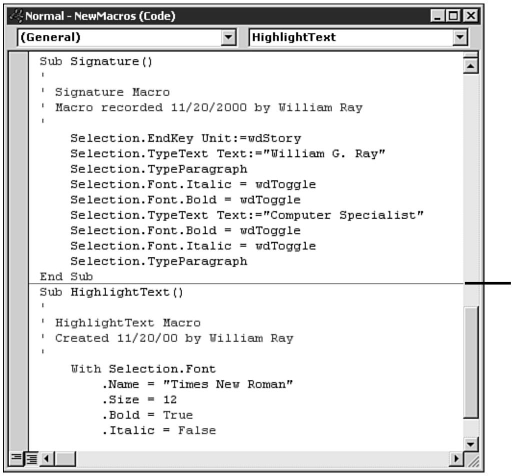
UNDERSTANDING PROCEDURES

VBA macros are stored as *procedures*. A procedure is a collection of VBA statements and declarations. A *statement* is a single line of VBA code that performs an action. A *declaration* is a special kind of statement, which declares a VBA language element, such as a procedure or a variable.

Each procedure must have a unique name. The rules for naming a procedure are the same as the ones for naming a macro. When you record a macro, the name you give the recorded macro becomes the procedure name. The first line of a procedure, known as the declaration, identifies the name of the macro:

```
Sub Signature ()
```

Figure 3
In Full Module view, the code window shows all your macro code continuously.



```
Sub Signature()
'
' Signature Macro
' Macro recorded 11/20/2000 by William Ray
'
    Selection.EndKey Unit:=wdStory
    Selection.TypeText Text:="William G. Ray"
    Selection.TypeParagraph
    Selection.Font.Italic = wdToggle
    Selection.Font.Bold = wdToggle
    Selection.TypeText Text:="Computer Specialist"
    Selection.Font.Bold = wdToggle
    Selection.Font.Italic = wdToggle
    Selection.TypeParagraph
End Sub
Sub HighlightText()
'
' HighlightText Macro
' Created 11/20/00 by William Ray
'
    With Selection.Font
        .Name = "Times New Roman"
        .Size = 12
        .Bold = True
        .Italic = False
```

This statement has three parts:

- **Sub**—The VBA keyword that identifies the beginning of a Sub procedure. All recorded macros are Sub procedures.
- **Signature**—The name of the procedure. If you edit the name on this line, you change the name of the macro.
- **Parentheses ( )**—Indicate that this procedure contains no *arguments*. Arguments are used by more advanced VBA programmers to pass information from one procedure to the next. Macros generated by the macro recorder have no arguments, but VBA still requires the parentheses.

Note

Procedure names follow the same rules as macro names. Names can be up to 80 characters long, must start with a letter, and can't contain spaces or most other characters.

The last statement in a VBA procedure is simple:

```
End Sub
```

The End Sub statement marks the end of a VBA procedure. Every procedure must begin with the Sub statement and end with the End Sub statement.

Note

All recorded macros are *Sub* procedures. A Sub procedure is simply a sequence of VBA statements that you can execute. There are also function and property procedures. *Function* procedures are used to calculate a result, such as a numeric calculation or a text value. *Property* procedures define the attributes of custom objects that you define in class modules. Function and Property procedures are advanced topics, outside the scope of this book, that are not likely to be used in simple macros.

To find more information on function and property procedures, use the online help to search either, "Function Procedures," or "Property Procedures."

VBA creates every recorded macro as a single procedure. As you gain experience in VBA programming, you can create additional procedures that can be used by your macros. For example, if you want to use the same set of statements in many macros, you can enter those statements as a procedure and call the procedure from many different macros. Creating reusable procedures can save you typing and make your VBA code easier to maintain.

Creating a Macro in the Visual Basic Editor

As you become more experienced with VBA, you might prefer to type your code directly into the code window, without first recording it. To create a new procedure in the VBE, follow these steps:

1. Activate the code window in which you want to create a new procedure.
2. Choose Insert, Procedure. The Add Procedure dialog box appears, as shown in Figure 4.

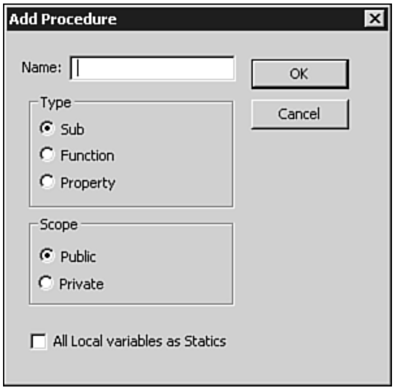


Figure 4
Use the Add Procedure dialog to name a new VBA procedure.

3. Type a name for the procedure in the Name text box, following the rules for naming procedures and macros (letters and numbers only, beginning with at least one letter).
4. Select one of the three procedure types (sub, function or property). Most simple macros are Sub procedures.

5. Select the scope. Most macros are Public, which means that they are visible to all the modules in the current project. If you select Private, the procedure does not appear on the Macro Name list of the Macros dialog box.

> **Note**
>
> When you create more complex VBA code, you might create a procedure that can be called by other procedures in the same module. By declaring the procedure as Private, you can prevent users from running the procedure directly from the Macros dialog box.

6. Check All Local Variables as Statics only if you want the variables in your macros to retain their values between the times you call the macro. This is an advanced option, and it is usually not the recommended choice.

EDITING WITHIN THE VISUAL BASIC EDITOR

After the code window is open to the procedure you want to edit, it works in much the same way as Word or any other editor. However, the VBE provides many tools that make this code window especially good at editing VBA code.

The Standard toolbar for the VBE has buttons for many of the most commonly used commands in the VBE. Some of these buttons, such as Cut, Copy, and Paste, work the same way they do in a Word document.

However, even though most of these buttons work as you would expect, some require a bit more explanation due to their idiosyncrasies:

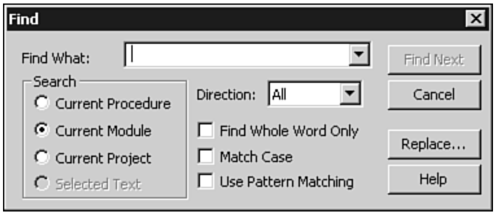
- The View Microsoft Word button switches to the Microsoft Word window, without closing the VBE.
- The Save button saves the current project, which is the template or document that contains the active code window.
- The Find button displays the VBE Find dialog box, shown in Figure 5. Use this dialog box to search for and replace text in your macro code.

VBA SYNTAX—THE RULES FOR ENTERING CODE

VBA code is governed by a *syntax*—a set of rules—for entering your code. After you begin editing your recorded code or typing your own VBA code, you must follow the VBA syntax rules or errors will occur, preventing your macros from running.

Type one statement per line in the VBE. You can enter more than one statement per line by using a colon to separate the statements. This can, however, make your code harder to read and provides no performance benefit. For example, the following line of code contains two statements:

```
Documents.Add : Selection.TypeText
```



Find

Find What:

Search

○ Current Procedure

● Current Module

○ Current Project

○ Selected Text

Direction: All

□ Find Whole Word Only

□ Match Case

□ Use Pattern Matching

Figure 5
Use the Find dialog box to search for text in the Visual Basic Editor.

The same function can be performed by the following code, which is easier to read:

```
Documents.Add
Selection.TypeText
```

You can also use "white space" to make your code easier to read. White space refers to any use of blank lines, tabs, and spaces. By leaving blank lines between sections of code, you can easily break the code into related sections. Also, by adding additional spaces to some lines of code, you can separate out the critical items, making them easier to spot.

Some VBA statements are long and do not fit entirely on the width of the screen. When you edit a long line of code, you can choose between two editing approaches:

- Type the entire statement on one line. The VBA editor scrolls to the left or right as you edit the line.
- Type the line continuation character—the underscore (_)—and then press Enter to continue the statement on the next line.

  For example, the following statement appears entirely on one line:

  ```
  ActiveDocument.Tables.Add Range:=Selection.Range, NumRows:=1, NumColumns:=2
  ```

  You can use the line continuation character to break up the statement over several lines of code, making it easier to read:

  ```
  ActiveDocument.Tables.Add Range:=Selection.Range, _
    NumRows:=1, _
    NumColumns:=2
  ```

There is no functional difference between these two forms of the statement. It's completely a matter of your preference.

Making Your Code More Readable

One of the best ways to make your VBA code easy to read and understand is to use comments. A *comment* is a line of code that does not perform any action, but which you can use to display information about the code that you have recorded or edited.

Word's macro recorder automatically adds a few comment lines to your recorded macros, as you can see from the examples in this chapter:

```
'
' Signature Macro
' Macro recorded 10/6/00 by William Ray
'
```

Comment lines begin with an apostrophe ('). When you run a VBA macro, VBA ignores any commented lines, skipping on to the next regular line of code. Comments are great tools for documenting the purpose of your macro, when it was created or modified, or other details that might not be obvious when reading the VBA code. As your macros grow in length and complexity, it's a good idea to add comments throughout the macro, describing the purpose of each section of the code. The general convention for using comments is to place the comment on the line before the code it is documenting. This allows you to read the comment, be aware of the comment, and then read the code that the comment discusses. This is the convention recommended by Microsoft.

You can also place a comment on the same line as your code, in the following manner:

```
Selection.Bold = True    'Turn on bold formatting
```

In this style, any text that appears after the apostrophe is treated as a comment. The code preceding the comment executes as usual. You can also use comments to experiment with and test your VBA code. What if you want to know how your macro would work if one or more lines of code were removed from the procedure? Rather than delete the lines of code, you can temporarily disable them by typing an apostrophe at the beginning of the line, turning it into a comment. If you decide later that you want to reactivate the line of code, all you have to do is remove the apostrophe, and the VBA code is returned to normal use.

In longer macros, you might want to comment out a large block of code lines, but it is easy to make an error while typing all those apostrophes (to say nothing of tedious). The VBE provides some code editing tools discussed later in this chapter that make it easy to perform this operation quickly.

Another way to comment code is to use the Comment Block button found on the Edit toolbar. The Edit toolbar contains several tools for editing code in the code window.

Select the lines of code you want to comment out; then click the Comment Block button. VBE adds an apostrophe at the beginning of each selected line, turning the line into a comment. To restore the lines of code to normal use, select them again and then click the Uncomment Block button.

Another way to make your code easier to read is to use indenting. Programmers have traditionally used indenting to emphasize standard programming structures, such as loops and other program control blocks. You can indent a line of code by pressing Tab one or more times before you type the code. When you press Enter after typing the code, the code window automatically aligns the next line at the same indented position. If you don't want that line to be indented, press Backspace or Shift+Tab to reduce the indenting.

The Edit toolbar has two buttons that enable you to indent or unindent a block of code quickly. Select the lines of code you want to change; then click the Indent button to increase the indenting or the Outdent button to reduce the indenting.

UNDERSTANDING VBA STATEMENTS: OBJECTS, METHODS, AND PROPERTIES

Learning to recognize and control objects, methods, and properties is perhaps the most important skill in learning to be an effective VBA programmer. The *Word Object Model*, which declares the relationship among all the objects that make up the functionality of Word, defines all objects in Word. Similarly, there are object models for Excel, PowerPoint, Word, and Access, as well as for some applications that don't even use VBA as a macro language. After you are familiar with VBA programming in Word, you only have to learn the object models of other programs to write VBA macros for them.

Statements perform the real work of VBA. A statement is a single line of code that performs some action. You can use many different kinds of statements in a VBA macro, but the most common type of statement is one that performs work using the objects, methods, and properties of Word.

Word is organized into a system of many different objects that VBA can control. Objects act as the nouns in the VBA language. Here are just a few examples of the objects in Word:

- The Word application
- A document
- A paragraph
- A font
- The current selection

Each object in Word has a collection of methods and properties that Word can use to control the object. A property is an attribute of an object that usually describes the object in some way. For example, a Font object has a Size property, and a Document object has a Name property. You can use VBA to read and to change the properties of an object. One way to think of properties is as the adjectives of the VBA language. The following sample of code from a recorded macro shows one way of using properties:

```
Selection.Font.Italic = wdToggle
```

This statement is the VBA equivalent of clicking the Italic button or pressing Ctrl+I. The effect is to set the Italic property of the Font object to the opposite value. The statement has several components:

- Selection is the object that represents the currently selected text, at the time the statement executes.
- The Font property of the Selection object results in a Font object, which in turn has its own properties.

- The Italic property of the Font object refers to the current state of italics for the selected text.
- The assignment operator (=) indicates that you want to assign a new value to the Italic property.
- The predefined constant wdToggle indicates that you want to switch the italic state to the opposite of its previous state.

Note that periods are used to connect the object to its properties or methods. This notation, sometimes referred to as dot notation, is characteristic of working with objects in VBA.

Note

The preceding example of the Italic property toggles italic on or off, but what if you simply want to turn on a toggle-based function, like Italic, regardless of whether the current selection is enabled or not? The statement in the following example turns on italic:

```
Selection.Font.Italic = True
```

In this case, the statement sets the Font.Italic property to the value True, which has the effect of turning on italics. As you might expect, the opposite of True is False, so setting the Font.Italic property to False turns italics off.

True and False are examples of the Boolean data type, which VBA uses to represent logical values.

This is an example of how can modify a recorded macro to perform exactly the function you want.

A *method* of an object is a built-in capability to perform a task, and most objects have multiple methods. A method is an action that the object can perform. For example, a Document object has a Save method and a Close method. As you can see, the names of the methods and properties of an object are usually descriptive. Continuing the interpretation that objects are the nouns of this language and that properties act as the adjectives for VBA, you could say that methods perform the role of verbs. Here are two examples of the use of methods:

```
Selection.TypeParagraph
Selection.TypeText Text:="Bill Ray"
```

In the first example, the TypeParagraph method of the Selection object is the equivalent of pressing Enter, which inserts a paragraph mark at the location of the selection.

In the second example, the TypeText method is used to type at the location of the selection. Of course, the method needs to know what text you want to type, so it requires that a value be assigned to the argument Text, which provides the required information. The argument (Text := “Bill Ray”) indicates what text Word should type with the TypeText method.

The syntax for this statement is

```
Selection.TypeText(Text As String)
```

This means that the TypeText method requires one argument, named Text, which is of the data type String. The same action could be achieved with the following statement:

```
Selection.TypeText "Bill Ray"
```

Omitting the argument name might seem convenient, but it can make your macros more difficult to read as they become more complex.

Note

The := operator is useful and not too well documented. When a Sub or Function requires a large number of arguments and you really only need to pass one or two arguments to get your job done, the := argument assignment operator can save typing many commas that are required when arguments are passed in order.

Objects are often organized into *collections*. Word contains many collections of similar items, such as paragraphs, words, and styles. For example, the Documents collection consists of all the documents that are currently open. The Paragraphs collection contains all the paragraphs in the active document. There are many more examples of these collections. Generally (with a few exceptions), when an object name is plural, it refers to a collection, and when it is singular, it refers to a single object.

As an example of using a collection, consider the VBA code you get if you record the act of creating a new document:

```
Documents.Add DocumentType:=wdNewBlankDocument
```

This statement uses the Add method to add a new document to the collection of currently open documents. The type of document is defined by the DocumentType argument, using a predefined constant, wdNewBlankDocument.

Another Object-Based Example

Believe it or not, just about anything you can think of can be described using object terminology. As an example I am going to describe myself using object-based terms. I am the object named Author and am a member of the Humans collection. My properties include HairColor, EyeColor, Height, Weight, and so on. My methods include Eat, Sleep, and Write. Using this example, it is easy to understand the roles of collection, object, property, and method.

No one can reasonably expect to memorize all the objects in the Word Object Model, but experience teaches you to look in the right places for information about an object and its attributes. In the next section, you'll learn how to use the VBE to learn and work with the object model.

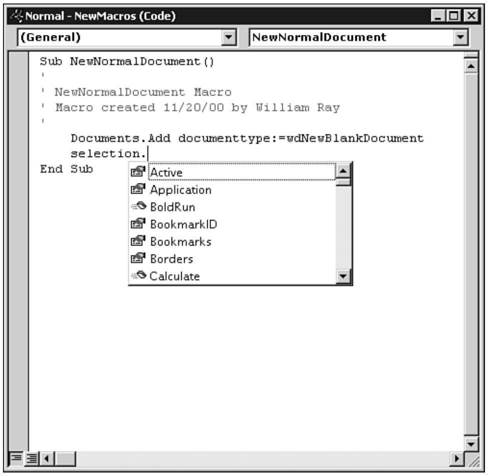
USING THE VBE TO CREATE ACCURATE CODE

The Visual Basic Editor contains many tools to make your coding process easier and more accurate than it would be if you just had to type text into a text editor. In this section, you'll see how the VBE helps you find the methods and properties that are available for Word objects, and helps you write error-free code.

USING THE PROPERTIES/METHODS LIST

One of the best ways to learn about objects, methods, and properties is to use the built-in help that the VBE provides. You can see lists of methods and properties for any object, and even get the VBE to do much of the typing for you.

When you type the name of an object and then type the period that follows it, the Properties/Methods list appears, as shown in Figure 6. In that example, the list shows the properties and methods for the Selection object. If you type the first few letters in the property or method you are looking for, the list selection moves to that item, or at least to one whose spelling starts with the same few letters. You can also use your up and down arrow keys to move through the list, or scroll the list with your mouse. After you have selected the item you want to use, press Tab to let VBE type the text of the item into the code window.



```
Sub NewNormalDocument()
'
' NewNormalDocument Macro
' Macro created 11/20/00 by William Ray
'
    Documents.Add documenttype:=wdNewBlankDocument
    selection.
End Sub
```

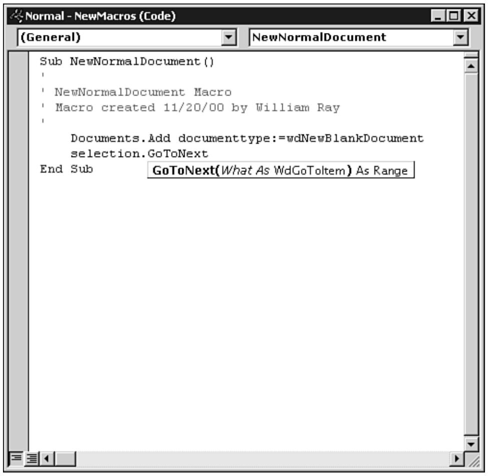
Figure 6
The Properties/Methods list shows the properties and methods available for the Selection object.

Tip from
Bill Ray

If you do not want to have the Properties/Methods list automatically display, select Tools, Options and go to the Editor tab. Remove the check from the Auto List Members box.

Some methods have one or more arguments, which you can use to provide extra information for the method. After you select or type the name of the method, the arguments are listed as a Quick Info tip, as seen in Figure 7. In this instance, there is only one argument, named What. The What argument tells Word what item (page, field, comment, and so on) you want to go to. The Quick Info tip also uses the “As” keyword to identify the data type of the argument. In this case, the What argument is declared as the type WdGoToItem, a predefined VBA type.

Figure 7
A Quick Info tip lists the arguments for the GoToNext method.

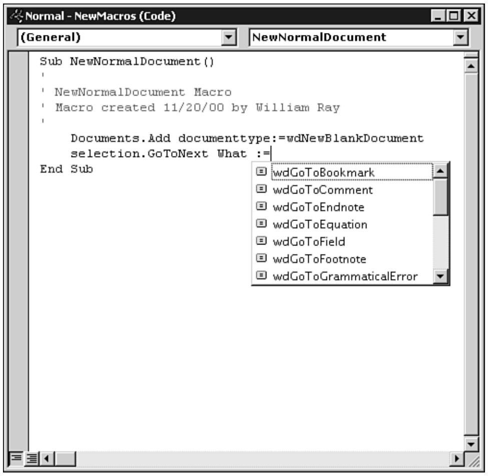


```
Sub NewNormalDocument()
'
' NewNormalDocument Macro
' Macro created 11/20/00 by William Ray
'
    Documents.Add documenttype:=wdNewBlankDocument
    selection.GoToNext
End Sub
```

GoToNext(*What As* WdGoToItem) As Range

Many arguments can be filled in using predefined constants. These constants represent literal values such as numbers, text, or another data type, but your code is much easier to read and understand if you use the descriptive constant. When a constant is available for an argument, a Constant list appears, as shown in Figure 8. Once again, you can select from the list by typing the first few characters of the list item, or selecting from the list by using the mouse and then pressing Tab.

Figure 8
The Constant list shows the choices for the What argument of the GoToNext method.



```
Sub NewNormalDocument()
'
' NewNormalDocument Macro
' Macro created 11/20/00 by William Ray
'
    Documents.Add documenttype:=wdNewBlankDocument
    selection.GoToNext What :=
End Sub
```

After you have typed all the text of a line of code, press Enter. The VBE accepts the line of code and moves the insertion point to the next line, unless an error is in your code.

➔ To learn how to handle errors in your VBA code, **see** "Using Automatic Syntax Checking," **p.17**.

Using Complete Word to Ensure Accurate Code

Another way to make your VBA coding faster and more accurate is to use the Complete Word feature. Any time you are in the middle of typing a VBA keyword or object name, you can press Ctrl+Space to complete the word.

For example, after you have typed **Sel**, if you press Ctrl+Space the editor types the complete word **Selection** in the code window. If you haven't typed enough to identify the word you're trying to type, the VBE displays a list of words spelled similarly. Select the word on the list and then press Tab to finish typing the word in the code window.

Using the Edit Toolbar

In addition to getting the assistance you've learned about in this section as you type new VBA code, you can get the same information at a later time using the VBE's Edit toolbar.

- To see a list of the properties and methods for an object, select the object and click the List Properties/Methods button or press Ctrl+J.
- To see a list of constants that can be used as an argument, select an existing constant and then click the List Constants button or press Ctrl+Shift+J.
- To see a list of the arguments or other information available for an object or method, select the item and click the Quick Info button or press Ctrl+I.

Using Automatic Syntax Checking

When you type a line of VBA code and press Enter, the VBE automatically checks the syntax of the statement you typed. The syntax is the set of rules that govern how you can use the VBA language.

Note

It is possible to turn off automatic syntax checking by selecting Tools, Options, Editor, and unchecking the Auto Syntax Check check box. This is probably not a good idea, though. Although the syntax checking can seem annoying at times, it can catch enough errors to save you a lot of time and frustration while writing macros.

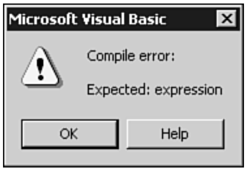
If your line of code doesn't contain any syntax errors that the VBE can catch, you can continue typing the next line of code. If there is an error, however, the VBE produces an error message, such as the one shown in Figure 9.

In this example, the user intended to enter a statement such as

```
Selection.GoToNext what:=wdGoToField
```

By pressing Enter before completing the statement, the user causes a syntax error to occur, since VBA needs an expression, such as the value wdGoToField, to assign the argument named *what*.

Figure 9
A compile error occurred when VBA tried to process this line of code.



Microsoft Visual Basic

Compile error:

Expected: expression

As a second example, a syntax error would occur if you typed the following line of code and pressed Enter:

```
Selection.TypeText."Bill Ray"
```

The error message, “Expected: identifier or bracketed expression,” provides some clue to the error, although the exact error might not seem obvious. The period after the TypeText method is the error that produces the message. The TypeText method requires you to supply an argument so that VBA knows what you want to type, and you have to use a space to separate the argument from the method. In this example, the correct line would be

```
Selection.TypeText "Bill Ray"
```

In some cases, you can look at the line of code you just typed and see the error you have made. When this happens, click OK and edit the line. If you can’t see your error, or you don’t understand the message, select Help to get more detailed information about the error.

Caution

Just because you don’t get a compile error when you type your VBA code, don’t think your code is guaranteed to work as expected. Two other classes of errors that the VBE may not catch are likely to affect your programs.

Runtime errors occur while your macro is executing. For example, consider the following statement:

```
Documents.Add "c:\Data\Expenses.doc"
```

This statemnt should cause VBA to open the file named Expenses.doc, which is stored in the Data folder on your hard drive. The syntax of the statement is correct, so it should work perfectly. But, what if you delete the document, then try to run the macro that contains this statement. An error occurs, and your macro stops running.

Another more subtle type of error is a *logic error*. With a logic error, your macro executes without producing any error messages, but it doesn’t do what you expect it to. Supposed you assign a value to a variable named Count:

```
Count = 1
```

Later in your macro, you want to increase the value of Count by one:

```
Count = Counr + 1
```

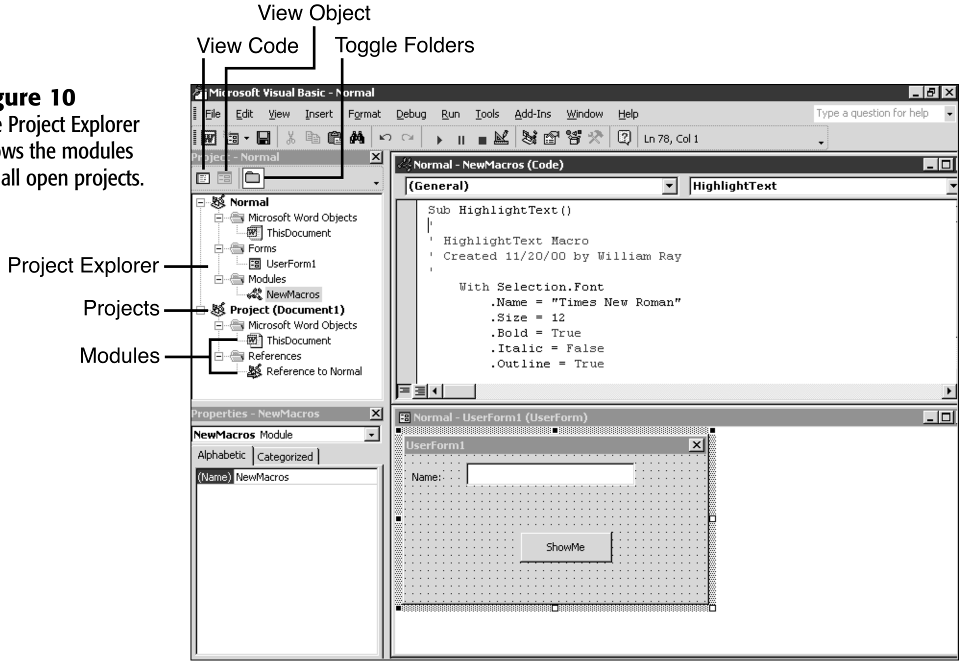
Notice that the variable name was misspelled. This mistake is not a syntax error, and it will not even cause a run-time error. It will, however, cause your macro to produce different results than those you intended.

Every programmer makes mistakes, and the only way to reduce or eliminate errors is by testing your macros thoroughly.

UNDERSTANDING PROJECTS AND MODULES

As you have seen throughout this chapter, VBA macros are collections of statements, stored in procedures. Procedures are stored in modules, and modules are stored in projects. Now that you have worked with statements, procedures, and modules, the best way to understand this organization is to work your way down from the project level.

Each Word template or document contains exactly one project. When you are running the VBE, you can see the project for each open document or template in the Project Explorer, as shown in Figure 10.



```
Sub HighlightText()
'
' HighlightText Macro
' Created 11/20/00 by William Ray
'
    With Selection.Font
        .Name = "Times New Roman"
        .Size = 12
        .Bold = True
        .Italic = False
        .Outline = True
```

Figure 10
The Project Explorer shows the modules for all open projects.

Note

If the Project Explorer is not visible in the VBE, click the Project Explorer button on the Standard toolbar, or press Ctrl+R.

In this example, the Project Explorer shows the projects for the Normal template, which is always open, and the document named Document1. Of course, if you have more documents and templates open, they all appear in the Project Explorer.

The modules are organized into different categories within the project and are displayed beneath each folder in the outline:

- **Standard Modules**—Contain the VBA procedures for recorded macros and other macros you enter directly into these modules. Standard modules appear under the Modules folder.
- **Microsoft Word Objects**—Used to store code that can respond automatically to events, such as creating a new document, opening a document, or closing a document.
- **Form Modules**—Contain the VBA code to respond to events that occur in a user-designed form. For example, you can write a procedure that runs when the user clicks the OK button on your custom form.
- **References**—Indicate a dependency on another template or library file. In Figure 10, Document1 is based on the Normal template, so the reference shows that the macros in the Normal project are available within this project.
- **Class Modules**—Contain VBA code for creating custom objects and defining the custom methods and properties of those objects.

To display the code window for a module, select the module in the Project Explorer and click the View Code button or press F7.

To display a form for editing, select the form in the Project Explorer and click the View Object button or press Shift+F7.

SAVING CHANGES TO YOUR MACROS

When you make changes to macros—by recording new macros or by editing your VBA code—you should save your work often. Because VBA code is stored in templates and documents, all you have to do is to save the file that contains the code.

From within the VBE, simply select a module in the project Explorer, or a code window in the project you want to save. Then click the Save button on the Standard toolbar or press Ctrl+S.

Caution

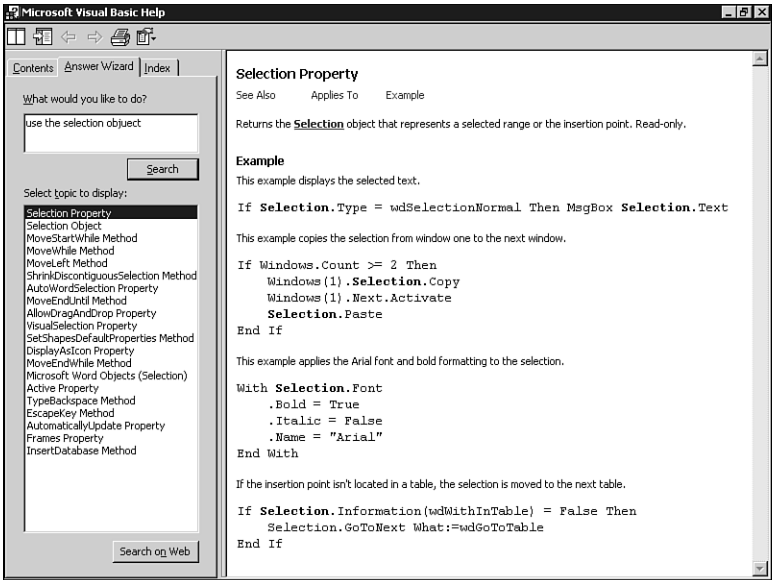
Because errors are common in macro programming, it is important to get in the habit of saving your work often. In particular, if you have edited your code in any way, you should save your changes before trying to run the macro. As your macros become more complex, it is possible to create a macro that causes an application error, in which case you won't have another chance to save your work.

UNDERSTANDING THE VBA HELP SYSTEM

As you continue to learn the VBA language and the Word Object Model, you'll need a lot of help picking up the details. Fortunately, VBA comes with a lot of help for the language, as well as the objects, methods, and properties you'll need to learn.

A quick way to get help with a topic in VBA is to position the insertion point anywhere in the term for which you need more information and press F1. VBA Help appears, with the help topic selected, as shown in Figure 11.

Figure 11
The VBA help topic for the Selection object describes its purpose and shows examples of its use in code.



use the selection objuect

Selection Property

Returns the **Selection** object that represents a selected range or the insertion point. Read-only.

Example

This example displays the selected text.

```
If Selection.Type = wdSelectionNormal Then MsgBox Selection.Text
```

This example copies the selection from window one to the next window.

```
If Windows.Count >= 2 Then
    Windows(1).Selection.Copy
    Windows(1).Next.Activate
    Selection.Paste
End If
```

This example applies the Arial font and bold formatting to the selection.

```
With Selection.Font
    .Bold = True
    .Italic = False
    .Name = "Arial"
End With
```

If the insertion point isn't located in a table, the selection is moved to the next table.

```
If Selection.Information(wdWithInTable) = False Then
    Selection.GoToNext What:=wdGoToTable
End If
```

In most cases, the VBA help topics include samples of the topics' use in code, as well as links to related objects, properties, and methods.

You can also ask the Office Assistant for help while you're writing VBA code. Activate the assistant by choosing the Help, Microsoft Visual Basic Help command, and type your question. Then choose Search to display a list of topics that might be of interest to you.

Tip from
Bill Ray

If you need to use Help to see code samples, it's much easier to use the sample text from Help than to type it yourself. Highlight the text, right-click it and select Copy from the menu that displays. Then paste it into your procedure. From there you need only modify the code to suit your needs.

Troubleshooting

What to Do When Automatic Conversion of Macros Does Not Work

The automatic conversion of WordBasic macro code into VBA doesn't always work perfectly, especially if your WordBasic macros are complex. Test your converted macros carefully to verify that they work the same way they did in WordBasic. If they don't, you might have to edit the macros, or re-create them in VBA.

What to Do When the Properties/Methods List Does Not Display When You Enter Code into a Module

If you want the Properties/Methods list to automatically display, select Tools, Options and go to the Editor tab. Place the check in the Auto List Members box.

What to Do When You Get an Error Message After You Enter a Line of Code

The VBE's automatic syntax checking tries to identify errors each time you finish typing a line of code. When this occurs, look carefully at the line you have just typed, and try to identify and correct the error. There are several common causes for these errors:

Misspelled language keyword:

```
If x < 100 Thenn
```

instead of:

```
If x < 100 Then
```

Missing expression in a calculation or assignment:

```
x = x +
```

instead of:

```
x = x + 1
```

Failure to include separators, such as spaces or periods, in a statement:

```
StatementBold := True
```

instead of:

```
Statement.Bold := True
```

Leaving out a required argument:

```
Selection.TypeText
```

instead of:

```
Selection.TypeText Text:=strMyVariable
```

How to Access the Visual Basic Editor from the Desktop

Don't forget that VBA is a hosted language. To use it, you must be running a hosting application, such as Word or another Office program. You can't just start the VBE from Windows and start writing code.

BONUS WORD CHAPTER

Taking More Control of Your Macros

In this chapter

by Bill Ray

Special Edition **Using**

que®

Recording Dialog Boxes and "With" Statements

The macro recording examples found in the SE Using Word 2002 Web Chapter 1, "Getting Started with VBA," were pretty simple. Additionally, while the macro recorder does a reasonably good job of capturing actions recorded you may be in for a surprise, however, when you record a macro that uses a built-in dialog box. As you'll see in the example that follows, you might get much more VBA code than you're interested in.

Of course, using dialog boxes while you record is convenient. By choosing one command, you can choose many preferences or options at a time. Also, in some cases there aren't any toolbar buttons or keyboard shortcuts for the features you want to record.

Suppose that you need a macro to emphasize selected text by changing the font size to 14 points and setting the text to bold and italic. Follow these steps to record a macro that accomplishes this:

1. Select the text you want to format.
2. Double-click the REC text on the status bar to display the Record Macro dialog box.
3. Give the macro a name, such as `Emphasize`, in the Macro Name text box and click OK to begin recording.
4. Choose Format, Font.
5. Select Bold Italic in the Font Style list box.
6. Select 14 in the Size list box.
7. Click OK to apply the formatting.
8. Click the Stop Recording button on the Stop Recording toolbar to end the recording session.

To see the code that you've recorded, follow these steps:

1. Press Alt+F8 to display the Macros dialog box.
2. Select the macro name (Emphasize) that you just recorded in the Macro Name list.
3. Choose Edit to display the macro in the VBE.

You should get something like this:

```
Sub Emphasize()
'
' Emphasize Macro
' Macro recorded 12/10/2000 by William Ray
'
    With Selection.Font
        .Name = "Times New Roman"
        .Size = 14
        .Bold = True
        .Italic = True
        .Underline = wdUnderlineNone
        .UnderlineColor = wdColorAutomatic
        .StrikeThrough = False
        .DoubleStrikeThrough = False
        .Outline = False
        .Emboss = False
        .Shadow = False
        .Hidden = False
        .SmallCaps = False
        .AllCaps = False
        .Color = wdColorAutomatic
        .Engrave = False
        .Superscript = False
        .Subscript = False
        .Spacing = 0
        .Scaling = 100
        .Position = 0
        .Kerning = 0
        .Animation = wdAnimationNone
    End With
End Sub
```

Notice two important things about this macro. First, VBA has used the With…End With statement to capture the settings in the Format Font dialog box. By using the With keyword, followed by the reference to the font object of the current selection (Selection.Font), every line of code that follows is treated as if it were a statement that began with Selection.Font. The End With statement signals the end of the structure where this rule applies.

The second thing to notice is that everything in the Font dialog box was recorded. The biggest problem with this is that the macro doesn't do exactly what you had in mind when you recorded it. For example, if you select some text currently formatted with the Arial font, running the macro will change the font to Times New Roman because that was the font when you recorded the macro.

To make this macro do what you want, all you have to do is to delete those lines that you don't want in the macro. After deleting the unneeded lines, the macro should look something like this:

```
Sub Emphasize()
'
' Emphasize Macro
' Macro recorded 12/17/2000 by William Ray
'
    With Selection.Font
        .Size = 14
        .Bold = True
        .Italic = True
        .Underline = wdUnderlineNone
    End With
End Sub
```

Tip from
Bill Ray

If you're not sure which lines to delete, you can comment the lines out and test the macro. As soon as you're sure that the macro works the way you want it to, you can delete the commented lines.

To comment out a block of lines, select the lines in the code window, then click the Comment Block button on the Debug toolbar. Click the Uncomment Block to remove the comments, restoring the lines of code.

Declaring and Using Variables

Just about any useful macro has to manage data. In VBA, you can work with many different types of data, including text, numbers, and logical data, as well as many custom data types defined in Word. It's often useful to reuse the same data items several times in a single macro. That's where variables come in.

A *variable* is a temporary storage location for a piece of data. You can think of a variable as a box in which you put the data until you want to come back and use it again. As its name implies, you can also change the value of a variable.

One of the most common operations with a variable is assignment. Using the assignment operator, you can put a value in the variable for later use. The following statement assigns some text to a variable named MyText:

```
MyText = "Welcome to the world of VBA programming."
```

From this point on in the same macro, you can use the variable name anywhere you would otherwise use a text value:

```
Selection.TypeText MyText
```

You can also use variables to store numeric data. The following example shows how you can assign values to variables using mathematical calculations:

```
''Increase price by 20%
 Price=Price*1.20

'''''Calculate Vacation based on years employed
'Five days vacation per year of employment
Vacation=YearsWithCompany*5
```

After a variable has been assigned a numeric value, it can be used in calculations. The following example assigns multiple numeric values to variables and then uses the variables in a calculation:

```
BaseSalary = 60000
Increase = 0.025
NewSalary = BaseSalary + (BaseSalary * Increase)
Selection.TypeText "Your new salary will be " & NewSalary
```

Variables are useful tools for storing long or multiple line strings. For example, if you wanted to create a string to be used with a *message box* that had multiple lines, you would use the following statements:

```
MessageText = "This operation may take several moments to complete."
MessageText = MessageText & vbCrLf & "Please wait ..."
```

Notice the use of the & character when combining strings. The process of combining strings is called *concatenation*. The other thing to note in the last sample code statement is the use of vbCrLf. vbCrLf is a VBA-defined constant that has the value of Chr(13) + Chr(10). The Chr(13) and Chr(10) represent a carriage return and a line feed. In other words, these are the characters needed for a new line.

Tip from
Bill Ray

You can use the VBA constant vbNewLine instead of vbCrLf. Microsoft introduced vbNewLine in recent versions of VBA to replace the existing constant, vbCrLf.

➔ For more information on working with message boxes or to see examples of VBA constants, **see** "Displaying Information with Message Boxes," **p. 10**

➔ For more information on constants, **see** "Using Constants," **p. 9**

Normally, VBA enables you to define a variable as you need it. You can just assign a value to the variable the first time you use it. However, most programmers prefer to declare their variables before using them. Declaring your variables makes your programs more organized and enables you to specify exactly which type of data you want to use in a given variable.

You can declare a variable in a VBA procedure using the Dim statement. The following statements declare several variables, specifying their data types:

```
Dim Authorname as String
Dim NumberOfCopies as Integer
Dim InterestRate as Long
Dim Info
Dim HireDate as Date
```

Note

The declaration for the variable named Info does not include a data type in this example. When you do not supply a data type, VBA assigns the default data type of *variant*.

By declaring a variable with the appropriate data type, you help VBA store the variable's information in the most efficient format, improving macro performance and reducing macro errors. The following rules apply when naming variables:

- Variable names can have as many as 255 characters.
- The name can contain letters, numbers, and underscores.
- Spaces and punctuation marks are not allowed.
- The first character must be a letter.

Understanding Data Types

You can use many different data types in VBA. Although a complete discussion of data types is beyond the scope of this book, here are a few of the most commonly used types:

- **String**—These variables can store text. Use these variables for any information you want to type in a document or use in a message. You might also use String variables for values that look like a number but that won't be used in calculations, such as phone numbers and ZIP Codes.
- **Integer**—These variables can store whole numbers, between –32,767 and 32,767. You can't use the Integer type if you need a wider range, or if you need decimal values.
- **Long**—These variables can store whole numbers between –2,147,483,648 and 2,147,483,647
- **Single** and **Double**—These variables can store very large (and very small) numeric values, including decimal values. This is the most common numeric data type to use if you require more options than an Integer variable can provide.

Note

Single values are floating-point numbers with a maximum value of 3.402823E38 and a minimum value of –3.402823E38. The smallest positive fraction you can represent as a single value is 1.401298E-45, and the largest negative fraction is –1.401298E-45.

Double values are floating-point numbers with a maximum value of 1.79769313486232E308 and a minimum value of –1.79769313486232E308. The smallest positive fraction you can represent as a double value is 4.94065645841247E-324, and the largest negative fraction is –4.94065645841247E-324.

In this notation, the letter E represents an exponential power of 10, so that 1.5E4 is the same as 1.5×10 to the fourth power, or 15,000.

The Double data type provides a wider range of values and greater precision than the Single type. The Single type has the advantage of using less memory than the Double type.

- **Date**—These variables can store date and time values. VBA has a variety of functions for performing calculations with dates and times.
- **Variant**—These variables can store just about any data type. When you use a variable without declaring it, or when you declare a variable without specifying a data type, the variable is treated as a Variant. Although Variants are convenient, they also use the most memory and can be the slowest performers of the data types. The Variant type is the default used if no type is declared.
- **Object**—These variables can represent the many objects that make up Word and other Office applications. Just as objects have methods and properties, an object variable has the same methods and properties as the object to which it refers.

When naming variables, you might want to consider using a naming convention that ties the name of the variable with its data types. This makes it easier to maintain code and variables in large procedures because you don't have to return to the beginning of the procedure to look at a Dim statement to determine what data type you assigned to a variable. Table 1 lists recommended naming prefixes for use with variables.

TABLE 1 NAMING PREFIXES FOR VARIABLES

Data Type	Short Prefix	Long Prefix
Array	a	ary
Boolean	f	bln
Byte	b	byt
Currency	c	cur
Date/Time	dt	dtm or dat
Double	d	dbl
Integer	i	int
Long	l	lng
Object		obj
Single		sng
String	s	str
Variant	v	var

WORKING WITH ARRAYS

Most high-level programming languages allow you to create arrays for storage of information. An *array* is a group of variables having the same data type and sharing a common name. The elements of the array are identified by an index number. The syntax for defining an array is

```
Dim array_name(n) As type
```

where *n* is the items in the array.

For example, if you wanted to create an array that could hold five elements, you would use the following statement:

```
Dim sngTaxTable(4) as Single
```

Note that the number within the parentheses is 4, not 5. This is because the first index number is by default 0. Arrays are useful for working with similar information. Let's say that you needed to process 20 names. You could create 20 variables, meaning that you would need 20 Dim statements. Or you could create an array to hold the names using the following statement:

```
Dim sNames(19) As String.
```

Most of the time you will work with single dimension arrays like the one in the previous example. However, *multidimensional arrays* are supported in Visual Basic. For example, a two-dimensional array could be thought of as similar in structure to a spreadsheet and would be defined as

```
Dim sngTable(3, 3) As Single
```

This statement creates a four by four array. Another option when declaring arrays is to not give it a size. You may need to have the flexibility of defining the size at runtime. If an array is declared without a size, it is a *dynamic array*. The syntax for a dynamic array is

```
Dim dyn_array() As type
```

After declaring the array, size it at runtime using *ReDim*:

```
ReDim dyn_array(array_size)
```

The *array_size* argument represents the new dimensions of the array. To retain the values of the array, be sure to include the reserve word *Preserve* after the *ReDim* statement.

REQUIRING VARIABLE DECLARATION WITH OPTION EXPLICIT

Although VBA doesn't normally require that you declare your variables before using them, it's a good habit to get into. You can even change VBA's rules so that declaration is required.

Requiring variable declaration is a good idea because it can help you identify many potential errors in your code before they cause other problems. If you use required declaration, VBA checks all the variables in your code to make sure that they were declared. If you have any undeclared variables, a compiler error occurs, and you are notified of the undeclared variable. This usually occurs when you have made a typing error, misspelling the name of a variable. If variable declaration is not required, this kind of error can be difficult to identify.

If you don't tell VBA to require variable declaration, you can simply use a new variable any time you want to in your code, without declaring it. This has the appearance of convenience, but can lead to many errors.

To require variable declaration, type the following statement in the Declarations section of each module:

```
Option Explicit
```

If this statement is in the module, VBA checks for undefined variables each time the code is compiled, such as when you run a macro for the first time after editing it.

Even better, you can follow these steps to customize your Visual Basic Editor (VBE) so that it places this line of code in all new modules:

1. In the VBE, choose Tools, Options, Editor.
2. Check the Require Variable Declaration check box.
3. Click OK.

From this point forward, the VBE adds the Option Explicit statement to all newly created modules. This has no effect on any existing modules.

Note

Even if you choose not to use Option Explicit to require variable declaration, VBA always requires you to declare object variables and array variables.

USING CONSTANTS

Variables are defined as placeholders for nonstatic information. When you need a placeholder for static information, create it as a constant. Constants are used for two reasons. One reason is to hold a value referenced numerous times that does not change during the run of the program but may change in the future—for example, a tax rate. Another reason is to make a program more readable. For example, TaxRate is easier to read than .08167.

To declare a constant and set its value, use the Const statement. After a constant has been declared, it cannot be assigned a new value. If you wanted to declare, for example, a constant to store the value of a sales tax, you would use the following statement:

```
Const SALESTAX As Long = .06231
```

Constants can be declared as one of the following data types: Boolean, Byte, Integer, Long, Currency, Single, Double, Date, String, or Variant. Because you already know the value of a constant, you can specify the data type in a Const statement. Many programmers follow a convention of using all caps for names of constants. This convention makes it easy to tell the difference between variables and constants in your code.

COMMUNICATING WITH THE USER

Many times it's useful to give some information to the user while your macro is running. You might want to inform the user of the progress of a long macro, or tell him or her that the macro found a problem in the document. You can use a message box to present such messages to the user.

Besides telling the user something, you might want to ask a question. It might be as simple as whether the user wants to continue with a long process, or what name the user wants to put in the signature block of a letter.

As you will see in the next couple of sections, you can use both message boxes and input boxes to ask these simple questions. For more complex data input, you can design custom dialog boxes, also known as *forms*, complete with list boxes, check boxes, and other controls.

DISPLAYING INFORMATION WITH MESSAGE BOXES

The easiest way to present a simple message to the user is by using the MsgBox statement to display a message box. This simple, one-line statement is loaded with options to show messages and ask questions in a variety of ways.

The MsgBox statement has the following syntax:

```
MsgBox Prompt, Buttons, Title, Help, Context
```

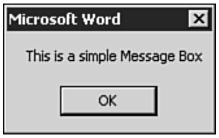
This statement includes several arguments:

- **Prompt**—This argument is a string expression, such as literal text or a string variable, that represents the message you want to display. The prompt is required any time you use the MsgBox statement.
- **Buttons**—This optional argument is a numeric value that determines what kind of buttons you want to appear, such as OK, Cancel, Yes, No, Abort, Retry, or Ignore. This value can also determine whether you want to display an icon in the message box, indicating a critical message icon, for example, or a question icon. If you omit this value, the message box will have an OK button and no icon. See Table 2 for more information on the values you can use for the Buttons value.
- **Title**—This optional argument is a string expression that represents the title area of the message box. If you omit it, “Microsoft Word” appears as the title.
- **Help**—This optional argument is the name of a help file that contains context-sensitive help for the message box. If you include a help file, you must also include a context.
- **Context**—This optional argument is a numeric value that represents a help topic within the help file you identified. The author of the help file must provide the context number for you.

Here are a few examples of using message boxes. In the simplest case, all you have to provide is the prompt for the message. All other arguments are optional. The following statement creates the message box shown in Figure 1:

```
MsgBox "This is a simple Message Box"
```

Notice that it has just an OK button and the standard message box title.



This is a simple Message Box

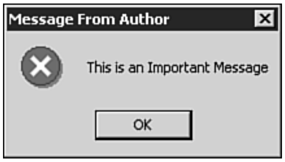
Figure 1
It takes only a single line of VBA code to display a message.

USING THE MSGBOX STATEMENT’S BUTTONS AND TITLE ARGUMENTS

To personalize the message box a little further, you can use the Buttons and Title arguments. The next statement creates the message box shown in Figure 2:

```
MsgBox "This is an Important Message", vbCritical, "Message from the Author"
```

Notice that this message includes the critical icon and a customized title.


Message From Author

This is an Important Message

Figure 2
A critical message gets the user's attention.

Notice the use of the predefined constant vbCritical in the previous example. VBA provides several constants for use in message boxes, as listed in Table 2. You can combine these values by adding them together.

TABLE 2 BUTTON CONSTANTS FOR THE MSGBOX STATEMENT

Constant	Description
vbOkOnly	OK button only; the default selection.
vbOkCancel	OK and Cancel button.
vbAbortRetryIgnore	Abort, Retry, and Ignore buttons.
vbYesNoCancel	Yes, No, and Cancel buttons.
vbYesNo	Yes and No buttons.
vbRetryCancel	Retry and Cancel buttons.
vbCritical	Critical Message icon.
vbQuestion	Warning Query icon.
vbExclamation	Warning Message icon.
vbInformation	Information Message icon.
vbDefaultButton1	First button is default.
vbDefaultButton2	Second button is default.
vbDefaultButton3	Third Button is default.
vbDefaultButton4	Fourth button is default.
vbApplicationModal	The user must respond to the message before continuing work in Word but can switch to another program; selected by default.
vbSystemModal	The user can't use any program on the computer until he or she responds to the message box; should be used with extreme caution and is not recommended.

GETTING INFORMATION FROM A USER WITH MSGBOX

You've probably seen a message box that has more than one button on it when you try to exit an application without saving your work. How does the application know which button you selected? This can be done because MsgBox can be used as a function, as well as a simple statement. A function is an expression that returns a value. The following is an example of the MsgBox function that returns which button is selected by the user:

```
IResult=MsgBox("This operation may take several moments. Continue?", _
vbYesNo+vbQuestion)
```

Two things about the MsgBox function in this example are different from the MsgBox statement. First, parentheses are used around the arguments. This is a common characteristic of functions in VBA. Second, the results of the function are assigned to a variable, in this example one named IResult. By testing the value of this variable, you can decide how your macro should proceed. In this case, the result will either be yes or no (vbYes or vbNo).

Note

If you don’t want to return a value from a function, you drop the parentheses. In this statement, no value is returned:

```
MsgBox "This is a test."
```

In this statement, a result is returned:

```
—IResult = MsgBox("This is a test. Do you wish to
continue?", _
vbOkCancel)
```

Consider the macro in Listing 1. In this example, the MsgBox function asks whether the user wants to continue. If the user answers Yes, the macro presents one message, but if he answers No, it displays a different message. If you’re not sure how the If statement works, see “Understanding VBA Control Structures,” later in this chapter.

Listing 1 The MessageBoxesExamples Macro Demonstrates the Use of MsgBox to Display Information and to Ask the User a Question

```
Sub MessageBoxesExamples()
    Dim intResult As Integer
    MsgBox "This is a simple Message Box"
    MsgBox "This is an Important Message", vbCritical, _
      "Message from the Author"
    intResult = MsgBox("Do you want to continue?", vbYesNo)
    If intResult = vbYes Then
        MsgBox "Enjoy the rest of the program."
    Else
        MsgBox "Thanks for coming. See you next time!"
    End If
End Sub
```

Notice the use of the predefined constant vbYes for comparison with the result of the message box. Table 3 lists the predefined constants available for the values that can be returned by the MsgBox function.

Table 3 Constants for Values Returned by the MsgBox Function

Constant	Description
vbOk	OK
vbCancel	Cancel
vbAbort	Abort
vbRetry	Retry
vbIgnore	Ignore
vbYes	Yes
vbNo	No

Getting More Information with Input Boxes

Although message boxes are great for displaying messages and asking simple questions, sometimes you need more information from the user than a simple yes or no. What if you want to ask the user's name, so you can type the name in several locations of a contract? You can use an input box to ask for this kind of information.

The InputBox function has a syntax similar to the MsgBox function:

```
result = InputBox (Prompt, Title, Default, xPos, yPos, Helpfile, Context)
```

First, examine each of the arguments:

- **Prompt**—This argument is a string expression representing the message or question you want to display. The prompt is required.
- **Title**—This argument is a string expression that represents the title area of the message box. If you omit it, "Microsoft Word" appears as the title.
- **Default**—This argument is a string expression entered as the value in a text box answer if the user doesn't type anything. This saves the user some time if it is likely that the same answer will be used often. If you omit this value, the text box will be empty.
- **xPos** and **yPos**—These arguments are optional numbers that enable you to position the input box on the screen. They represent the horizontal and vertical position of the upper-left corner of the input box, relative to the corner of the screen. If you omit these numbers, the input box is automatically positioned near the center of the screen. If xPos is omitted, the dialog box is horizontally centered. If yPos is omitted, the dialog box is vertically positioned approximately one-third of the way down the screen. To ensure that your macros work on a wide variety of Windows screen resolutions, you should generally omit these arguments.
- **Help File**—This optional argument is the name of a help file that contains context sensitive help for the message box. If you include a help file, you must also include a context.
- **Context**—This optional argument is a numeric value that represents a help topic within the help file you identified. The author of the help file must provide the context number for you.

Note

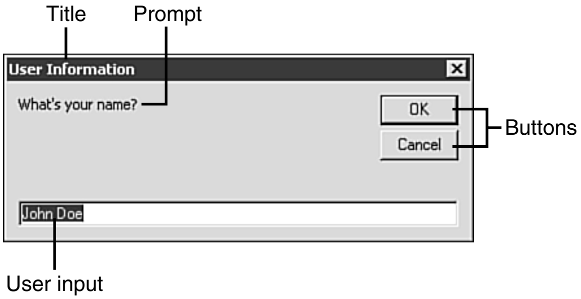
You are not likely to have access to the help context information for a help file unless you, or someone you know, has created the help file. To create your own help files, you need a third-party tool.

The InputBox function returns a string result representing the text that the user entered in the text box. The macro in Listing 2 asks for the user's name and then uses the variable txtYourName to display a message using the name. Figure 3 shows the input box that this macro presents.

LISTING 2 USE THE INPUTBOX FUNCTION TO ASK THE USER FOR TEXT INFORMATION

```
Sub InputBoxes()
    Dim txtYourName As String
    txtYourName = InputBox("What's your name?", _
       "User Information", "John Doe")
    MsgBox "Hello, " & txtYourName
End Sub
```

Figure 3
Use an input box to ask the user a question.



Notice that the name John Doe appears in the text box as the default. The user can type over this name with his or her own name.

UNDERSTANDING VBA CONTROL STRUCTURES

Many VBA macros run in a simple fashion. Starting with the Sub statement, they execute every statement in the macro, from start to finish. This is fine as long as you want the macro to do the same thing every time, but sometimes you need a macro that's a little smarter than that. VBA provides a wide variety of program control structures that enable you to change the way the statements are executed, depending on conditions that are found while the macro is running.

USING THE IF…THEN…END IF STATEMENT

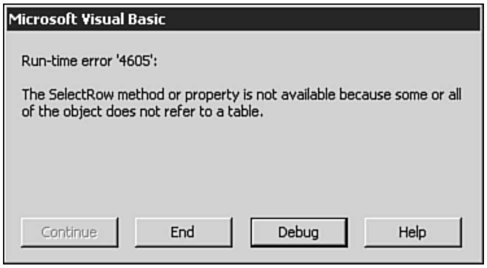
Perhaps the simplest and most commonly used control structure in VBA is the If…Then…End If statement. This statement enables you to execute a group of VBA statements only if some condition is true. Consider an example of why this might be necessary.

Suppose that you work with tables a lot, and you have recorded a simple macro that deletes the current row of the table you're using. That macro might look something like this:

```
Sub DeleteTableRow()
    Selection.SelectRow
    Selection.Rows.Delete
End Sub
```

This simple macro, assigned to a shortcut key, looks like a real timesaver. There's one problem, though. If you run the macro when you don't have the selection inside a table, an error occurs, as shown in Figure 4.

Figure 4
Runtime errors can be confusing to the users of your macros.


Microsoft Visual Basic

Run-time error '4605':

The SelectRow method or property is not available because some or all of the object does not refer to a table.

As you can see, the error message, although accurate, isn't necessarily the easiest thing to understand. By selecting End, you can cancel the execution of the rest of the macro. If you select Debug, you are dropped into the Visual Basic Editor, with the line where the error occurred highlighted. This can be confusing to a user who doesn't even know that he or she is using a macro.

Fortunately, there is a simple solution. You just need to check for whether the selection is in a table before attempting to delete a table row. This is where the If…Then…End If statement comes in. The following example tests the Information property of the Selection object to see whether the selection is in a table. If the result is True, the statements are executed. If the result is False, VBA skips over the statements, and no error occurs.

```
Sub DeleteTableRow()
    If Selection.Information(wdWithInTable) = True Then
        Selection.SelectRow
        Selection.Rows.Delete
    End If
End Sub
```

Note

You can use the Information property of the Selection object to find many different kinds of information. In the DeleTableRow example, the constant wdWithinTable indicates that you want to know whether the selection is within a table. As another example, you can use the constant wdCapsLock to determine whether the Caps Lock key is in effect. There are more than 30 predefined constants indicating different types of information you can determine with the Information property. For a complete list, see the help topic for the Information property.

You can improve on this example even further by using the optional Else clause of the If…Then…End If statement. The statements in the Else clause are executed only if the tested condition is False. The completed macro, shown in Listing 3, uses the Else clause to present a message to the user, reminding him or her to position the insertion point in a table before running the macro.

LISTING 3 USE THE IF…ELSE…THEN STATEMENT TO TEST FOR A CONDITION BEFORE EXECUTING STATEMENTS THAT MIGHT CAUSE A MACRO ERROR

```
Sub DeleteTableRow()
    If Selection.Information(wdWithInTable) = True Then
        Selection.SelectRow
        Selection.Rows.Delete
    Else
        MsgBox "Please position the insertion point " & _
            "in a table before running this macro."
    End If
End Sub
```

Note

The indention shown here with the If statement is not required but is highly recommended. The indentions make the statement easier to read.

A common use of If statements is to test the result of a MsgBox function. This code illustrates an example of testing a MsgBox function’s result:

```
Dim iResult As Integer

iResult = MsgBox("This operation may take several moments.", _
     vbOKCancel + vbInformation)

If iResult = vbOK Then
'statements to be executed if user clicks OK.
Else
    MsgBox "Procedure cancelled."
End If
```

Tip from
Bill Ray

When designing If statements, place the most likely test result first. This improves performance because VBA tests each condition until it finds the one that matches the item being tested. Placing the most likely test result last could affect performance in large If statements.

USING THE SELECT CASE STATEMENT

When you need to choose between more options than just True and False, you can use the Select Case statement. This statement is useful for selecting from among several, or even many, different options.

Suppose, in preparing invoices, that you need to add a statement to each invoice concerning sales tax. This message varies in each of the several states in which you do business, including some in which no sales tax is charged. The Select Case statement can be used to enter the text according to the response you get from an input box.

The macro in Listing 4 uses the InputBox function to request the state of the customer. Because most of the customers in this example are in New York, a default value of “NY” has been used in the input box. The result of the input is stored in the variable *txtState*.

The Select Case statement chooses its action depending on the value of the *txtState* variable. Each Case statement is compared to the variable. If it matches, the statement following the Case is executed, and a value is assigned to the variable *txtOutput*. If no match is found, the Else clause assigns a value to the variable, handling all other cases. Finally, the variable is used by the TypeText method to type the message into the document.

LISTING 4 USE THE SELECT CASE STATEMENT TO CHOOSE OPTIONS FROM A LIST OF ALTERNATIVES

```
Sub SalesTax()
    Dim txtState As String
    Dim txtOutput As String
    txtState = InputBox("What state do you live in?", _
         "Sales Tax", "NY")
    txtState = UCase(txtState)
    Select Case txtState
    Case "NY"
        txtOutput = "Please include 5.25% Sales Tax. "
    Case "NJ", "CT"
        txtOutput = "Please include 4.5% Sales Tax. "
    Case "PA"
        txtOutput = "Please include 6% Sales Tax. "
    Case Else
        txtOutput = "Sales Tax is not required in " & txtState & ". "
    End Select
    Selection.TypeText txtOutput
End Sub
```

The Select Case statement example tests for uppercase states. Users have been known to be a little unruly and may not enter the state in uppercase. Note the use of the UCase function in conjunction with the *txtState* variable. You can use the UCase function to convert the value of *txtState* to uppercase. This simplifies your work as far as the values you need to include in the Select statement.

Tip from
Bill Ray

When designing Select statements, place the most likely test result first. This improves performance because VBA tests each condition until it finds the one that matches the item being tested. Placing the most likely test result last could affect performance in large Select statements.

USING FOR…NEXT LOOPS

A common requirement in macros is to repeat one or more statements many times. This process is often referred to as *looping*. VBA provides several looping statements, each of which meets a different programming requirement.

The For…Next loop is useful when you want to repeat a block of statements a set number of times. Here is the syntax of the For…Next loop:

```
For counter = startvalue To finishvalue
statement        ...
Next counter
```

In this structure, *counter* is a variable that increases by 1 each time through the loop, starting with *startvalue*, and ending with *finishvalue*. The statements between the For and Next statements are performed each time through the loop.

For example, suppose that you want to print a document that shows all the printable characters in a particular font. Because the printable characters in most fonts are those ANSI values between 32 and 255, you can use the For…Next loop to create a table with each row representing one of the printable characters. The macro in Listing 5 creates a new document, adds a table to the document, and then adds a row for each character in the ANSI character set.

LISTING 5 USE THE FOR…NEXT STRUCTURE TO REPEAT VBA STATEMENTS A PREDETERMINED NUMBER OF TIMES

```
Sub CharacterSet()
    Dim intCounter As Integer

    Documents.Add DocumentType:=wdNewBlankDocument
    ActiveDocument.Tables.Add Range:=Selection.Range, _
        NumRows:=1, _
        NumColumns:=2, _
        DefaultTableBehavior:=wdWord9TableBehavior, _
        AutoFitBehavior:=wdAutoFitContent
    Selection.TypeText Text:="Number"
```

Listing 5 Continued

```
    Selection.MoveRight Unit:=wdCell
    Selection.TypeText Text:="Character"
    Selection.MoveRight Unit:=wdCell

    For intCounter = 32 To 255
        Selection.TypeText intCounter
        Selection.MoveRight Unit:=wdCell
        Selection.TypeText Chr(intCounter)
        If intCounter < 255 Then
            Selection.MoveRight Unit:=wdCell
        End If
    Next intCounter
End Sub
```

Besides the looping statements, this macro contains several statements that demonstrate the use of Word's object model.

- The Documents.Add method creates a new document. The constant wdNewBlankDocument indicates that a blank document should be created, rather than an e-mail message or a Web page.
- The ActiveDocument.Tables.Add method adds a new table to the active document, at the location of the selection, containing 1 row and 2 columns.
- The Selection.MoveRight method moves to the next cell in the table because the Unit argument is assigned the constant wdCell. This is the same as pressing the Tab key when you are editing a table.

For more information on these and other Word objects, see the online help.

Using Do…Loop Statements

The For…Next loop is great if you know how many times you want to go through a loop, but sometimes there is no way to know at programming time how many times you'll need to execute the statements. This is where the Do…Loop statement comes in.

The syntax of the Do…Loop statement is simple:

```
Do While condition
statement
...
Loop
```

In this syntax, *condition* is an expression that can be interpreted as either True or False. If the condition is True, the loop's statements are executed and then the condition is checked again. If the condition is still true, the statements are executed again. This process is repeated until the condition becomes False. When the condition tests False, the statements are skipped, and VBA resumes execution with the first statement after the Loop statement.

There is an important concept to remember about the Do…Loop. Among the statements within the loop, you must provide some way for the condition to become False. Otherwise, you will have an endless loop. When you run code that has an endless loop, there is nothing to make the code stop executing, and it runs "forever." The only way to stop a program caught in an endless loop is to press Ctrl+Esc, which breaks the execution of the macro. Of course, when you stop a running macro by pressing Ctrl+Esc, you have no control over exactly how many statements of the macro will have been executed.

For example, suppose that you need to search a document for all occurrences of the word "employee," and underline them. The macro in Listing 6 repeats the find operation and then underlines each occurrence, until no more occurrences of the target word are found.

LISTING 6 USE THE DO…LOOP STRUCTURE TO REPEAT VBA STATEMENTS UNTIL A CONDITION BECOMES FALSE

```
Sub FormatAll()
    Selection.HomeKey wdStory
    Selection.Find.ClearFormatting
    With Selection.Find
        .Text = "employee"
        .Replacement.Text = ""
        .Forward = True
        .Wrap = wdFindAsk
        .Format = False
        .MatchCase = False
        .MatchWholeWord = False
        .MatchWildcards = False
        .MatchSoundsLike = False
        .MatchAllWordForms = False
    End With
    Do While Selection.Find.Found
        Selection.Words.Underline = wdUnderlineSingle
        Selection.Find.Execute
    Loop
End Sub
```

In the line

```
Do While Selection.Find.Found
```

the expression Selection.Find.Found returns the True or False condition of whether the text was found. Although you might not know how many occurrences of "employee" you will find, you certainly know that eventually you will find the last one, after which the Found property will be False. The statement Selection.Find initiates the Find operation that then sets the value of the Selection.Find.Found property.

In addition to the Do…Loop that you have already seen, you can use three variations on the logic. In the first example, you can move the While test to the end of the loop. This guarantees that the statements in the loop will execute at least once. The While test is performed only after the statements have executed for the first time.

```
Do
statement
...
Loop While condition
```

The next variation uses the keyword Until instead of While. In this version of the loop, the statements are performed only if the condition is False, and the loop is repeated until the condition becomes True.

```
Do Until condition
statement
...
Loop
```

The final variation of the Do…Loop statement moves the Until test to the end of the loop. Once again, this technique ensures that the statements run at least once because the test is not performed until the statements have executed at least once.

```
Do
statement
...
Loop Until condition
```

Using For Each…Next Loops

VBA provides a looping structure for working with collections of an object the For Each…Next statement. Its syntax is fairly simple:

```
For Each objVar In Collection
    statements
    ...
Next objVar
```

It is important to remember two things, however, about the For Each…Next loop:

- Collection represents one of the built-in object collections in Word, such as Documents, Windows, Paragraphs, Fonts, and so on.
- You must declare the object variable *objVar* using the correct type for an object in the collection.

The macro in Listing 7 illustrates the use of For Each…Next to loop through the collection of open documents. To print a list of the open documents, the macro creates a new document and then types the name of each open document into the new document.

Listing 7 Use the For Each…Next Statement to Repeat VBA Statements for Each Member of a Collection

```
Sub ListOpenDocuments()
    Dim objDoc As Document, strNewDocName As String

    If Documents.Count > 0 Then
        Documents.Add
        strNewDocName = ActiveDocument.Name
        For Each objDoc In Documents
            If objDoc.Name <> strNewDocName Then
                Selection.TypeText objDoc.Name
                Selection.TypeParagraph
            End If
```

LISTING 7 CONTINUED

```
        Next objDoc
    Else
        MsgBox "No documents are currently open"
    End If
End Sub
```

The collection in this case is called Documents. This collection is always available to your Word VBA macros, and it represents all the open documents at the time you run your macro. If no documents are open, the collection still exists, but it has zero members. The Count property of any collection contains the number of elements in the collection.

Notice that you must declare the variable *objDoc* as the type Document. Remember, the Documents collection is made up of individual Document objects.

The macro tests the Count property of the Documents collection, so that it can show a message box if there are no open documents. If any documents are open, the macro creates a new document and then types the Name property of each open document into the new document. Because you wouldn't want to include the name of the new document that contains the names that will be printed, the example compares the Name property of each document to the name of the new document, which was stored in the variable *strNewDocName*.

TROUBLESHOOTING

WHAT TO DO IF YOU GET AN ERROR MESSAGE SAYING A VARIABLE HAS NOT BEEN DEFINED

You have either typed `Option Explicit` in the General Declarations section of your module or you have selected to require variable declarations through the Options dialog box. Because of this, you have to use a Dim statement to declare your variables. You tried to use a variable that had not been declared. Make sure to declare all variables before using them and check the spelling of variables carefully.

WHAT TO DO WHEN YOU TYPE THE FIRST LINE OF YOUR IF STATEMENT AND YOU GET A SYNTAX ERROR WHEN YOU PRESS ENTER

The message you see should tell you exactly what you did wrong. It states that it expected a Then or GoTo. You forgot to type `Then` at the end of the first line of your If statement. This is one of the most common typos in VBA.

WHAT TO DO IF THE CODE YOU RUN GIVES UNEXPECTED RESULTS

The first thing to remember is that the VBE performs syntax checking when you press Enter. The statement you have has the correct syntax. What has probably happened is that you have a logic error. If your MsgBox function has, for example, vbOkCancel as its argument and you test for vbYes and vbNo, you are not going to get the results you expect because you are testing for the wrong results. Look at your MsgBox statement and then look at the If statement and make sure that you are testing for the right results.

Creating a Real-World Solution with VBA and Word

In this chapter

by Bill Ray

Planning an Automated Solution in Word 2002

If you've read the other two VBA-related SE Using Word 2002 Web Chapters, "Getting Started with VBA," and "Taking More Control of Your Macros," then you should be getting more comfortable with VBA language concepts such as objects, properties, and methods. You should also have had some experience working with the various VBA statements and be able to declare variables, add logical control, display message boxes, and so on. The question that is probably before you now is, "How do I build a solution from scratch?"

Before you start automating a task or process, you should understand the requirements for that task or process and the benefits that can be gained through automation. By evaluating the requirements, you can determine the degree of automation that you need and the approach the design should take. Some important design features you should consider are

- **Document Layout**—What are the requirements for paper size, portrait or landscape orientation, margins, headers, and footers? Don't forget to account for international factors, such as metric paper sizes, if you work in a large organization.
- **Formatting and Styles**—What fonts do you want to use? What are all the different paragraph styles, such as indenting, topic headings, bullets, numbering, or any other special formatting?
- **Special Elements**—Does the document require any special items, such as graphics for a logo, repetitively used text, tables, or anything else that you would be likely to use in this document?
- **User Input**—How do you expect the user to work with the document? Can he or she enter text anywhere in the document, or are there requirements for specific text in certain locations or sequences?
- **Standards**—How do all these elements tie into your organization's document standards?

Identifying and documenting these requirements helps you determine the best strategy for designing your solution. For example, some documents require consistent formatting and page layout but accept text entered wherever the user wants it. In this case, you can design a template with the appropriate styles but little or no VBA code. In other cases, it might be important that information is entered in specific locations on a detailed form so that the data can be transferred to another file automatically. In this case, you can either create a protected form or use VBA code to gather and type the information through custom dialog boxes.

One key to developing effective and flexible solutions in Word is to let Word do as much of the work as possible. Developers who have programmed in other computer languages are often tempted to write VBA code to do everything, when they can rely on template design, styles, AutoText, and many other features to do more of the work, minimizing the VBA code they have to create and maintain.

UNDERSTANDING THE ROLE OF TEMPLATES

After you assess your requirements, you can start the hands-on development of your solution by creating a template. The template contains the formatting and layout of your application, together with the macro code and custom forms that control the automation.

Remember that every Word document is based on a template. Many users create most of their documents on the global template, called Normal.dot. To create more specific solutions, however, it is often a good idea to create a specific template that contains the characteristics of any special document, such as a letter, fax cover, office form, or sales proposal.

You can build much of the design of an application into the template, without writing unnecessary VBA code. The elements that you can capture in the template design for your solution are as follows:

- **Page Layout**—Set the paper size, margins, and other overall document layout requirements of your application. If necessary, divide the template into sections to accommodate settings that change throughout the document.
- **Styles.** Create styles to meet your requirements for fonts and paragraph formatting.

➔ For more information about working with styles, **see** Chapter 11, "Streamlining Your Formatting with Styles," **p. 331**.

- **Text and Graphics**—You might have a requirement for text that appears on every document, such as memo or fax cover headings, headers and footers, or a return address. You might want a graphics element, such as a logo, to appear at a predetermined location. You can enter any such text and graphics into your template at design time so that they appear on every document created with the template.
- **AutoText**—The user might need to enter set passages of text at various locations throughout the document. Creating these text items as AutoText entries in the template makes them available for insertion by the user at any time. Your VBA code can also insert the contents of AutoText entries, based on user requirements at a later time.

➔ For more information about working with AutoText, **see** Chapter 10, "Automating Your Documents," **p. 295**

- **Bookmarks**—You can identify important locations in the template by defining bookmarks. A bookmark can represent the spot where text—such as a memo heading—must be entered. You can also define a bookmark to cover a range of text, which your macro can use for copying or deleting.

Tip from
Bill Ray

You may not think of deleting text as an operation that your macro would need to perform on a template, but some templates that you use may contain standardized text. For example, a template might contain a long passage of text, such as a disclaimer, which is only required in some instances. Depending on the user's selections in the opening dialog box, the macro might delete the text.

➔ For more information about working with bookmarks, **see** Chapter 23, "Footnotes, Bookmarks, and Cross-References," **p. 721**

- **VBA Code and Customizations**—After you have created the template and customized its features, you can begin creating the VBA code and other customizations specific to this template. If you store menu, toolbar, or keyboard customizations in the template, they will automatically be in effect any time you are using the template.

➔ For more information about working with templates, **see** Chapter 12, "Templates, Wizards, and Add-Ins," **p. 357**

Consider the example of creating a template to produce memos for the ABC Corporation. The employees of this organization create a lot of memos, and they find it time-consuming to create them from scratch. An automated template can address several challenges of memo creation:

- **Consistency of Design**—Not everyone's Word skills are consistent, and some people don't take care to follow the company's design standards. Memos therefore appear with different fonts, margins, bullets, and other features. An automated template will result in memos that follow the company's design.
- **Saving Time**—Those who do follow the company standards spend a lot of extra time putting their work in the correct format. They repeat the same actions, such as typing heading text, creating tables, and applying formatting, each time they create a new memo. If an automated template is available, however, they can eliminate these redundant actions.
- **Cutting Cost**—Besides reducing the time spent on repetitive tasks, an automated template will cut the costs of time spent correcting errors and responding to support calls from users who need help with basic features of Word. It can also reduce the cost of test printouts and revision and reformatting.
- **Automating HTML Memos**—As more of the internal documents of ABC Corporation are being published on the company's intranet, users need a way to quickly create the HTML version of their memos and save them in the correct location on the company's network. This eliminates the need for someone to translate documents from traditional formats to HTML.

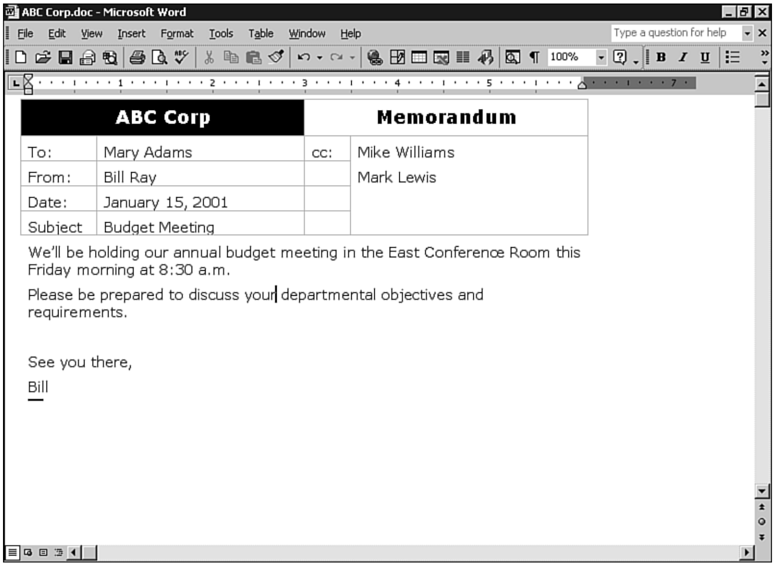
As a result of the creation of an automated memo template, users of all skill levels can create company memos quickly and in a fashion consistent with company design standards.

Note

Word 2002 comes with a variety of sample templates that you can use as starting points for your template design. Word has templates for creating letters, memos, fax covers, rèsumès, and other standard documents.

If your organization's design standard is similar to the design of these templates, you can modify them to meet your requirements. Otherwise, you might find that it is more efficient to create your own templates from scratch.

An example of the ABC Corporation memo is shown in Figure 1.



ABC Corp		Memorandum	
To:	Mary Adams	cc:	Mike Williams Mark Lewis
From:	Bill Ray		
Date:	January 15, 2001		
Subject	Budget Meeting		

We'll be holding our annual budget meeting in the East Conference Room this Friday morning at 8:30 a.m.

Please be prepared to discuss your departmental objectives and requirements.

See you there,

Bill

Figure 1
The CorpMemo template provides all the design and automation for ABC Corporation's memos.

Consider several important items in the design of the template:

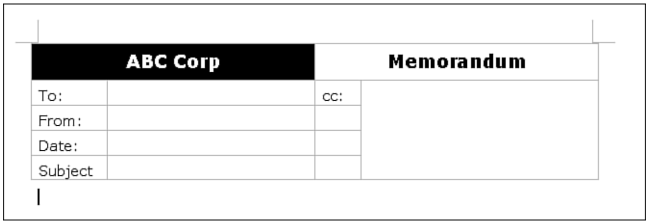
- The headings for To, From, Date, and Subject must be aligned consistently, and the information entered to the right of each heading must also be aligned.
- The date should be entered automatically, but the user must be able to edit the date.
- The cc list is optional. If no cc's are entered, the cc label should not appear on the memo.
- The standard font is Arial, 12 point. Paragraphs are left aligned, with an extra space added after each paragraph.

Creating the Basic Template

To begin creating your template, create a new document based on the Normal template or on a template similar to the memo you want to create. Enter the text, graphics, table, and other formatting you want in the new template.

Figure 2 shows the new memo template after the basic text elements have been entered. The memo headings have been entered into a table, with the table widths adjusted to match the desired alignment. A larger area for the cc list has been created by merging three table cells.

After you have created the basic layout, save your template under a new filename.



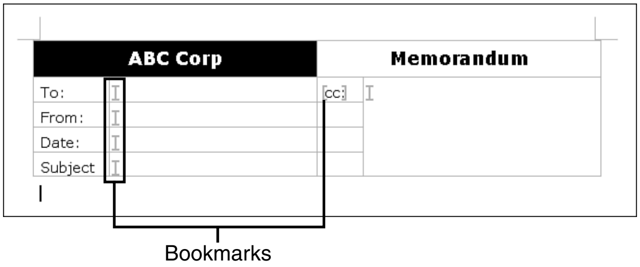
ABC Corp		Memorandum	
To:		cc:	
From:			
Date:			
Subject			

Figure 2
The CorpMemo template uses a table to align the memo headings and to accommodate user input.

Using Bookmarks for Later Coding

You can mark a location in your template by creating a bookmark. Bookmarks provide an easy way for you to enable your VBA code to enter text at a specific location. You can also use bookmarked locations for formatting, copying, or a destination for the cursor.

Figure 3 shows the CorpMemo template with bookmarks displayed. The macro uses several bookmarks:



ABC Corp		Memorandum	
To:		[cc:]	
From:			
Date:			
Subject			

Figure 3
Bookmarks identify the locations for entry of key data in the memo.

- The bookmarks named bkTo, bkFrom, bkDate, bkSubject, and bkCC identify the locations where the VBA code will enter text, based on the user's input.
- The bookmark named bkCCLabel contains the text of the label next to the cc area of the memo. If the user doesn't enter any cc text, our VBA code deletes this label.

Tip from
Bill Ray

You can name your bookmarks anything you want. However, beginning each bookmark name with a prefix such as *bk* makes it easy to distinguish bookmark names from other names in your code.

Designing the User Interface

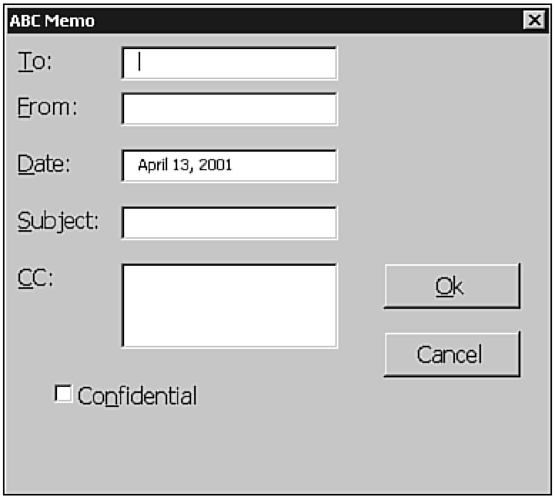
After you have completed your template design, it's time to begin creating your VBA code. As you can see, it is common to invest quite a bit of time into the template design, before even writing the first line of code.

The Visual Basic Editor (VBE) provides form design tools, which you can use to create custom dialog boxes. These dialog boxes have several advantages over requiring the user to enter text directly into a document:

- Your VBA code can enter the input text in the correct location, or even at multiple locations throughout a document.
- You can make data entry easy for the user, by adding list boxes, check boxes, option buttons, and other controls to the forms.
- You can validate the data entry, making sure that all data has been entered before the document is created.

Custom dialog boxes, also known as *forms*, are created in the VBE. You can create a custom form by adding controls—such as text boxes, labels, and check boxes—to a blank form. The VBE also enables you to install additional custom controls available commercially.

Plan your form by considering what data is required and what types of controls you need to get data from the user. Figure 4 shows the form that the CorpMemo template will use to gather data from the user.



Date: April 13, 2001

Figure 4
A custom dialog box makes data entry for a corporate memo easy and reliable.

Four types of controls are used in the form:

- **Text Boxes**—These controls enable the user to enter the text for the To, From, Date, Subject, and cc list.
- **Command Buttons**—These controls are used for the OK and Cancel buttons. The user clicks one of these buttons to close the form, choosing either to create a memo or to cancel the operation.

- **Labels**—These controls are used to display text on the form. The user can't edit label text, so it's good for showing instructions or identifying areas of the form.
- **Check Boxes**—These controls are referred to as toggle controls. By clicking the check box, the user alternates between placing and removing a check mark in the box. Check boxes are used to turn options and features on and off.

CREATING A CUSTOM FORM

The first step in designing the form is to add a form to the template's VBA project. Follow these steps:

1. With your template open, activate the VBE by pressing Alt+F11.
2. Open the Project Explorer window, if it isn't already open.
3. Select the TemplateProject item from the Project Explorer window that corresponds to the template on which you are working.
4. Choose Insert, UserForm. A new form is added to the project and is open for editing.

A big part of form design involves changing the properties of the form and its controls. You can use the Properties window to view and change these properties. If the Properties window is not displayed, choose View, Properties Window to display the window.

When you select the form, or any object on the form, that object's properties are displayed in the Properties window. You can change a property by selecting that property and entering a new value for it.

You should immediately change two properties for any form when you begin working on it:

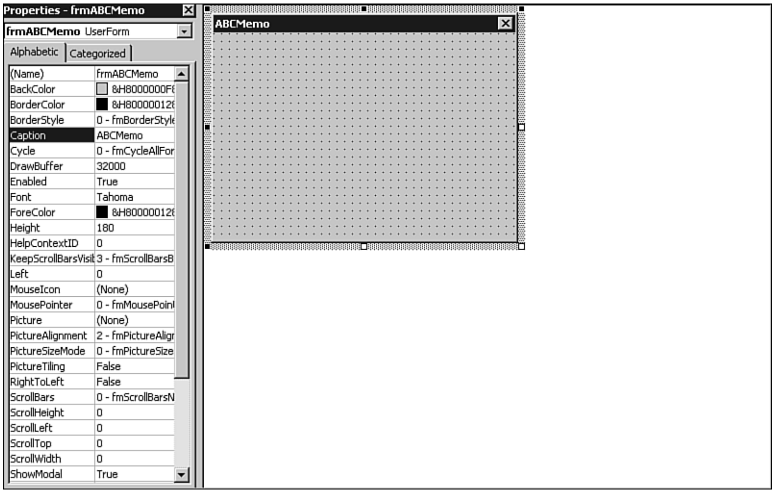
- **Name**—The Name property identifies the form in your VBA code. Using a descriptive name makes the code easier to read. In the CorpMemo template, the form is named frmABCMemo. Note the use of the prefix frm to indicate that this is the name of a form.
- **Caption**—The Caption property of a form contains the text that appears in the title bar of the form. This should be descriptive text that identifies the purpose of the form to the user.

Figure 5 shows a new form after its Name and Caption properties have been set. The Name property of the form has been set to frmABCMemo, and the Caption property has been set to ABC Memo.

Tip from
Bill Ray

There are two ways that you can resize a form. One way is to drag the sizing handle of the form until the form is the size you want. Another way is to set the form's Height and Width properties.

Figure 5
Use the Properties window to modify the properties of a form or its controls.



(Name)	frmABCMemo
BackColor	&H8000000F8
BorderColor	&H80000012(
BorderStyle	0 - fmBorderStyle
Caption	ABCMemo
Cycle	0 - fmCycleAllFor
DrawBuffer	32000
Enabled	True
Font	Tahoma
ForeColor	&H80000012(
Height	180
HelpContextID	0
KeepScrollBarsVisit	3 - fmScrollBarsB
Left	0
MouseIcon	(None)
MousePointer	0 - fmMousePoin
Picture	(None)
PictureAlignment	2 - fmPictureAligr
PictureSizeMode	0 - fmPictureSize
PictureTiling	False
RightToLeft	False
ScrollBars	0 - fmScrollBarsN
ScrollHeight	0
ScrollLeft	0
ScrollTop	0
ScrollWidth	0
ShowModal	True

ADDING CONTROLS TO THE FORM

To add controls to a form, you must display the control Toolbox. If the Toolbox is not displayed, choose View, Toolbox to display it.

You can place a variety of controls on your form. Each control is designed for a specific use. Table 1 describes the controls and their uses as well as provides recommended naming prefixes for the controls.

TABLE 1 BASIC CONTROLS

Control	Naming Prefix	Description
Label	lbl	Used to display static text.
Text Box	txt	A box in which a user can enter text or numbers.
Combo Box	cbo	Combination of a text box and a list box.
List Box select from.	lst	Used to display a list of text strings that the user may
Check Box	chk	A box that specifies whether an option is on or off (True or False).
Option Button	opt	A button used to select one of a group of mutually exclusive settings.
Toggle Button	tog	A button used to show whether an item is selected.
Frame	fra	A border used to organize and group other controls. For example, option buttons are often placed within a frame.
Command Button	cmd or btn	A command button signals to a user that a command will execute when the button is clicked. Examples of command buttons are OK and Cancel.

TABLE 1 CONTINUED

Control	Naming Prefix	Description
Tab Strip	tab	A control that contains a collection of one or more tabs. Each tab of a tab strip is a separate object that users can select.
Multipage	mul	Presents multiple screens of information as a single set.
Scroll Bar	hsb or vsb	A horizontal or vertical scrollbar used for changing numeric values.
Spin Button	spn	A pair of buttons used for incrementing or decrementing a displayed value.
Image	img	Used to display an image.

After you have decided which controls you need for your form, you can use the Toolbox to add the selected controls to your form, following these steps:

1. Click the Toolbox on the control you want to add to the form.
2. Click the form at the location where you want the control to appear.
3. Drag the control to position and size it accordingly.
4. Enter and change information in the control's properties to fit its specific use.

ADDING THE LABELS AND TEXT BOXES

The first control to place on the form is a label for the To: line of the memo headings.

1. Click the form to select it. If the Toolbox is not visible, choose View, Toolbox to display it.
2. Click the Label control on the Toolbox.
3. Point to the desired location on the form and click to place the control on the form.
4. Using the mouse, adjust the size and position of the control as necessary. As shown in Figure 6, use the *sizing handles*, the boxes that appear around the control when it is selected.

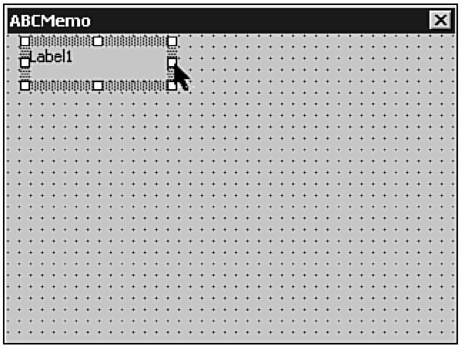


Figure 6
You can resize a label on your custom form using its sizing handles.

5. While the control is selected, use the Properties window to change the Caption property of the control. Click the Caption property and type `To:`.

The next control to add to the form is a text box, in which the user can type the name of the person who is to receive the memo.

1. Click the Text Box control.
2. Point to the desired location on the form and click to place the control on the form.
3. Use the mouse to adjust the size and position of the text box.
4. Change the Name property of the text box, using the Properties window. Change the Name to txtTo. The prefix of *txt* serves as a reminder that this control contains text.

Tip from
Bill Ray

The VBE provides a grid that aids you in the placement of controls in your grid. You'll notice that when you add a control on the form that it aligns itself to the nearest grid marks. To control whether the control automatically aligns itself to the grid, select Tools, Options and go to the General tab. The Align Controls to Grid check box controls this feature.

Following the same steps, add the following labels and text boxes to the form:

1. Add a label and change its Caption property to `From:`.
2. Add a text box and change its Name property to `txtFrom`.
3. Add a label and change its Caption property to `Date:`.
4. Add a text box and change its Name property to `txtDate`.
5. Add a label and change its Caption property to `Subject:`.
6. Add a text box and change its Name property to `txtSubject`.
7. Add a label and change its Caption property to `cc:`.
8. Add a text box and change its Name property to `txtCC`.
9. Change the MultiLine property of the txtCC text box to `True`, so that it can accommodate more than one line of entries in the cc list. Change the size of this text box to give more room for the display of the names that will be entered.

Tip from
Bill Ray

You can use the position of one control to set the alignment of other controls. Select the controls to align. Then select the control you want to use as the basis for alignment. Select Format, Align. Pick the type of alignment you want to perform from the Align submenu.

After you have a set of controls aligned, you may choose to group them together so that when you move them they stay aligned. You can group controls by selecting the controls and choosing Format, Group.

Using Shortcut Keys in Custom Forms

To set the shortcut or accelerator key for a control, you need to enter a value for the Accelerator property of the control. For example, if you had a label with the caption of From:, you would probably want the accelerator key to be F. Enter `F` as the value of the Accelerator property and you'll see the caption of the label appear as <u>F</u>rom:.

In our sample form, we set the first letter of each label as the accelerator key.

Adding the Check Box

At the bottom of the form you are going to add a check box that allows a user to select whether he wants to add the text CONFIDENTIAL to the top of the document. To add the check box to the form, complete the following steps:

1. Click the Check Box control.
2. Click to place the control on the form.
3. Use the mouse to adjust the size and position of the check box as needed.
4. Change the Name property of the check box to `chkConfidential`. The prefix of *chk* serves as a reminder that this control is a check box.
5. Change the Caption property to `Confidential`.
6. Set the Accelerator property to `n`.

Adding the Command Buttons

The last two controls are the command buttons that the user can click to close the custom form.

1. Click the Command Button control in the Toolbox.
2. Point to the location on the form where you want the command button to appear and click the form to place the button.
3. Use the mouse to adjust the size and position of the control.
4. Change the Name property of the button to `btnOk` and the Caption property to `OK`. Set the accelerator key to `O`.
5. Add a second command button, changing its Name property to `btnCancel` and its Caption property to `Cancel`. Figure 7 shows the completed form.

Now that you have created the custom form and you have named the form and its controls, you are ready to begin writing the VBA code that automates the form.

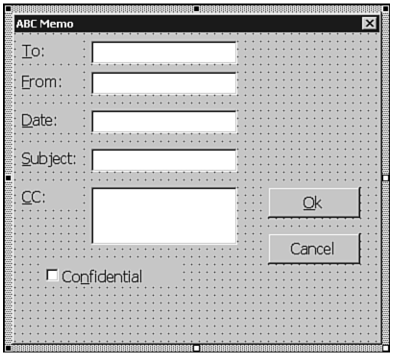


Figure 7
The completed form in Design view.

Controlling a Form with VBA

Several steps are involved in using a form under the control of VBA. It's a good idea to think about each of these steps every time you design a form:

1. Your macro should initialize the form by setting the properties of some of its controls before it displays the form. For example, you might want to make sure that the text boxes in the form are blank, or you might want to set them to a default value.
2. Next, your macro displays the form so that the user can interact with it.
3. You can choose to respond to the events that occur while the form is open. VBA automatically monitors the events that occur while a form is displayed, such as entering text into a control, moving from one control to another, or clicking a command button.
4. You can dismiss the form by using the Hide method. This method makes the form disappear but keeps the form in memory so that you can read the data that the user entered into the form.
5. Now your VBA code can read the data that the user entered and perform the desired actions. In the example, the text that the user entered in the text boxes will appear in the new memo document being created.

Where should all this VBA code go? In the example, the code falls into three categories:

- The main part of the code is stored in the document module that is a part of every template. This code executes automatically every time a user creates a new memo based on the template.
- Some of the code belongs in the event procedures of the form. In this case, all that is needed is some code to respond to the user clicking the OK and Cancel buttons. You have to write the code that closes the form when the user clicks OK or Cancel.
- We'll also use a standard module to contain a global variable to keep track of which button the user clicked. Later, this same module can be used to contain a macro for saving an HTML version of the memo.

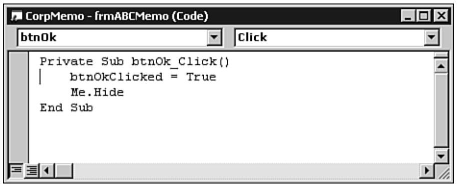
Creating the Macros for OK and Cancel

The first step is to create the macros that respond to the events of the user clicking the OK and Cancel buttons.

Each form that you create is represented by a code module in the Project Explorer window. This module has the same name as the form itself. In the example, the module's name is frmABCMemo. To edit the code module for a form, select the form in the Project Explorer window and then click the View Code button in the Project Explorer or press F7.

In the code window for a form module, you can edit the event procedures for the form and for the controls on the form. In Figure 8, the procedure for the Click event of the btnOk object is displayed. VBA automatically names this procedure btnOk_Click. To edit this |procedure, select btnOk in the Object list and then select Click in the Procedure list. This procedure is automatically called when the user clicks the OK command button.

Figure 8
The Click procedure for btnOk runs when the user clicks the OK button.

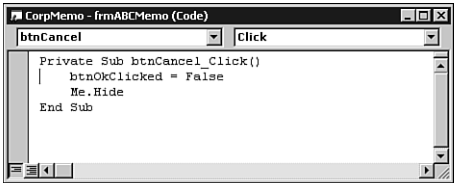


```
Private Sub btnOk_Click()
    btnOkClicked = True
    Me.Hide
End Sub
```

As you can see, the code for this procedure is simple. The first line sets the value of a variable named *btnOkClicked* to True. This variable is used to determine whether the user clicks OK or Cancel. The next step of code development is to declare this variable as a public variable so that the main macro can read its value. The second line of code calls the Hide method of the form to dismiss the dialog box. The VBA keyword Me refers to the form object that contains the running VBA code. Therefore, the expression Me.Hide closes the current form.

The code for the Click event of the Cancel button is similar to the code in the btnOk_Click procedure. Figure 9 shows the Click procedure for btnCancel. The only difference is that the variable *btnOkClicked* is set to False.

Figure 9
The Click procedure for btnCancel runs when the user clicks the Cancel button.



```
Private Sub btnCancel_Click()
    btnOkClicked = False
    Me.Hide
End Sub
```

Declaring Variables

To use the variable *btnOkClicked* in the main macro, the variable must be declared as a public variable. The declaration statement must appear in a standard module in the template, not in the form module. Follow these steps to add a standard module to the template project and to declare the variable.

1. Select the TemplateProject object for the memo template in the Project Explorer window.
2. Choose Insert, Module. A new module appears under the Modules branch of the TemplateProject object.
3. Select the new module and press F7 to open the module's code window.
4. Type the following line of code in the code window to declare the public variable:

```
Public btnOkClicked as Boolean
```

This statement declares that the variable named *btnOkClicked* is public, so that it can be used by any procedure in the current project, including the form module and other event procedures. This also ensures that the variable holds its value as long as the project is open. The data type of the variable is Boolean, which means that it can have only two values, True and False.

Tip from *Bill Ray*

The reserve word Public can also be used with constants. The syntax to create a public variable is

```
     Public Const constant_name = value
```

Note

You can have as many code windows open as you want in the VBE, but you might find that this gets confusing after a while. After editing the VBA code in any code window, you can close the window and reopen it whenever you want.

Your code is saved only when you save the template that contains the code. Remember to click the Save button on the VBE toolbar often to save your code.

CREATING THE MAIN MACRO

Now for the main event. The next step in the solution development process is to create the macro that displays and responds to the custom form. Because you want to display the custom form each time a user creates a new memo document, you can take advantage of the built-in event model of the template's document object.

Each template or document project contains a module called ThisDocument, which is displayed under the Microsoft Word Objects branch of the Project Explorer window. To edit the code in this module, select ThisDocument in the Project Explorer window and press F7 to open the module's code window.

➔ To get more information on the code window, **see** the Bonus Word Chapter entitled, "Getting Started with VBA."

Select the Document object in the Object list of the code window. The Procedure list now contains three event procedures:

- **Document_New**—This procedure runs automatically each time a new document is created based on this template.
- **Document_Open**—This procedure runs automatically each time a document based on this template is opened.
- **Document_Close**—This procedure runs automatically each time a user attempts to close a document based on this template.

Because this macro needs to run each time a new document is created, select the Document_New procedure. Listing 1 shows the code for this procedure.

LISTING 1 THE DOCUMENT_NEW PROCEDURE

```
Private Sub Document_New()

    With frmABCMemo
        .txtTo = ""
        .txtFrom = ""
        .txtDate = Format(Date, "mmmm d,yyyy")
        .txtSubject = ""
        .txtCC = ""
        .txtTo.SetFocus
    End With

    frmABCMemo.Show

    If btnOkClicked = True Then
        If frmABCMemo.chkConfidential = True Then
            Selection.HomeKey Unit:=wdStory
            Selection.Font.Bold = True
            Selection.ParagraphFormat.Alignment = wdAlignParagraphCenter
            Selection.TypeText Text:="CONFIDENTIAL"
            Selection.TypeParagraph
            Selection.TypeParagraph
        End If
        Bookmarks("bkTo").Range.Text = frmABCMemo.txtTo
        Bookmarks("bkFrom").Range.Text = frmABCMemo.txtFrom
        Bookmarks("bkDate").Range.Text = frmABCMemo.txtDate
        Bookmarks("bkSubject").Range.Text = frmABCMemo.txtSubject
        If Len(frmABCMemo.txtCC) > 0 Then
            Bookmarks("bkCC").Range.Text = frmABCMemo.txtCC
        Else
            Bookmarks("bkCCLabel").Range.Delete
        End If
        Selection.EndKey wdStory
        Options.DisplayGridLines = False
        With ActiveDocument.ActiveWindow.View
            .ShowBookmarks = False
            .ShowAll = False
        End With
    Else
```

Listing 1 Continued

```
        ActiveDocument.Close wdDoNotSaveChanges
    End If
End Sub
```

Understanding the Macro Code

The easiest way to understand the code for the Document_New procedure is to break it into parts. The following sections describe in detail what is being accomplished with the statements of the procedure. By reviewing the code statements, you can understand the logic and flow of the procedure.

Initializing Text Boxes and Displaying the Form

The first part of this procedure initializes the text boxes in the form. Using the With…End With block makes it easy to address the controls in the form. Most of the text boxes are initially set to empty strings, ensuring that no leftover text remains from the last time you ran the macro. The only exception is the txtDate text box, which is initialized to the current date, in a full date format. The last statement in the With…End With block calls the SetFocus method to move the insertion point to the txtTo text box when the form appears.

As soon as the initialization is complete, the form's Show method is used to display the form. Listing 2 shows the initialization code for this procedure.

Listing 2 Statements Initializing Text Boxes and Displaying the Form

```
With frmABCMemo
        .txtTo = ""
        .txtFrom = ""
        .txtDate = Format(Date, "mmmm d,yyyy")
        .txtSubject = ""
        .txtCC = ""
        .txtTo.SetFocus
    End With

  frmABCMemo.Show
```

Reviewing the Code for When OK or Cancel Is Clicked

While the form is displayed, the Document_New procedure is suspended. As the user interacts with the form, the event procedures of the form are executed. In the example, the VBA code has been written for only two of these event procedures: the Click events of the OK and Cancel buttons.

After the user dismisses the form by clicking one of the two buttons, VBA continues by executing the next line of code after the frmABCMemo.Show statement. Your code can check the *btnOkClicked* variable to tell which button was clicked.

If *btnOkClicked* is True, the user clicked OK, so the main body of the rest of the procedure should be executed. Listing 3 shows the code used for testing the value of variable *btnOkClicked*.

LISTING 3 IF STATEMENT USED TO TEST VALUE OF btnOkClicked

```
If btnOkClicked = True Then
    ...
    Else
        ...
  End If
```

TESTING WHETHER CONFIDENTIAL IS CHECKED

The first thing that is done after testing whether *btnOkClicked* is True is to perform another If statement. In this statement, you are testing whether there is a check in the chkConfidential check box control. If there is, you are directing the procedure to type CONFIDENTIAL at the top of the document. If not, the procedure moves on to the next statement. Listing 4 shows the If statement used to check the Value property of the chkConfidential control.

LISTING 4 IF STATEMENT USED TO DETERMINE WHETHER CONFIDENTIAL IS CHECKED

```
If frmABCMemo.chkConfidential = True Then
        Selection.HomeKey Unit:=wdStory
        Selection.Font.Bold = True
        Selection.ParagraphFormat.Alignment = wdAlignParagraphCenter
        Selection.TypeText Text:="CONFIDENTIAL"
        Selection.TypeParagraph
        Selection.TypeParagraph
End If
```

Note

Note that we could have used the bookmark object method (described in the next section) instead of a manual insertion method. This would have required that "CONFIDENTIAL" be included in the form, bookmarked, and deleted if the check box was not checked.

USING THE BOOKMARK OBJECTS

Earlier in this chapter, bookmarks were added to the template. At this point in the procedure, the bookmarks are used to place text into the document from the form. For each of the text boxes in the form, you can set the Text property of the Range associated with each bookmark. This has the effect of entering the text from the form at the location of each bookmark. Listing 5 illustrates the code used to insert the text from the text boxes on the form to the bookmarked locations in the document.

LISTING 5 WORKING WITH BOOKMARK OBJECTS

```
Bookmarks("bkTo").Range.Text = frmABCMemo.txtTo
        Bookmarks("bkFrom").Range.Text = frmABCMemo.txtFrom
        Bookmarks("bkDate").Range.Text = frmABCMemo.txtDate
        Bookmarks("bkSubject").Range.Text = frmABCMemo.txtSubject
```

In the case of the cc list, if the user didn't type any text in the txtCC text box, the cc: label should be deleted from the document. The Len() function can be used to determine the length of the text in the txtCC text box. If the length is 0, the Delete method can be called to remove the text. Listing 6 shows the If statement used to determine whether cc: needs to be deleted.

LISTING 6 TESTING THE VALUE OF THE TEXT PROPERTY OF THE TXTCC CONTROL

```
If Len(frmABCMemo.txtCC) > 0 Then
     Bookmarks("bkCC").Range.Text = frmABCMemo.txtCC
Else
     Bookmarks("bkCCLabel").Range.Delete
End If
```

SETTING THE INSERTION POINT AND DOCUMENT DISPLAY OPTIONS

After you have the text inserted into your document, the procedure needs to move the insertion point to the end of the document and set the display to hide table gridlines, bookmarks, and other nonprinting characters. Listing 7 shows the code used to move the insertion point and set the display of the document.

LISTING 7 SETTING THE INSERTION POINT AND DOCUMENT DISPLAY OPTIONS

```
Selection.EndKey wdStory
        Options.DisplayGridLines = False
        With ActiveDocument.ActiveWindow.View
             .ShowBookmarks = False
             .ShowAll = False
      End With
```

REVIEWING THE CODE FOR WHEN CANCEL IS CLICKED

If *btnOkClicked* is False, the user clicked Cancel, so the active document should be closed without saving.

```
ActiveDocument.Close wdDoNotSaveChanges
```

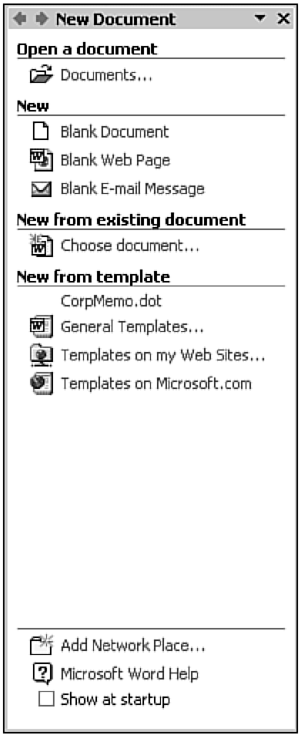
Now that the Document_New procedure has been created, the next time a user creates a new document based on this template, the procedure runs, displaying the custom dialog box and updating the newly created document.

Running the Customized Application

Now that you've created a template, designed an input form, and written the VBA code to automate the process, you're ready to run the application. Because you've place the main portion of code in the template's Document_New procedure, the application runs automatically whenever you create a new document based on the template.

1. Choose File, New. Word displays a variety of ways to create a new document using the New Document side pane, as shown in Figure 10.

Figure 10
Select the template you want to use for your new document from the New from Template section of the New Document side pane (after you select File, New).

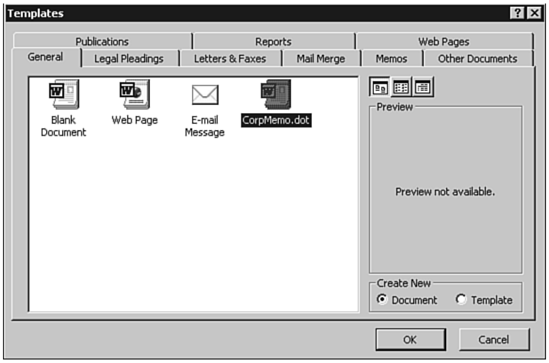


2. If the template you want to use is listed, click on its name. If it is not listed, click on General Templates and select the desired template, as shown in Figure 11.

Word creates a new document based on your selected template and automates the process by running the VBA code that you placed in the Document_New procedure.

Tip from
Bill Ray

Microsoft provides many Word templates on its Web site. When you click the Templates on Microsoft.com link, Word uses your Web browser to connect to the Microsoft Office Template Gallery, which contains templates that you can use and modify. These templates serve many purposes, from corporate uses such as staffing and management, marketing, and business forms, to personal uses, such as rèsumès, educational reports, fundraising, and hobby forms.



Create New
● Document ○ Template

Figure 11
The Templates dialog box shows all the templates available for creating new Word documents.

Creating an HTML Version of a Memo

Because the memos will also be published on the company's intranet, one of Word's formatting themes can be used to provide the HTML formatting. A simple macro in a standard module can automate this procedure.

To create the macro, open the code window of a standard module in the template. Because a standard module was added to the template for the definition of the *btnOk-Clicked* variable, all you have to do is to double-click the module to open the code window.

Enter the macro shown in Listing 8. This macro begins by applying the Corporate theme to the document, changing its appearance for the intranet. The next section displays the Save As dialog box, after initializing the dialog box with a default filename and setting the file type to HTML. The Show method displays the dialog box, so the user can change the filename or select a different location.

After the user dismisses the dialog box, the view is set to Web Layout, so the user can preview and edit the HTML file.

Listing 8 The SaveMemoAsHTML Subprocedure Creates an HTML File Based on the Current Memo

```
Sub SaveMemoAsHTML()

    ActiveDocument.ApplyTheme Name:="corporat 011"
    With Application.Dialogs(wdDialogFileSaveAs)
        .Name = "HTMLMemo"
        .Format = wdFormatHTML
        .Show
    End With
    ActiveWindow.View.Type = wdWebView
End Sub
```

Deploying Your Application

When you create an application like the memo template created in this chapter, you might want to deploy it throughout your organization. Whether your users work on standalone computers or on a network, you have a variety of strategies you can use to organize the installation. Probably the best procedure is to distribute the template, which includes all the forms and code required for the memo.

➔ For information on setting up global and document templates, **see** Chapter 12, "Templates, Wizards, and Add-Ins," **p. 357**.

Creating Custom Toolbars and Menus

There are several ways to execute your VBA macros. The most common is to use the Tools, Macro, Macros command to run a macro. You can also run VBA code automatically in response to events, such as creating a new document or clicking a button on a form. Another way to run a macro is by using a customized menu or toolbar button. Toolbar and menu customizations are stored in templates. If you customize the CorpMemo template, you can use that customization to run the macro whenever you edit an ABC corporate memo.

➔ For more information about customizing templates, **see** Chapter 27, "Customizing Word," **p. 823**.

Troubleshooting

How to Display a Grid on Your Form

Select Tools, Options and go to the General tab. Place a check in the Show Grid box.

What's Wrong with My Variable if I Am Not Getting the Right Value?

You forgot to use the Public reserve word when defining the variable.

Get the Installation Right

Many applications, like the example in this chapter, require your VBA code to open templates, documents, or other files. Make sure you install all the required files in the correct directories as required by your application design, especially when you distribute your application to other users.

Check Your Options

Some macro code works differently on different computers, depending upon certain options the user has selected. For example, the Tools, Options command contains the option Use Smart Cut and Paste on the Edit pane. This option can cause differences in spacing between words if your macro performs cut, copy or paste operations.

To work around this, you can set the state of the option the way you want to run your code, then restore its value when you're done. Here's an example of how to do this:

```
Dim SmartState
SmartState = Options.SmartCutPaste
Options.SmartCutPaste = False
'my code goes here
Options.SmartCutPaste = SmartState
```

I've used the variable SmartState to capture the original setting of the Smart Cut and Paste option, then I've turned the option off, so that my code operates reliably. When I'm done, I reset the state of the option back to its original value, so that the user will get the behavior that he or she expects.

You can apply this principle to many of the feature in the Options command, as well as the AutoCorrect features.

BONUS WORD CHAPTER

Using Word's Online Collaboration Tools

In this chapter

Special Edition Using

que®

UNDERSTANDING THE ROLE OF ONLINE COLLABORATION

Business is about communication. Communication by memo, phone, fax, e-mail, and now, over the Internet. Much of that communication is team oriented. Groups must accomplish their jobs, and they need ways to effectively communicate with each other and often share documents. To add to the communication headaches, team members may be scattered across multiple time zones and continents. Currently, most document sharing and discussion is accomplished via e-mail. But managing e-mail threads, documents, and document versions is a manual juggling act at best.

With Word 2002 you have another means for team communication—online collaboration. *Online collaboration* is the process of using a Team Web Site on a Web server to collect all the documents, discussion, to-do lists, and any other information needed for the successful completion of a team project. All team members can participate in all phases of the project as needed. The Team Web Site provides a central location for the storage and review of critical team documents and a central clearinghouse for communication. And Word provides the collaboration tools.

Note

Even though the documents are on a Web server, they are not accessible by everyone on the Internet or your organizational intranet. Only people with assigned access rights can read the documents and leave comments. These people usually must log on with a user ID and password. Also, different rights can be given to different team members and different areas of the Team Web Site.

So why use online collaboration for team communication and document review? Consider some of these advantages:

- Online collaboration occurs on a Web server. That means anyone, anywhere, anytime can work on project documents or communicate via discussion threads. It is much easier to obtain input from remote offices or employees—no more having to juggle everyone's schedule. Documents and information are available to every team member immediately.
- No more tracing versions. You no longer have to worry whether you have the latest version of a document or were included on the e-mail distribution list for it. The latest version is online in one place. Any subsequent versions are posted online in one place.
- No specialized software is needed to participate in online collaboration. Anyone with a fairly recent browser (version 4 or later of Netscape Navigator or Internet Explorer) can use the Team Web Site. Of course, Internet Explorer 5 or later is recommended.
- You can expand your pool of document reviewers because any relevant team document can now be available globally around the clock. This can mean that more people in more places can review and comment on a document in a shorter time, making for a higher-quality final document—and one that has a chance of going out the door faster.

- This is a cheaper process for reviewing documents. It occurs over the Internet. No more long distance or conference calls. No more faxing or using express delivery services to deliver the draft document or the comments.
- All team communication is now centralized in one place. No more wondering whether you or a fellow team member has the latest project information. As long as everyone agrees to use the Team Web Site as a central repository, every team member is up to date.

Using online collaboration can lead to some potential business benefits, including

- Streamlining and speeding up the team communication and document review process for any given group of people.
- Any group that must routinely collaborate on documents can now do so more quickly and efficiently than the competition, which can translate directly into increased sales or winning more contracts.
- Personnel can spend less time on the document review process and more time in other mission-critical areas.

There is one thing online collaboration cannot handle: real-time editing of a document by several people at once. In other words, a document cannot be directly changed in real-time via online collaboration. Reviewers can only leave comments, which the document author may use in a later revision session.

Communicating with Word: The Importance of Consistency and Perspective
Consistency is an important factor in any document you produce. This is more difficult to maintain when there are several contributing authors. Apart from the smooth flow of information from the beginning to the end of the document, you also need to watch that the writing style does not change from author to author. The important areas to watch include the voice and the perspective.

A document written from more than one perspective—for instance, in third person and first person—indicates that there may be more than one opinion present. When you are writing a business document, you want to convey the opinion held by that business and none other. Choose an appropriate perspective and make sure that all the authors use it consistently. Generally speaking, a first-person active voice has more impact and energy than a third-person passive voice.

UNDERSTANDING OFFICE WEB SERVER

Online collaboration via a Team Web Site is made possible by the installation of OWS (Office Web Server, also called Office Web Server extensions) onto a Microsoft Web server. The Office Web Server extensions are similar in function and installation to FrontPage extensions. Installation of OWS adds the following features on the Web server:

- Web discussions
- Web notification and subscription
- Provisioning of a single Team Web Site

- Tools for administering OWS, including user rights and access
- A framework for using Web components

INSTALLING OWS

OWS is present on the Office XP CD-ROM. Minimum requirements for installation include

- Windows NT 4 Server or Workstation with Service Pack 6 or Windows 2000 Server or Professional
- 200 MHz Pentium processor (133 MHz for NT 4 Workstation)
- 64 MB RAM for NT or 2000 Workstation; 128 MB for NT 4 or 2000 Server; 192 MB recommended for 2000 Server
- 70 MB disk space plus 4 MB for each provisioned Web site

OWS requires Microsoft Internet Information Server (IIS) version 5 or later as the Web server running on Windows NT 4 or Windows 2000. OWS will not install under Windows 95, 98, or Me.

OWS installs a default database, MSDE, a simplified version of SQL Server 7, to store and manage the list data, discussion threads, and other Team Web Site metadata. OWS can also use SQL Server 2000 or SQL Server 7 to store data.

It is beyond the scope of this chapter to detail installation and setup of OWS, IIS, or a Web server. Contact your Information Technology (IT) department to set up OWS for your team or company. As an alternative, consider using an external ISP already running OWS for hosting your Team Web Site. Check the Microsoft Web site for a list of ISP's offering provisioned Team Web Sites.

CREATING A WORD 2002 TEAM SITE

Installing OWS onto a Web server is only the first step in the Team Web Site installation process. By default, during installation, one Web site is automatically provisioned. *Provisioning* is the process of adding Team Web Site functionality to a specific Web site on the IIS server. Provisioning provides the following capability to a Web site:

- A Home Page with links to all the Team Web Site functions
- A default discussion forum
- The means to manage and create any type of list, such as to-do lists
- The means to add, remove, and grant rights to team members
- Document management tools, including uploading
- An engine to search your Team Web Site and documents
- The means to build and manage lists of announcements, events, and links to Team-related Web sites

- The capability for each user to filter and sort his view of any list
- An address book for all relevant team contact information
- A way to create, schedule, and assign tasks

Each of the functions listed previously is available on a provisioned Team Web Site. The functions are accessible to anyone with a browser, and in many cases, from within Word. No special tools, such as a Web page editor, are needed to use these functions. After provisioning, the Team Web Site can be immediately customized for your team needs.

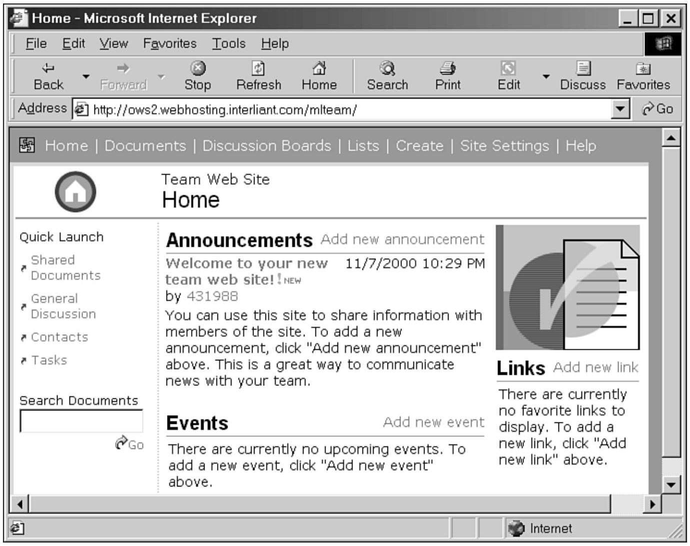
CUSTOMIZING THE TEAM WEB SITE

The provisioning process described in the previous section adds a lot of functionality to the Team Web Site. But it is only functionality with potential: You have to use the tools provided to mold the default Team Web Site into a central location for your team project. After you've done some initial customization, then you can invite your fellow team members to join and start using your Team Web Site to successfully complete your project.

DEFAULT LAYOUT OF A PROVISIONED TEAM WEB SITE

After you've been assigned a Team Web Site, you may need a user ID and password to access the site. You can obtain these from your IT department or ISP. Your Web site will have a standard HTTP address, also obtainable from your IT department or ISP. Type the address into your browser (preferably Internet Explorer 5 or higher), enter your user ID and password if prompted, and press Enter. Your default Team Web Site home page should look something like Figure 1.

Figure 1
A default, provisioned Team Web Site before customization.



Address: http://ows2.webhosting.interliant.com/mlteam/

Team Web Site
Home

Announcements

Welcome to your new team web site! NEW 11/7/2000 10:29 PM
by 431988
You can use this site to share information with members of the site. To add a new announcement, click "Add new announcement" above. This is a great way to communicate news with your team.

Events

There are currently no upcoming events. To add a new event, click "Add new event" above.

Links

There are currently no favorite links to display. To add a new link, click "Add new link" above.

From here you have links to all the tools you need to customize your Team Web Site. So where do you begin to customize?

Planning for Your Team Web Site

Before customizing your Team Web Site, you may want to do some basic planning. The focus of each project is different. Some projects are more document oriented, and your Team Web Site may centralize your team comments and document editing processing. In this case, customizing document management should be one of the first things you do for your Team Web Site. Some projects are more communication oriented: You need a place to collect everyone's thoughts and related documents for summary. So setting up a customized discussion forum would be more important and so on.

When you've decided which sections of your Team Web Site to customize, make some global changes to your Team Web Site settings and then proceed to your area of focus.

Site Settings

To make global changes, click Site Settings at the top of your default home page (refer to Figure 1). You can modify four major areas from here:

- Web Site Settings, including your team site name and home page layout
- Web Administration, including adding users, inviting them, creating a subweb, and site administration
- User Information and its display
- Modify Site Contents, from which you can customize all the content sections or lists on your Team Web Site

Your team site name replaces "Team Web Site" (refer to Figure 1) on the home page, and your site description becomes the top paragraph on your home page. The home page layout lets you drag and drop any list, default or customized, to or from the home page and select whether to display that list in the central column (defaults to Announcements and Events) or right column (defaults to the Links list). Figure 2 shows a customized team site with a site description added, several lists moved around, and the default Tasks list added.

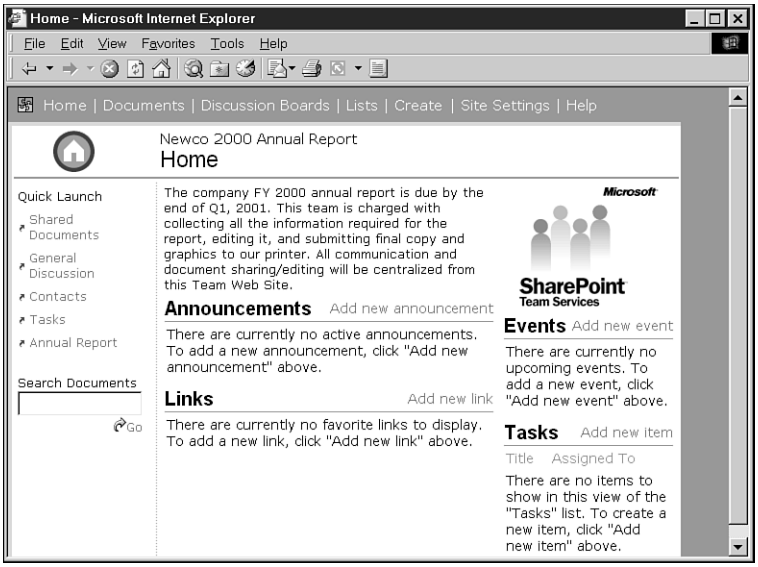
Note

The Web administration module of the Team Web Site is designed with ease-of-use in mind. But if you have a particularly large project to manage with many team members or if you aren't comfortable with the technology, consider giving the administration task to your IT department. For small projects, though, almost anyone can perform the basic administrative tasks.

From the Site Settings page under Web Administration, click Adding Users. In the Add a User page, enter a username and password in the top section. In the bottom section, User Role, you can assign rights, using the categories shown in Figure 3.

For your first administrative task, you need to add your team members as users.

Figure 2
Naming your Team Web Site and customizing the lists that display on your home page.



Newco 2000 Annual Report
Home

The company FY 2000 annual report is due by the end of Q1, 2001. This team is charged with collecting all the information required for the report, editing it, and submitting final copy and graphics to our printer. All communication and document sharing/editing will be centralized from this Team Web Site.

Announcements

There are currently no active announcements. To add a new announcement, click "Add new announcement" above.

Links

There are currently no favorite links to display. To add a new link, click "Add new link" above.

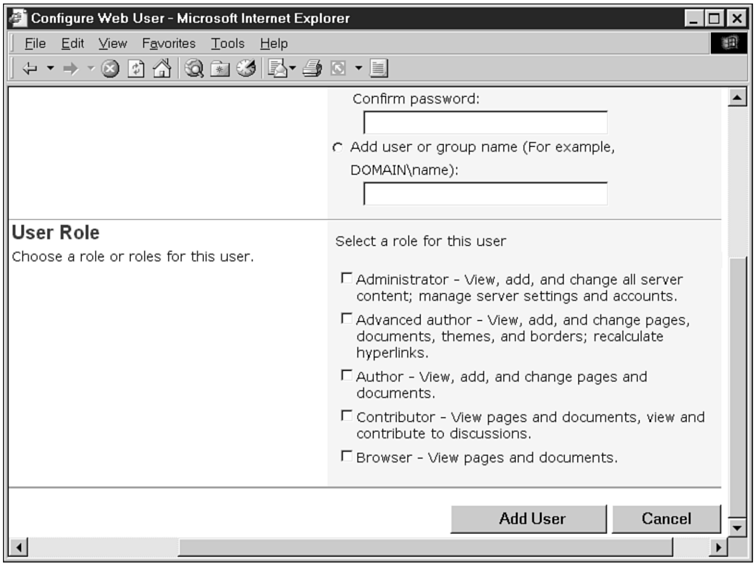
Events

There are currently no upcoming events. To add a new event, click "Add new event" above.

Tasks

There are no items to show in this view of the "Tasks" list. To create a new item, click "Add new item" above.

Figure 3
Assigning rights to each team member.


Configure Web User - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Add User
Cancel

Confirm password:

○ Add user or group name (For example, DOMAIN\name):

User Role
Choose a role or roles for this user.

Select a role for this user

□ Administrator - View, add, and change all server content; manage server settings and accounts.
□ Advanced author - View, add, and change pages, documents, themes, and borders; recalculate hyperlinks.
□ Author - View, add, and change pages and documents.
□ Contributor - View pages and documents, view and contribute to discussions.
□ Browser - View pages and documents.

After you've added one or more users, you can invite them to the Team Web Site. You need each team member's e-mail address, and each member must already be in the Team Web Site list of users for verification. You can add some personalized text to the invitation and tell each team member her rights level.

Another site administration function you may want to consider is creating a subweb. A *subweb* is simply another related Web site branching under your main Team Web Site. A subweb is useful when

- You have a large, multifaceted project.
- You want someone else to share administration duties.
- You want to create a separate workspace for specialists, such as a graphic artists discussion and document area.

To create a subweb, you first need to name it. Use a short, descriptive name, such as Artists. You can also choose a different administrator for the subweb and create the subweb as either another provisioned Team Web Site or a blank Web site. A folder named Artists with the new Web site will be created under your main Team Web Site.

You can also perform further site administration from the Web administration module. Site administration mostly duplicates Web administration and also lists all your subwebs.

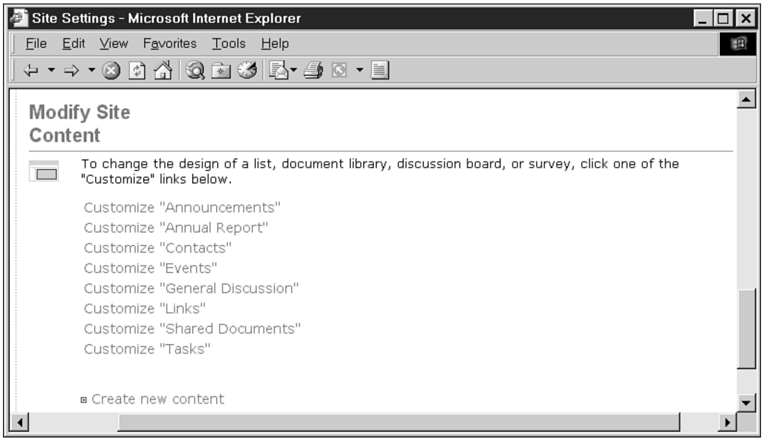
From the third Site Settings module, User Information, you can edit your personal information, such as changing your password or managing your subscription lists.

➔ To find more information about subscriptions, **see** "Holding a Document Discussion," **p. 24**.

You can also list all users, including their full name, username, e-mail address, and any notes entered when the user was created. You cannot edit user information from this screen, although there are links to the editing pages.

Finally in Site Settings, you can modify the content of all sections of the Team Web Site or add new content as shown in Figure 4.

Figure 4
Customizing your Team Web Site from Site Settings.


Site Settings - Microsoft Internet Explorer
File Edit View Favorites Tools Help

Modify Site Content

To change the design of a list, document library, discussion board, or survey, click one of the "Customize" links below.

Customize "Announcements"
Customize "Annual Report"
Customize "Contacts"
Customize "Events"
Customize "General Discussion"
Customize "Links"
Customize "Shared Documents"
Customize "Tasks"

Create new content

When you choose to customize, each list customization page has three common areas:

- General Settings
- Columns
- Views

From General Settings you can change

- The name and description of the list
- Whether to display a link to the list in the left-hand navigation bar on the home page, also called the Quick Launch Bar
- Access to the list, including read access, edit access, and design access

You can also delete the list or Web site section from General Settings. For the Shared Documents list, you can add a URL to a default document template for creating new pages for your Team Web Site, if you want.

Using the Column Settings you can change the name for each column, add default values and default formats, where applicable, and reorder the columns. The content of each column is specific for the list type. For instance, the Events list has default columns for event times and locations. The Contact list has default columns for name, address, phone number, and so on.

The view items you can customize depend on which list you're editing. By default, all items for any given list are displayed. But you can modify any listed view or create your own view. View items you can customize include

- The name of the view
- The Web page address of the view
- Whether this view is the default
- Which columns of information are displayed and their order
- The sort order of the information within each column
- The capability to apply a Boolean logic filter to the view
- A limit on the number of items listed in the view

If you don't want to customize an existing list, you can create new content from scratch from the Site Settings page. New content you can add includes

- A new document library
- Surveys
- Custom discussion boards
- A new list of hyperlinks
- Custom lists

You can also import existing lists from spreadsheets as the basis for a new custom list.

Site Settings give you control over the content and display of the content on your Team Web Site. You can precisely tailor the entire site to meet the needs of your team project.

Adding Lists, Discussions, and Documents to Your Team Web Site

Four key areas of your Team Web Site are detailed in the following sections:

- Custom lists, which give you complete flexibility in displaying a list of any type of information. This includes graphical information that may be important to the team.
- Discussion forums, which gives the team a central online meeting place to discuss critical topics.
- Lists of links, which are a great way to link to reference material important to the team or project.
- Uploading documents to your Team Web Site, which allows you to create a central repository for key team documents.

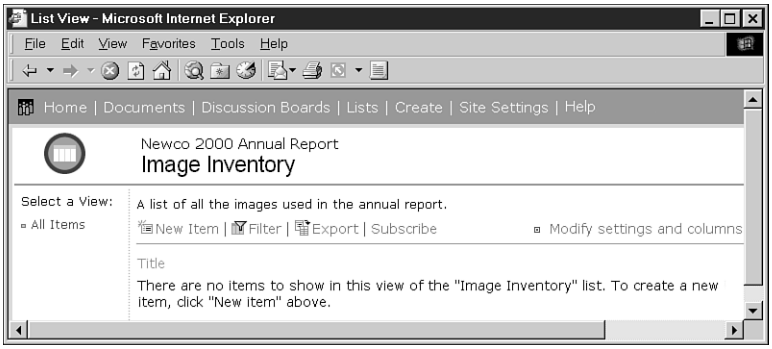
Creating a Custom List

If you need a list and the structure of your list isn't similar to the default lists (Events, Announcements, Tasks, or Contacts) in the Team Web Site, you can create one from scratch:

1. Click Create on the top toolbar.
2. Click the Custom Lists link.
3. In the New List page, name your list. You'll probably want to use a name descriptive of the list purpose, like Image Inventory.
4. Add a basic description for your list.
5. Finally, decide whether to put a link to the list in the Quick Launch navigation bar on the right side of the home page.
6. Click Create.

A blank custom list is created as shown in Figure 5.

Figure 5
The newly created custom list.


List View - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Home | Documents | Discussion Boards | Lists | Create | Site Settings | Help
Select a View:
All Items
New Item | Filter | Export | Subscribe
Modify settings and columns
Title

Newco 2000 Annual Report

Image Inventory

A list of all the images used in the annual report.

There are no items to show in this view of the "Image Inventory" list. To create a new item, click "New item" above.

Next you need to define a structure for your list:

1. Click Modify Settings and Columns.
2. On the Customize List page, scroll to the Columns section. By default, a single-line Title is listed as a column. For this example, this line should be edited.
3. Click Title and fill out the form as shown in Figure 6. Click OK when you are finished.

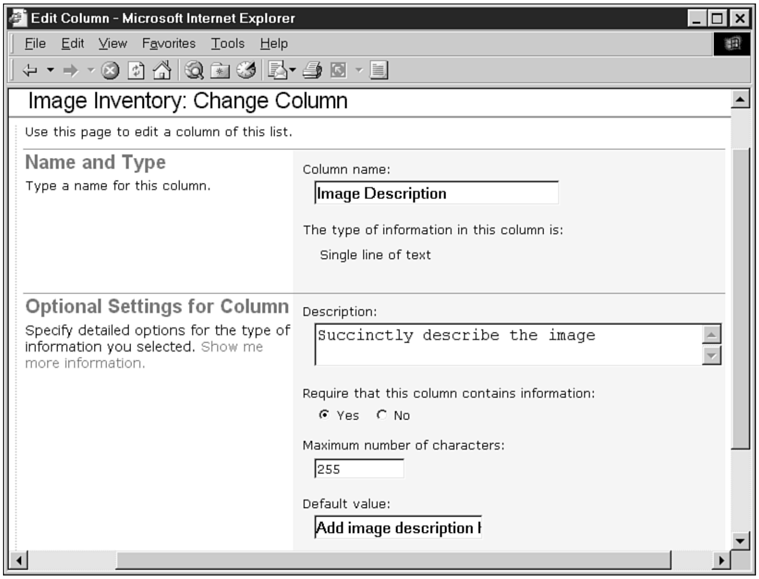

Edit Column - Microsoft Internet Explorer
File Edit View Favorites Tools Help

Image Inventory: Change Column

Use this page to edit a column of this list.

Name and Type

Type a name for this column.

Column name: Image Description

The type of information in this column is: Single line of text

Optional Settings for Column

Specify detailed options for the type of information you selected. Show me more information.

Description: Succinctly describe the image

Require that this column contains information: ● Yes ○ No

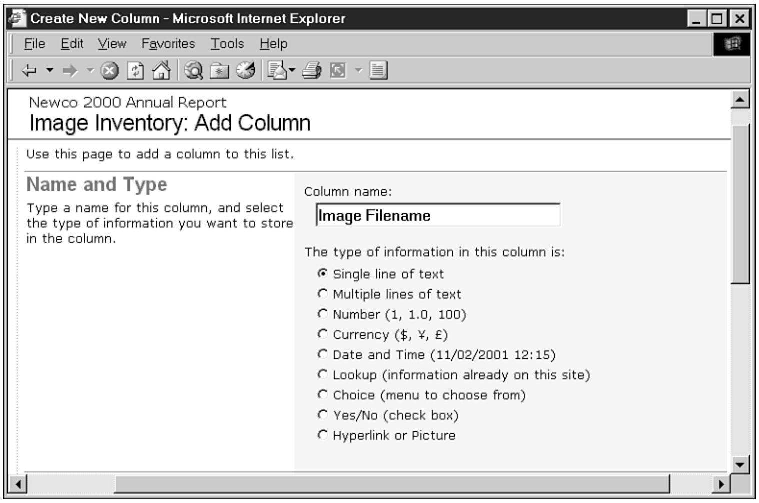
Maximum number of characters: 255

Default value: Add image description I

Figure 6
Modifying the Title column in the customized list.

4. Click Add a New Column.
5. Fill out the Name and Type as shown in Figure 7. Pay close attention to the data type formats available here. Using the correct data type for your column minimizes how much manual formatting you need to do and ensures that the correct data type is entered when you populate your list with data.
6. Under Optional Settings for Column, make sure that the field is required and type `filename.type` into the default field where filename is the name of the file and type is the file's extension. Click OK.
7. Add a third column, Server Path to Image, also a single line of text.
8. Add a final column, Thumbnail of Image, and choose Hyperlink or Picture for the Name and Type. The Options List changes to include a selection for Hyperlink or Picture. Choose Picture and click OK.

One possible column structure might look like Figure 8.


Create New Column - Microsoft Internet Explorer
File Edit View Favorites Tools Help

Newco 2000 Annual Report

Image Inventory: Add Column

Use this page to add a column to this list.

Name and Type

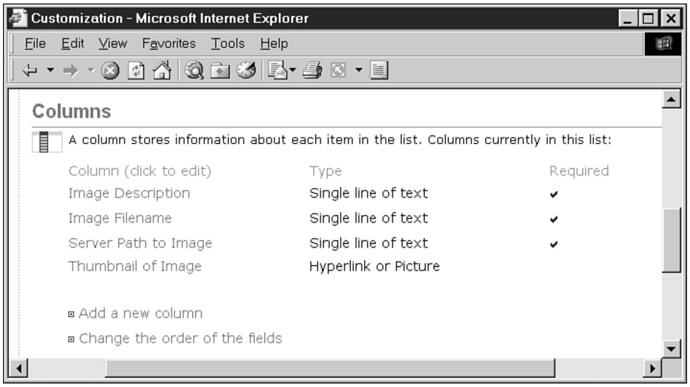
Type a name for this column, and select the type of information you want to store in the column.

Column name: Image Filename

The type of information in this column is:

- ● Single line of text
- ○ Multiple lines of text
- ○ Number (1, 1.0, 100)
- ○ Currency ($, ¥, £)
- ○ Date and Time (11/02/2001 12:15)
- ○ Lookup (information already on this site)
- ○ Choice (menu to choose from)
- ○ Yes/No (check box)
- ○ Hyperlink or Picture

Figure 7
Adding another column to a custom list.


Customization - Microsoft Internet Explorer
File Edit View Favorites Tools Help

Columns

A column stores information about each item in the list. Columns currently in this list:

Column (click to edit)	Type	Required
Image Description	Single line of text	✓
Image Filename	Single line of text	✓
Server Path to Image	Single line of text	✓
Thumbnail of Image	Hyperlink or Picture	

- Add a new column
- Change the order of the fields

Figure 8
A possible completed structure for the custom list.

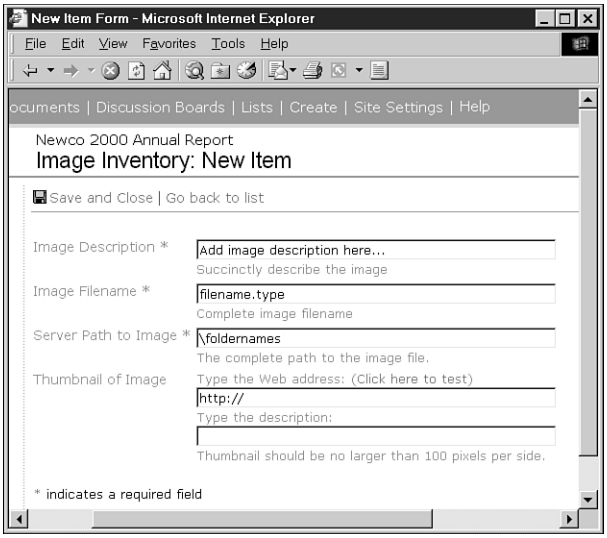
Click the Back button on your browser or the Go Back to return to your custom list and click the link near the top of the page. Next you need to populate the list with information:

1. Click New Item. Data can be entered in the New Item screen (see Figure 9).

Note

Notice that the data field for the thumbnail image starts with a Web address. Use the Web address of your Team Web Site to complete the path to the thumbnail image.

2. Click Save and Close when you're finished to return to the custom list.
3. Repeat steps 1 and 2 until your list is complete.


New Item Form - Microsoft Internet Explorer
File Edit View Favorites Tools Help
ocuments | Discussion Boards | Lists | Create | Site Settings | Help
Save and Close | Go back to list

Newco 2000 Annual Report

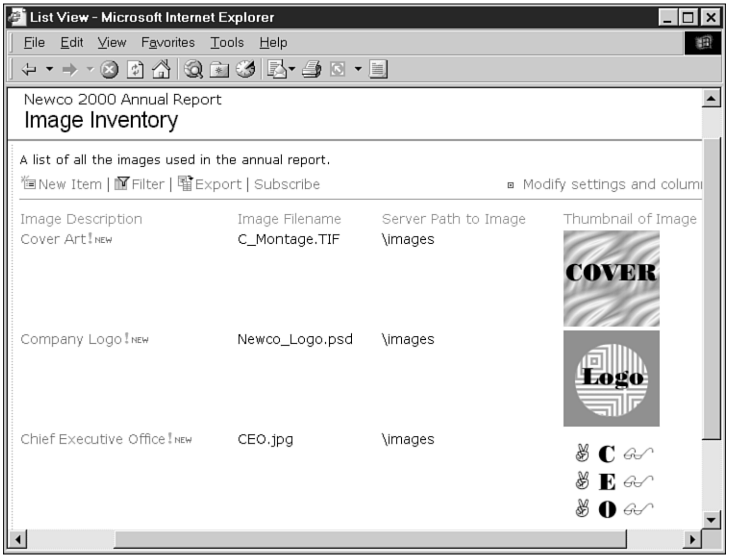
Image Inventory: New Item

Image Description *	Add image description here... Succinctly describe the image
Image Filename *	filename.type Complete image filename
Server Path to Image *	\foldernames The complete path to the image file.
Thumbnail of Image	Type the Web address: (Click here to test) http:// Type the description: Thumbnail should be no larger than 100 pixels per side.

* indicates a required field

Figure 9
Entering new data into the custom list.

A possible custom list with a few sample data items is shown in Figure 10.


List View - Microsoft Internet Explorer
File Edit View Favorites Tools Help
New Item | Filter | Export | Subscribe
Modify settings and colum
COVER
Logo
C
E
O

Newco 2000 Annual Report

Image Inventory

A list of all the images used in the annual report.

Image Description	Image Filename	Server Path to Image	Thumbnail of Image
Cover Art! NEW	C_Montage.TIF	\images	
Company Logo! NEW	Newco_Logo.psd	\images	
Chief Executive Office! NEW	CEO.jpg	\images	

Figure 10
A sample custom list.

If the data for your list is already contained in a spreadsheet, you can easily import that information into your list. This saves manually retyping data or having to define your list structure, column by column.

1. Click Create at the top of the Team Web Site home page.
2. Click Import Spreadsheet.

3. In the New List screen, type a name (required field) for the list and a description.
4. Browse to the location of the spreadsheet you want to import.
5. Decide whether to add a link to your new list on the Quick Launch menu.
6. Click Import.
7. A dialog box requests the range to import. Click the range (A1:D5, for instance) of cells to import and click Import.

The contents of the spreadsheet are imported, and a new list is automatically created in your project and populated with the data from your spreadsheet.

Using the Discussion forums

Team members can communicate with each other using two discussion features on the Team Web Site: a discussion board or forum and document discussion.

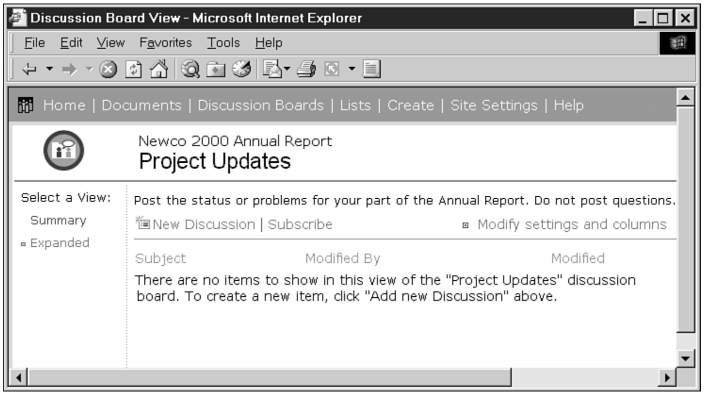
➔ To find more information about document discussion using Word 2002, **see** "Holding a Document Discussion," **p. 24**.

When a Team Web Site is provisioned, a default forum, General Discussions, is created. Forums on the Team Web Site are like forums typically found elsewhere on the Internet or in newsgroups. Any team member can post a comment or question for others to read, or respond to existing comments, forming a discussion thread.

You can set up more than one discussion forum on your Team Web Site. To create a new forum

1. Click Discussions from the top navigation bar on the Team Web Site home page.
2. Click New Discussion Board.
3. Give the forum a name and type in a description.
4. Decide whether to include your new forum in the Quick Launch navigation bar and click Create.

Figure 11 shows a sample new discussion forum, Project Updates.


Discussion Board View - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Home | Documents | Discussion Boards | Lists | Create | Site Settings | Help
Select a View:
Summary
Expanded
New Discussion | Subscribe
Modify settings and columns
Subject
Modified By
Modified

Newco 2000 Annual Report

Project Updates

Post the status or problems for your part of the Annual Report. Do not post questions.

There are no items to show in this view of the "Project Updates" discussion board. To create a new item, click "Add new Discussion" above.

Figure 11
A sample Project Updates custom discussion forum.

To add new postings, click New Discussion.

You can further customize (assuming you have the proper rights) your new forum by clicking Modify Settings and Columns. Items you can customize from here include

- Forum name
- Forum description
- Columns, including adding, deleting, rearranging, and renaming (The two default columns are Subject and Text.)
- Views

You can set the view to an expanded view or to Summary. The Summary view excludes the Text column.

You can also subscribe to this forum, which means you'll be notified by e-mail every time this forum changes. More on subscriptions can be found later in this chapter in the section "Understanding Subscriptions."

ADDING LINKS

You can build a list of hyperlinks on your Team Web Site. These can be links to external Web sites or internal portions of your Team Web Site.

Note

You can find the complete Web address to a specific section of your Team Web Site by modifying the settings for that section. The Web address is listed in the General Settings section.

To begin creating links, click Links from the Team Web Site home page. As with discussion forums or lists, you can customize many aspects of your links page.

Tip

You don't have to limit your list of hyperlinks to strictly text links. You can also use pictures as clickable hyperlinks. Pictures combined with the comments column of the default Links setup may make a much more interesting list of links. If you decide to use pictures, remember to keep them small if any of your team members access your Team Web Site with a modem.

Add as many text or image hyperlinks as you need for your project.

ADDING DOCUMENTS TO YOUR DOCUMENT LIBRARY

Almost every team project involves documents for reference or review. You can upload documents to the Team Web Site

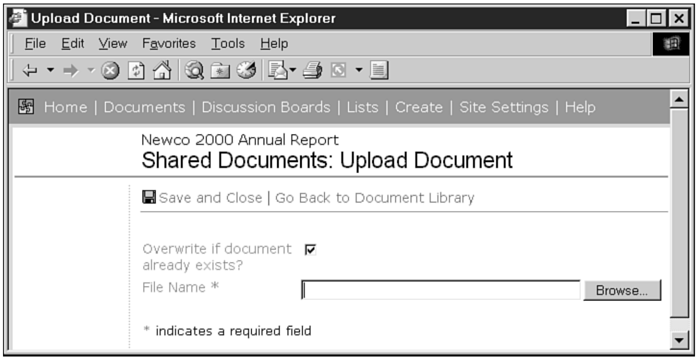
- From a browser
- From a Web folder from within Word using File, Save As
- From within Word using the Document Library Web component

By default, the Team Web Site has one document library in a folder named Shared Documents, but any number of document libraries can be created and customized using the same techniques from lists, discussion forums, and links.

To upload a document to a document library from within a browser

1. Click on Documents at the top of the Team Web Site home page.
2. Click on the document library where you want to place your document.
3. Click Upload Document.
4. Click Browse and browse to your document (see Figure 12).

Figure 12
Uploading a document to a Team Web Site document library.


Upload Document - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Home | Documents | Discussion Boards | Lists | Create | Site Settings | Help
Save and Close | Go Back to Document Library
Browse...

Newco 2000 Annual Report
Shared Documents: Upload Document

Overwrite if document already exists? ☑
File Name *

* indicates a required field

5. Click Save and Exit.

Your document is uploaded, and you are returned to the main page of your document library.

You can also upload documents using a Web folder that leads to your Team Web Site. Note that you may be prompted for a username and password before your document is uploaded.

➔ To find more information about Web folders, **see** the Bonus Word Chapter entitled "Adding Interactivity with Hyperlinks, Web Forms, and Smart Tags."

➔ To find more information about using the Document Library Web component in Word 2002, **see** "Managing Team Web Site Documents with the Document Library View Web Component," **p. 21**.

USING WORD TO MODIFY THE TEAM WEB SITE

So far we've concentrated on using a browser to add to or modify the contents of a Team Web Site. Word gives you even more tools to manage your Team Web Site documents and lists by letting you add Web components to your Team Web Site Web pages.

WEB COMPONENTS

A Web page is typically composed of textual information and other elements including images and hyperlinks. Because a Web page lives on a Web server, however, it is capable of containing other, more dynamic elements. These elements add more functionality or convenience of use to the individual Web page or even the entire Web site. Microsoft has named these elements *Web components*. Each Web component generally has a single, specialized function that otherwise cannot easily be added to the page. These Web components can be inserted into Web pages built in Word. The two Web components available in Word for use with Team Web Sites are:

- List View, which lets you easily generate and manage lists of items.
- Document Library View, which manipulates and sorts all the documents in use.

More Web components are available on the Web and can easily be downloaded for use in the Web pages of your Team Web Site (go to `http://officeupdate.Microsoft.com`).

Note

Web components have similar functionality to some components that FrontPage can insert, including Pivot Table, Chart, and Hover Button.

WEB PAGE COMPONENT COMPATIBILITY

Web components require support from the Web server to function. In other words, a Web component on a Web page is useless and will not function unless the Web server has the proper extensions installed. These extensions are called Office Web Server (OWS). OWS extensions, also called the next generation FrontPage extensions, at this time run only on the Internet Information Server (IIS), Microsoft's Web server.

Note

In the past, Microsoft has modified FrontPage extensions to work with other Web servers. If you or your business uses a different Web server, check with Microsoft to see whether OWS is available for other Web servers. Another alternative is to purchase Web hosting services with companies already running OWS on IIS.

Also note that the Web components, List View and Document Library View, only function if the team collaboration components of OWS are activated on the Web server.

MANAGING LISTS WITH THE LIST VIEW WEB COMPONENT

Word provides the List View Web component to help you manage the lists on a Team Web Site. Lists can be used for scheduling, to-do items, tasks, lists of goals, or any other information the team needs. The List View Web component lets you build, modify, filter, and sort multiple lists as you build a list for a Web page for your Team Web Site.

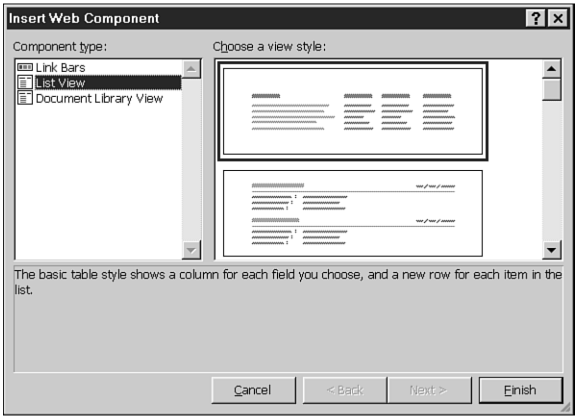
To add the List View Web component to your Web page, do the following:

1. Open the Web page you want to place the List View component into.
2. Immediately save your Web page to a Web server with OWS installed and active.

Note

You cannot build and save your page on your local hard drive or network server while you're working with Web components and upload it or save it to the Team Web Site later. The Web component is not active and editable except from the Web server.

3. From the menu, choose Insert, Web Component.
4. From the Insert Web Component dialog box (see Figure 13), click on List View in the left-hand window.



The basic table style shows a column for each field you choose, and a new row for each item in the list.

Figure 13
Placing the List View Web Component into your Web page.

5. The right-hand window displays a list of the view styles available. A *view style* is the format used to view your list items. Clicking on a view style brings up some descriptive text in the lower window. Scroll through the list and choose the view style most suitable for your list. Click Finish.

The List View Web component is inserted into your Web page and immediately requests customization information.

A listing of generic list types pops up first (see Figure 14). Choose the list type closest to the list information you want to manage and click OK.

From the List View Properties box that appears, you can further customize your list by clicking on any of the following buttons:

- List: This box lists your current customizations next to each button. Clicking the Lists button takes you back to the listing of generic list types.

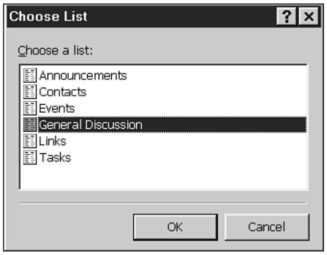
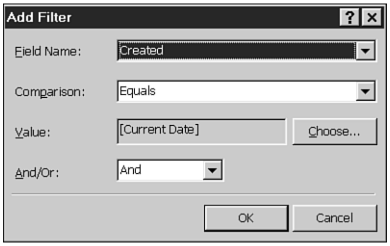


Figure 14
Choose the type of list to add.

- Fields: You can choose which list fields to include in your list from the Fields button and reorder them. Click OK to return to the List View Properties box.
- Sort: Clicking the Sort button from the List View Properties box lets you further sort list fields. Clicking the Change Sort button allows you to sort in ascending or descending order as reflected by the yellow arrow next to the field name. Click OK to return to the List View Properties box.
- Filter: You can apply various filters, including Boolean, to your lists through the Filter button. Click Add to begin customizing your filter (see Figure 15).



Field Name: Created
Comparison: Equals
Value: [Current Date]
And/Or: And

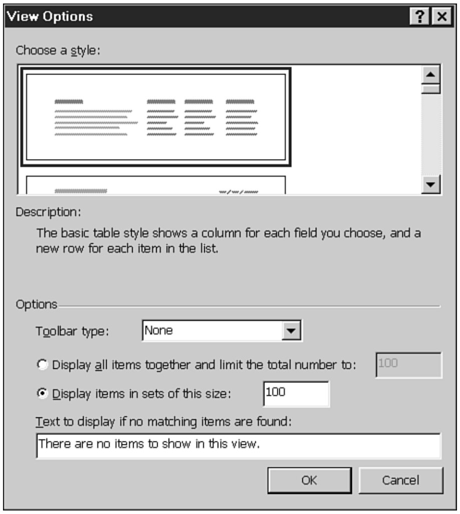
Figure 15
Choosing your filtering criteria.

Choose a field from the Field Names list and an operand from the Comparison list. The Value for comparison varies depending on which field is selected. For instance, the Created field lets you choose a date for the value. The ID field lets you enter free text into Value.

Finally, choose whether to compare this filter with the next filter in the Filter Criteria list using And or Or. Click OK when you're finished to add the criteria to your Filter list. To add more filters to your Filter criteria list, continue to click Add. Note that you can also remove and modify existing criteria as you build your Filter list. Click OK when you've added all the criteria you want.

- Options: The last customization option for List View Properties is the Options button. Clicking this brings up the View Options dialog box as shown in Figure 16.

Figure 16
Changing more List View options.



View Options

Choose a style:

Description:

The basic table style shows a column for each field you choose, and a new row for each item in the list.

Options

Toolbar type: None

○ Display all items together and limit the total number to: 100

● Display items in sets of this size: 100

Text to display if no matching items are found: There are no items to show in this view.

From here, you can again change the list style by scrolling through the visual list of styles. Below this you can select a Summary or Full toolbar to display with the List View or select None to leave the toolbar out. The toolbars show up on your Web page as text links. The Full toolbar is composed of the following links:

- **New Discussion**—You can add open a new thread on a Team Web Site discussion board.
- **Filter**—Reapplies the filter after changes.
- **Subscribe**—Monitor the list for changes.
- **Modify Settings and Columns**—Lets you customize the list further on your Team Web Site.

The Summary toolbar contains only two items: General Discussion (takes you to the Web site discussion board) and Add New Discussions (open a new thread).

Note

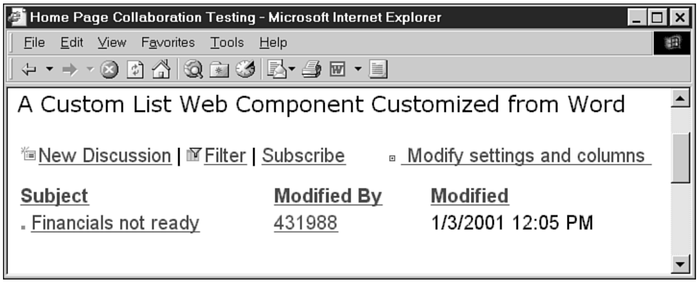
The contents of the Summary or Expanded toolbar change depending on the list type.

Finally, you can decide how many records to display in the list at one time and what message to display if no list items make it through your filters. Click OK to return to the List View Properties dialog box.

After customizing the List View, click OK to insert your customized Web component into your Web page. Save your Web page.

When you open your Web page in a browser, the output for the List View may look like Figure 17, shown with Full toolbars and a new discussion thread.

Figure 17
The List View Web Component with Full toolbars viewed with a Web browser.


Home Page Collaboration Testing - Microsoft Internet Explorer
File Edit View Favorites Tools Help

A Custom List Web Component Customized from Word

New Discussion | Filter | Subscribe   Modify settings and columns

Subject	Modified By	Modified
Financials not ready	431988	1/3/2001 12:05 PM

MANAGING TEAM WEB SITE DOCUMENTS WITH THE DOCUMENT LIBRARY VIEW WEB COMPONENT

Documents are often at the center of team collaboration. You may have research documents that require reading, a draft report that needs reviewing and comments, or multiple documents that have to be summarized. You need tools to keep a handle on the ever-growing list of documents. One of those tools is the Document Library View Web component. As with the List View component, the Document Library View component can be embedded into your Web page and customized.

The Document Library View Web component gives you versatility in managing documents. Some of its functionality includes

- Filtering your document set
- Sorting your document set
- Deleting and adding documents
- Listing all your documents

To add the Document Library View Web component to your Web page, perform the following steps:

1. Open the Web page you want to place the Document Library View component into.
2. Immediately save your Web page to a Web server with OWS installed and active.
3. From the menu, choose Insert, Web Component.
4. From the Insert Web Component dialog box, click on Document Library View in the left-hand window.
5. The right-hand window displays a list of the view styles available. Scroll through the list and choose the view style most suitable for your needs. Click Finish.

The Document Library View Web component is inserted into your Web page and immediately requests customization information:

1. A list of document libraries (see Figure 18) pops up. The initial default is Shared Documents. Click OK.

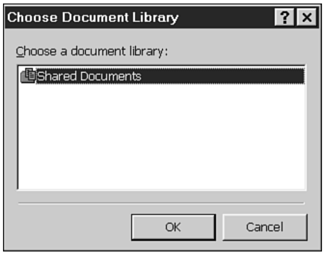
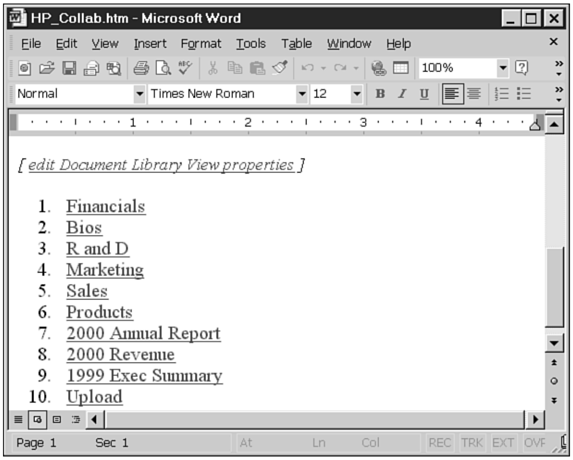


Figure 18
The list of Document Libraries available through the Document Library View Web component.

2. From the List View dialog box you can sort, filter, and change other options such as list style and toolbar style types.
3. Click OK to apply your customizations to the Document Library View Web component.

The Document Library View Web component now lists all documents contained in the Shared Documents library. You can modify the view anytime you open the Web page in Word. Figure 19 shows a sample Document Library View Web component in Word.


HP_Collab.htm - Microsoft Word
File Edit View Insert Format Tools Table Window Help
100%
Normal
Times New Roman
12
Page 1 Sec 1 At Ln Col REC TRK EXT OVR

[edit Document Library View properties]

1. Financials
2. Bios
3. R and D
4. Marketing
5. Sales
6. Products
7. 2000 Annual Report
8. 2000 Revenue
9. 1999 Exec Summary
10. Upload

Figure 19
The Document Library View Web component in a Web page viewed from within Word.

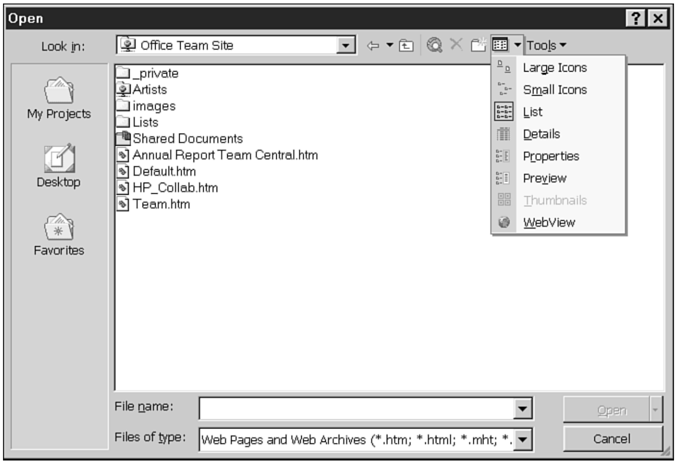
Note

Notice the Edit Document Library hyperlink in Figure 19. This hyperlink only displays from within Word and is the means for changing the Web component properties. This hyperlink does not display when the page is viewed in a browser from your Team Web Site.

USING WEB VIEW WHEN SAVING DOCUMENTS TO THE TEAM WEB SITE

You can save documents from Word in any format directly to your Team Web Site. You can also open documents from your Team Web Site in Word. Also, if you are not sure about the filename of a Web page you want to open but you remember what it looks like, use a Web View. A *Web view* works like a page preview: It displays a portion of the document content in a window as you select a filename. To use the Web view

1. From Word, choose File, Open and browse to your Team Web Site. You may need to enter authentication information.
2. Change the Files of Type list box at the bottom of the Open dialog box to Web Pages and Web Archives or All Word Documents.
3. Click on the View icon in the Open dialog box toolbar, as shown in Figure 20.



Look in: Office Team Site

Files of type: Web Pages and Web Archives (*.htm; *.html; *.mht; *.

Figure 20
Activating Web view from the File Open dialog box toolbar.

4. Choose Web View.
5. Click on a Web page filename, for instance, Default.htm. A reduced view of the Web page appears in the right-hand frame, as shown in Figure 21.

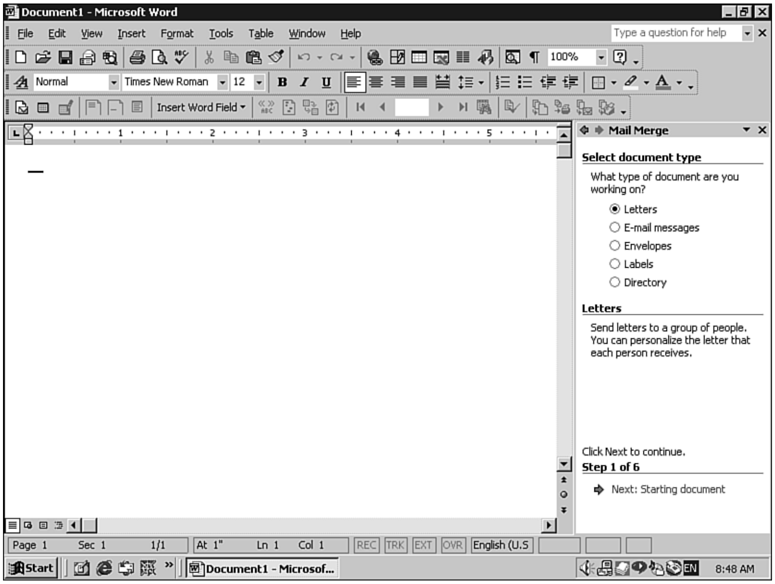
The Web view only displays the text and graphics on a Web page. Web bots (code snippets inserted by FrontPage) or Web components do not display in the Web View window. You can scroll up and down the Web View window to view other parts of the Web page.

Note

Sometimes you may have trouble activating the Web view. An alternative method to display the Web view is to select a Web page filename and repeatedly click the View button. This cycles through all the views, including Web view.

Finding Components on the Web

More components can be added from the Microsoft site and other third-party sites such as MSNBC or MSN. To locate more Web components, choose Tools, Tools on the Web from the menu and follow the directions from the Web page.


Document1 - Microsoft Word
File Edit View Insert Format Tools Table Window Help
Type a question for help
100%
Normal
Times New Roman
12
Insert Word Field
Mail Merge
Page 1 Sec 1 1/1 At 1" Ln 1 Col 1 REC TRK EXT OVR English (U.S
Start
Document1 - Microsof...
8:48 AM

Select document type

What type of document are you working on?

- ● Letters
- ○ E-mail messages
- ○ Envelopes
- ○ Labels
- ○ Directory

Letters

Send letters to a group of people. You can personalize the letter that each person receives.

Click Next to continue.

Step 1 of 6

Next: Starting document

Figure 21
The Web view of a Web page from a Team Web Site.

HOLDING A DOCUMENT DISCUSSION

You can hold two types of discussions from your Team Web Site:

- An exchange of comments in a discussion thread on a discussion forum
- A document discussion

A *discussion forum* is run as a specialized list on the Team Web Site and is oriented around a particular topic. Comments and responses are posted on the discussion forum for all to read. A *document discussion*, also called a Web discussion, centers on a particular document and requires a specialized set of tools. Comments and responses are saved in a special database on the Office Web Server (OWS) and associated with that particular document. Any document on or off the Team Web Site can be discussed as long as all parties have access to it and have signed in to the Discussion database. Also the Document Discussion tools are available from Word 2002, Word 2000, and Internet Explorer 4 or later.

Another key part of document discussions is the *subscription* capability. When you subscribe to a document discussion, you are automatically notified by e-mail whenever a change is made to the document or document discussion comments.

GETTING SET UP FOR A DOCUMENT DISCUSSION

A document discussion requires the support infrastructure of a Web server with OWS extensions installed.

➔ To find more information on installing OWS or outsourcing to a Web hosting company, **see** the section "Understanding Office Web Server" **p. 3**.

Next, everyone who is to participate in document discussions must have user accounts set up on the OWS. Then the document or documents to be discussed can be uploaded to the Team Web Site on the OWS.

Note

The documents being discussed don't have to be uploaded to the OWS. But it does make document management and control easier. Documents being discussed only have to be accessible to all interested parties.

Finally, everyone has to log in to the discussion server to save his comments. To log in to the discussion server from Word, do the following:

1. Choose Tools, Online Collaboration, Web Discussions from the Word menu. This opens the Discussion toolbar at the bottom of the Word window, as shown in Figure 22.

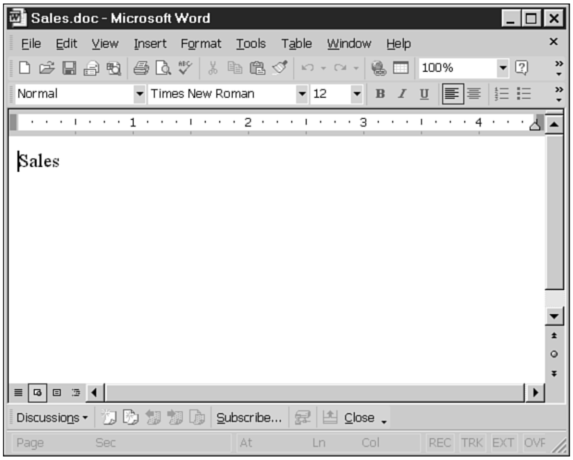
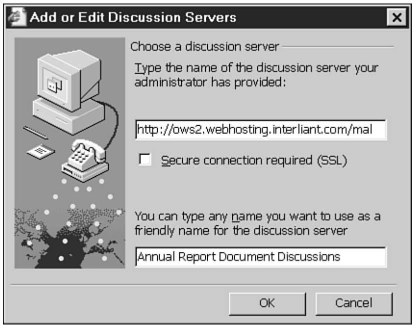

Sales.doc - Microsoft Word
File Edit View Insert Format Tools Table Window Help
100%
Normal
Times New Roman
12
B I U
Sales
Discussions
Subscribe...
Close
Page Sec At Ln Col REC TRK EXT OVR

Figure 22
The Web Discussion toolbar in Word 2002.

Note

The Web Discussion toolbar cannot be opened and closed like other toolbars and is not listed in the toolbar list in Word. It can only be opened from the menu.

2. Click Discussions, Discussion Options in the Web Discussion toolbar.
3. Fill out the location of your Team Web Site on the OWS in the Add or Edit Discussion Servers screen, as shown in Figure 23. Click OK.


Add or Edit Discussion Servers

Choose a discussion server

Type the name of the discussion server your administrator has provided:

http://ows2.webhosting.interliant.com/mal

□ Secure connection required (SSL)

You can type any name you want to use as a friendly name for the discussion server

Annual Report Document Discussions

Figure 23
Selecting a Discussion database for your document discussion comments.

That's it. You can load the document you want to discuss and begin reading or inserting comments.

Tip from
Woody Leonard

When creating a document in collaboration with others, you have opportunities to make the document more effective. Take advantage of all the eyes looking at the document and have all the contributors proofread each other's portions. The proofreading should be more than grammatical and spell checking.

Because the contributors have their own specialties, each will have strength in her own sections but may be unfamiliar with the other contributors' portions. Use this to your advantage by having them make sure that everyone explains herself thoroughly.

DISCUSSING A DOCUMENT

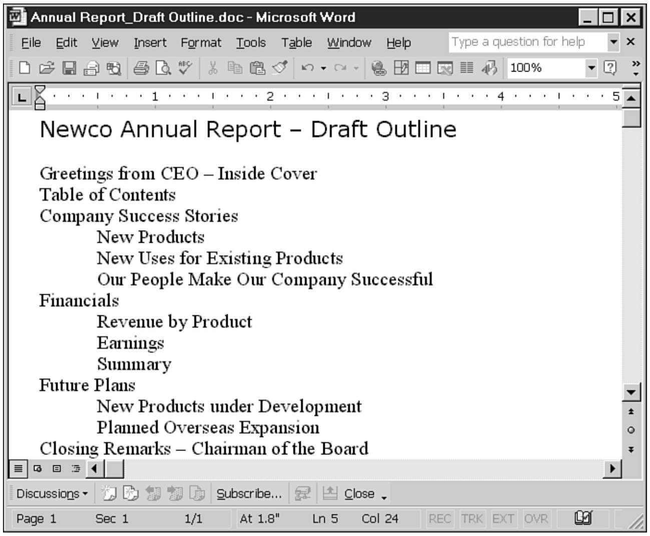
Once you've uploaded a document to discuss to your Team Web Site and invited all concerned parties, you can open the document in Word and begin the discussion. Figure 24 shows a sample document with the Web Discussion toolbar open.

Comments fall into two categories: comments about the entire document and comments about a specific part in the document.

To add a comment about the entire document

1. From the Web Discussion toolbar, click Insert Discussion About the Document.
2. Add a title and the text of the comment as shown in Figure 25. Click OK.

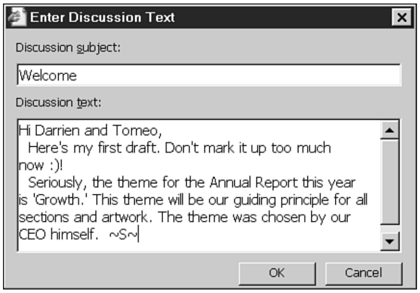
Figure 24
A sample team discussion document with the Web Discussion toolbar open at the bottom.


Annual Report_Draft Outline.doc - Microsoft Word
File Edit View Insert Format Tools Table Window Help
Type a question for help
100%
Discussions
Subscribe...
Close
Page 1 Sec 1 1/1 At 1.8" Ln 5 Col 24 REC TRK EXT OVR

Newco Annual Report – Draft Outline

Greetings from CEO – Inside Cover
Table of Contents
Company Success Stories
New Products
New Uses for Existing Products
Our People Make Our Company Successful
Financials
Revenue by Product
Earnings
Summary
Future Plans
New Products under Development
Planned Overseas Expansion
Closing Remarks – Chairman of the Board

Figure 25
Adding a comment about a document as part of a document discussion.



Discussion subject: Welcome

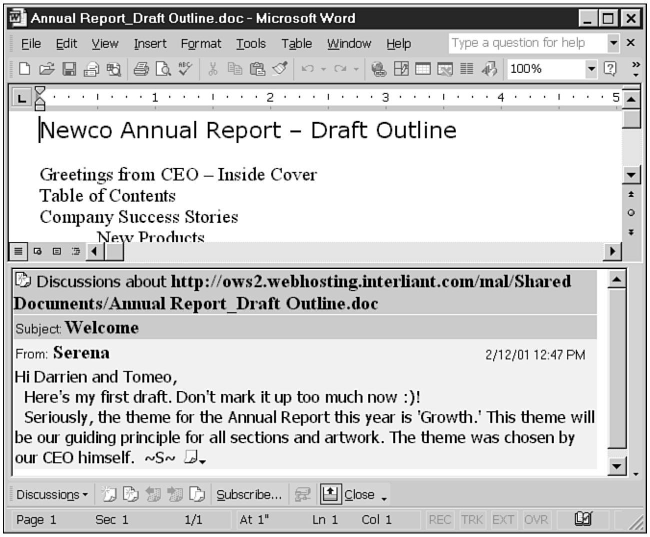
Discussion text:
Hi Darrien and Tomeo,
Here's my first draft. Don't mark it up too much now :)!
Seriously, the theme for the Annual Report this year is 'Growth.' This theme will be our guiding principle for all sections and artwork. The theme was chosen by our CEO himself. ~S~

This comment is recorded in the database and displayed in the discussion window below the document (see Figure 26).

Notice two yellow icons embedded in the comment. The one in the upper left-hand corner is the Insert Discussion About the Document icon, telling us what type of discussion comment this is. Clicking on this icon brings up the Enter Discussion Text dialog box (refer to Figure 25).

From the other icon appended to the end of text, you can

- Reply to the comment
- Edit the comment (original author only)
- Delete the comment (original author only)
- Close the discussion window
- Close item and replies, which also closes the discussion window


Annual Report_Draft Outline.doc - Microsoft Word
File Edit View Insert Format Tools Table Window Help
Type a question for help
100%
Discussions
Subscribe...
Close
Page 1 Sec 1 1/1 At 1" Ln 1 Col 1 REC TRK EXT OVR

Newco Annual Report – Draft Outline

Greetings from CEO – Inside Cover
Table of Contents
Company Success Stories
New Products

Discussions about **http://ows2.webhosting.interliant.com/mal/Shared Documents/Annual Report_Draft Outline.doc**

Subject: **Welcome**

From: **Serena** 2/12/01 12:47 PM

Hi Darrien and Tomeo,
Here's my first draft. Don't mark it up too much now :)!
Seriously, the theme for the Annual Report this year is 'Growth.' This theme will be our guiding principle for all sections and artwork. The theme was chosen by our CEO himself. ~S~

Figure 26
An opening welcome comment for the document discussion.

You can also comment on a specific part of the document. To do so just click on the spot in the document where you want to add a comment and select Insert Discussion in the Document from the Web Discussion toolbar. Type a title for your comment and click OK. This adds a yellow comment icon to the document.

Note

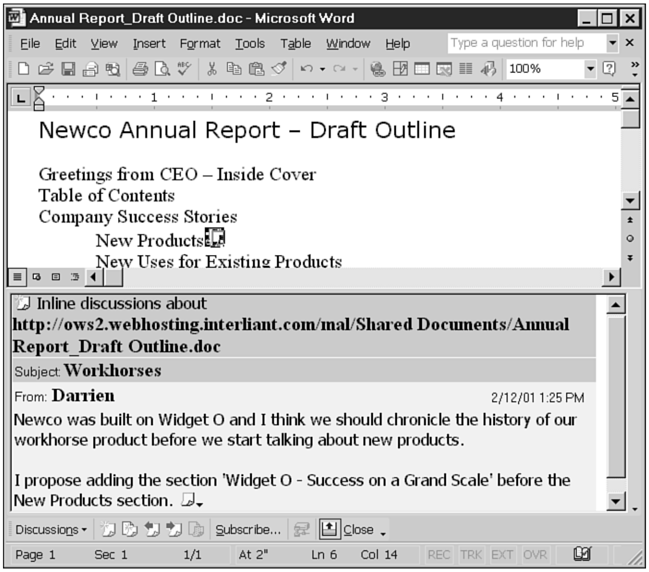
The added icons (except for inline comments) and their related text are not visible unless the Web Discussion toolbar is open in Word while the document is being viewed and you are logged in to the discussion server.

Whenever someone other than the author re-opens the document, they will see the inline comment as shown in Figure 27.

Again the icons in the discussion window let you edit, reply, delete, or close the discussion window.

You can respond to the inline comment by clicking Reply. The title for the Insert Discussion Text dialog box is already filled in with the topic of that discussion. In this example it's "Re: Workhorses." Type your response and click OK. Your response is added to the discussion thread as shown in Figure 28.

Figure 27
An inline comment requesting a specific change at a specific location in the document.


Annual Report_Draft Outline.doc - Microsoft Word
File Edit View Insert Format Tools Table Window Help
Type a question for help
100%
Discussions ▾ Subscribe... Close
Page 1 Sec 1 1/1 At 2" Ln 6 Col 14 REC TRK EXT OVR

Newco Annual Report – Draft Outline

Greetings from CEO – Inside Cover
Table of Contents
Company Success Stories
New Products
New Uses for Existing Products

Inline discussions about **http://ows2.webhosting.interliant.com/mal/Shared Documents/Annual Report_Draft Outline.doc**

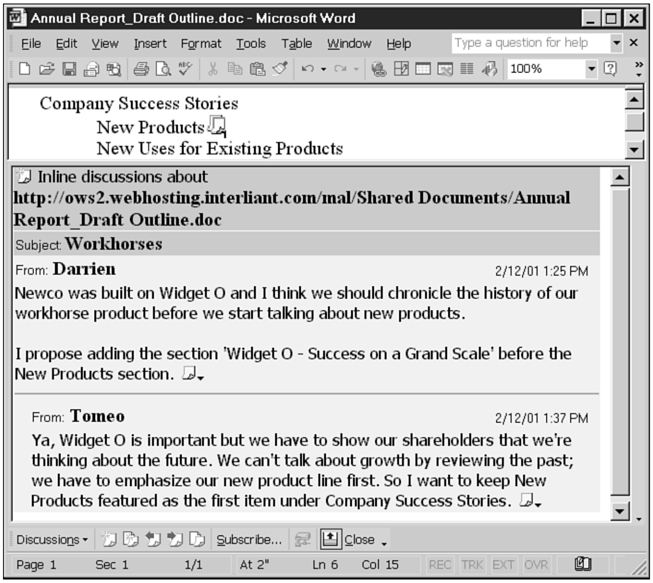
Subject: **Workhorses**

From: **Darrien** 2/12/01 1:25 PM

Newco was built on Widget O and I think we should chronicle the history of our workhorse product before we start talking about new products.

I propose adding the section 'Widget O - Success on a Grand Scale' before the New Products section.

Figure 28
A response to an inline discussion comment.


Annual Report_Draft Outline.doc - Microsoft Word
File Edit View Insert Format Tools Table Window Help
Type a question for help
100%
Discussions ▾ Subscribe... Close
Page 1 Sec 1 1/1 At 2" Ln 6 Col 15 REC TRK EXT OVR

Company Success Stories
New Products
New Uses for Existing Products

Inline discussions about **http://ows2.webhosting.interliant.com/mal/Shared Documents/Annual Report_Draft Outline.doc**

Subject: **Workhorses**

From: **Darrien** 2/12/01 1:25 PM

Newco was built on Widget O and I think we should chronicle the history of our workhorse product before we start talking about new products.

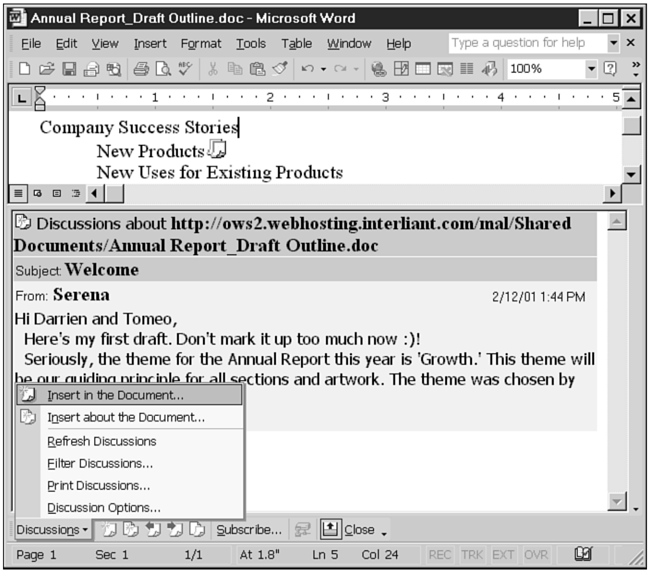
I propose adding the section 'Widget O - Success on a Grand Scale' before the New Products section.

From: **Tomeo** 2/12/01 1:37 PM

Ya, Widget O is important but we have to show our shareholders that we're thinking about the future. We can't talk about growth by reviewing the past; we have to emphasize our new product line first. So I want to keep New Products featured as the first item under Company Success Stories.

Also notice that two more yellow buttons with arrows are now active in the Web Discussion toolbar, Previous and Next. When more than one comment has been added, you can use these to jump from comment to comment in the document. Also the General Discussions button becomes active so that you can switch the contents in your discussion window between general comments about the documents to specific inline comments.

Also during the course of a discussion, any author can click Discussions on the Web Discussion toolbar for more options (see Figure 29).



Annual Report_Draft Outline.doc - Microsoft Word

Company Success Stories
New Products
New Uses for Existing Products

Discussions about **http://ows2.webhosting.interliant.com/mal/Shared Documents/Annual Report_Draft Outline.doc**

Subject: **Welcome**

From: **Serena** 2/12/01 1:44 PM

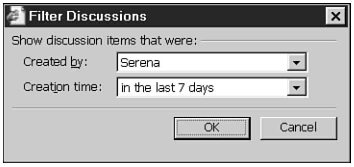
Hi Darrien and Tomeo,
Here's my first draft. Don't mark it up too much now :)!
Seriously, the theme for the Annual Report this year is 'Growth.' This theme will be our guiding principle for all sections and artwork. The theme was chosen by

File Edit View Insert Format Tools Table Window Help
Type a question for help
100%
Insert in the Document...
Insert about the Document...
Refresh Discussions
Filter Discussions...
Print Discussions...
Discussion Options...
Discussions
Subscribe...
Close
Page 1 Sec 1 1/1 At 1.8" Ln 5 Col 24 REC TRK EXT OVR

Figure 29
Discussion options available while adding or reading document discussions.

The two top selections in Figure 29, Insert in the Document and Insert About the Document, duplicate their matching controls in the Web Discussion toolbar. The lower four additional choices include the following:

- **Refresh Discussions**—This command retrieves and displays discussion comments made by anyone else since you loaded the document into Word. Any comments you make are immediately displayed in your view of the document.
- **Filter Discussions**—Sometimes it is necessary to filter the comments by creator or for a given time period. The Filter Discussions dialog box, shown in Figure 30, enables you to select from the list of comment creators and filter for time intervals ranging from all (anytime) to six months.



Filter Discussions

Show discussion items that were:

Created by: Serena

Creation time: in the last 7 days

Figure 30
You can filter discussions by participant or time.

- **Print Discussions**—Prints a list of all the comments made on a given document. The complete path to the document under discussion is printed at the top of the first page. Each comment contains the following information:

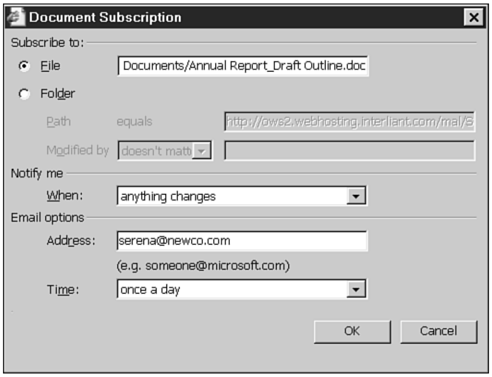
- The location of each inline comment by page number and line number
- The subject of the comment
- The author of the comment
- The text of the comment
- The date and time the comment was posted

- **Discussion Options**—From this dialog box (refer to Figure 23), you can select which discussion server to use and choose which information to display in a comment. Clicking the Remove button deletes the displayed discussion server. If you click the Add or Edit button from the Discussion Options dialog box, you can add a new discussion server or edit the path to an existing server. You can also choose to Show Closed Discussions. A *closed discussion* is a single comment or thread which has been removed from the rest of the discussion.

After a document discussion has started, you can monitor changes by subscribing to a document.

Understanding Subscriptions

After the five buttons for creating or navigating comments comes the Subscribe button. Opening a *subscription* or *Web notification* establishes a request that you automatically be notified by e-mail whenever a comment is left about any document (or folder of documents) to which you have a subscription. Clicking the Subscribe button brings up the screen shown in Figure 31.



Document Subscription

Subscribe to:

● File: Documents/Annual Report_Draft Outline.doc

○ Folder

Path equals http://ows2.webhosting.interliant.com/mal/S

Modified by doesn't matt

Notify me

When: anything changes

Email options

Address: serena@newco.com

(e.g. someone@microsoft.com)

Time: once a day

Figure 31
Activating a subscription to monitor document comments.

Notice from Figure 31 that you can open a subscription to the current document or use the filters under Folder to be notified based on

- All the files in a given folder or path on the discussion server
- Any comments left by a specific author

When you choose to subscribe to a folder, you have the option of being notified whenever a document is modified by anyone (doesn't matter) or by a specific person.

You can be notified whenever

- Anything changes.
- A new document is created.
- A document is modified.
- A document is deleted.
- A document is moved.
- A discussion item is inserted or deleted.

Under E-mail Options, you need to type the valid e-mail address to which notification should be sent. Time periods of notification include

- When a change occurs
- Once a day
- Once a week

Clicking OK places your subscription with the discussion server, and you receive a message back about whether a subscription was successfully started. You receive an e-mail whenever a comment is left according to the frequency you requested.

Disconnecting from the Discussion and Removing the Discussion Pane

The three rightmost buttons on the Web Discussion toolbar are

- **Stop Communication with Discussion Server**—This cancels the connection to the discussion server and is only active (it has red highlights) when the connection to the discussion server is active.
- **Show/Hide Discussion Pane**—This toggles the appearance or disappearance of the discussion pane. Toggling the discussion pane off lets you view more of the document.
- **Close**—This closes the discussion interface and eliminates it from the screen.

Also note the small down arrow next to the Close button. From here you can customize your Web Discussion toolbar, adding or removing options as you would when you customize any other toolbar.

Troubleshooting

After typing in the address for your discussion server, you get an error message, and you're not connected. What can you do?

This could be caused by several factors:

- The address must start with "http" and represent a complete URL, including the address to the folder for your specific Team Web Site. For example:
  `http://ows.reliant.com/mal`
- The discussion server is installed as part of Office Web Server extensions. Contact your ISP or IT department to make sure that OWS is installed and the discussion server is active. Also confirm the HTTP address leading to your Team Web Site.

YOU WANT TO SEE YOUR COLLEAGUE'S REVIEW COMMENTS IN A WEB DISCUSSION.

Make sure that you're looking at the newest set of comments. From the Web Discussion toolbar, choose Discussions and then Refresh Discussions. This refreshes all the discussion comments on your screen.

YOU RECEIVED A MESSAGE THAT SUBSCRIPTIONS ARE NOT AVAILABLE WHEN TRYING TO SUBSCRIBE TO CHANGES. HOW CAN YOU FIX THIS?

The subscriptions option requires an active SMTP (mail) server. This server may be down or have undergone some changes since OWS was installed. Contact your IT department.

YOU TRIED TO ADD A WEB COMPONENT TO YOUR DOCUMENT IN WORD, BUT IT DOESN'T WORK. WHAT'S WRONG?

Your page must be saved to a Web server with Office Web Server extensions active for the Web component to work. The Web component will not be active from your local hard drive, a network drive, or a Web server without an active OWS installation.

BONUS WORD CHAPTER

Using Word to Develop Web Content

In this chapter

by Michael Larson

Special Edition Using

que®

Web Page Development: Word's Strengths and Weaknesses

The Internet and its underlying technologies are a fixture in most businesses today. The preferred method for disseminating information within many corporations is now the company intranet. An *intranet* is an internal Web site designed for use within an entire organization or departmental unit. Parts of it may be open to public consumption on the Internet, but generally access is restricted to employees or other parties important to the organization. Some intranets have grown into *extranets*, which are networks of organizations that share a common interest.

With the increased emphasis on authoring documents for use in a Web environment, comes the need for Web authoring tools. And Word 2002 has much to offer in the Web arena.

Word 2002 and the Web

Word 2002 represents the next evolutionary step of incorporating Web technology into an Office product. Microsoft has continued to work under the hood to tune up the Web capabilities of this product. Some of the new Web-related capabilities in Word 2002 include

- A more robust translation to Web pages
- The choice to use simpler Web page code
- A new Web Archive file format
- E-mail based team collaboration
- Better support for Cascading Style Sheets
- The capability to easily incorporate picture bullets into Web pages

Microsoft built a wide variety of new Web technologies and languages into its Web page format in Word 2000. Word 2002 continues the use of this new alphabet soup of technologies, including

- XML (eXtensible Markup Language)
- CSS (Cascading Style Sheets)
- VML (Vector Markup Language)
- JavaScript and VBScript (Visual Basic Script)
- Support for the more versatile PNG (Portable Network Graphics) image file format

You don't need to know how to use each or any of these technologies to build or edit Web pages in Word 2002. The main effect of continuing to use these technologies is to enhance the capability of browsers to display the data, improve formatting, and increase the scope of graphical object types that can be included in Web pages.

This means that, in many situations, it doesn't matter anymore whether you save a document as a .doc binary file or a Web page. Either file format looks the same in both

Word 2002 and a properly equipped browser. All the information normally contained in the .doc binary file format is also included in the Web page format and vice versa. Microsoft calls this interchangeability of file formats *round-tripping*. You can use any Web page created in Word 2000 or Word 2002 to completely regenerate the binary .doc format. This was not possible with Word 97.

Note

Roundtripping applies only to Web pages created in Word. The document looks the same whether it's a Word-created Web page or a Word .doc file. If any other Web page is edited in Word, it may or may not look like it originally did after it has been saved from Word.

The beefed-up Web page format can now display the majority of Word features. These supplementary technologies increase the capability of HTML to display and manage more than is possible by using HTML alone.

Note

With Word 2000, you could save a .doc file as a .htm file that supported roundtripping. With Word 2002, you can still save to the roundtripping .htm file as well as two other Web file formats: simple (also called filtered) HTML and Web Archive file format. The simple HTML format is useful when you require file compatibility with older browsers. The Web Archive file format gives you the option of combining all Web page linked files, such as images, into a single file. This greatly simplifies file exchange and management.

Word 2002's Enhancements for Web Publishing

Word 2000 introduced several new Web-related enhancements to Word including Web folders and a Web page wizard. These capabilities are still present in Word 2002 along with some new enhancements for Web publishing:

- Link bars make it easy to build a simple navigation system between multiple Web pages.
- The capability to save a filtered Web page reduces the file size of a Web page and improves browser compatibility.
- Hyperlinks can now include *ScreenTips*, short text descriptions that pop up when your mouse pointer hovers over a link.

➔ To learn how to build ScreenTips into hyperlinks, **see** the Bonus Word Chapter, "Adding Interactivity with Hyperlinks, Web Forms, and Smart Tags."

These additions are most useful when you want to save existing Word documents as Web pages. If you want to create an entire Web site with many Web pages or have more control over the HTML source code, consider using FrontPage 10 as your primary authoring and Web site management tool.

Word 2002's HTML Shortcomings and Benefits

Word 97 suffered from an incapability to effectively translate many Word features into a Web page. When you saved a Word document as a Web page, many formatting items were simply not included in the resulting Web page. It was also often confusing as to which feature would be enabled in the Web page and which wouldn't. Word 2000 corrected most of these problems by beefing up the HTML source code used to save the Web page. Some weaknesses in Word's ability to translate all its features to Web pages, however, still existed and have carried over into Word 2002. The following features do not translate to a Web page:

- Versioning
- Passwords
- Word file headers/footers
- Columns (though the text is unaffected)

No text or formatting information is lost in the conversion to the Word 2002 Web page format if you use roundtripping. When using versioning, only the latest version number of the Word document is included in the HTML source. The lack of support for passwords lies in that, typically on a Web site, the Web server controls passwords, rather than individual documents (or pages). The lack of support for columns and headers/footers occurs because Web browsers simply have no functionality (that is, there is no HTML equivalent) to display these formatted items. When the Web page is reloaded into Word 2002, however, columns and headers and footers are restored. Because these "translation" problems are due to shortcomings in HTML or other Web technology, Microsoft simply cannot create a version of Word that is 100% compatible with Web pages.

Tip from
Michael Larson

The Web pages Word 2002 generates using roundtripping are not easily interchangeable with those created in other word processors or HTML editors. The languages incorporated into Word 2002 Web pages—XML and VML especially—simply don't have equivalents in most other programs. To avoid problems with programs that don't have support for XML and VML, save your Word documents as filtered Web pages.

Word 2002 is ready for prime time as a tool for Web page creation. Using Word to create Web pages has three advantages:

- If documents already exist as Word .doc files, saving them as Web pages is a one-step operation.
- You don't need to learn how to use a separate Web page editor or Web site management tool, such as FrontPage.
- You have multiple Web page format selections: roundtripping (Save a Web page), simple HTML code (Filtered Web page), and a single file, encapsulating format (Save as Web Archive).

These advantages are most useful when you're generating only a few Web pages or Web pages with a single purpose (particularly on a company intranet), or don't have the resources or time to learn FrontPage or other Web content creation tool.

Tip from
Michael Larson

Choose your tools based on the job you want to do. If you plan on using many flashy Java, JavaScript, VBScript, or dynamic HTML effects, then using a tool specifically designed to implement such elements would be a good idea. If you choose to provide information in a more conservative manner, then Word can do the job. Word's roundtripping HTML feature is useful if you plan to mirror your pages with print communications and you will be updating both the online and print versions simultaneously.

HTML Roundtripping: Advantages and Disadvantages

Microsoft introduced Roundtripping with Word 2000 and essentially the functionality remains unchanged in Word 2002. The term "roundtripping" simply refers to the capability of Microsoft Word to save most of the functionality and richness of a Word document into a Web page. Roundtripping has a single, unique advantage: It is the only format to retain most Word formatting, functionality, and document information in a Web page. In other words, this format retains the most compatibility with the Word .doc file format. If you must use a Web page format and need to retain maximum compatibility with Word, then use the Web page format in Word.

Roundtripping has several disadvantages:

- The generated Web page file size is large, and the HTML source code is complex.
- The heavy use of XML, VML, and so forth reduces compatibility with browsers, especially older ones.
- The Web page format generates an extra folder that stores images and other support files. This makes the file less portable and increases file management tasks.

For the company intranet, using the Word .doc format is often easier than using the Web page format, especially if all users have access to Word or a Word viewer.

Note

Word viewers for all Windows platforms are available on the Microsoft Web site at `http://www.microsoft.com/office/downloads.htm`.

Another workaround for the disadvantages of roundtripping is to save the Word document as a filtered Web page (smaller file size and simpler HTML) or as a Web archive file (only a single file; no subfolders to manage). Both of these options are new in Word 2002 and are listed in the File, Save as dialog box.

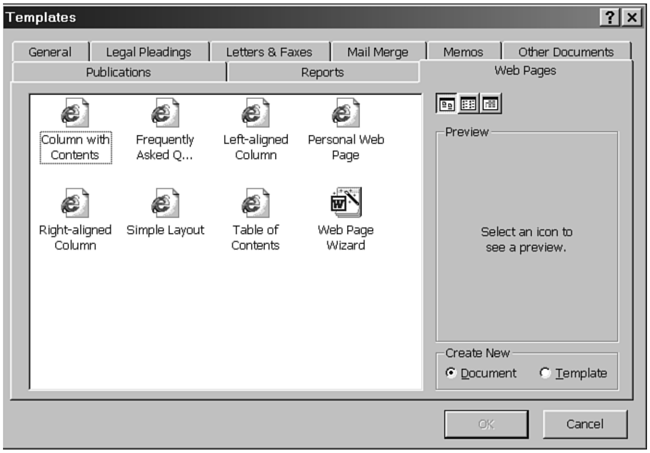
CREATING A SINGLE WEB PAGE IN WORD

In Word 2002, creating a Web page is no different from creating a Word document. You do not need to open up a special environment or think differently about the contents of your page.

BUILDING A WEB PAGE FROM SCRATCH

To begin building from a blank Web page, click on the Blank Web Page link in the New Document window in the task pane. If you prefer a page that has some basic formatting, choose General Templates from the New from Template section in the New Document window and choose from among some preformatted Web page types, as shown in Figure 1.

Figure 1
A list of preformatted Web pages.



After you open the preformatted page, just replace the placeholder text with your own content in the same way you would if editing the template of a typical Word document.

Tip from
Michael Larson

The main advantage of using template pages is that you can concentrate on content rather than worry about the formatting. To maintain the formatting, select only the placeholder text you want to replace in a given section. If you select the whole page when you intend to delete the placeholder elements, you may lose some of the other formatting elements. Also note that you can create your own templates and add them to the list of Web page templates as discussed in the section "Creating Your Own Web Page Templates" later in this chapter.

PREVIEWING THE WEB PAGE

As you are building your Web page, you can view or preview your Web pages using Web Layout view and Web Page Preview.

Web Layout view (choose View, Web Layout from the menu or click the Web Layout icon in the status bar) presents your document like a Web page. The only difference between Normal view and Web Layout view is that the background texture or image, normally present in a Web page, is turned off in Normal view. All other items display identically.

Web Page Preview enables you to preview your Web page in a browser without first having to save the file. Click File, Web Page Preview on the menu to initiate the process. The file in Word is opened in your default browser for viewing. This ensures that what you are building in Word is indeed being displayed the same way in the browser.

Note

Remember that just because your Web page looks good in one browser doesn't mean that it'll look good in all browsers. Unless you know everyone will be accessing your Web page with the same browser and version, it is a good idea to test your Web pages out with the latest versions of Microsoft and Netscape browsers as well as earlier versions, if possible. Note that if some things do not show up in one browser (the scrolling marquee, for instance, is not supported by Netscape browsers), you may need to remove that element or build browser-specific pages.

In Figure 2, you can see the effect of browser type on the Web page display of a table with a background color.

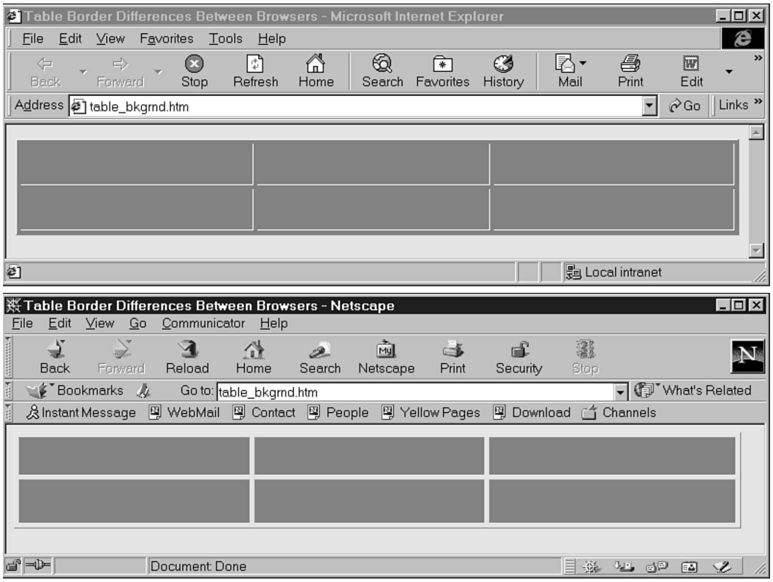


Table Border Differences Between Browsers - Microsoft Internet Explorer

Address table_bkgrnd.htm

Table Border Differences Between Browsers - Netscape

Go to: table_bkgrnd.htm

Document: Done

Figure 2
Table borders display differently between Netscape Navigator (bottom) and Internet Explorer (top) when a table background color is used.

The table size and table color background (red) are the same, but the borders, despite being the same in Word, display differently.

➔ To see a list of browser-specific tags, **see** "Web Browsers and Platforms," (Chapter 6) in the book *HTML 4 Unleashed* (Rick Darnell, ISBN: 1-5752-1299-4, Sams Publishing.)

Saving as a Web Page

With a Word .doc file, you can embed many elements, such as graphics and sounds, directly into the .doc file so that you have only one file to worry about. Web pages, however, do not directly embed many of the items displayed in browsers. For a graphic, the Web page contains only the path to the image file, not the image file itself. The same is true for sounds, videos, Java applets, and many other items. So what happens to these associated files when you save a Web page from Word 2002?

For example, imagine that you're saving your Web page to a local drive under the filename of HomePage.htm using Save as Web Page.

Note

You can save an open document as a Web page using File, Save As and choosing Web page from the Save as type drop down box at the bottom. Or there is a separate choice on the File menu of Save as Web Page. Both produce the same result.

First, the actual Web page, HomePage.htm, is stored in the folder you selected on your local hard drive. Next, a folder called HomePage Files is created under the folder containing HomePage.htm.

Note

The name of the folder containing ancillary files will track with the filename you select for your Web page. So if you use Save as Web Page to save a file named Earnings, the created subfolder will be named Earnings Files.

In the HomePage Files folder, you will find the following:

- Two files, filelist.xml and header.htm, contain information necessary for roundtripping (that is, for translating HomePage.htm back to a .doc file). These files are not used when HomePage.htm is viewed in a browser.
- All the bitmap image files are saved as image001.gif, image002.jpg, image003.png, and so forth.
- Parts or all of drawing objects may also be saved as .gif files. The remainder will be displayed as VML code in the HTML source.
- Sound and video files are not moved to the HomePage Files folder. Although your Web page will properly play the sound or movie on your local hard drive, anyone else accessing your Web page will not receive any audio or video.

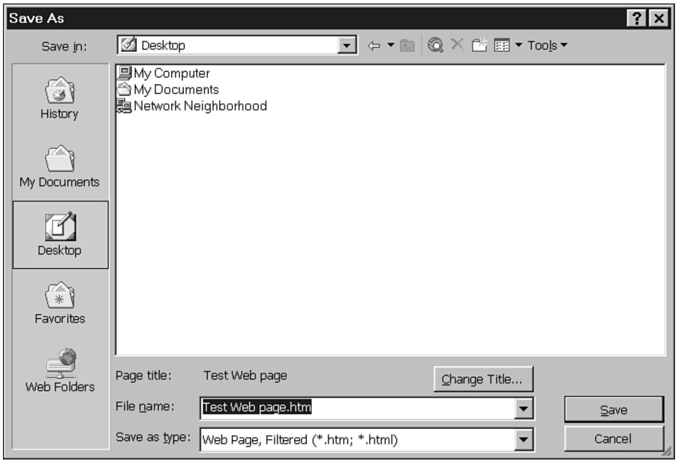
Tip from
Michael Larson

Save your Web page before inserting the multimedia elements. Doing so instructs Word to create the HomePage Files folder. Then copy the audio, video, and Java applets to the HomePage Files folder before inserting them in your Web page. This ensures that the files remain linked to HomePage.htm.

SAVING TO FILTERED WEB PAGE FORMAT

As mentioned earlier, the Web Page format in Word is complex and a larger file size than simpler HTML pages. Word 2002 now offers you the capability to save your Web page in the simpler HTML format called the Filtered Web Page format. To save a document as a filtered Web page, follow these steps:

1. Create your document from scratch or load an existing document.
2. Chose File, Save As from the menu. This brings up the Save As dialog box.
3. From the Save as Type drop-down list box at the bottom of the Save As dialog box, select Web Page, Filtered (*.htm, *.html) as shown in Figure 3.



Save in: Desktop

My Computer
My Documents
Network Neighborhood

Page title: Test Web page
File name: Test Web page.htm
Save as type: Web Page, Filtered (*.htm; *.html)

Figure 3
Saving a Word document as a filtered Web page.

4. Click Save. A dialog box warns you about the Office-specific tag removal.
5. Click Yes to continue to save your document as a filtered Web page.

The primary advantage of the filtered format is simpler HTML code and a smaller file size that improves loading speed from a Web server. It is also easier to edit the filtered Web page in an HTML editor, such as FrontPage. A serious disadvantage is that fewer Word elements are saved than a typical Web page format. In addition to items that are not saved in the Web page format, you may also lose the following:

- WordArt is saved as a static bitmap image file.
- Document properties are lost. This does not affect any content.
- Font formatting.
- Page setup information.
- Custom list (numbered or bulleted) styles.
- Legacy frames are converted to tables.
- Custom underline styles and colors.
- Tabs.

Tip from
Michael Larson

If you're working on a new document, make sure to save it first in another, more complete file format (.doc or Web archive) to retain information that won't be saved in the Filtered Web Page format. Don't choose Save as Web page as a more complete file format since it has the same file extension as the filtered Web page, leading to possible confusion.

SAVING TO WEB ARCHIVE FILE FORMAT

Microsoft introduced the Web Archive file format in Word 2002 to address a common problem when saving Web pages in Word: a subfolder is created for every Web page saved. Every time you want to move the Web page, you have to remember to move the subfolder. Moreover, if you save several files as Web pages, you accumulate just as many subfolders.

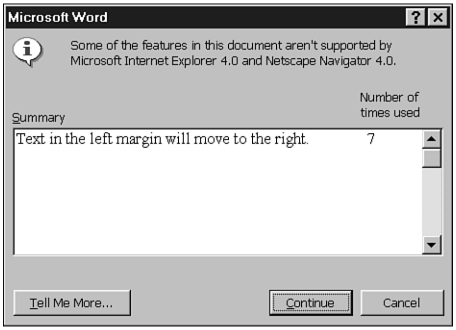
The Web Archive File format (.mht or .mhtml) stores all files associated with a Web page (images, sound files, XML files, and so on) into a single archive file that a browser can read. This makes the files much easier to transport and maintain and eliminates the proliferation of subfolders. But it has a significant drawback: This file format requires Internet Explorer or Netscape Navigator version 4 or later to display the file.

To save your document as a Web archive, follow these steps:

1. Choose File, Save As from the menu.
2. From the Save as Type drop-down list, select Web Archive (*.mht, *.mhtml).
3. Click Save. If certain items will not display properly in the resulting archive, you may see a dialog box similar to Figure 4 listing the problems.
4. If you receive the dialog box, click Continue to finish the save. Clicking Tell Me More leads you to Online Help topics about the Web Archive format.

VIEWING HTML SOURCE CODE

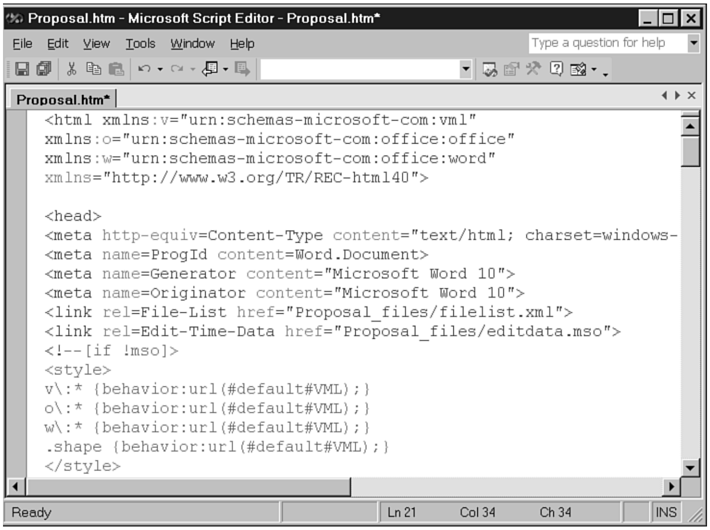
Whether you save a file as a Web page, filtered Web page, or Web archive, the page information is saved as *HTML source code*. HTML source code is the actual code that a browser translates to display a Web page. In Word 2002, this is not only HTML but also includes XML, CSS, VML, and the scripting languages. If you want to access this code—for instance to modify the JavaScript—you can access the code from the menu under View, HTML Source. The source code is displayed in Microsoft Script Editor window, as shown in Figure 5.



Some of the features in this document aren't supported by Microsoft Internet Explorer 4.0 and Netscape Navigator 4.0.

Summary	Number of times used
Text in the left margin will move to the right.	7

Figure 4
Saving a Word document in Web Archive format.



```
<html xmlns:v="urn:schemas-microsoft-com:vml"
xmlns:o="urn:schemas-microsoft-com:office:office"
xmlns:w="urn:schemas-microsoft-com:office:word"
xmlns="http://www.w3.org/TR/REC-html40">

<head>
<meta http-equiv=Content-Type content="text/html; charset=windows-
<meta name=ProgId content=Word.Document>
<meta name=Generator content="Microsoft Word 10">
<meta name=Originator content="Microsoft Word 10">
<link rel=File-List href="Proposal_files/filelist.xml">
<link rel=Edit-Time-Data href="Proposal_files/editdata.mso">
<!--[if !mso]>
<style>
v\:* {behavior:url(#default#VML);}
o\:* {behavior:url(#default#VML);}
w\:* {behavior:url(#default#VML);}
.shape {behavior:url(#default#VML);}
</style>
```

Figure 5
Viewing the HTML source code of a Web page in the Microsoft Script Editor.

Note

The Script Editor is not part of the default installation. The first time you activate this feature, it initiates autoinstallation so keep your Word 2002 or Office XP CD-ROM handy.

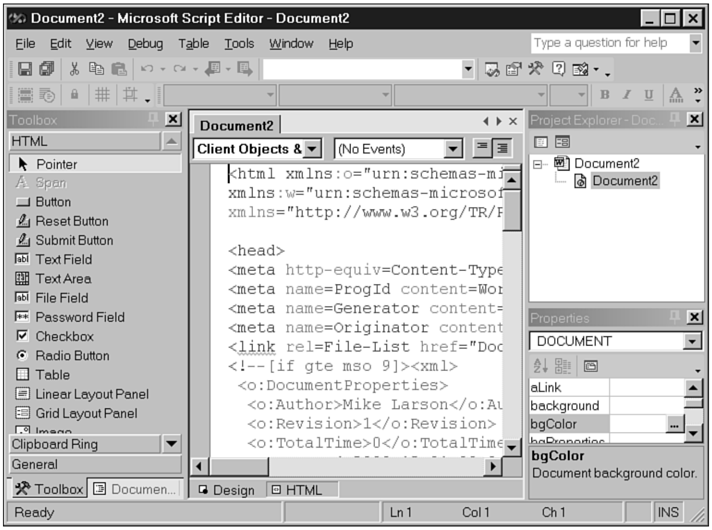
As far as the HTML source code is concerned, you can search and edit the HTML source code from within the Script Editor as you want. To save your changes, choose File, Exit from the menu.

Directly modifying the HTML source from within the Script Editor is not recommended unless you are an experienced programmer or Web developer, or if you want to learn more about HTML by experimenting on pre-built code.

USING WEB SCRIPTING

Web pages commonly use scripting languages to define actions and objects accessible from within a browser. The most common languages in use are JavaScript (Microsoft's version is called JScript) and VBScript (Visual Basic Script). To add JavaScript or VBScript code to a new Web page, do the following:

1. Choose Tools, Macro, Microsoft Scripting Editor from the menu. Or use the keyboard shortcut, Alt+Shift+F11. The Microsoft Script Editor opens up as shown in Figure 6, although with a different view from typical HTML source viewing.



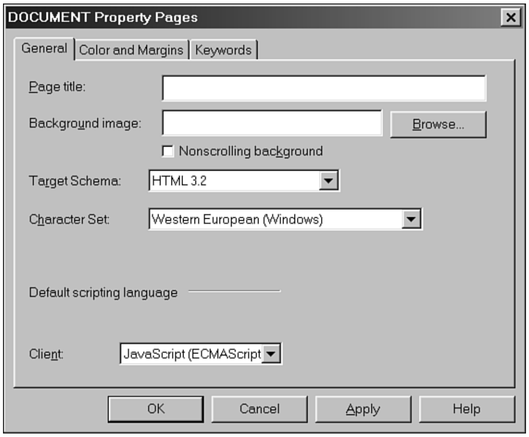
```
<html xmlns:o="urn:schemas-mi
xmlns:w="urn:schemas-microsof
xmlns="http://www.w3.org/TR/F

<head>
<meta http-equiv=Content-Type
<meta name=ProgId content=Wor
<meta name=Generator content=
<meta name=Originator content
<link rel=File-List href="Doc
<!--[if gte mso 9]><xml>
 <o:DocumentProperties>
  <o:Author>Mike Larson</o:Au
  <o:Revision>1</o:Revision>
  <o:TotalTime>0</o:TotalTime
```

Figure 6
Adding script code to a Web page using the Microsoft Script Editor.

Note

The Script Editor view for typical HTML source (refer to Figure 18.6) does not open the Toolbox, Properties window, or Project Explorer window used to create or edit scripting code as seen in Figure 18.7.

2. To choose the default language (either VBScript or JavaScript) in which to write a script, choose View, Property Pages from the menu (Shift+F4 from the keyboard). This affects scripts for which you do not explicitly specify a language. As shown in Figure 7, you can choose between VBScript and JavaScript (ECMAScript) for the Client.



Page title:

Background image:

☐ Nonscrolling background

Target Schema: HTML 3.2

Character Set: Western European (Windows)

Default scripting language

Client: JavaScript (ECMAScript

Figure 7
Choosing the scripting language to use in a Web page.

Note

Client-side scripts execute within the browser after the Web page has loaded; they are self-contained.

The scripting language can also be changed in the Properties window of the Script Editor under defaultClientScript.

3. After you've selected a scripting language, scroll to the insertion location in the Web page. Right-click and choose Insert Script Block, Client (see Figure 8). This command is also available from the menu under Edit, Insert Script Block, Client. A *scripting container* is simply the code defining the start and end of the text of the script within a Web page. You fill in the middle of the container with your desired scripting functions.

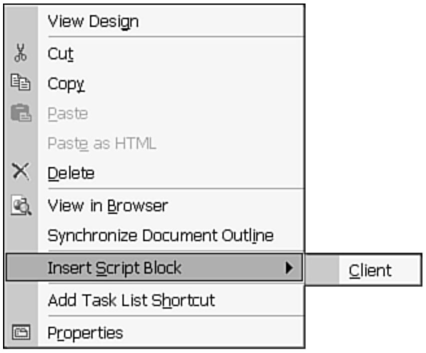


Figure 8
Inserting a blank scripting container into a Web page.

A typical blank scripting container for JavaScript would appear as follows:

```
<script language=javascript>
<!--
```

```
//-->
</script>
```

4. After the container is created, you can start typing code. If you prefer to select scripting functions from a list, click ShowMemberList on the Text Editor toolbar. An object member list appears as shown in Figure 9. Scroll and choose members as needed.

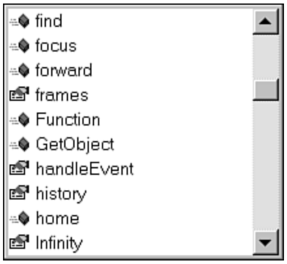


Figure 9
Choosing scripting members from a pop-up list.

5. After you've selected a member, you can click on the ParameterInfo iconin the Text Editor toolbar to view the proper syntax for a member.

The Microsoft Script Editor includes many of the amenities found in sophisticated programming environments, such as debugging and the capability to insert breakpoints. You can completely test and debug your script from within the Script Editor.

Note

You may need to install the debugging features in the Script Editor before use. The Debug menu consists of only one item, Install Web Debugging, if debugging features aren't installed.

When the script works to your satisfaction, save the file and exit the Script Editor to return to Word.

Caution

Word generates the HTML code. If you resave the Web page from within Word, any manual changes you made from the Script Editor may be overwritten and lost.

THE TECHNOLOGIES USED IN WORD 2002 WEB PAGES

Whenever you save a Word document as a Web page or use Word to build a new Web page, you are building a page of HTML source code that a browser can read and interpret to display the various elements in the Web page. As described earlier, in Word 2002, four other technologies (XML, CSS, VML, and scripting languages and supplementing traditional HTML code.

These new scripting technologies have some advantages and disadvantages. The advantages include

- Web pages can now contain almost anything that a Word 2002 document can.
- It no longer matters whether you save a file as .doc or as .html; a document looks the same in Word 2002 as it does in a version 5.0 browser.
- XML greatly improves the organization and searchability of the Web page.
- VML reduces the bandwidth required to send a graphical image from a Web server to a browser. This improves the browser page load time, improves image quality, and helps to reduce Internet or intranet network congestion.

The disadvantages to these new technologies include

- If you want to edit the HTML code outside Word 2002, you need to be a programmer (that is, be able to understand the scripting languages). Also, rather than just having to only know HTML, you need to understand VML, CSSs, and XML, as well as how these four languages interact with each other. This can be daunting even for many experienced Web designers and programmers.
- These technologies are not compatible with earlier browser versions (versions 4 and earlier) still in use on the Internet, though the majority of Internet visitors are using Internet Explorer 5. Telling Word 2002 to implement only those Web page features common to version 4 browsers helps somewhat, although it reduces the effectiveness of using Word 2002 to build Web pages, especially roundtripping ones. In the meantime, this technology can be used to its fullest on an intranet or extranet where most people can be motivated or required to have version 5 or higher browsers.

Component Overview

Each component language used in Word 2002 Web pages has a distinctive role. Here is a short overview of each of the five languages:

- **HTML**—Hypertext Markup Language is the *lingua franca* (medium of exchange) of the World Wide Web. Almost every Web page is built with this language. HTML, a simple formatting and organizational language, is ideal for the display of text, simple graphics, and hyperlinks. It is not suited at all for precise page layout (such as you might find in a magazine) or complex data organization. The appeal of HTML is its ease of use and universal acceptance. HTML 4 is the current version of this language.

Caution

Do not confuse the language version of HTML, currently at 4.0, with the browser versions already mentioned.

- **CSS**—Cascading Style Sheets are used to define the layout of a document precisely. Style sheets are more powerful than the styles found in Word because style sheets can also specify page layout. A style sheet can be a separate document, or it can be embedded in each HTML page. Because browsers have different capabilities in how they interpret these styles, they interpret what they can and ignore the rest; that is, they cascade down in their interpretation and display what they are able to. Netscape Navigator 3 and Microsoft Internet Explorer 2 browsers or earlier cannot read these style sheets.
- **XML**—EXtensible Markup Language is more robust and extensible (hence its name) than HTML. You can define new tags and their uses at any time and in any way by referencing them in an associated text document. The strength of XML is its capability to use these new tags to identify specific information. For instance, if you're talking about a golf wood on a Web page, your page can include golf-specific tags. This distinguishes your page from the many others that may be dealing with carpentry; these would probably use woodworking-specific tags. People who want to look for Web pages about golf woods can now filter out the woodworking pages or vice versa. This technology will vastly improve the users' abilities to find specific-subject Web pages and will open the Internet up to even more data mining.
- **VML**—Vector Markup Language uses text to define geometric shapes, colors, line widths, and so forth. These words are then interpreted and displayed as graphical images in browsers that understand VML (currently only Microsoft Internet Explorer 5). No matter what size circle you want to display, you use the same amount of text to define it.
- **JavaScript or VBScript**—Both of these script-style programming languages are in common everyday use on the Web right now. They are both robust. JavaScript is supported by the vast majority of browsers; VBScript is supported by only Microsoft Internet Explorer browsers. These languages enable you to program interactivity into Web pages.

All (or most) of these languages are used to build Word 2002 Web pages However, don't let this plethora of languages dismay you. You do not need to understand one line of code of any language to use Word 2002 to build effective Web pages.

ADDING CONTENT TO WEB PAGES IN WORD

A Web page and a Word document are similar in that their predominant content is usually words, sentences, and paragraphs. But Web pages include content not typically found in Word documents. In this section, we'll look at how to apply the following to a Web page:

- Hyperlinks
- Horizontal rules
- Scrolling text (also called a *marquee*)
- Fonts

- Framesets
- Images and background texture graphics and colors
- Graphics formats
- The Web palette
- Web Page Templates
- A good Web page title

Lastly, we'll look at how to create a group of linked Web pages, essentially the contents of a small Web site, using the Web Page Wizard.

Building Hyperlinks

One of the most powerful technologies on a Web page is the hyperlink. Through a simple click, you can be transported almost anywhere on the Internet. Hyperlinks are discussed in much greater detail in Chapter 19. For now, you need to know the basics of building a hyperlink.

A *hyperlink* is the combination of some descriptive text or an image and the location (also called the address) of a Web page or object. Any text or image on a Web page can hold a hyperlink. Hyperlinks most commonly point to:

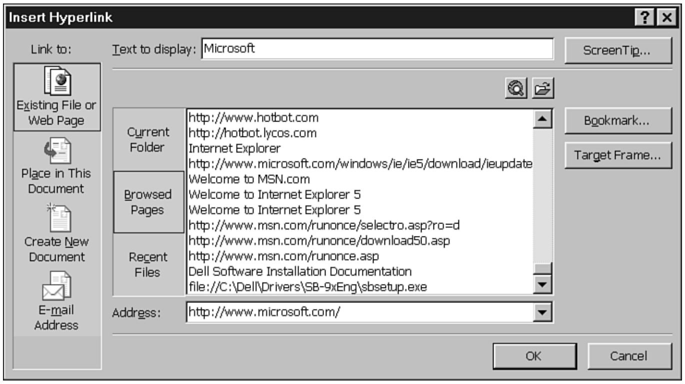
- Web pages
- Media objects such as sounds, video, or pictures
- E-mail addresses

Building a Text Hyperlink

Any text phrase, word, or part of a word can be included in a hyperlink. To build a hyperlink, you need some text in a document and the exact location where you want the hyperlink to lead when it is clicked:

1. From a document in Word, highlight a text phrase. For instance, highlight "Microsoft" in the phrase, "For more information, visit the Microsoft home page."
2. Click the Insert Hyperlink button on the Standard toolbar to open the Insert Hyperlink dialog box, shown in Figure 10. You can also reach the Insert Hyperlink dialog box by using the keyboard shortcut, Ctrl+K, or right-clicking on the selected text and choosing Hyperlink from the shortcut menu.
3. In the top box, labeled Text to Display, the text you highlighted is displayed. In the Address field, enter the Web URL for the site to which you want the text to link.
4. Click OK. When you return to your document, the text you highlighted is now a blue color and underlined, indicating that it is now an active hyperlink.

Figure 10
Filling in the Insert Hyperlink dialog box.



Text to display: Microsoft

http://www.hotbot.com
http://hotbot.lycos.com
Internet Explorer
http://www.microsoft.com/windows/ie/ie5/download/ieupdate
Welcome to MSN.com
Welcome to Internet Explorer 5
Welcome to Internet Explorer 5
http://www.msn.com/runonce/selectro.asp?ro=d
http://www.msn.com/runonce/download50.asp
http://www.msn.com/runonce.asp
Dell Software Installation Documentation
file://C:\Dell\Drivers\SB-9xEng\sbsetup.exe

Address: http://www.microsoft.com/

Note

Hyperlinks most often link to URLs (Uniform Resource Locators). A URL is simply the Internet address of a Web page. A URL can be very specific and lengthy, pointing toward a single file buried deep within a Web site. Or it can be as simple as microsoft.com. Most Web browsers no longer need the protocol header (http://) or the www (World Wide Web) portion of the address to find a common Web site.

Building a Hyperlinked Image

The process for building a clickable or hot image is similar to building a text hyperlink:

1. Select any clip art, picture, Drawing object, or WordArt within a Web page.
2. Click Insert Hyperlink on the toolbar to display the Insert Hyperlink dialog box. The Text to Display line will be dimmed because there is no text.
3. Type the address for the link in the Address box at the bottom.
4. Click OK to complete the hyperlink.

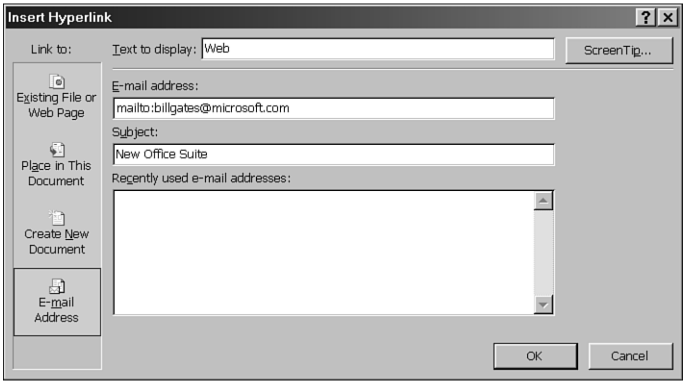
The picture will not look any different. If you view the Web page using File, Web Page Preview, the default cursor changes to a hand with a pointing finger when it hovers over the image to indicate it is now clickable.

Building an E-mail Hyperlink

Besides referencing other pages, hyperlinks can open and preaddress blank e-mail messages:

1. Select the text or choose an image for the hyperlink.
2. Click on the Insert Hyperlink button to display the Insert Hyperlink dialog box.
3. In the lower-left corner of the Insert Hyperlink dialog box, click on E-mail Address.
4. Enter the e-mail address, as shown in Figure 11.

Figure 11
Hyperlinking to an e-mail address.



Text to display: Web

E-mail address: mailto:billgates@microsoft.com

Subject: New Office Suite

Notice how the phrase `mailto:` is automatically added to the beginning of your e-mail address.

5. Click OK to complete the link.

When the link is clicked, a blank, preaddressed e-mail is opened. The e-mail hyperlink is a convenient means for letting visitors to your Web page send you feedback or questions.

➔ The process of working with hyperlinks in Word is described in greater detail in the Bonus Word Chapter, "Adding Interactivity with Hyperlinks, Web Forms, and Smart Tags."

Inserting Horizontal Lines in a Web Page

Word documents don't normally use horizontal lines (also called horizontal rules) to separate topical sections. This is, however, a commonly used convention in Web pages because there is no established page length for Web pages. To add horizontal lines to a Web page, follow these steps:

1. Click on the line in your page where you want to insert the horizontal line.
2. Choose Format, Borders and Shading from the menu.
3. Click the Horizontal Line button at the bottom of the resulting dialog box.
4. A list of lines from which to choose is displayed and should look similar to Figure 12.
5. Choose a line and click OK to insert it into your Web page.

Note

Horizontal lines can be either the HTML element (<HR> tag) itself or, more commonly, a decorative graphic line.

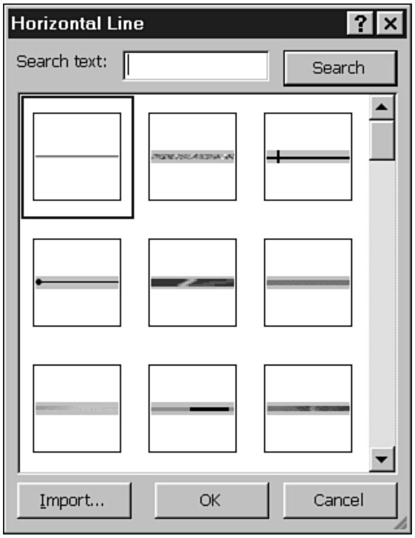
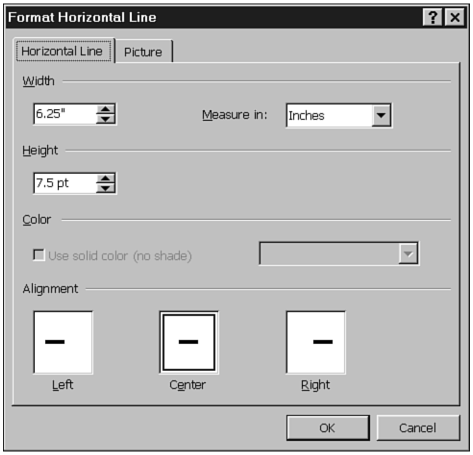


Figure 12
Select a horizontal line to use.

After you've inserted a horizontal line (HTML tag or a graphic) that you like, you can edit the line by following these steps:

1. Select the horizontal line by clicking it; then right-click it and choose Format Horizontal Line from the menu. The Format Horizontal Line dialog box, shown in Figure 13, displays.



Width: 6.25" Measure in: Inches

Height: 7.5 pt

Color: □ Use solid color (no shade)

Alignment: Left [✓ Center] Right

Figure 13
Changing the properties of a horizontal line.

2. The current width, height, and alignment of the line are displayed. Color will also be displayed (and can be changed) if the horizontal line is the HTML tag and not a graphic image.

3. The next tab in this box, Picture, (this tab will not be present for the HTML tag generated line) lets you change some of the picture qualities including brightness and contrast. You can also crop the line from here.
4. Click OK to accept your changes to the horizontal line.

Tip from
Michael Larson

After you have the line looking just as you want, use the Clipboard to copy and paste the line when you want to use it again. For consistency, use the same horizontal line throughout your Web site or section of a Web site.

Inserting Scrolling Text

Scrolling text is also called *marquee text*. Scrolling text marches stock-ticker style in a line across your Web page.

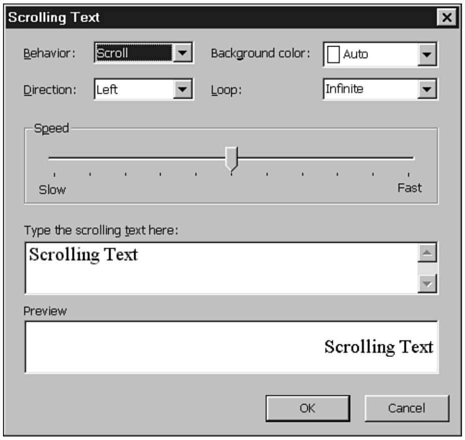
Note

Before using scrolling text, remember that it is not supported in Netscape browsers and displays only partially or not at all. If you expect your Web page to be viewed with Netscape browsers, either don't use this feature or don't put any critical information in it.

You can find scrolling text on the Web Tools toolbar. (Right-click on a toolbar and select Web Tools if this toolbar isn't open.)

The ScreenTip for this control is labeled Scrolling Text. When you click on it, the Scrolling Text dialog box appears, as shown in Figure 14.

Figure 14
Adding Scrolling Text to a Web page.



Scrolling Text

Behavior: Scroll
Background color: Auto
Direction: Left
Loop: Infinite

Type the scrolling text here: Scrolling Text

Preview: Scrolling Text

To make the scrolling called text move how you want and display your text, do the following:

1. Set your text behavior to Scroll (the default), Slide (which scrolls the text only once and stops on the opposite side of the screen), or Alternate (which bounces the text back and forth—like a Ping-Pong ball—between the left and right margins of your page).
2. For the Scroll and Alternate settings, you can choose how many loops to scroll or bounce with the Loop settings. Your only choices with this control are 1 through 5 times and Infinite.
3. Choose a background color for your control from the list. (Sorry, no custom colors are on this list.)
4. Choose the direction of your scroll: Left or Right.
5. The slider control in the middle of the box controls the speed of the scroll (or bounce).
6. Finally, type in the text you want to scroll. The window at the bottom of this box provides a preview of your scrolling text as you change the various options.
7. Click OK to place the marquee in your Web page.

After the marquee is inserted called into your Web page, you can make more changes to it:

- Grab the bounding box with your mouse to change the length or height of the marquee (the text stays centered).
- Format the scrolling text using any of the standard font-formatting tools in Word.
- If you right-click the marquee, the context menu shows three options: Stop (stops the marquee from scrolling or bouncing), Play (starts the marquee), and Properties (brings up the Scrolling Text dialog box again).
- To make the entire marquee a clickable hyperlink, select the marquee by choosing Design Mode from the Web Tools toolbar. Choose Insert, Hyperlink from the menu. Insert the correct Web address and click OK.

Note

Marquees are great attention-grabbing called devices for pointing people to new or important information. They can be irritating or distracting, however, if placed on a page where people also have to read a lot of text.

USING FONTS

You can apply any font available on your computer to the text on your Web page. You can also select any font size or weight you want just as you would for a Word document.

Note

HTML does not support using embossing or engraving in font styles and won't be displayed in any browser.

Remember one important caveat when you choose fonts: The computer displaying any Web page you create must have the same fonts installed as your machine. If, for example, you choose Andale Mono as your font style under Windows, and your Web page is viewed on a Macintosh computer or under Unix, those viewers will not see the text displayed with the Andale Mono font. Their browsers will look for the font and when it is not found, the browser will substitute a default font available on that machine. If the Andale Mono font is important to your Web page in terms of spacing or emphasis, both of these features will be lost on machines that don't have the Andale Mono font installed.

There are a few workarounds that allow you to use a variety of fonts and have them display across multiple platforms and browser types:

- Manually mark up the font tag to support multiple fonts.
- Convert text to graphical images.
- Embed Web fonts into your Web page.

Manually Modifying the Font Tag

The majority of computers in use have three basic fonts installed: serif, sans serif, and monospaced. Examples of these fonts on the Windows platform include Times New Roman, Arial, and Courier, respectively. If, for example, you apply the Arial font to a line of text, the HTML code will appear as follows:

```
<font face="Arial">
```

However, on the Macintosh, the default sans serif font is Helvetica, and on many Unix installations, it is simply sans serif. You can manually add these font names to the font tag as follows:

```
<font face="Arial, Helvetica, Sans serif">
```

You cannot tell Word to support multiple font names when it applies fonts to a Web page. You must make these changes within the HTML source using a global search and replace. Note that the font tag may be written differently within the source depending on whether you're working in a Web page, filtered Web page, or Web archive. Again, Word may overwrite any manual changes you make to the HTML source if you later open and save the Web page from Word.

Note

Cascading style sheets (CSS files) or embedded style tags can be used to apply font information to text. Any fonts you specify with a style tag or CSS file must still be present on the viewer's computer to be rendered.

Converting Text to Graphics

If you want to use a font that is not commonly found across multiple platforms, you have the option to add the text to your Web page as a graphical element. This completely eliminates the need for any user to have your font on his machine. There are a couple of serious side effects of this approach though:

- If you display a large amount of text as graphics, the Web file size increases dramatically, slowing down page load for users who have a slow connection.
- A search engine cannot index any text converted to graphics. The search engine simply sees an image; it cannot "read" the text.

Using text-as-graphics only for headers keeps your page size down. To make the text of the graphic searchable, you can add the text of your image to a Web page comment or a metatag or manually edit the image tag to include the title attribute. Examples of these approaches are displayed here:

```
<!—Add the text of your graphic as part of a comment here —>
<meta name="Description" content="Text of your graphic here">
<img src="images/text.gif" title="Text of your graphic here">
```

Comments and metatags will not display in a Web browser but a text indexer can read them. The metatag information must be placed between the <head> and </head> tags in your Web page. Again, all these changes must be made manually in the HTML source.

To create a graphic from text, you can use a bitmap editor such as Paint or Adobe Photoshop. Or you can create your text using WordArt and save your Web page as a filtered Web page. The WordArt is converted to a bitmap (.gif or .jpg) graphic during the save operation.

➔ For more detail on saving to the Filtered Web Page format, **see** the section earlier in this chapter, "Saving to Filtered Web Page Format," **p. 9.**

Embedding Web Fonts

Microsoft has developed means for Web page authors to embed fonts directly into a Web page. When a browser reads a Web page, it accesses the embedded fonts on that page to display the text. So as an author, you no longer have to worry about which machines have which fonts. Using embedded fonts does increase the Web page file size somewhat, and not all browsers, especially older ones, can read the embedded Web fonts.

Word 2002 unfortunately does not have the capability to insert embedded Web fonts into a Web page. If you want to use embedded Web fonts, visit Microsoft at `http://msdn.microsoft.com/workshop/c-frame.htm?/workshop/author/default.asp`.

Using Frames

Frames enable you to display more than one Web page at a time in a Web browser, with each page in its own custom-sized window or frame. You can have as many frames as you want, but in practical use most people use two or three. The most common use for frames is for navigation: the smaller, left-hand pane has a list of links. Clicking a link in this navigation pane displays the page content in the main, larger frame to the right. Some people choose to use a third frame across the top or bottom of the page for additional navigation choices or advertising.

Frames have a number of disadvantages:

- Older browsers, usually version 2 or earlier, do not support frames. These browsers are now rare on the World Wide Web and are virtually nonexistent on company intranets.
- Hyperlinking becomes more complex. You need to add some special text to your hyperlinks when working with frames, though this is done transparently in Word. If hyperlinks in frames are used improperly, you can get the dreaded “hall of mirrors” effect, where a frameset is duplicated several times in the browser window, making each window very small and almost unreadable.
- Frames eat up valuable screen real estate on smaller screens such as those on laptops and mobile computing devices.

Adding Frames

You can create a set of frames on a Web page, either by choosing Format, Frames, New Frame Page from the menu or by opening the Frames toolbar. From the Frames toolbar, shown in Figure 15, you can choose to put a frame on any side of the existing page (above, below, left, or right).

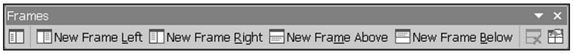


Figure 15
Adding frames to a Web page.

Note

A *frameset* is just a container page that tells the browser how to split the screen real estate among the pages in the set. For instance, a browser displaying two Web pages as frames is actually using three Web pages: one Web page for display in each frame and the third frameset page defining how the screen is split up between the other two pages.

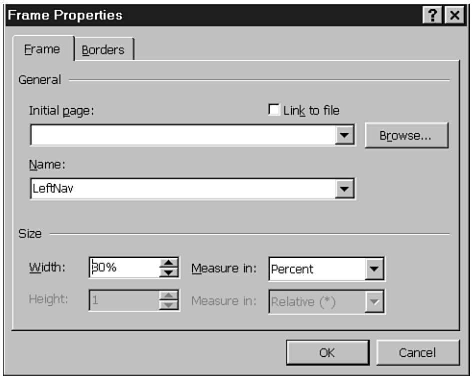
Modifying Frames

After you have a frameset, you can grab the bar between frames and resize each frame manually.

You can also access the properties of each frame by right-clicking a frame and choosing Frame Properties or choosing a same-named button from the Frames toolbar. The settings chosen in the Frame Properties dialog box are specific for one frame:

1. From the Frame tab, shown in Figure 16, you can name your frame and numerically resize it.

Figure 16
Naming and resizing a frame from the Frame Properties dialog box.



Frame Properties

Frame | Borders

General

Initial page:

□ Link to file

Name: LeftNav

Size

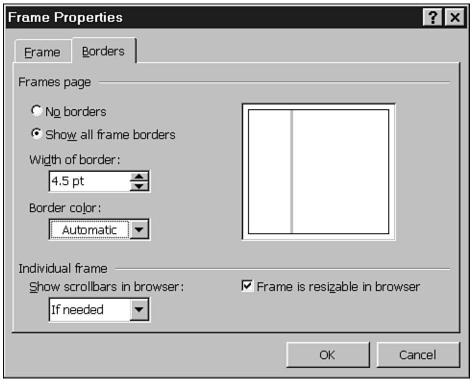
Width: 30% Measure in: Percent

Height: 1 Measure in: Relative (*)

2. You can choose an Initial Page (the Web page that will actually be displayed in the frame) for the frame by choosing an existing Web page from the drop-down box or browsing to a file. Name the frame based on its function to make it easier to build hyperlinks between frames. For instance, in Figure 18.18, the name LeftNav indicates both the location of the frame in the browser window and its function as a navigation frame.
3. You can resize the frame based on Percent, Inches, or Relative (that is, proportionately). For example, in percentage mode, you can define the current frame as 30%. Word resizes this window to take up 30% of the screen and automatically resizes the second frame to take up 70%. Using Inches simply resizes the frame to whatever width you enter. If you choose Relative, then you are defining each frame in proportion to each other. A setting of one means a one-to-one (1:1) proportion between frames or each frame will fill half the screen. A setting of 2 (2:1 proportion) means the first framez will be twice as large as the second or fill two-thirds the width of the frame.

The Borders tab of the Frame Properties dialog box, shown in Figure 17, enables you to modify the properties of the frame border.

Figure 17
Changing Frame Border properties.



Frame Properties

Frame | Borders

Frames page

○ No borders

● Show all frame borders

Width of border: 4.5 pt

Border color: Automatic

Individual frame

Show scrollbars in browser: If needed

☑ Frame is resizable in browser

1. You can choose No Borders to turn off border display or Show All Frame Borders to leave the border visible. The small representation of the frameset on the right updates itself based on your selection.

Tip from
Michael Larson

It is often more attractive to turn off the border display. But do this only if you are absolutely certain your frameset is readable under a wide variety of conditions. Having an attractive page is of little value if a visitor can't read it.

2. For displayed borders, you can change the color of the border and its thickness in point size.
3. In the lower part of the Borders tab, you can decide what to do about scrollbars. Scrollbars can be always shown, always hidden, or shown as needed.

Caution

Be cautious about turning off scrollbars completely. Even though the contents of one frame may not change (your navigation frame, for instance), it may not display the same in every browser, especially if you use text. Another browser may choose a larger default font. If you don't enable the scrollbars, your visitor won't be able to read the contents of your frame. Also visitors using laptops or mobile computing devices with small screens may need the scrollbars just to view the entire content of a frame.

4. The Borders tab has a Frame Is Resizable in Browser check box. If this is checked, users can grab the frame border and resize the frames from within their browsers. If this is unchecked, then the frame size is fixed and unchangeable. You may want to leave frames resizable so that your visitors can move them around for a better view, if necessary.

SAVING FRAMES

Each of the files in a frameset must be saved as a separate Web page. To save the container page, use File, Save as Web Page from the menu. To save each frame page, right-click in any blank area of the frame and choose Save Current Frame As.

If you later decide that you don't want one or more frames, you can select a frame in Word by clicking in it and then choose the Delete Frame button from the Frames toolbar. That frame disappears, although its file is unaffected if it was saved. The Delete Frame option also shows up on the Format Frames menu when a frameset is open.

BUILDING A TABLE OF CONTENTS FOR FRAMES

You can automatically build a table of contents for your set of frames. This works effectively only if you have built your Web page using heading styles (Headers 1 through 9) to separate sections of text. This feature can be accessed from the Frames toolbar or from the menu under Format, Frames, Table of Contents in Frame.

Note

Adding a table of contents in a frame adds an additional frame to your frameset. So if you have two frames before you begin building the table of contents, you will have three after the operation. If you just want two frames, a left frame of hyperlinks and a main frame containing documents, open your primary Web page containing the headers and add the table of contents by clicking the appropriate button or choosing the menu command.

The Table of Contents in Frame feature builds a navigation frame with hyperlinks to each of your headers. It is handy for building a simple navigation system for lengthy Web pages, with a hyperlink to each header on the page. The Table of Contents in Frame feature does not work across multiple Web pages. In other words, you can't automatically build a table of contents for more than one Web page.

Tip from
Michael Larson

The table of contents built for a given frame is formatted with the default style. If your Web site uses a theme, you need to apply the theme to the Table of Contents page by choosing Format, Theme from the menu and selecting your site theme.

Adding Images, Background Images, and Background Color

Besides text, images are the second most common element found on typical Web pages. When used correctly, images can reinforce text messages, add a dash of color, or add content not possible with text. Images can also be used as page backgrounds to make text more readable and reduce the glare common with a white background on monitors. The background color of the page can also be changed to reduce glare or add more interest to the page.

Adding Graphical Images to a Web Page

Adding a picture to a Web page in Word is no different from adding an image to a Word document.

1. Place your cursor where you want to place the graphic.
2. Choose Insert, Picture and the location of an existing graphic or tools to create a graphic (WordArt, for example) from the menu.
3. Choose or create your image and click OK to insert it into the page.

After the graphic is placed, you can resize it by clicking on it, grabbing the bounding boxes, and dragging the image larger or smaller.

Tip from
Michael Larson

Bigger is not always better for images. Larger images can dramatically increase the load time for your Web page if your audience only has access to low bandwidth, such as a modem. Make the image only as large as needed a Web page to make your point.

➔For more information on working with images in Word, **see** Chapter 14, "Getting Images into Your Documents," **p. 437.**

USING SMALLER IMAGES TO CREATE A BACKGROUND PAGE TEXTURE

A small image repeated or *tiled* across the background of a Web page can give the illusion of texture, such as paper, cloth, stone, wood, and so one. To add a background texture to your Web page in Word, do the following:

1. Choose Format, Background, Fill Effects from the menu. This brings up the Fill Effects dialog box.
2. Click the Texture tab to bring up the default list of textures as shown in Figure 18.

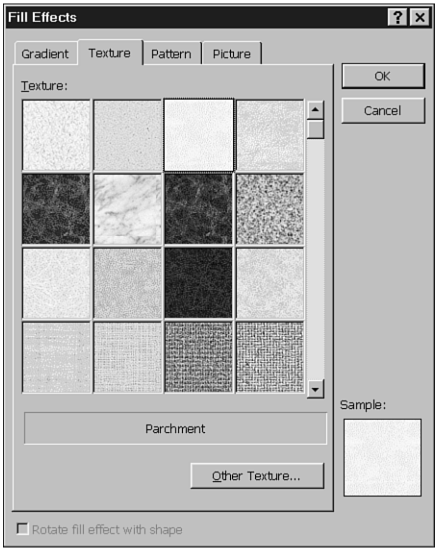


Figure 18
Accessing the background texture graphics in Word.

3. Scroll through the list until you find the texture you want. Click on it and click OK. The texture will now fill the background of your Web page.

Whenever you use a background texture, make sure that it contrasts with your font color. Dark text is best read against light backgrounds and vice versa. Also, background textures that contain too much definition or sharp patterns can make the page text difficult to read.

Tip from
Michael Larson

The texture image files are specially designed to seamlessly tile—that is, tile without showing any boundaries between each image—across your Web page. It is often best to use images specially designed to be background textures; otherwise, the boundary between each tile becomes visible and spoils the illusion.

Adding Background Color to a Web Page

In addition to background images, you can also add a background color to a Web page.

1. With your Web page open, choose Format, Background from the menu.
2. You can choose one of the colors displayed or click More Colors to bring up more options.
3. After you've found a suitable color, click OK to apply it to your Web page background.

As with textures, you want your background color to have good contrast with your font color.

Tip from
Michael Larson

Generally, it is a good idea to limit yourself to using about three colors. Your background color should be one that shows your text with good contrast. You should then reserve the second color for your main text color. You can use the third color to add accents and highlight those things you want to grab the audience's attention. Using too many colors is distracting to the eye and detracts from the overall message. This would be a good time to use the company colors if they fit the profile. If the colors are difficult to work with, you can use just one or two of the colors and use a white background.

➔ For more information on working with background colors and fills in Word, **see** Chapter 14, "Getting Images into Your Documents," **p. 437.**

Using Special Graphics Formats

The most common graphics file formats used in Web pages are GIF (Graphics Interchange Format) and JPG (Joint Photographic Experts Group, also JPEG). Word automatically exports all images to these two formats when you save as a filtered Web page.

Note

As of the writing of this book, the new JPG2000 file format has been completed and approved. This format uses wavelet compression to greatly compress images—that is, reduce image file size—with minimal loss of image quality. Expect tools for this new format to appear in 2001. Currently, no browsers can read or display this new image format, though this should quickly change.

Word also supports the display of two additional graphics file formats in Web pages: VML (Vector Markup Language) and PNG (Portable Network Graphics).

Using VML Graphics

Vector images are defined by equations. As such, they scale perfectly to any size. This is in contrast to bitmap images, where each pixel has a defined position and color value. Bitmap images scale poorly because the graphics program must interpolate pixels as the image dimensions are changed. Items created in Word using the Drawing tools are drawn as vector objects. When you save as a Web page or a Web archive from Word, the graphical object is defined by the VML language. The primary advantage of using VML is economy of size, especially if you're using large images. But a significant disadvantage is that vector objects can only be displayed by Internet Explorer version 5.0 or later.

To toggle the use of VML in Web pages, go to Tools, Options; click the General tab; then click the Web Options button. Under the Browsers tab, Rely On VML for Displaying Graphics in Browsers may or may not be checked depending on the target browser version selected.

Using the New PNG File Format

The specification for the PNG format has been in place for several years. The idea behind this format is to solve the primary weaknesses of .gif and .jpg files: GIF can support only 256 colors, and JPG gains its small file size using a lossy compression scheme, that is, as you make your file smaller, you lose photo clarity and resolution as image data is discarded. Also GIF supports transparency and animation, but JPG does not. PNG supports 24-bit color, supports transparency, and uses a file compression scheme that does not reduce the file size at the expense of image clarity. The main reason that this format is not widely used now is because older browsers cannot read the PNG format.

To toggle the use of PNG in Web pages, go to Tools, Options; click the General tab; then click the Web Options button. Under the Browsers tab, Allow PNG as a Graphics Format may or may not be checked depending on the target browser version selected.

The Web Page Palette

When you are working with a video card displaying 24-bit color, all RGB (Red-Green-Blue) combinations are available for use. In the Web world, however, many browsers use a 216-color palette called the "Web" or "Netscape" palette. Whenever possible, limit your .gif images to these 216 colors for optimal display.

Note

You cannot change the palette in .jpg files; this file format is 24-bit color by definition.

Pixels present in a .gif image that are not one of the 216 colors present in the palette are dithered. *Dithering* is where the graphics program makes a best guess about what color to assign a pixel color not contained in the image palette. Because you want optimal control over the appearance of your image, stick with the 216-color palette rather than depending on a dithering algorithm. Many modern paint or graphics programs have the Web palette built into them. You can find a table listing all 216 colors of the Web palette and their hex and decimal equivalents at `http://www.lynda.com/hexv.html`.

WORKING WITH LINK BARS

Word 2002 now includes special elements to add more functionality or convenience of use to an individual Web page or even the entire Web site. Microsoft has named these elements *Web components*. Each Web component generally has a single, specialized function that otherwise cannot easily be added to the page.

Note

For those of you familiar with FrontPage, Web components have similar functionality to some components such as Pivot Table, Chart, and Hover Button.

Web components require support from the Web server to function. In other words, a Web component on a Web page is useless and will not function unless the Web server has the proper extensions installed. These extensions are called Office Web Server (OWS). OWS extensions, also called the next generation FrontPage Extensions, at this time run only on the Internet Information Server (IIS), Microsoft's Web server. OWS ships with the Office XP suite of software applications.

Note

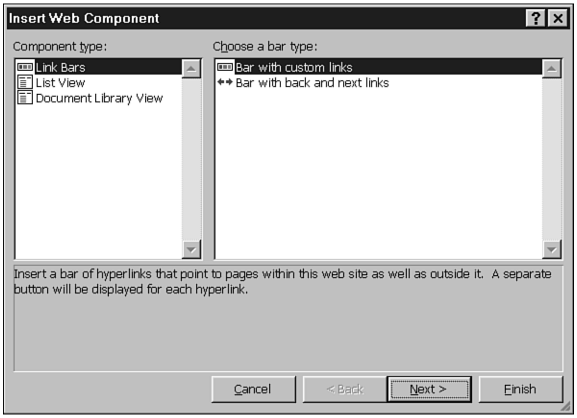
In the past, Microsoft has ported FrontPage extensions to other Web servers. If you or your business use a different Web server, check with Microsoft to see whether OWS is available for your server. Another alternative is to purchase Web hosting services with companies already running OWS on IIS.

For any Web site, building and maintaining a navigation system is a time-consuming task, especially as the number of Web pages on the site grows. The Link Bar Web component can, however, greatly ease this routine maintenance task. The Link Bar Web component lists all the Web pages in a site and adds the Web page titles as hyperlinks to a list. You can then choose which pages to include in the Link Bar, the format of the links, and where to place the Link Bar on the Web page. The Link Bar component then automatically builds the navigation bar and its links and places it into your Web page. Anytime you need to change the navigation system on your Web page, just edit the Link Bar. You can easily change your Link Bar from text to graphical links and choose from all of the Web page themes available to Word. And you don't have to write or edit a single HTML tag.

To build and customize a Link Bar for your Web page:

1. Open the Web page where you want to place the Link Bar.
2. Immediately save your Web page to a Web server with OWS installed and active.
3. From the menu, choose Insert, Web Component.
4. From the Insert Web Component dialog box, click on Link Bar in the left-hand window.
5. The right-hand window displays the two styles available: Bar with Custom Links and Bar with Back and Next Links (see Figure 19). The bar with custom links is more useful for navigating a Web site. The bar with back and next links is more useful if you want to set up a presentation or tutorial. Choose a style and click Next.

Figure 19
Choosing a Link Bar presentation style.



Insert a bar of hyperlinks that point to pages within this web site as well as outside it. A separate button will be displayed for each hyperlink.

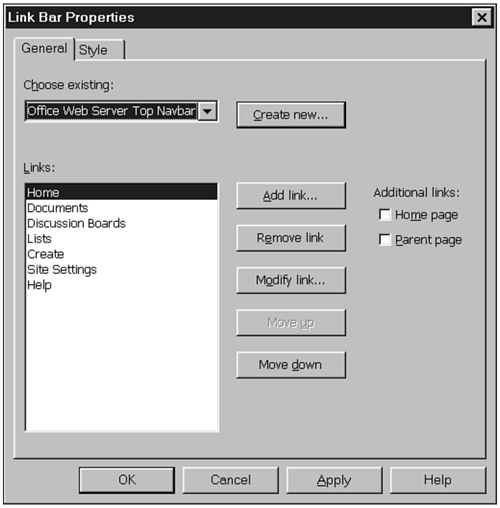
6. Choose a theme for your Link Bar. If your Web page already incorporates a theme, then choose Use Page's Theme and click Next.
7. Choose the orientation for your Link Bar, whether to place your list of links across the top of the page or down the left side. Click Finish.

The Link Bar is inserted into your Web page, and you immediately are prompted to customize the Link Bar.

1. Use the Create New Link Bar dialog box that pops up after you insert the Link Bar into your Web page to name the Link Bar and click OK.

   The Link Bar Properties, General tab lists all the pages by title in the Links window for the Web site folder containing the Web page you are working on (see Figure 20).

Figure 20
Choosing links and their order for the Link Bar.



Link Bar Properties

General | Style

Choose existing: Office Web Server Top Navbar

Links:
Home
Documents
Discussion Boards
Lists
Create
Site Settings
Help

Additional links:
☐ Home page
☐ Parent page

Tip from
Michael Larson

The text on each Link Bar button is taken from the Web page title. The Web page title can be manually changed after the Link Bar is generated if you don't like the page title name as the text for a Link Bar button. Future updates of the Link Bar will not alter your modifications.

You can add, delete, or modify links. You can also change the order in which the links are displayed by moving them up or down the Links window. If the Web site Home page (usually index.htm) is not listed, you can add a link to the Link Bar by selecting the Home Page check box. You can also choose to link back to a parent page that links to your open Web page.

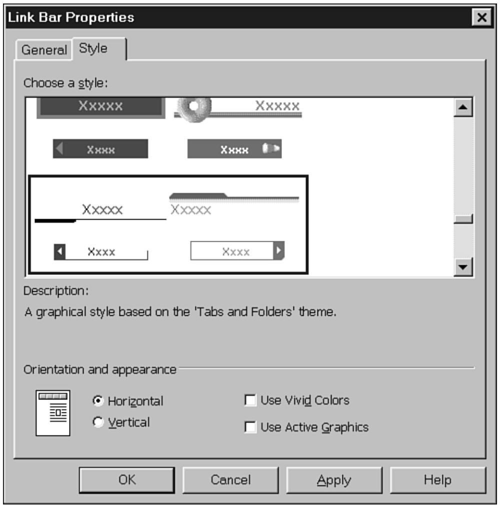
2. To add a link to the Link Bar, click the Add Link button. From the Add to Link Bar dialog box, you can browse to other Web pages on the Web server or add a link to an e-mail address. Click OK to add the link to the Link window and thus to your Link Bar.
3. Click on the Style tab (see Figure 21) to change the Link Bar style and orientation.

Tip from
Michael Larson

Keep the theme consistent between all the pages on your Web site to give users a visual cue that they are still located on your site.

4. When you've finished customizing your Link Bar, click OK. Word automatically builds the hyperlinks and places the generated Link Bar on your Web page (see Figure 22). Save your Web page.

Figure 21
Changing the style and orientation of the Link Bar.



Link Bar Properties

Choose a style:

Description:
A graphical style based on the 'Tabs and Folders' theme.

Orientation and appearance

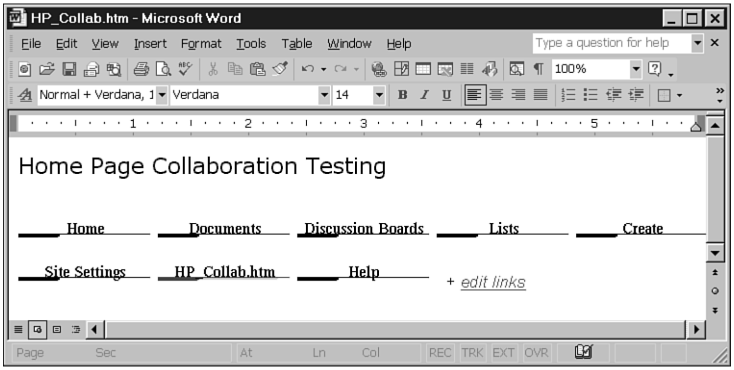
● Horizontal
○ Vertical

□ Use Vivid Colors
□ Use Active Graphics

Note

The hyperlinked "edit links" text appended to the end of the generated Link Bar in Word will not show up when your Web page is viewed with a browser. Clicking "edit links" brings up the Link Bars Properties dialog box for further editing when needed.

Figure 22
A generated Link Bar in Word.



HP_Collab.htm - Microsoft Word

Home Page Collaboration Testing

Home Documents Discussion Boards Lists Create

Site Settings HP_Collab.htm Help + edit links

CREATING YOUR OWN WEB PAGE TEMPLATES

If you don't like any of the Web templates that come with Office or need a special template for your organization, you can easily build your own.

To build a new Web page template, do the following:

1. From the New Document window, choose General Templates to bring up the Templates dialog box.
2. Click Create New Template in the bottom-right corner.

3. Open an existing template or a blank Web page, whichever suits your needs best.
4. Save your template immediately.

Note

Notice that when you choose File, Save As from the menu, Word prompts you to save the template as a .dot file. You can use this format or save a template as a Web page, .htm, file.

Your template is saved to the path defined by Tools, Options, File Locations. The default for this is the Windows/Application Data/Microsoft/Templates folder on the drive where Word 2002 is installed.

This path, of course, assumes that the default installation path for Word 2002 was used. This template will now show up as a Web template in the list of Web page templates in the Templates dialog box.

5. Add whatever items you need to make a basic template. These might include
   - A company logo
   - A background texture or color
   - Font colors and styles
   - A basic text outline
   - Standard hyperlinks, such as one to your Web site home page

   For instance, if your company has established colors and fonts, you could use those in your Web page template. You could use a company color as a background color to your Web pages. Your legal department might require you to include a hyperlink to a disclaimer in small text at the bottom of every page. These are the types of elements to include in a template to maintain consistency through all the Web pages on your site.
6. Save the template again with the items you just added.

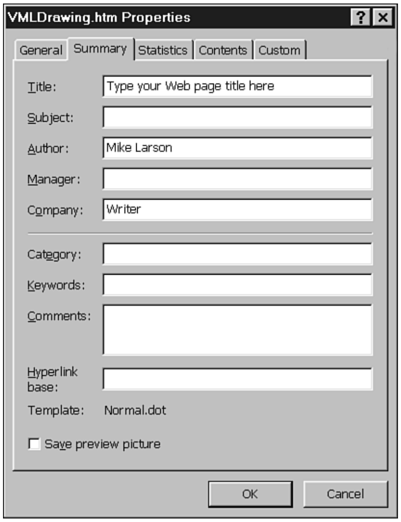
THE IMPORTANCE OF THE WEB PAGE TITLE

When you create a Word document, you can consider giving the page a title for ease of searching. When you create a Web page, the title is even more important for the following reasons:

- Whenever you open a Web page in a browser, the title of the page is displayed across the top title bar of the browser. This is an important visual aid to your visitors because it tells them what site they're on and what part of that site they're currently in.
- The search engines on the Internet (Lycos, Alta Vista, Excite, and so forth) often use the information in the title to index a Web page. Some search engines use the title exclusively to index the contents of a Web page. If your Web page has no title or an unremarkable title, people will have a difficult time finding your Web page via search engines. This phenomenon is also true on intranets if a local search engine is used.

Word uses the document's title (from the document properties) as the HTML title. You enter the title for a Web page by selecting File, Properties from the menu, as shown in Figure 23.

Figure 23
Adding a title to a Web page.



Title: Type your Web page title here
Subject:
Author: Mike Larson
Manager:
Company: Writer
Category:
Keywords:
Comments:
Hyperlink base:
Template: Normal.dot
□ Save preview picture

Notice that you can also enter keywords here. Search engines also use keywords when indexing files.

The Web Page Wizard

When you've began building individual Web pages, your next inclination might be to build a group of related pages or a Web site. Word provides the Web Page Wizard to conveniently step you through the process.

To open the Web Page Wizard, click on General Templates in the New Document task pane. From the tabbed Templates box that appears, choose the Web Pages tab and the Web Page Wizard is one of the selections.

After you open the wizard and click past the brief introductory screen, you can start building your group of Web pages.

1. The next screen asks you to name your Web site and the location (a Browse button is available) where you want to save your files. Note that the Web site name is used to create a subfolder of the same name for file storage. You can save your set of Web pages anywhere: on your hard drive, a network drive, a Web Folder, or a Web server if you have the proper permissions.
2. From the next screen in the wizard, you can choose what type of navigation system to use on your site: a vertical frameset (small left frame with links to the larger right frame), a horizontal frameset (small top frame with links to the larger lower frame), or placing links on a separate, unframed page.

3. On the next screen, you list the pages you want to include in your set of Web pages. You can add blank Web pages, existing files (which can also be non-Web pages), or template files. When you choose to add a template file, a blank template is actually displayed on the screen for you to inspect. You can also remove files from your set from this screen.
4. You can specify in what order you want your links to appear in your navigation system from the next screen. The wizard automatically builds a named link to each page and places it in the smaller vertical or horizontal frame (see step 2) or on a separate page. You use the Move up or Move Down buttons to organize your list of Web pages. You can also rename individual pages from here.
5. From the next to last page in the wizard, you can decide whether to add a visual theme to your set of Web pages and choose from the list of available themes. The wizard applies the same theme to all Web pages in your set. Non-Web pages are not themed. You also have the option of choosing no visual theme.

From the last page of the wizard, click Finish. The wizard creates blank and template pages and saves them to your subfolder. It also copies existing files to your subfolder. Then it creates your page of navigation links in a file named default.htm. Default.htm is loaded into Word, ready to be updated with content.

Tip from
Michael Larson

Also consider using the Web Page Wizard to link together groups of non-Web page files for easy access. For instance, you can build a report that includes an Excel spreadsheet, several Word docs, and a PowerPoint presentation and link them together using the Web Page Wizard. Then you only have to track default.htm, which you'll probably want to rename to something more meaningful, as a summary page and link to your other related files.

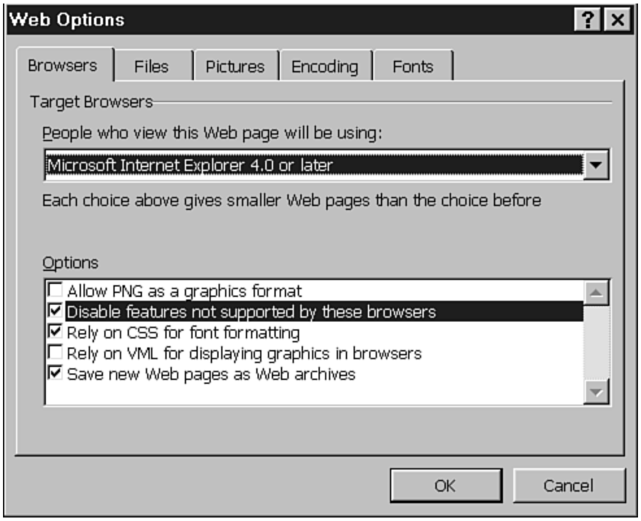
ADVANCED WEB OPTIONS

Advanced Web options enable you to change the default display of your Web pages, what languages and features are supported, and how Web pages are saved.

To access these options, from the menu, choose Tools, Options; click the General tab; and finally click the Web Options button at the bottom. The opening Web Options screen should look like Figure 24.

CUSTOMIZING THE APPEARANCE OF YOUR WEB PAGE

Under the Browsers tab for Web Options, you can set your target browser. The target browser is based on version number and runs from Internet Explorer 3 and Netscape Navigator 3 up through Internet Explorer 6. Select your default target browser based on the audience viewing your Web pages: to reach the widest audience on the Internet, use the lowest version numbers. You might choose Internet Explorer 6 as your target browser on a company intranet where everyone has standardized on the latest browser and you need these capabilities to support the content in your Web pages.



People who view this Web page will be using:

Microsoft Internet Explorer 4.0 or later

Each choice above gives smaller Web pages than the choice before

Options

- ☐ Allow PNG as a graphics format
- ☑ Disable features not supported by these browsers
- ☑ Rely on CSS for font formatting
- ☐ Rely on VML for displaying graphics in browsers
- ☑ Save new Web pages as Web archives

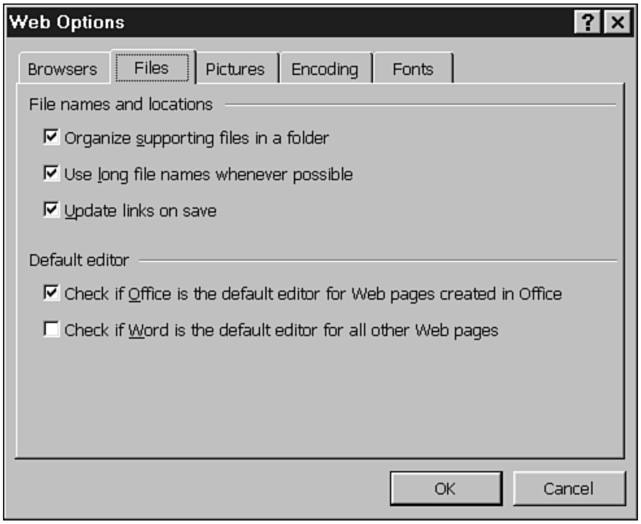
Figure 24
Setting Web page appearance using Web Options in Word 2002.

Each target browser setting enables or disables a set of supported features, including

- Allow PNG as a graphics format. This new format is not widely supported on the Internet yet but has advantages over GIF and JPEG files.
- Disable features not supported by these browsers. For instance, no VML would be used in Web pages generated by Word 2002 because version 4 browsers do not understand VML. If you clear this box, then every Web page feature built into Word 2002 is used without regard for whether any browser version can support it
- Rely on CSS for font formatting. Only version 3 browsers cannot understand CSS. I recommend leaving this on for the greatest flexibility in changing Web page formatting.
- Rely on VML for displaying graphics in browsers. VML reduces overall Web page size but browsers prior to version 5 do not understood it.
- Save new Web pages as Web archives. This format has the advantage of storing all the files in a single file. You can, of course, override this option each time you save your Web page.

Selecting Web Page File Options

From the Files tab of the Web Options dialog box, shown in Figure 25, you can change some filename options and make choices about Word 2002 being your default Web page editor.



File names and locations

☑ Organize supporting files in a folder

☑ Use long file names whenever possible

☑ Update links on save

Default editor

☑ Check if Office is the default editor for Web pages created in Office

☐ Check if Word is the default editor for all other Web pages

Figure 25
Setting filenames, locations, and default editor options using Web Options in Word 2002.

The first check box in the File Names and Locations section asks whether you want to organize supporting files in a folder. When Word 2002 saves a Web page, it sends many (though not all) supporting files—such as graphics—to a separate folder. This box should be unchecked only if you already have a set of folders set up with your Web page supporting files.

The Use Long Filenames Whenever Possible check box is checked by default. The only operating system that does not support long filenames is DOS (with or without Windows 3.x). Unless you have a large number of people using this operating system, leave this check box checked.

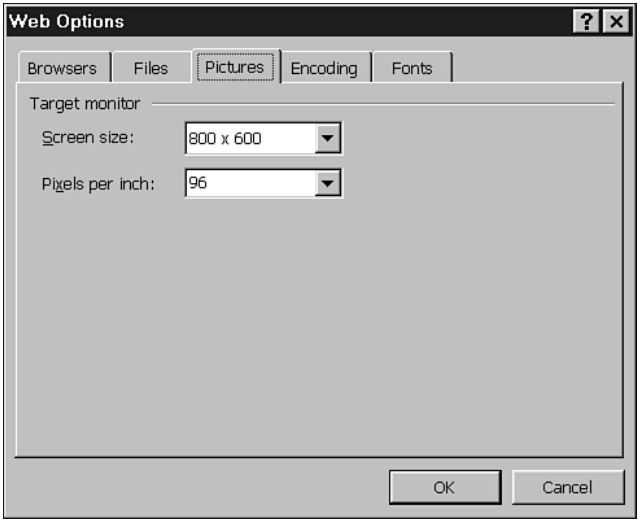
The final check box in this section, Update Links On Save, updates links to supporting graphics and components in your Web page. It does not update or check hyperlinks.

The Default Editor portion of the Files tab under Web Options enables you to decide whether you want

- Office to be the default editor for Web pages created in Office (checked by default).
- Word to be the default editor for all Web pages (not checked by default).

CHANGING PICTURE SIZE

The Pictures tab, shown in Figure 26, defines the target monitor you want for your Web page.



Screen size: 800 x 600

Pixels per inch: 96

Figure 26
Choosing your target monitor size and resolution for display of your Web pages.

The target monitor refers to the screen resolution you want to optimize your Web pages for. Screen resolution is expressed in pixels, usually as width × height. The most common screen resolutions in use today on PCs are

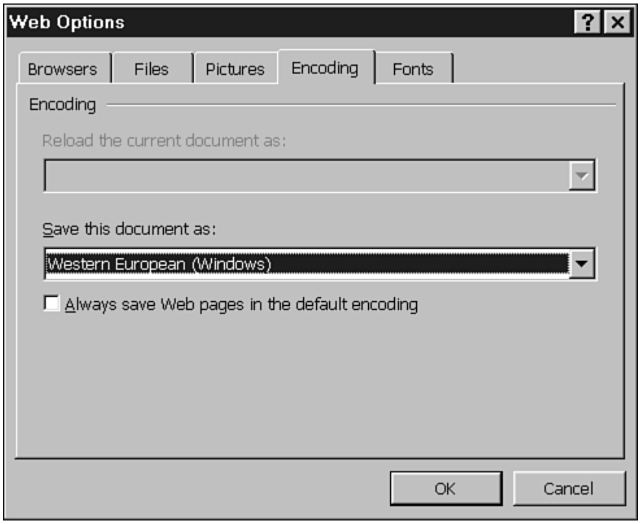
- 800×600
- 1024×768
- 1280×1024

The larger the number, the more pixels and thus more information are displayed on the screen. If you choose a target monitor size of 800×600 for building your Web pages in Word and view the resultant page at 1280×1024, much of the screen will be empty space with most of the information crowded to the left side of the screen. On the other hand, if you design your pages at 1280×1024 and view it at 800×600, you will have to endlessly scroll to the right to see all the information. The default of 800×600 is suitable for most uses, unless you're sure that the majority of your audience uses other screen resolutions.

You can also change the pixels per inch of your target monitor. Again the default of 96 is suitable for most uses. Using higher values greatly increases the size of your graphics and increases your Web page load time. Using a value of 120 slightly increases the detail in your Web page. A value of 72 gives you smaller Web graphics, but your Web page will have a slightly coarser appearance.

CHANGING LANGUAGE ENCODING

The Encoding tab in the Web Options dialog box, (see Figure 27), enables you to choose the language code page from those installed on your machine. Choose the appropriate code page for the language you are using to build your Web page.



Reload the current document as:

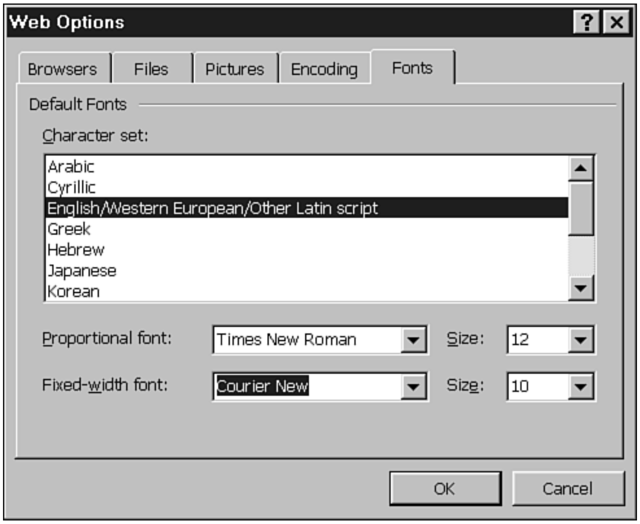
Save this document as: Western European (Windows)

☐ Always save Web pages in the default encoding

Figure 27 Choosing the language code page using Web Options.

CHANGING THE DEFAULT FONTS

You can set the default proportional and fixed-width fonts for your Web page from this next Web option as shown in Figure 28.



Character set:

- Arabic
- Cyrillic
- English/Western European/Other Latin script
- Greek
- Hebrew
- Japanese
- Korean

Proportional font: Times New Roman Size: 12

Fixed-width font: Courier New Size: 10

Figure 28 Changing the default font for your Web page.

Use common fonts for your defaults. If you use fonts in your Web pages that aren't installed on your viewers' PCs, their browsers can't render your fonts and will substitute their own default fonts.

TROUBLESHOOTING

HOW DO I SIMPLIFY MY WEB PAGES AND ELIMINATE EXTRA FILES?

A Web page saved from Word contains many complicated scripts, XML, and several extra files. You want to simplify this file and eliminate the extra files so that you don't have to keep track of them anymore.

To eliminate the complicated scripting and XML, you need to save your document as a filtered Web page:

1. With your document open in Word, choose File, Save As from the menu.
2. From the Save as Type list at the bottom, choose Web Page, Filtered.
3. Save your document.

To eliminate the extra files, save your document as a Web page archive:

1. With your document open in Word, choose File, Save As from the menu.
2. From the Save as Type list at the bottom, choose Web Archive.
3. Save your document.

There is no single file format available to simplify the HTML and eliminate the extra files associated with Web pages.

WHAT TO DO IF YOU NEED TO REARRANGE FRAMESETS

You have a frameset composed of a smaller page on the left and a larger, main window to the right. You want to move the left frame to the top so that you will have a smaller, top frame and a larger, main frame below it. How do you do this?

With your frameset open in Word

1. Right-click in the left frame and choose Frame Properties. Make a note of the filename (it will usually have a .htm extension) listed in the Initial Page combo box. Cancel out of the Frame Properties dialog box.
2. Open the Frames toolbar.
3. Click in the main window (right frame). From the Frames toolbar, click New Frame Above to place a new frame above the main frame.
4. Click in the smaller, left window. From the Frames toolbar, click Delete Frame. The selected frame disappears.

   You should now have a two-frame window with an upper, empty frame and the lower main frame.
5. Right-click in the upper, empty frame and choose Frame Properties.
6. Type the filename from step 1 (or Browse to it) into the Initial Page combo box. Click OK.

The upper, empty frame should now contain the content formerly seen in the left frame.

7. Adjust the borders between the frames by dragging or using Frame Properties.

What to Do if You Are Unable to Add a Table of Contents

You have a two-page frameset. You want to have Word build an automatic table of contents. You added the content you wanted to index in the main frame. In the smaller frame, you clicked Table of Contents in Frame on the Frames toolbar. The message `Error! No table of contents entries found` *showed up. What happened?*

The Table of Contents (TOC) is built based on any text contained in heading tags. To fix this,

1. Go back to your main page, select the text you want to see in your TOC and apply a heading (Heading 1, Heading 2, and so on) style from the Style list box on the Formatting toolbar.
2. After you reapply the styles, click in the smaller frame and rebuild your TOC.

What to Do to Save Manual HTML Code Changes from Word

I made some changes to the HTML source in the Script Editor and saved it. I continued to work on the Web page in Word and saved the file. When I opened the Web page later, all my changes were gone. How can I keep my changes?

Word overwrites any changes you make as it regenerates the HTML source every time it opens the document. To avoid losing your changes, create them last. After saving your Web page in Word, do not open the page again in Word. If you must make further changes, use a text editor, such as Notepad, or an HTML editor application, such as FrontPage.

BONUS WORD CHAPTER

Adding Interactivity with Hyperlinks, Web Forms, and Smart Tags

In this chapter

by Michael Larson

www quehelp com

Special Edition Using

que®

EXPLAINING HYPERLINKS

By now, you've surely clicked on hundreds, if not thousands of hyperlinks in the course of your work (or leisure) on the Web. Usually, when you click a hyperlink, you "go" somewhere; that is, another Web page loads into your browser. Did you ever wonder what was actually happening when you clicked that link?

Behind the scenes, in its most basic form, a *hyperlink* is an HTML tag containing some display text (or image) and an address. The HTML code for a hyperlink to Microsoft's home page would look like this:

```
<a href="http://www.microsoft.com">Microsoft's Home Page</a>
```

The hyperlink tag, also called an *anchor*, begins with `<a href`, which tells the browser that a hyperlink is starting. The address, `http://www.microsoft.com`, is specified next. An angle bracket, `>`, ties off the end of the address. Any text between this angle bracket and the closing anchor tag, `</a>`, is the display text. So the browser would display the text `Microsoft's Home Page` as a hyperlink, usually as blue, underlined text. Clicking on the hyperlink loads the address into your browser, and the Microsoft home Web page is loaded.

Fortunately, you don't have to know HTML to build hyperlinks in Word 2002. Understanding the basic parts of a hyperlink—the address and the display text—will, |however, help you more easily use the hyperlink capabilities in Word 2002.

CONNECTING WITH HYPERLINKS

The most powerful aspect of hyperlinks is the ease with which they enable you to connect to information that has already been created. You don't have to re-create the information or even get the information and place it into your document. You just need to build a link. This leads to all kinds of terrific synergy. For instance, if you read a detailed article online, you may include a summary of the article in your Web page with a link to the complete article. You can link to topically related multimedia files (pictures, sounds, video, animations, and so forth) that may be available.

You can build links to information as it exists in other file formats. For instance, you can include a link to content in a Word document, an Excel spreadsheet, or a PowerPoint presentation. You can also link to any non-Office document.

Caution

Be aware that people on other platforms may not be able to read documents in formats other than HTML. For instance, Unix users cannot read Office documents. If possible, always provide your information as Web pages and offer other file formats as a convenience for those who can use them. Make sure to provide clues in your text when you are linking to non-HTML pages to warn people that the link leads to another file format.

An online appendix can now be a list of hyperlinks to Web sites or other parts of the intranet that contain information related to your topic. You can even build entire Web sites that are nothing but lists of links, though you must be careful to organize the lists for ease of navigation. One major Internet search site, Yahoo!, is indeed primarily an extremely well organized and complete list of hyperlinks. Because of how well it is organized, Yahoo! routinely ranks among the ten most-visited Internet sites.

The best way to learn about the power of hyperlinks is to learn to build them and start using them in your Web pages or Word documents. The following sections show you how to do just that.

BUILDING HYPERLINKS

As mentioned earlier, hyperlinks are primarily composed of two parts: the display text or image (on which you actually click) and the network address or *URL* (Uniform Resource Locator, just fancy jargon for an Internet address).

Word offers several ways to build a hyperlink. Follow these steps to insert a text hyperlink into a Word document:

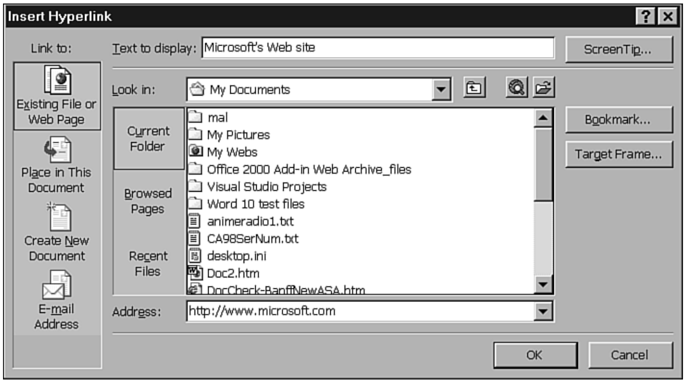
1. In a blank Word document, type the following: `More information can be found at Microsoft's Web site.`
2. Use your mouse to select the words "Microsoft's Web site." This will be the display text for your hyperlink.

To build the hyperlink, choose from any of these options:

- Click the Insert Hyperlink button (a globe with a chain link) on the Standard toolbar.
- Use the keyboard shortcut, Ctrl+K.
- Choose Insert, Hyperlink from the menu.
- Right-click the selected text and choose Hyperlink from the context menu.

The Insert Hyperlink dialog box appears, as shown in Figure 1.

Figure 1
You have many different ways to add an address to a link in the Insert Hyperlink dialog box.



Text to display: Microsoft's Web site

Look in: My Documents

Address: http://www.microsoft.com

4. In the Text to Display box at the top of the dialog box you'll whatever text you had selected when choosing to insert the hyperlink. In this case you should see, "Microsoft's Web site."
5. In the Address box, enter the URL for the Web site, namely `http://www.microsoft.com`.
6. Click OK at the bottom, and the text "Microsoft's Web site" should now be blue and underlined (the default style for hyperlinks) denoting a hyperlink in your document.

That's all there is to building a basic text hyperlink.

If you test your link and it doesn't work, see "What to Do if Your Hyperlink Does Not Load" in "Troubleshooting" at the end of this chapter.

If you're not sure of the Internet address to which you want to link, just click the Web Page icon on the right side of the screen. This opens your default browser and enables you to browse to the Web page of your choice. Closing the browser inserts the URL into the Address box of the Insert Hyperlink dialog box.

Of course, you probably noticed many more options in the Insert Hyperlink dialog box. Let's go over each of the available options in more detail.

As you could see in Figure 1, there are four buttons running down the left side of the Insert Hyperlink dialog box:

- Existing File or Web Page
- Place in This Document
- Create New Document
- E-mail Address

Choosing any one of these selections changes the information displayed to the right of it.

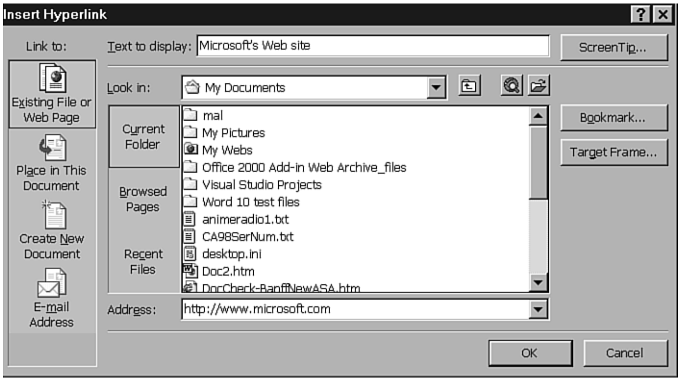
Linking to an Existing File or Web Page

The default selection, Existing File or Web Page, displays the information shown in Figure 1. You can type in a URL or click the down arrow to choose one from the list below it. Three selection buttons immediately to the left of the list control the contents of this list:

- Current Folder
- Browsed Pages
- Recent Files

Current Folder (the default) shows the contents of your current folder and lets you browse to any place on your hard drive or networked drives or any place accessible from your computer directory tree.

You can also choose to hyperlink to pages by clicking the next button down the list, labeled Browsed Pages, which displays the titles of all the HTML pages you have recently browsed, as shown in Figure 2.



Text to display: Microsoft's Web site

Look in: My Documents

mal
My Pictures
My Webs
Office 2000 Add-in Web Archive_files
Visual Studio Projects
Word 10 test files
animeradio1.txt
CA98SerNum.txt
desktop.ini
Doc2.htm
DocCheck-BanffNewASA.htm

Address: http://www.microsoft.com

Figure 2
Building hyperlinks to Web pages and Help files you have recently browsed.

Notice that the list displays mostly the Web page titles, only using the actual Internet address if a name is not available. Selecting a title displays the address in the Address Name box. Also notice that these Web pages can be

- Local files
- Web pages on the Internet
- Help pages

Tip from
Michael Larson

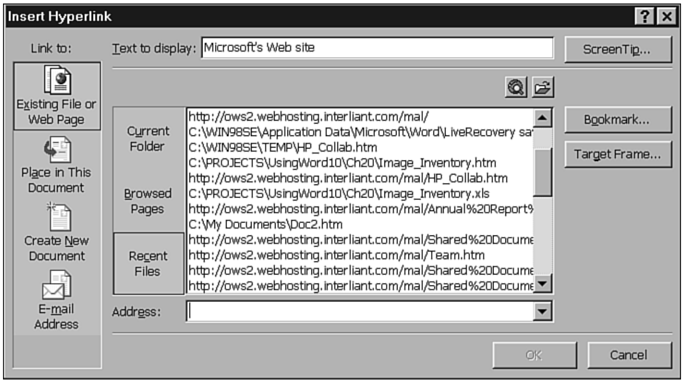
If you routinely access certain parts of online Help, building a list of hyperlinks to those parts in a separate document might considerably ease the process of opening your Help topic. You can also e-mail a document containing the hyperlink to colleagues, and it will work on their systems (assuming they also have Word 2002 installed). Note, however, that not all Microsoft Office–related Help is installed by default, especially Help for VBA (Visual Basic for Applications, the programming language in Office XP). Be careful to consider whether your intended user of the links will have access to the file(s) you link to.

Clicking the Recent Files button shows a list of the most recently accessed files on your computer, as shown in Figure 3.

This list includes files you have recently created or edited on your computer. Clicking on a filename in the box displays it in the Address box.

Caution

Building a hyperlink to a file on your local hard drive may be useful only to you. Remember that if you e-mail a document containing a local hyperlink to anyone, or use this document with the local hyperlink anywhere else, the hyperlink will not lead you or anyone else back to the file on your hard drive. If you're going to use the document containing the local hyperlink anywhere else, it's better to build the link to a file on a shared network file server or make the file available on a Web site.



Text to display: Microsoft's Web site

```
http://ows2.webhosting.interliant.com/mal/
C:\WIN98SE\Application Data\Microsoft\Word\LiveRecovery sa
C:\WIN98SE\TEMP\HP_Collab.htm
C:\PROJECTS\UsingWord10\Ch20\Image_Inventory.htm
http://ows2.webhosting.interliant.com/mal/HP_Collab.htm
C:\PROJECTS\UsingWord10\Ch20\Image_Inventory.xls
http://ows2.webhosting.interliant.com/mal/Annual%20Report%
C:\My Documents\Doc2.htm
http://ows2.webhosting.interliant.com/mal/Shared%20Docume
http://ows2.webhosting.interliant.com/mal/Team.htm
http://ows2.webhosting.interliant.com/mal/Shared%20Docume
http://ows2.webhosting.interliant.com/mal/Shared%20Docume
```

Address:

Figure 3
Building hyperlinks to recently accessed files on your computer.

If the file you want to link to is not in this list, you can click on the Browse for File icon on the upper-right side of the Insert Hyperlink box. This brings up the Link to File dialog box, which resembles a standard Office XP File Open or File Save dialog box. You can use the drop-down list box at the bottom of this dialog box to select files based on a specific file type.

Tip from *Michael Larson*

Standard file operations are available in the Link to File dialog box, including deleting, renaming, printing, changing views of the file list, and working with folders.

After you have chosen your file, click OK to return to the Insert Hyperlink dialog box. This screen now displays the path to your file in the Address box.

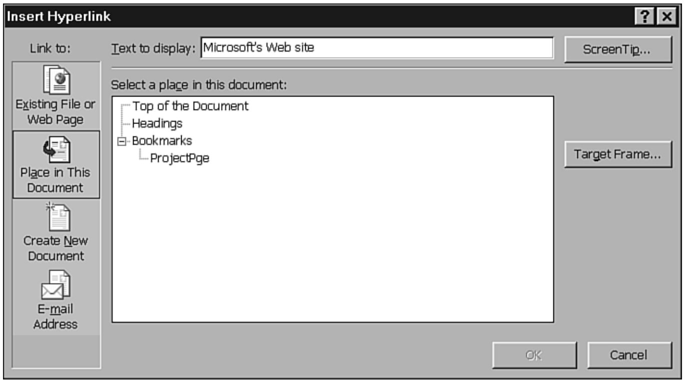
LINKING TO A LOCATION IN THE CURRENT DOCUMENT

Not only can you build hyperlinks to Web pages or documents outside your current document, you can build hyperlinks to refer to specific sections within your current document. This builds a link that enables readers to immediately jump to another section of your document. Clicking Place in This Document displays a tree structure for your current document, as shown in Figure 4.

Tip from *Michael Larson*

Before building this type of hyperlink, you do not necessarily have to select the display text. You can start by placing your cursor in the document where you want the hyperlink to be and then open the Insert Hyperlink dialog box to continue from there.

Figure 4
Building hyperlinks to bookmarks and other sections of the same document.



Text to display: Microsoft's Web site

Select a place in this document:

- Top of the Document
- Headings
- Bookmarks
  - ProjectPge

Notice that Figure 4 shows three areas inside your document to which you can build hyperlinks:

- Top of the document
- Headings (all headers in the document)
- Bookmarks (all bookmarks in the document)

Tip from
Michael Larson

You can also build a hyperlink to a bookmark or a header by clicking the Bookmark button on the right side of the Insert Hyperlink dialog box (with Existing File or Web Page selected as the Link to option). This brings up the Select a Place in this Document box that displays the same tree content as shown in Figure 4.

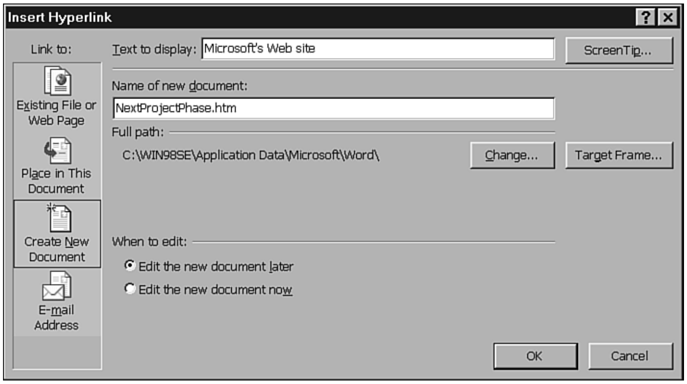
To build a hyperlink using this technique, headers and bookmarks must already be present in the document.

You can click on any of these options, then OK, and the name of the bookmark or header, preceded by the # sign, is displayed in the Address box at the bottom of the Insert Hyperlink dialog box. The # sign indicates a link to a bookmarked section in the current document. Clicking OK builds a hyperlink to that internal portion of your document.

Linking to a New Document

Use the Create New Document option to create a brand-new document during the process of creating the hyperlink. In other words, you can link to a file that doesn't yet exist. When you select this option, the Insert Hyperlink dialog box changes to look like Figure 5.

Type in the name of the new, not-yet-created file in the Name of New Document box. If you don't like the default path, click Change to bring up the Create New Document dialog box and choose the location for storing your new file. When you finish this, you can choose whether to edit your new file immediately or later and click OK to create a hyperlink to this new file.



Text to display: Microsoft's Web site

Name of new document: NextProjectPhase.htm

Full path: C:\WIN98SE\Application Data\Microsoft\Word\

When to edit:

● Edit the new document later

○ Edit the new document now

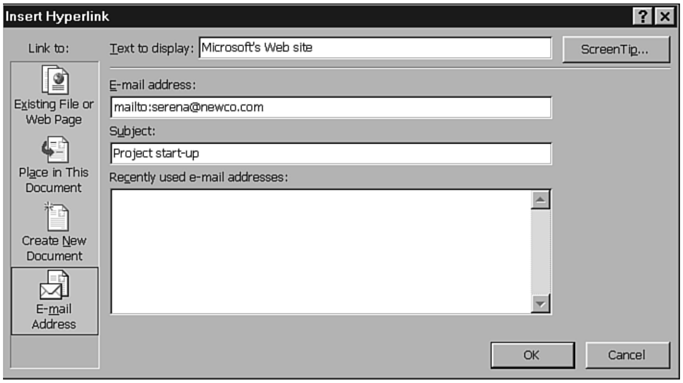
Figure 5
Hyperlinking to new, nonexistent documents.

Note

If the filename you select matches an existing file, Word will warn you that the existing file will be overwritten if you continue. If you don't want this to happen, be careful to answer 'No' to the warning.

E-MAIL ADDRESS

Several special types of hyperlinks do more than just take you to another page. One of these opens up and addresses a blank e-mail message. This is an excellent mechanism to provide visitors to your page a means of giving you feedback. You can easily build one of these e-mail hyperlinks by clicking on the E-mail Address button at the bottom of the Insert Hyperlink dialog box. Figure 6 shows the E-mail Address screen.



Text to display: Microsoft's Web site

E-mail address: mailto:serena@newco.com

Subject: Project start-up

Recently used e-mail addresses:

Figure 6
Building an e-mail hyperlink.

To build an e-mail hyperlink:

1. Type the e-mail address into the E-mail Address box. You can also choose an e-mail address from the list in the lower part of the screen, which shows any recently used e-mail addresses.

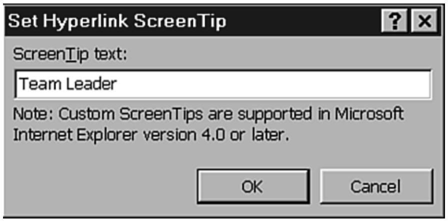
Note that the text “mailto:” is automatically entered before the e-mail address you type. The word “mailto:” is necessary to make the browser understand this is an e-mail hyperlink.

2. If you didn’t select any text before bringing up the Insert Hyperlink dialog box, the e-mail address you enter becomes the default display text in the Text to Display field. You can change the display text to be anything you choose.
3. If you want the e-mail message to have a preaddressed subject, you can type one in on the Subject line. Using a default subject line means your visitors don’t have to think one up. Also, it is a good way to insure a key phrase shows up so you can properly respond or route the email.
4. After you fill in the information you want, click OK to build the e-mail hyperlink.

INCLUDING SCREENTIPS IN HYPERLINKS

Internet Explorer 4 and later versions support ScreenTips for hyperlinks. A *ScreenTip* is a text comment that displays whenever a mouse pointer hovers over a hyperlink. This is the same idea as ToolTips, the text that displays whenever your mouse hovers over a button on a toolbar. The Insert Hyperlink dialog box enables you to add ScreenTips to your hyperlinks. Click the ScreenTip button in the upper-right corner of the Insert Hyperlink dialog box to bring up the Set Hyperlink ScreenTip dialog box shown in Figure 7.

Figure 7
Adding ScreenTips to hyperlinks.



ScreenTip text: Team Leader

Note: Custom ScreenTips are supported in Microsoft Internet Explorer version 4.0 or later.

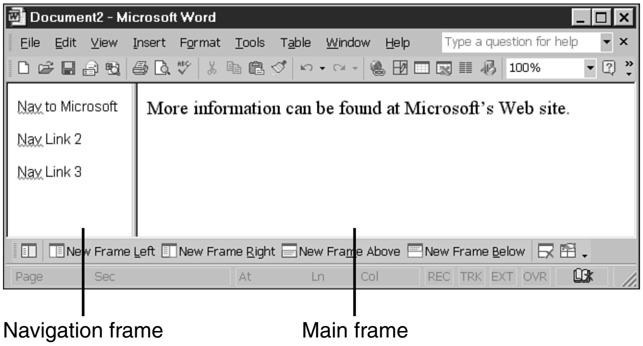
Enter the text you want to display in the ScreenTip and click OK. You can use up to 256 characters in your ScreenTip, including extended ASCII characters.

USING HYPERLINKS IN FRAMES

Often, you’ll want a link in one window of a frameset, a container page that tells the browser how to split up the screen real estate, to display a page in another, larger frame. Hyperlinks between frames require some special code to tell the browser which window of a frameset to display to the target document. Again, Word 2002 adds this special code so that you don’t have to think about it.

➔ A more detailed discussion of frames, framesets, and their properties can be found in the Bonus Word Chapter entitled, “Using Word to Develop Web Content.”

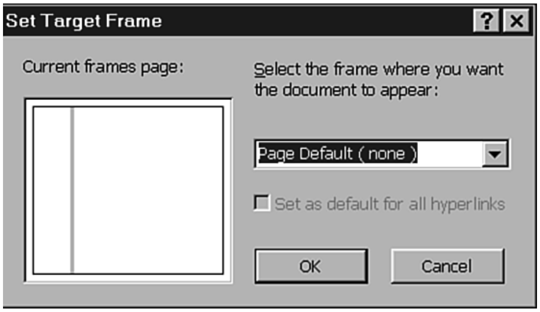
Imagine that you have a frameset containing two frames, as shown in Figure 8.



More information can be found at Microsoft's Web site.

Figure 8
An example of a frameset.

The left frame is for navigation links, whereas the right frame is the main window for displaying your Web pages. The idea is that you want to be able to click on a link in the navigation frame to load a page into the main frame. In the navigation frame, select some text and choose Insert, Hyperlink from the menu or use the keyboard shortcut, Ctrl+K. From the Insert Hyperlink dialog box, click the Target Frame button. The Set Target Frame dialog box (see Figure 9) appears, displaying a miniature representation of your current frameset.



Current frames page:

Select the frame where you want the document to appear:

Page Default (none)

☐ Set as default for all hyperlinks

Figure 9
Building hyperlinks between frames.

Clicking on a frame in the Current Frames Page section of the dialog box causes that frame's name to appear in the accompanying list box. The list also contains more selections in addition to the names of each frame displayed:

- **Page Default**—This option automatically sends the hyperlink to the default frame, if one is assigned. If none has been assigned, then the word in parentheses after the option is "none." If a frame has been assigned, the frame name appears in parentheses. Do not use this option if no page default has been assigned; clicking a link with this undefined target only opens a new instance of your browser so that you will have two open copies of your page, both displaying the same frameset.
- **Same Frame**—The destination is the same frame that the link is in. For this scenario, where we want the navigation frame to open a page in the main frame, this would be

undesirable. This option would be useful, though, for any links that might be contained in pages displayed in the main frame of this example.

- **Whole Page**—This option destroys the frameset. In other words, rather than display the page in the right frame, the page is again displayed using the entire browser window. This is useful if you need the whole browser window to display a large graphic or if you go to a part of a Web site that no longer requires the frameset for navigation.
- **New Window**—This option is similar to the Whole Page option, except it launches a second instance of your browser and displays the page in it. One copy of the browser now has the unchanged frameset in it, and the other new browser displays the page referenced in the hyperlink. This keeps your original Web page on the user's desktop even if the user browses elsewhere.
- **Parent Frame**—This final option loads the page into whichever frame is defined as the parent frame. If none is defined, this option has the same effect as the Same Frame option.

After you choose the target option for your hyperlink, you can also enable the Set as Default for all Hyperlinks check box to make your selected target the frameset's default frame. Regardless, click OK to build the hyperlink and then save your file.

Tip from
Michael Larson

It's always a good idea to test the hyperlink immediately after creating it (especially the first one built in a set of frames). Then you know that you've selected the correct options and can build your entire navigation list thereafter with confidence.

Inserting Hyperlinks Automatically

Word 2002 includes two ways to build hyperlinks automatically:

- You can type a URL, network address, or e-mail address directly into the body of your document.
- You can use cut-and-paste or drag-and-drop between documents or Office applications.

Whenever you type an e-mail address, a network path, or an Internet URL into the body of a Web page or Word document, Word makes it a hyperlink automatically. The display text is the same as the URL, network address, or e-mail address. You do not need to open the Insert Hyperlink dialog box or do anything extra to create the hyperlink. You may, however, want to edit the hyperlink to change the display text. You will learn more about editing hyperlinks in the "Editing Hyperlinks" section later in this chapter.

This automatic feature can be toggled on and off. From the menu bar, select Tools, AutoCorrect Options, and then click the AutoFormat as You Type tab from the dialog box that appears. In the Replace as You Type section, check or clear the Internet and Network Paths with Hyperlinks box.

You can also build a hyperlink between two Word documents (at least one must be saved as a Web page) or any other open Office applications (except Access) using standard cut-and-paste operations. First, select some text in the originating document. Then copy it to the Clipboard (Ctrl+C from the keyboard or Edit, Copy from the menu), select the location for the hyperlink in the other open Word document or Office application, and choose Edit, Paste as Hyperlink from the menu. The text you copied is now a hyperlink back to the starting document. This is certainly the easiest way to build hyperlinks between Office documents.

More Types of Hyperlinks

You don't have to restrict yourself to building hyperlinks between Web pages on the Internet or your company intranet. You can hyperlink to several other areas on the Internet. You can also link to many different file types besides Web pages.

Building Hyperlinks to Other Types of Internet Locations

The most common types of hyperlinks include links to other Web pages, frames, or e-mail addresses. Several less common hyperlinking options include building hyperlinks to

- FTP (File Transfer Protocol) sites
- Newsgroups
- Telnet sessions

Note

Including a hyperlink to an FTP site, a newsgroup, or a telnet session will spawn external applications when it is clicked. The FTP hyperlink may open your default FTP client, the hyperlink to a newsgroup will open your default newsreader (probably Outlook Express), and the hyperlink to a telnet session will spawn your default telnet application (probably either Telnet or HyperTerminal).

Because these options are not automatically available in Word 2002, you need to manually enter the URL in the Insert Hyperlink dialog box. The following sections provide sample code for each type of hyperlink.

Hyperlinks to FTP Sites

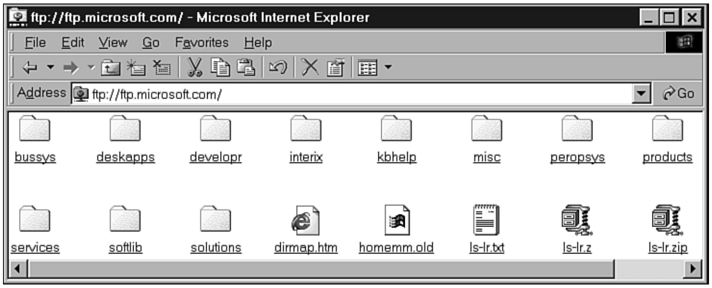
An *FTP site* is an Internet file repository. It has no fancy Web pages or graphics, just folders filled with files. Many FTP sites have public areas where visitors can browse and download files. These sites are often called anonymous FTP sites. FTP, File Transfer Protocol, is the actual protocol used to transfer files between computers and has become synonymous with the sites containing the files.

A hyperlink to an FTP site is the same as one to a Web site except that the `http://` is replaced by `ftp://`, and the URL usually starts with `ftp` (instead of the usual `www`). To build a hyperlink to a FTP site (in this example, we'll use Microsoft's), just open the Insert Hyperlink dialog box, type in the display text, and in the URL box, enter the FTP URL. In this case enter:

```
ftp://ftp.microsoft.com
```

Click OK. Now when you click on this hyperlink, you should see something like Figure 10.

Figure 10
The root directory to Microsoft's FTP site.



You can build hyperlinks to files on FTP sites that, when clicked, download files to a visitor's computer. Because the FTP protocol is used only to transfer files, this is often the fastest way to download a file.

HYPERLINKS TO NEWSGROUPS

Newsgroups are forums for discussing specific topics on the Internet. The collection of newsgroups, more than 80,000 at last count, is referred to as Usenet. Each newsgroup is devoted to a single topic, usually defined by its title. For instance, `misc.invest.stocks` or `alt.invest.penny-stocks` discuss investment topics.

You can provide hyperlinks to newsgroups that discuss topics related to the subject of your Web page. The syntax is similar to that used for building an e-mail hyperlink. To build a hyperlink to the newsgroup `misc.invest.stocks`, your display text in the Insert Hyperlink dialog box might be "Talk about stocks on Usenet," and your address would look like this:

```
news:misc.invest.stocks
```

Clicking on a hyperlink to a newsgroup opens the default newsreader (a program specifically designed for reading and posting messages to newsgroups), loads the current contents of the forum, and displays the messages posted over the last several days or weeks. The messages can then be browsed, read, or responded to as desired.

Caution

Not all NNTP servers receive all the newsgroups listed on Usenet. For instance, if your ISP's NNTP server does not carry `misc.invest.stocks`, you might try `misc.invest.mutual-funds` or `alt.invest` when building a hyperlink. Keep in mind that the audience for your links could have similar issues–the inability to access certain groups through their ISP or other Internet connection.

Note

Many organizations and companies sponsor newsgroups on their own NNTP servers. Microsoft has numerous newsgroups hosted on its own servers. In this case, the Address field in the Insert Hyperlink dialog box would require the entire address to the newsgroup as shown here:

```
news://msnews.microsoft.com.microsoft.public.office.misc
```

HYPERLINKS TO TELNET

Telnet is a means of accessing another computer on the Internet as a terminal. Windows comes with a program called Telnet that enables you to connect in this manner. Telnet is strictly a text environment and is thus used heavily by universities or other organizations that need to disseminate information. Some advantages of Telnet are speed and direct access to the operating system of the target computer. A disadvantage is the lack of a standard navigation interface, though most systems are easy to figure out or have excellent online help.

To build a hyperlink to start a Telnet session, this text would be added in the Address box of the Insert Hyperlink dialog box:

```
telnet::melvyl.ucop.edu
```

Note that there are two colons after telnet; this is not a typo. This launches the Windows Telnet application and loads the Telnet address. You are usually prompted for some type of login information at Telnet sites. Public sites will usually require a generic login so that anyone can have easy access. Private sites require an established user account (username and password) for access.

BUILDING HYPERLINKS TO OFFICE DOCUMENTS

As you saw earlier, it is possible to build hyperlinks to internal markers, bookmarks, and headers in a Word document. You can use this same idea and build links to similar markers in other Office documents, including

- Bookmarks in other Word documents
- Named ranges of cells in Excel spreadsheets
- Forms and reports in Access databases
- Specific PowerPoint slides

Unfortunately you cannot build links to specific objects (specific email, calendar items, and so forth) in Outlook.

Two techniques for building these hyperlinks are available. The easiest way is to do drag and drop:

1. Open the Word document or Web page to where you want to place the hyperlink.
2. Open the other application and it's file. Navigate to the specific portion to which you want to hyperlink.

Note

Both the Word document and the other application file (spreadsheet, presentation, and so on) must be saved before proceeding to build the hyperlink between them.

3. Rearrange Word and the other application so you can see both on the screen at the same time.
4. Using the right mouse button, click and drag from the application file over to Word and drop.

Tip from Michael Larson

In Excel, you need to grab the upper-right cell corner of the named range to pick it up. You cannot pick up the named range from any other location.

5. From the context menu that pops up, choose Create Hyperlink here.

You now have a working hyperlink inside your Word document. When the link is clicked, the other application opens, loads the file, and moves to the correct location inside it.

To use the second means to build a hyperlink to an Office document, follow these steps:

1. Before building the hyperlink make sure that you know the name of the Word bookmark, the name of the named range in the Excel spreadsheet, or the slide number in the PowerPoint presentation.
2. Select your display text, open the Insert Hyperlink dialog box, and enter the URL or file path to the Office document.
3. After the URL or file path, type in `#`.
4. Enter the name of the Word bookmark, the named range for an Excel spreadsheet, or the PowerPoint slide number.
5. Click OK to complete your hyperlink.

Here are sample hyperlink addresses to a bookmark in a Word document, an Excel worksheet, and a PowerPoint slide, respectively:

```
http://someserver.com/word.doc#bookmark
http://someserver.com/sheet1.xls#'Sales'!A1
http://someserver.com/present.ppt#5
```

Caution

These hyperlinks styles work only from documents loaded into Office applications; they do not work from a browser. In other words, you can jump from a Word document to an Excel chart to a PowerPoint slide and back. But if you convert any of these documents to Web pages, load them into a browser, and click the same hyperlinks, they will not work.

Using these hyperlinks makes it easy to link a group of Office documents together, perhaps, for example, for the purposes of a comprehensive report.

Hyperlinking and Embedded or Linked Objects

You can embed OLE (Object Linking and Embedding) objects (files from another program that are dropped into Word and display simply as an icon) or links to OLE objects within a Word document. You can build a hyperlink from these OLE objects by selecting the object and choosing Insert, Hyperlink from the menu. When the document containing the OLE object is saved as a Web page, the icon representing the embedded object is displayed and becomes an active hyperlink.

Caution

If the OLE object embedded in a Word document does not contain a hyperlink, when the document is saved as a Web page, only the icon will be seen when the Web page is displayed in a browser. The icon does not contain any functionality or links; it is only a graphic.

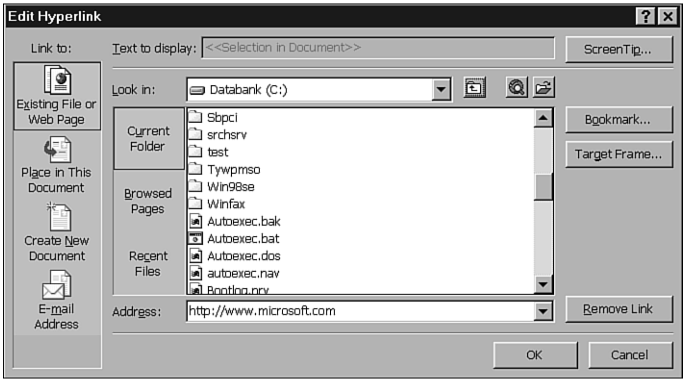
Editing Hyperlinks

Links go bad. Files get moved around or are deleted. Web sites come and go. When this happens, you need to be able to change or delete your hyperlinks. Again, Word 2002 makes this process easy. For example, if you wanted to change a hyperlink from `http://www.microsoft.com` to `http://home.microsoft.com`, simply:

1. Right-click anywhere over the hyperlink and choose Edit Hyperlink. (Do not left-click on the hyperlink—this activates the link.) Alternatively, you can use the keyboard arrow keys to place the cursor anywhere in the hyperlinked text and:
   - Choose Insert, Hyperlink from the menu
   - Click the Insert Hyperlink toolbar button
   - Use the keyboard shortcut, Ctrl+K.

   You do not need to select all the display text to edit a hyperlink.
2. Figure 11 shows the Edit Hyperlink dialog box. The address is displayed but not the display text. This dialog box closely resembles the Insert Link dialog box except for the addition of a Remove Link button at the bottom.
3. To edit your hyperlink, select the box containing the URL and enter your new URL, or choose a new URL using any of the many techniques already discussed in this chapter.
4. If you want to completely eliminate the hyperlink, click the Remove Link button at the bottom. The Edit Hyperlink dialog box disappears and so does your hyperlink, though the display text is unchanged.

Figure 11
Editing or deleting a hyperlink from the Edit Hyperlink box.



Edit Hyperlink

Text to display: <<Selection in Document>>

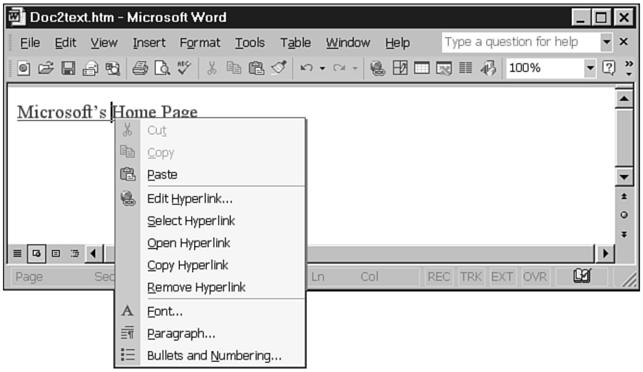
Look in: Databank (C:)

Address: http://www.microsoft.com

The hyperlink display text can be edited from the document as you would edit any other text. However, don't select and delete all the text; this removes the hyperlink. To replace all the display text of a hyperlink, select the entire display text and type in (or paste in) the new text.

More hyperlink editing choices are available from the pop-up menu when you right-click over a hyperlink (see Figure 12).

Figure 12
Right-clicking on a hyperlink brings up a context menu of choices for hyperlinks.



Doc2text.htm - Microsoft Word

Microsoft's Home Page

The list of the choices on this menu includes the following:

- **Edit Hyperlink**—Brings up the Edit Hyperlink dialog box, and you can proceed to edit or remove the hyperlink.
- **Select Hyperlink**—Selects all the text in the hyperlink. You can change all the display text or delete the display text and hyperlink.
- **Open Hyperlink**—Opens the link address in a new browser window.

- **Copy Hyperlink**—Copies the display text and URL to the Clipboard. You can select Paste to add this hyperlink to another point in your document or insert it into another document.
- **Remove Hyperlink**—Deletes the hyperlink but leaves the display text intact. This is the quickest way to delete a hyperlink.

If you click on a previously working link and get an error such as `File not Found (404)`*, see "What to Do if You Get a* `File Not Found` *Error" in the "Troubleshooting" section of this chapter.*

RELATIVE AND ABSOLUTE HYPERLINKS

The hyperlinks described so far have all had complete addresses. In other words, all the information required to navigate to a given point is present, no matter where the target is located, in the hyperlink itself. These types of hyperlinks are *absolute hyperlinks*. The following are examples of absolute hyperlinks:

```
http://www.microsoft.com
ftp://ftp.microsoft.com
```

Anyone, anywhere, can click on a hyperlink with this address and go to that point (assuming they're connected to the Internet).

Relative hyperlinks are hyperlinks that point to a place relative to the current location of the document containing the link on a Web site. For example, one folder down from the Web site root is a location relative to the Web site's root folder. Imagine that you're building a home page for `newco.com`. In that page, you want to build a hyperlink to a file, `project.htm`, which resides in a subfolder called `hq`. You can build an absolute hyperlink as follows:

```
http://www.newco.com/hq/project.htm
```

You can use a relative hyperlink, however, and save yourself some typing. In this case, the relative hyperlink would be:

```
hq/project.html
```

Both links take you to exactly the same place as long as the visitor is already at `newco.com`. The relative link only works relative to the rest of `newco.com`. If the home page is moved to a different folder, the relative hyperlink no longer works.

Using relative hyperlinks can save you a great deal of typing and has its own special nomenclature as far as denoting folders above or below the location of the file currently loaded into your browser. For instance, if you are editing a page stored in the `hq` subfolder and want to create a link to a file called `index.html` in the root folder, your relative hyperlink would be

```
..\index.html
```

If you were two folder levels down and wanted to link to this file in the root folder, your relative address would be

```
..\..\index.html
```

If you're going to use relative hyperlinks in documents on a Web server, it's always a good idea to set the hyperlink base of a document. The *base hyperlink* is the root URL under which all files are stored. For many Web sites, this will be `www.SomeDomain.com`. To set the hyperlink base, choose File, Properties from the menu and fill in the hyperlink base URL for a Web server at the bottom of the screen under the Summary tab. You can also use this feature to define a hyperlink base for your local hard drive or network server.

Adding Hyperlinks to Graphics and Toolbar Buttons

So far we've mostly talked about creating text hyperlinks inside Word. You can also attach hyperlinks to graphics and even to toolbar buttons. Graphics are a natural for hyperlinks since they're the first thing your eye is drawn to on a page. Adding hyperlinks to toolbar buttons lets you keep important hyperlinks available all the time, irregardless of which document you're working on.

Adding Hyperlinks to Graphics

As you may already know, you can make graphics into clickable images; that is, you can add hyperlinks to images. These images can be

- Imported graphics files
- Clip art
- Images imported with a scanner or digital camera
- Objects drawn using the Drawing tools in Word
- Graphics drawn using the WordArt tools

Charts or parts of charts drawn using Word 2002 chart tools cannot be used for hyperlinking.

Tip from *Michael Larson*

Even though you can't hyperlink directly to a chart, you can insert a bookmark in the Word document near the chart and hyperlink to that instead.

The technique for creating a hyperlinked image is identical to that used for creating a text hyperlink. After the image is in your document, select it and open the Insert Hyperlink dialog box. The only difference is that the box for entering the display text is now inactive. Otherwise, the box works the same. Enter your hyperlink information into the Address box and click OK.

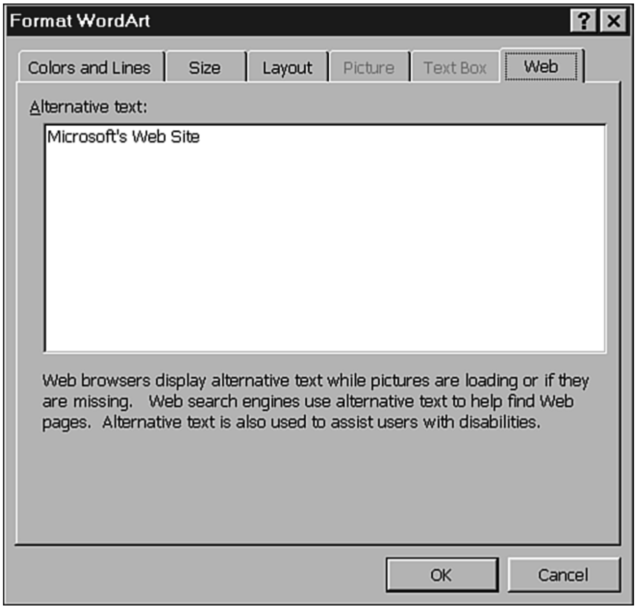
You can edit or delete the image hyperlinks just as you do for text hyperlinks. Select the graphic by right-clicking on it and choosing the appropriate editing option from the pop-up menu. These options include

- **Edit Hyperlink**—Brings up the Edit Hyperlink dialog box, and you can proceed to edit or remove the hyperlink.
- **Open Hyperlink**—Activates the link and opens the destination page in your default browser.
- **Remove Hyperlink**—Deletes the hyperlink but leaves the image intact.

It is a good idea to add an alternative text description of your graphic. In order to speed up the download of Web pages, many people browse with images turned off. If you have alternative text, then this text displays in the blank box where the graphic would normally display and gives users at least some idea of what is there. Also, alternative text for images is absolutely necessary for sight-impaired users on the Internet; special software can pick up this text and convert it to speech.

To add alternative text, right-click the graphic and choose Format Picture from the pop-up menu. On the Web tab, you can enter alternative text, as shown in Figure 13.

Figure 13
Adding alternative text to hyperlinked images.



Alternative text: Microsoft's Web Site

Web browsers display alternative text while pictures are loading or if they are missing. Web search engines use alternative text to help find Web pages. Alternative text is also used to assist users with disabilities.

The alternative text can be descriptive of the graphic (“a circle” or “picture of a cat,” for example), or it can be the URL of the image hyperlink.

Adding Hyperlinks to Toolbar Buttons

You can redefine any toolbar button into a hyperlink.

Tip from *Michael Larson*

You probably shouldn't redefine common toolbar buttons, such as the bold and italicize buttons, as hyperlinked buttons. Not only do you lose the original button functions, but you may inadvertently launch the hyperlink just out of habit.

A hyperlinked toolbar button provides a convenient way to always have instant access to any hyperlinkable document. To build a hyperlinked toolbar button:

1. Open the toolbar where you'd like to add your custom hyperlinked button.
2. Add a new button to the toolbar by clicking the small down-arrow at the end of the toolbar, choosing Add or Remove Buttons, then Customize.
3. Right-click over the new button on the toolbar and choose Assign Hyperlink.

Tip from *Michael Larson*

The right-click menu also displays other options for customizing the button, such as editing the text in the button or adding a new image to it. I highly recommend you use one of these options to customize your button.

4. The Assign Hyperlink dialog box comes up. You can type in a URL, choose one from several lists (selections are the same as those found in the Insert Hyperlink dialog box), or browse to a file on your hard drive or the Web.
5. Click OK to assign the hyperlink. Then click Close to remove the Customize box.

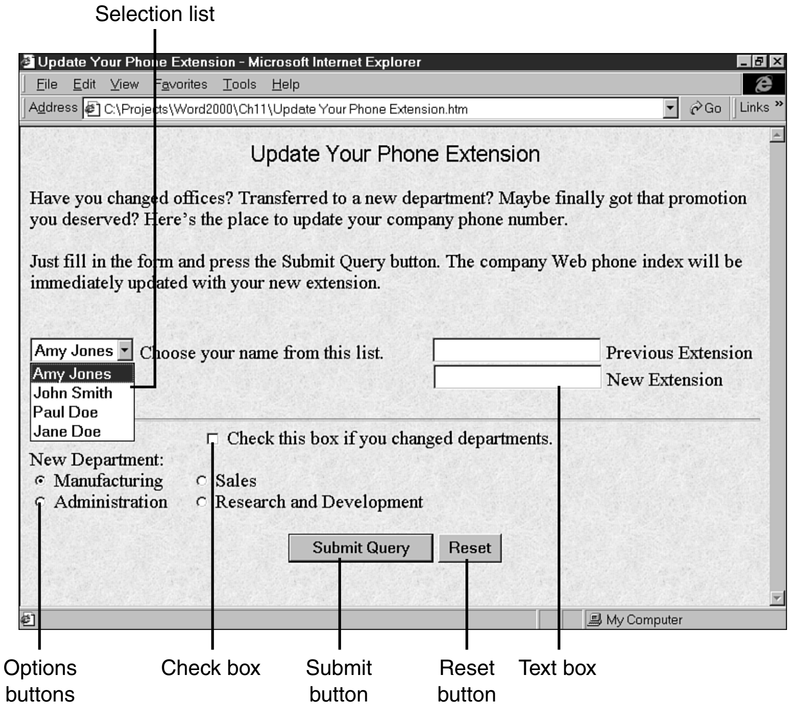
You now have a custom, hyperlinked button. When you click on the button, you'll launch the appropriate application, and your hyperlink will load the Web page or application file.

Building Web Forms in Word

If you've used a search engine on a Web site, you've used a simple Web form. Other typical uses of Web forms on the Internet include registration or order forms. A *Web form* is a specially set-off section of HTML code where a visitor can choose selections or enter text and send the results to a Web server for some action. Web forms contain all the standard buttons and boxes (called controls) you expect with any kind of form, including

- Text boxes
- Option (or radio) buttons
- Check boxes
- Selection lists or drop-down list boxes
- Submit and Reset buttons

The Submit button is unique to Web forms, though its apparent function is the same as any other button that is clicked to complete an operation, such as the OK button on a dialog box. A sample form using each of the form elements listed appears in Figure 14.



Selection list
Update Your Phone Extension - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Address C:\Projects\Word2000\Ch11\Update Your Phone Extension.htm
Go
Links
My Computer
Options buttons
Check box
Submit button
Reset button
Text box

Update Your Phone Extension

Have you changed offices? Transferred to a new department? Maybe finally got that promotion you deserved? Here's the place to update your company phone number.

Just fill in the form and press the Submit Query button. The company Web phone index will be immediately updated with your new extension.

Amy Jones ▾ Choose your name from this list.
Amy Jones
John Smith
Paul Doe
Jane Doe

Previous Extension

New Extension

□ Check this box if you changed departments.

New Department:
● Manufacturing ○ Sales
○ Administration ○ Research and Development

Submit Query | Reset

Figure 14
A sample Web form.

BUILDING WEB FORMS FROM SCRATCH

Because Word 2002 does not provide any means to generate Web forms automatically, you will need to build Web forms from scratch. Given the WYSIWYG nature of Word 2002 in terms of putting a Web form together, this is a relatively easy operation.

The controls you use to build a Web formare found on the Web Tools toolbar in Word 2002, as shown in Figure 15.

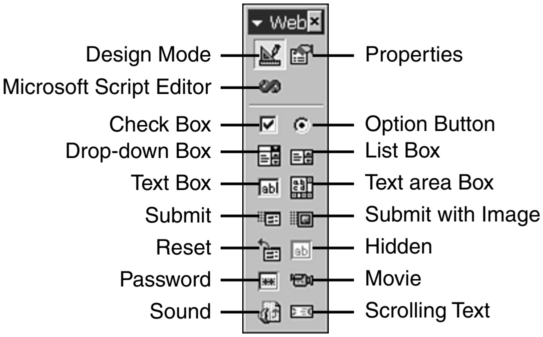
This toolbar contains all Web form controls listed earlier as well as several others:

- **Submit with Image**—Same as the Submit button, except you can use an image file for the Submit button rather than the standard gray button.
- **Text Area**—Same as a text box but allows multiple lines of text in a text box, rather than only a single line.
- **Hidden**—Sometimes you might want to pass the same information every time to the Web server, and you don't want a visitor to see it, such as a sales tax rate. Use this

control to enter that information. These controls do not display in the browser, hence the name Hidden.

- **Password**—A specialized text box that echoes asterisks rather than what is typed. Usually used for entering passwords.

Figure 15
Use the Web Tools toolbar to build Web forms.



From the Web Tools toolbar, you can also start the Microsoft Script Editor if you want to build scripts for the Web form.

Caution

Word 2002 has another toolbar, the Control Toolbox, which contains controls that appear to be identical to those in the Web Tools toolbar. Do not use the controls in the Control Toolbox to build Web forms. Even though the controls (option buttons, list boxes, and so on) look the same, they operate on a different principle and do not work with a Web server.

The Web Tools toolbar also has a button labeled Design Mode. Make sure that this is depressed before you begin building your form so that you will be operating in a WYSIWYG environment when you build your Web form. The first time you select a Web form control, Web form “Top of Form” and “Bottom of Form” markers are inserted into your Web page, as shown in Figure 18. All Web form controls must be placed between these markers to operate properly.

To add a form control to the form, place your cursor where you want the form control to appear and click on the appropriate element in the Web Tools toolbar. The form control appears on your form.

Now that the form control appears in your document, you can select it and drag it to adjust its location or size, the same way you would if it were a picture. Make sure that the form control remains within the top and bottom form markers.

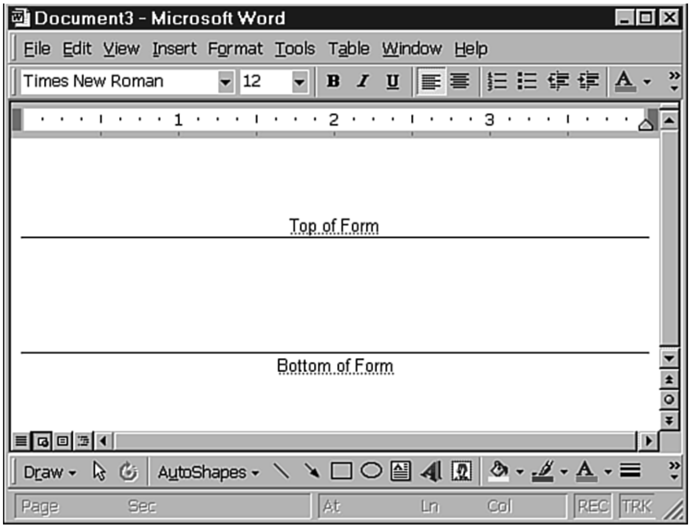

Document3 - Microsoft Word
File Edit View Insert Format Tools Table Window Help
Times New Roman
12
Top of Form
Bottom of Form
Draw
AutoShapes
Page Sec At Ln Col REC TRK

Figure 16
The form boundary markers in Word.

Note

You can use any Word formatting tools to align or place form controls or descriptive text. These include tabs, spacing, justification, and so on. Using a table to organize and align form controls is also a possibility.

If you are having trouble lining up form elements, see "What to Do to Make Your Form More Readable" in the "Troubleshooting" section of this chapter.

You can right-click over any form control and choose Properties to adjust any properties or parameters. It is beyond the scope of this text to describe these controls and their programmatic use in detail, but the properties will be familiar to anyone with experience building Web forms or programming.

THE WEB TOOLBAR

If you have used Word 97 or Word 2000, then you may already be familiar with the Web toolbar. This toolbar has not changed since its introduction in Word 97 as shown in Figure 17.

This toolbar contains much of the same functionality as a Web browser toolbar:

- The Forward and Back arrows take you through your browsing history. Because Word documents and other files can now contain hyperlinks, you can browse through previous Word documents as well as Web pages using these arrows.
- The Stop Current Jump button stops the loading of a Web page.
- The Refresh Current Page button reloads the page into Word.

- Clicking on the Start Page button loads your Start page as it is defined under the Go button on this toolbar.
- The Search the Web button loads your Search page as it is defined under the Go button on this toolbar.

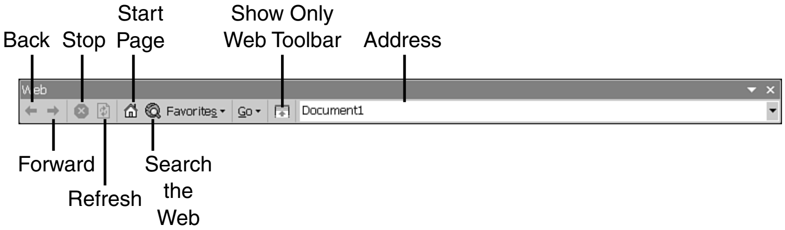


Figure 17
The Web Toolbar.

Note

The Start page and the Search the Web page are defined in the Internet browser properties and invoke the browser and connection if they are selected—these are not Word-specific features.

- The Favorites List is the same list of Internet (or other) addresses you see in the Microsoft Internet Explorer Favorites list.
- The Go button enables you to edit your Start and Search pages under Set Start Page and Set Search Page. You can also choose Open after clicking this button to pop up an address box. You can enter any Internet address into this address box. The Load Start Page and Load Search Page functions found here duplicate the Start Page and Search the Web buttons.

Tip from
Michael Larson

If you choose the Set Start Page or Set Search Page options, you are only given the option to select the current page as your new Start page or new Search page. You cannot browse to another page or choose another file.

- The Show Only Web Toolbar button toggles the display between showing only the Web toolbar and all selected Word toolbars.
- You can enter any Internet address, network path, or filename from a local drive in the address box. Clicking the arrow to the right of this box displays your browsing history. You can also select from this list to choose a particular file or address.

The Web toolbar is a convenient way to browse a set of hyperlinked Word documents or to launch your browser with your Start page or Search page loaded.

Tip from
Michael Larson

The Web toolbar can be customized like any other toolbar in Word. If you click the Add or Remove buttons at the end of the toolbar, click Customize, and then select All Commands in the Categories window, you can add anything. An especially useful addition to the Web toolbar might be the Web Page Preview button. For more information on customizing toolbars, see the section "Customizing Toolbars" in Chapter 31, "Customizing Word."

WEB FOLDERS

You can save your Word documents directly to a Web server using Web Folders. *Web Folders* are special folders that provide a direct shortcut to a Web server. They enable you to transparently upload files directly to your Web server or even work with your Web pages directly from the Web server. You can see Web Folders in Windows Explorer or the Open File dialog box in the same way that you can see any other folder on your hard drive. You can now work with the files on a Web server the same way you work with any other files or folders on your hard drive.

Web Folders require special support from the Web server, where the Web Folder stores your files. A *Web server* is a computer hosting a software package that serves up or delivers Web pages or other files in response to requests from browsers or other programs. You cannot use Web Folders with any Web server. The Web server must meet two criteria:

- It must be a Microsoft Internet Information Server (IIS) Web server. Web server software from other vendors don't support Web Folders.
- It must have the Office Web Server Extensions installed.

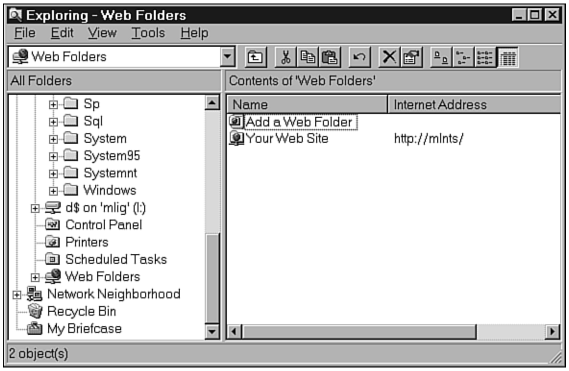
Also, you must have write-access to the Web server to be able to upload and modify files on the server. Contact whoever is in charge of the Web server to arrange these rights.

ADDING NEW WEB FOLDERS

When you have rebooted after installation of Office XP or Word 2002, you will notice a new object at the bottom of Windows Explorer on the left side called Web Folders (see Figure 18).

Clicking on Add a Web Folder under Web Folders opens the brief Add a Web Folder Wizard. The first screen, shown in Figure 19, requests the path to the Web server.

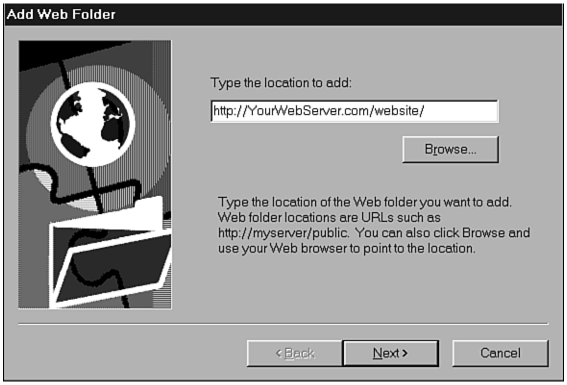
The path must begin with `http://`. If you save files to any location other than the root of the Web server, you need to add the appropriate subfolder information. Instead of manually entering the address to a Web server, you can also click Browse to open your browser and browse to the Web server. When you have filled out the URL to the Web server using either technique, click Next. The wizard confirms that the Web server is present, that it contains the Office Web Server Extensions, and that you have write-access to the server. If any information fails confirmation, you receive an error message and are unable to proceed with the creation of the Web Folder.



Name	Internet Address
Add a Web Folder	
Your Web Site	http://mlnts/

2 object(s)

Figure 18
The new Web Folders object installed by Office XP as seen in Windows Explorer.



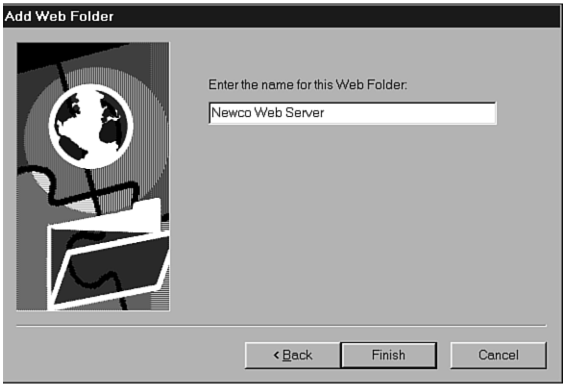
Type the location to add:

http://YourWebServer.com/website/

Type the location of the Web folder you want to add. Web folder locations are URLs such as http://myserver/public. You can also click Browse and use your Web browser to point to the location.

Figure 19
Entering the path to a new Web Folder.

If the information is confirmed (you may be asked for your userid and password), you are taken to the next screen, shown in Figure 20, where you can add a descriptive name to the Web Folder.



Enter the name for this Web Folder:

Newco Web Server

Figure 20
Naming your new Web Folder.

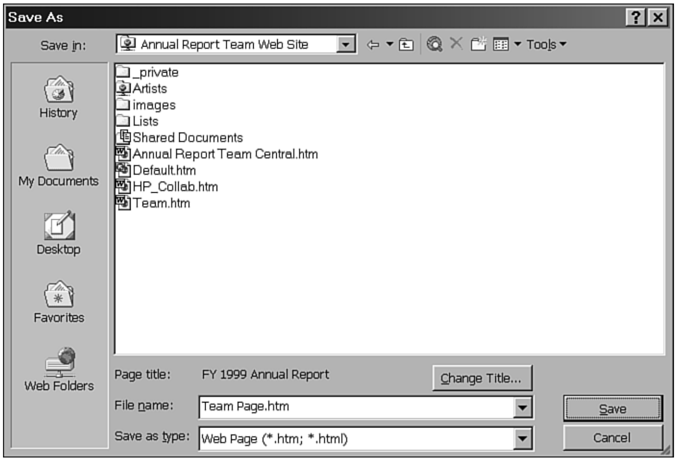
Click Finish to create the Web Folder. Your new Web Folder is saved under the descriptive name and is accessible from Windows Explorer and from within Word.

Sending Files to Your Web Server Using Web Folders

After your Web Folder is created, you can use it for saving or opening files. For example, imagine that you've just created a document in Word and you want to save it as a Web page to your Web server. After choosing File, Save as Web Page from the menu, you see the typical Save As file list box. Clicking on the Web Folders button in the lower-right corner of this box displays all available Web Folders.

If you click on a specific Web Folder, all the files in the root of the folder path and all the subfolders are listed, as shown in Figure 21.

Figure 21
Examining the contents of a Web Folder.



Save in: Annual Report Team Web Site

_private
Artists
images
Lists
Shared Documents
Annual Report Team Central.htm
Default.htm
HP_Collab.htm
Team.htm

Page title: FY 1999 Annual Report
File name: Team Page.htm
Save as type: Web Page (*.htm; *.html)

Clicking Save saves the file directly from Word to the Web server.

You can also use a similar procedure to open files for editing directly from a Web server. From within Word, choose File, Open from the menu, click on the Web Folders button in the lower-left corner of the Open box, click the Web Folder, select your file, and click Open.

Team Web Tools in Word

Oftentimes, you need to bring your team together to discuss a few documents. Or the team might need to summarize many documents into one, such as generating the company annual report. And your team might be wide-flung, across multiple continents or cities or even just a few buildings. So how can Word help?

Word has some tools that might just make that team interaction easier and more convenient. These include

- Team Web Sites
- Web components
- A document discussion interface

These tools mean any team member, anywhere, anytime can work on project documents or communicate via discussion threads or using the document discussion interface. It is much easier to obtain input from remote offices or employees—no more having to juggle everyone's schedule. Documents and information are available to every team member immediately. Also not everyone has to have Word to participate; only a browser, Internet Explorer 5 or later, is needed. And, if you use the Team Web Site, all of the team communication is centralized in one location.

Next, we'll look at where are these tools located and how you get started.

TEAM WEB SITES

Team Web Sites are made possible by the installation of OWS (Office Web Server, also called Office Web Server Extensions), present on the Office XP CD-ROM or free from Microsoft's Web site, onto a Microsoft Web Server. OWS requires Microsoft Internet Information Server (IIS) version 5 or later as the Web server running on Windows NT or Windows 2000. OWS will not install under Windows 95, 98, or Me.

The OWS provides basic Team Web Site functionality, including:

- A Home Page with links to all of the Team Web Site functions
- A default discussion forum
- The means to manage and create any type of list, such as to-do lists
- The means to add, remove, and grant rights to team members
- Document management tools, including uploading
- An engine to search your Team Web Site and documents
- The means to build and manage lists of announcements, events, and links to Team-related Web sites
- The ability for each user to filter and sort their view of any list
- An address book for all relevant team contact information
- A way to create, schedule, and assign tasks

The functions are accessible to anyone with a browser, and in many cases, from within Word. No special tools, such as a Web page editor, are needed to use these functions. The Team Web Site can be immediately customized for your team needs.

WEB COMPONENTS

Word gives you tools to manage your Team Web Site documents and lists by letting you add Web components to your Team Web Site Web pages. Each Web component generally has a single, specialized function that otherwise cannot easily be added to the page. These Web components can be inserted into Web pages built in Word. Two Web components are available in Word for use with Team Web Sites:

- List View lets you easily generate and manage lists of items
- Document Library View manipulates and sorts all of the documents in use

More Web components are available on the Web and can easily be downloaded for use in the Web pages of your Team Web Site.

Note

Web components have similar functionality to some components that can be inserted by FrontPage, including Pivot Table, Chart, and Hover Button.

More components can be added from the Microsoft site and other third-party sites such as MSNBC or MSN. To locate more Web components, choose Tools, Tools on the Web from the menu and follow the directions from the Web page.

THE DOCUMENT DISCUSSION INTERFACE

In addition to discussion forums on a Team Web site, you can hold document discussions using another set of tools in Word. During a *document discussion* comments and responses made by all parties are saved in a special database on the OWS and associated with that particular document. Any document on or off the Team Web Site can be discussed as long as all parties have access to it and have signed in to the Discussion database. Also the Document Discussion tools are available from Word 2002, Word 2000, and Internet Explorer 4 or later.

Another key part of document discussions is the *subscription* capability. When you subscribe to a document discussion, you are automatically notified by email whenever a change is made to the document or document discussion comments.

The Web Discussions toolbar can only be accessed via the menu by choosing Tools, Online Collaboration, Web Discussions from the Word menu.

You can find more details about how to use the OWS, Team Web Sites, Web components, and the Document Discussion interface by visiting the Special Edition Using site at [add URL here].

INTRODUCING SMART TAGS

To a word processor, a word is a word is a word. There are rules in Word 2002 for sentences (must start with a capital letter) and grammar and spelling. But these are word manipulations only; the word processor doesn't know the importance of one word or group of words over another. That takes human reasoning and understanding. But what if the word processor could automatically pick out certain word phrases and intelligently tag them as belonging to certain categories? Word 2002 can, and it takes the form of smart tags. Word 2002 adds a new capability called *smart tags* to automatically pick out or designate a word or short phrase in your document as belonging to a certain group, such as a person's name, and tagging it.

Which word or phrase is tagged depends on which smart tag capabilities you have activated in Word. Some of the smart tags available in Word are

- Proper names
- Dates
- Addresses
- Phone numbers

Not only can you tag these words, but you also can tell Word to undertake an action with the tagged words. For instance, you can use the smart tag to add a tagged name to your Outlook Contacts list.

Smart tags run in the background as you create your document. Based on the grammar checker, the smart tag algorithm acts like a trained dog, sniffing out the desired smart tag patterns and tagging them with a purple underline. The algorithm is not 100 percent accurate but is improving. For instance, the algorithm tags CD-ROM as a proper name, so you'll still have to do some manual clean-up and delete some smart tags.

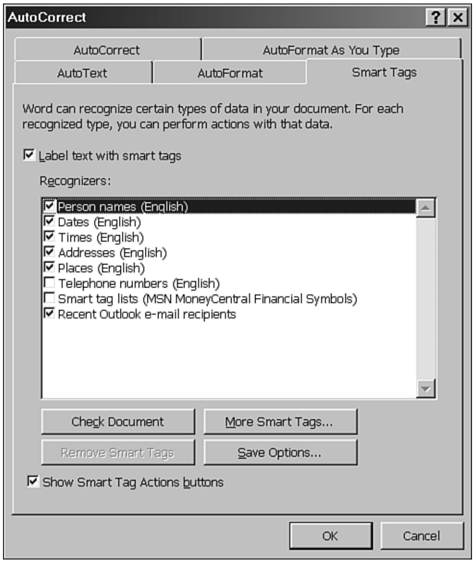
So why are smart tags important? With smart tags, Word automatically tags information (you don't have to read the document to find these data types) and then lets you do something with it immediately. You don't have to manually cut and paste the information; Word does it for you as a smart tag action. Also, if you save your Word document as a Web page, the smart tags are exported as XML (eXtensible Markup Language) tags, viewable in Internet Explorer 5 or higher. Other XML-aware applications can then retrieve the tagged information without a human being required to manually move the information.

Next, let's look in more detail at which smart tags are available and how to use, customize, delete, or even turn them off.

WORKING WITH SMART TAGS

Smart tags are accessible from the menu under Tools, AutoCorrect Options and the Smart Tags tab, as shown in Figure 22.

Figure 22
Smart Tag options available from the AutoCorrect box.



AutoCorrect

Word can recognize certain types of data in your document. For each recognized type, you can perform actions with that data.

☑ Label text with smart tags

Recognizers:

☑ Person names (English)
☑ Dates (English)
☑ Times (English)
☑ Addresses (English)
☑ Places (English)
☐ Telephone numbers (English)
☐ Smart tag lists (MSN MoneyCentral Financial Symbols)
☑ Recent Outlook e-mail recipients

☑ Show Smart Tag Actions buttons

The Label Text with Smart Tags check box is checked by default. Clear this if you want to turn off the entire smart tag capability.

Next is the Recognizers list. A recognizer is an algorithm designed to tag one particular data type, such as a phone number. From the Recognizers list you can decide which types of information you want Word to tag. Notice one of the options is the smart tag list. A *smart tag list* is a custom list of words or types of words that you want tagged. An example of a smart tag list is a list of stock ticker symbols. Creating custom smart tag lists requires programming skills, though the Smart Tag Software Development Kit is available for free from Microsoft.

The Check Document button scans the current active document and tags (or removes smart tags) according to the Recognizers list. This is much like running the spell checker.

The More Smart Tags button takes you to a list of other smart tags available at the Microsoft Web site. Note that smart tags may also be available from third-party vendors.

You can customize the way smart tags are saved with the Save Options button, as shown in Figure 23.

Only two options in the Save dialog box apply to smart tags: Embed Smart Tags, which is checked by default, and Save Smart Tags as XML Properties in Web Pages, which is not checked by default. Embed Smart Tags globally determines whether smart tags are saved. The XML feature is only useful if you want your smart tags to show up as XML in Web pages saved from Word.

Back at the AutoCorrect box (refer to Figure 25), the Show Smart Tag Action Buttons check box is used to turn the action buttons display on or off.

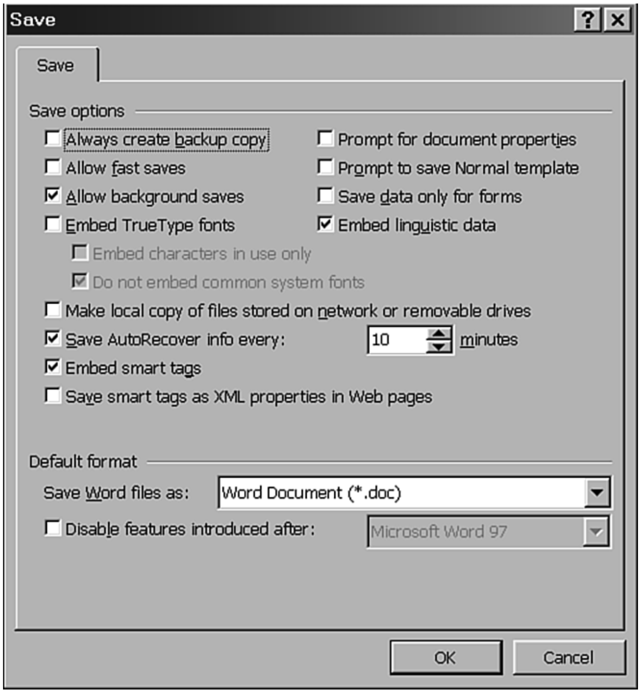


Figure 23
Save options for smart tags.

Save options

- ☐ Always create backup copy
- ☐ Allow fast saves
- ☑ Allow background saves
- ☐ Embed TrueType fonts
  - ☐ Embed characters in use only
  - ☑ Do not embed common system fonts
- ☐ Prompt for document properties
- ☐ Prompt to save Normal template
- ☐ Save data only for forms
- ☑ Embed linguistic data
- ☐ Make local copy of files stored on network or removable drives
- ☑ Save AutoRecover info every: 10 minutes
- ☑ Embed smart tags
- ☐ Save smart tags as XML properties in Web pages

Default format

Save Word files as: Word Document (*.doc)

☐ Disable features introduced after: Microsoft Word 97

So what does a smart tag look like? Figure 24 shows a tagged proper name and telephone number.

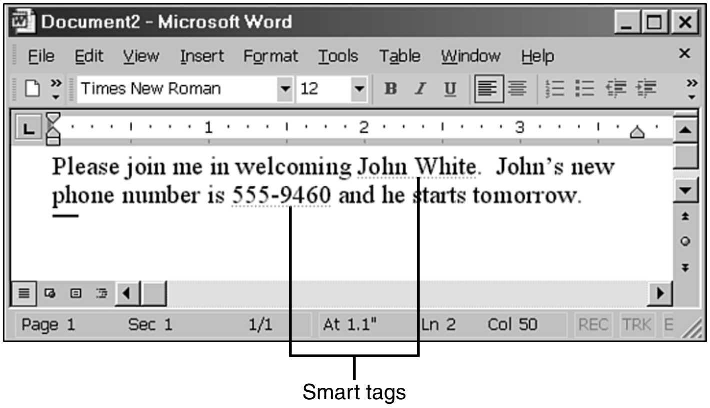


Figure 24
Smart tags in a Word document.

Please join me in welcoming John White. John's new phone number is 555-9460 and he starts tomorrow.

Page 1 Sec 1 1/1 At 1.1" Ln 2 Col 50 REC TRK E

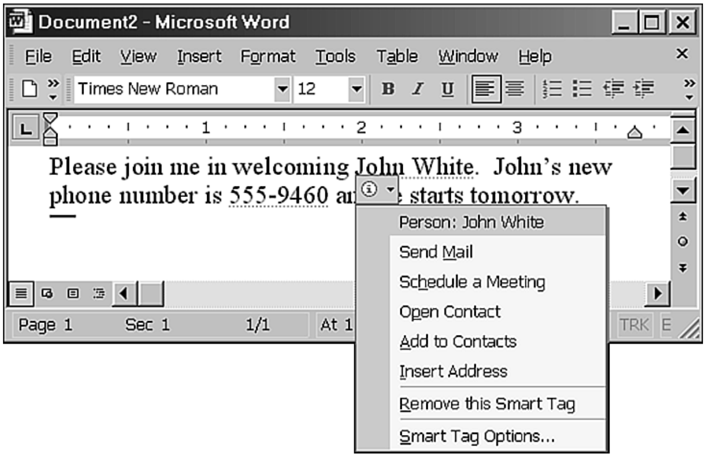
The smart tags for a proper name and a phone number are marked with a faint, dotted underline.

Tip from
Michael Larson

You can turn off the smart tag underline marker if you want. Go to Tools, Options, View, and clear the Smart Tags check box near the top. The smart tag is still present but is no longer visually marked.

If you hover your mouse pointer over a smart tag, the smart tag actions button appears. If you click on it, you'll see a menu of possible actions, as shown in Figure 25.

Figure 25
The smart tags action button and menu in a Word document.



Please join me in welcoming John White. John's new phone number is 555-9460 a[illegible] starts tomorrow.

The options available on the smart tags action button depend on the data type. For instance, the smart tag for the phone number contains options only for a single action, Add to Contacts, compared to the five action options available under a person's name. Clicking on an action item (such as those shown in Figure 25) automatically takes John White's name from the Word document and performs the requested action. For instance, if you choose Insert Address from the smart tag action menu, Word finds John White in your Outlook Contacts list, copies his complete address, and pastes it into your document all in one step. As you can see, smart tags can be a real time saver and an easy way to move information between applications.

Notice Smart Tags Options at the bottom of the smart tag action menu. Clicking this takes you back to the Autocorrect dialog box (refer to Figure 22), allowing you to change which data type is tagged. You can also reapply your smart tag changes by clicking the Recheck Document button, which is the renamed Check Document button from Figure 22.

Smart Tags and Web Pages

If you save your document as a Web page, Word converts any embedded smart tags to XML.

Note

Make sure that Word is set up to save the smart tags as XML. From the menu, choose Tools, AutoCorrect Options, Smart Tags; click the Save Options button; and check the Save Smart Tags as XML Properties in Web Pages box.

In XML, any word or phrase can be enclosed by a pair of custom tags, thereby defining the word. For instance, the following XML tags define blueberry as a type of pie:

```
<PIE>Blueberry</PIE>
```

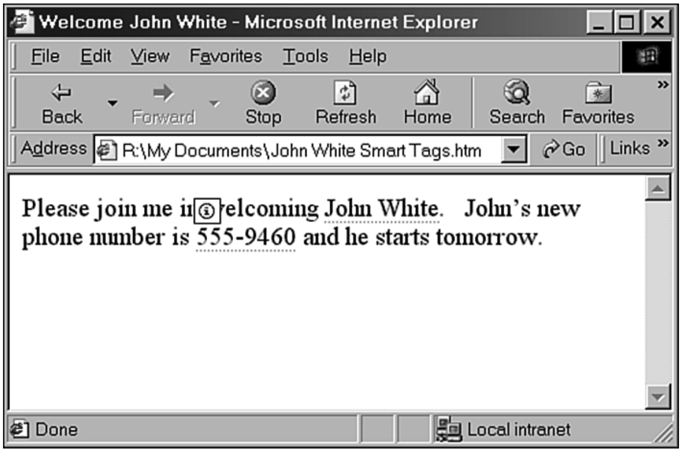
This differentiates blueberry from the actual berry or from the blueberry color or from any other use of blueberry. In a sense, smart tags are Word's version of XML. It is only natural that smart tags can be translated to XML in Web pages.

➔ For a more in-depth explanation of XML, see the Bonus Word Chapter entitled, "Using Word to Develop Web Content," in the section "The Technologies Used in Word 2002 Web Pages."

So what XML tags does Word actually use when it converts a smart tag? The XML tags for the phone number from our earlier example with John White (refer to Figure 24) look like this:

```
<st1:phone phonenumber="5559460">555-9460</st1:phone>
```

The Internet Explorer browser (version 5 or higher and only on computers with an Office XP installation) can also interpret these XML tags back into smart tags. Figure 26 shows our sample document in Internet Explorer 5 with the smart tag action button.



R:\My Documents\John White Smart Tags.htm

Please join me i ⓘ elcoming John White. John's new phone number is 555-9460 and he starts tomorrow.

Figure 26
A smart tag in Internet Explorer when viewing a Web page saved from Word with embedded smart tags.

Again, the smart tag is marked with a dotted, purple underline. Note that smart tags are not saved as XML tags if you save your Web page as a filtered Web page.

So why is saving smart tags as XML important or useful in a Web page? XML is becoming increasingly important as a way to add definition to the information on Web pages. Word provides a means for automatically tagging data types and generating XML.

SMART TAG LIMITATIONS

Smart tags are a great addition to Word, but they have some limitations. The algorithm that Word uses to define a particular data type is not perfect yet. It sometimes tags information as a proper name or place that really is not. You still have to manually scan a document and delete the incorrect smart tags. On the plus side, it is better to have an algorithm that tags too much information instead of missing some information that should be tagged.

Smart tags are not viewable in versions of Word earlier than 2002. Smart tags as XML in Web pages are not useful or viewable for browsers other than Internet Explorer 5 or higher.

TROUBLESHOOTING

WHAT TO DO IF YOU GET A `File not Found` ERROR AFTER CLICKING A HYPERLINK

You built and tested a hyperlink a while ago, and it worked fine. Now when you click on it, you get a File not Found (404) error. What does this mean?

It means that the page you linked to has been deleted, moved, or renamed. You will need to edit your link to point to the same material if you can find it elsewhere. If you can't find the revised page, you may need to remove the link.

WHAT TO DO IF YOUR HYPERLINK DOESN'T WORK

You built a hyperlink in a Word document to a file on your local hard drive. When you sent the Word document to a colleague down the hall, the hyperlink didn't work anymore. What happened?

If you build a hyperlink to a file on your hard drive, it has the drive letter embedded in it. Clicking on that link from any other computer causes the browser to look at the local drive on that computer, not the drive the link was originally built from. Instead, if the location of the hyperlink has a network share name, such as \\johnsdrive\WordDocs, use that when building the hyperlink. The hyperlink will then work from any computer on the network with access to that shared network drive.

WHAT TO DO TO MAKE YOUR FORM MORE READABLE

You're having trouble aligning form elements or making your form easy to read. How can you solve this?

Organize your elements within tables. You can easily control alignment and justification for cells or groups of cells. You can also use the table borders to set off various sections of your form for improved readability.

WHAT TO DO TO DISABLE SMART TAGS

I don't really need smart tags for the work I do. How can I turn them off?

To turn off Smart Tags, from the menu choose, Tools, AutoCorrect Options and click the Smart Tags tab. Clear the Label Text with Smart Tags check box near the top and click OK. All the smart tag recognizers (routines that sift out particular word patterns in your document) are now disabled.

Creating Forms

In this chapter

by Bill Camarda

Special Edition Using

que®

WORD'S FORMS CAPABILITIES: AN OVERVIEW

Word's sophisticated forms capabilities permit you to streamline a variety of business tasks that once required paper forms. You can create forms that enable users to choose among lists of options; forms that provide online help; even forms that guide users from start to finish. Best of all, users can fill in these forms without changing the underlying form itself. And if you're networked, you can use your network server or *intranet*—rather than some distant warehouse—as your central repository for forms.

WHEN TO USE WORD, WHEN TO USE ANOTHER TOOL

As you'll discover, building a printed or electronic form in Word 2002 is relatively easy. However, there may be times when it makes more sense to use another tool:

- Forms built in Word can perform calculations, but if your forms require extensive, complex calculations that go far beyond simple arithmetic, consider building them in Microsoft Excel rather than Word.
- Forms built in Word can transfer their data to a database such as Microsoft Access, but if integrating the information in your forms into a database is your central goal, consider building the forms in Access rather than Word.
- If the users who will fill out your online forms do not have access to Microsoft Word or Office, consider creating Web-based forms that can be accessed from a browser (see Web Chapter 6, "Adding Interactivity with Hyperlinks, Web Forms, and Smart Tags"). Alternatively, consider using a traditional forms program such as FormTool or FormFlow.

Note

The tools Word provides for building Web forms—available through the *Control Toolbox toolbar*—can also be used in standard Word forms.

APPLICATIONS FOR WORD'S FORMS FEATURE

You can use Word's forms feature to build three types of forms:

- Standard electronic forms that are filled out in Word by users whose responses are limited to specific areas and types of information
- Guided electronic forms where you display a series of questions and the forms can fill themselves in as the user provides answers
- Printed forms that can be completed with a typewriter or a pen

In addition to fairly obvious applications such as questionnaires and surveys, forms can also be used in more traditional word processing functions such as automated document production. Lawyers use forms to fill out contracts, whereas bankers use them to complete loan applications.

In fact, Word's forms feature can help you build any document that is largely repetitive except for small areas of specific, individual information.

Understanding the Workflow Associated with Building and Distributing Word Forms

Before you can begin building and using Word electronic forms, it helps to understand the workflow associated with them. In general, you should follow these steps in the order presented:

1. Plan your form. Understand its goals and the information it must elicit.
2. Build a skeleton of your form. Add all the text and images that won't change when a user fills out the form. Leave space for the form's interactive elements—the areas users will fill out. (At this stage, you might want to do preliminary testing of your form with colleagues who will ultimately be working with it.)
3. Add interactivity with form fields. In the spaces you've left for them, add form fields that provide your form's interactivity. These include text form fields that allow users to type text such as names and addresses; check box form fields; and drop-down form fields that allow users to choose from a list of options.
4. Protect your form. This prevents changes to its structure and functionality. Once you've done so, users can enter information only in the areas you've provided for them.
5. Distribute your form. There are several options for doing so. For example, you can save your form as a template, and store it in the network location you use for global templates. Users can then create new copies of the form whenever needed, by choosing the form from the File, New dialog box. Alternatively, you can provide the form as a Word document, stored on an intranet or sent by e-mail to users when they request it.

Planning Your Form

Careful planning can make the difference between an incomprehensible form that users fill out improperly (or not at all) and a clear, usable form that delivers the information you need rapidly and effectively. Here are some ideas for building effective, usable forms.

- **Be Clear About Your Goals**—As mentioned above, make sure you understand exactly what your form is intended to accomplish, and what information it must request in order to accomplish its goals.
- **Organize the Required Information Logically**—For example, place all contact details (name, address, phone, e-mail) in one area; place all financial details in another. Separate fields with headings that are as clear and self-explanatory as possible. Avoid redundancy: The user should never have to fill in the same information twice.
- **Think About How Users Will Work with Your Form**—For example, a form containing credit card information will typically allow the user to specify which type of

credit card is to be used (Mastercard, Visa); then the credit card number; only then the expiration date.

- **Emphasize Usability and Readability**—Use readable typefaces, avoid print that's too small, avoid "ALL CAPS" text, and if you use color, make sure to use high-contrast colors.
- **Leave Sufficient Space for User Information**—This is doubly important if you're creating a printed form that will be filled in by hand.
- **Use Conventions Your Users Will Expect**—For example, if you want users of an electronic form to select only one item from a list, use a drop-down box, not a series of checkboxes that allow them to (incorrectly) select multiple items.
- **Help Users Avoid Errors Up Front**—For example, if a form requires inputs to be in the form of dollars and cents, present a default setting in that format, use Word's tools for enforcing inputs—and give users help that explains specifically what you're looking for.
- **Test**—Try out the form with real live users—*and listen to what they tell you!*

Building the Skeleton of Your Form

Whether you ultimately want your forms to be used as hard copy or an online format, the first step is the same: creating a template containing the "shell" of the form. The shell is the text, layout, and formatting elements that remain constant whenever the form is used. To create a template from scratch, choose File, New; click General Templates on the New Document task pane; choose Template in the Create New area of the dialog box; and click OK.

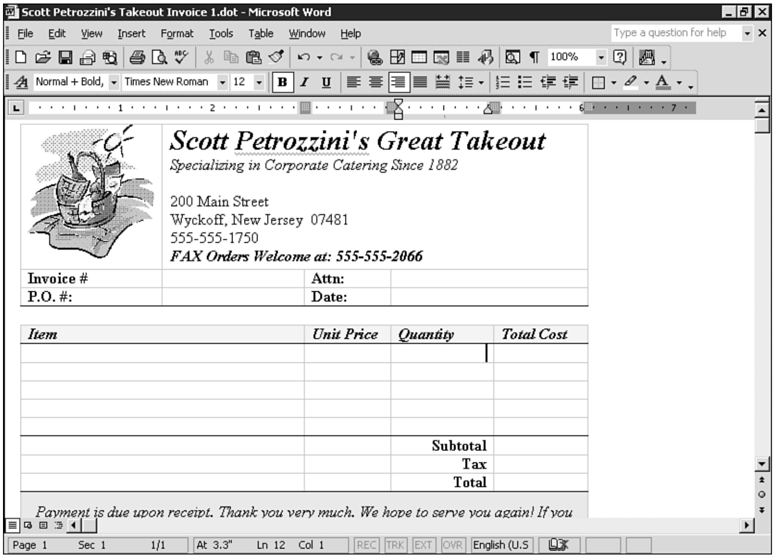
Note

If you originally created a printed form in Word and you now want to turn it into an electronic form that can be filled out from within Word, open the original Word file and resave it as a template.

After you create your template (or save an existing document as a template), you must create framework for your form. You can use all Word's editing, formatting, and drawing tools, just as if you were creating any other kind of document. Most forms make heavy use of the following features:

- Tables (see Chapter 13, "Tables: Word's All-Purpose Solution for Structuring Information ")
- Text boxes (see Chapter 16, "Word Desktop Publishing")
- Borders and shading (see Chapter 5, "More Day-to-Day Productivity Tools")

Leave empty spaces (or placeholder characters such as &&&&) for the areas of the form you want users to fill in. Later, you'll learn how to use form fields that transform those empty spaces with interactivity and automation, enabling users to enter information more quickly and accurately. Figure 1 shows the skeleton of a form with all structure, text, and graphics in place.


Scott Petrozzini's Takeout Invoice 1.dot - Microsoft Word
File Edit View Insert Format Tools Table Window Help
Type a question for help
100%
Normal + Bold, Times New Roman 12
B I U
Page 1 Sec 1 1/1 At 3.3" Ln 12 Col 1 REC TRK EXT OVR English (U.S

Scott Petrozzini's Great Takeout
Specializing in Corporate Catering Since 1882

200 Main Street
Wyckoff, New Jersey 07481
555-555-1750
FAX Orders Welcome at: 555-555-2066

Invoice #		**Attn:**	
P.O. #:		**Date:**	

Item	***Unit Price***	***Quantity***	***Total Cost***
		Subtotal	
		Tax	
		Total	

Payment is due upon receipt. Thank you very much. We hope to serve you again! If you

Figure 1
The skeleton of a form, awaiting the use of form fields.

Note

Your form template should also contain any macros and AutoText entries you can create to streamline filling out the forms later.

If you can't edit a form one of your colleagues has created, see "What to Do if Word Won't Allow You to Edit an Existing Form," in the "Troubleshooting" section of this chapter.

After you've built the skeleton for your form, make sure that you save it as a template under a new name, preferably a descriptive one. If your organization numbers its forms, you might include the new form number in the name.

By default, Word saves all templates in the Templates folder; saving your file here makes it appear in the General tab of the New dialog box. You can also save it in any of the Templates subfolders, such as Letters & Faxes, Memos, Publications, and so on, which correspond to the other tabs of the New dialog box.

To enable users to access the form easily across a network, store it in a location to which they have access, such as the Workgroup Templates folder set up on your network. Of course, no matter how you choose to distribute your electronic forms, you should password protect them so that they cannot be changed without authorization.

Note

If you have a form that you want everyone to fill in, you can send the template as an attachment on your electronic mail network. (Remember to add instructions on what to do with the form.)

Outlook 2002 has its own form-generating capability, and if you need strong support for messaging and workflow–especially if your organization also uses Microsoft Exchange–consider building the form with Outlook, instead of Word 2002.

Word provides the Send To option on your File menu, with options for routing your form or sending it as an attachment.

If you want to edit the form's structure after completing it, you must open the template itself, not a document created from the template. To make sure that you're doing so, change the Files of Type to Document Templates in the Open dialog box.

→ For more information about working with templates, **see** Chapter 12, "Templates, Wizards, and Add-Ins," **p. 357**.

ADDING INTERACTIVITY WITH FORM FIELDS

So far, you've only seen how to create a document that resembles a form—in other words, the skeleton of a form. In the rest of this chapter, you learn how to add interactivity and automation to your form with *form fields*. Form fields are special document elements that make it possible to fill out the form more easily and quickly, and to use the form's information after it's there.

WORKING WITH THE FORMS TOOLBAR

Working with forms requires access to a specific set of tools. Word 2002 has grouped the essential commands for creating and editing a form together in the appropriately named Forms toolbar. Enable this toolbar as you would any other: Choose View, Toolbars, and then choose Forms from the submenu. Figure 2 shows the Forms Toolbar.

Note

In Word 2002, displaying the Forms toolbar is the only way to work with form fields.

As with any other toolbar, you can drag it to any screen edge, if you want to, so that it doesn't interfere with your work.

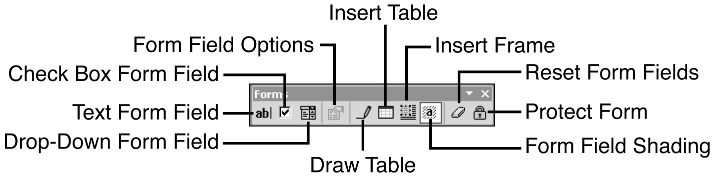


Figure 2
The Forms toolbar.

Ten buttons appear on the Forms toolbar:

- Text Form Field inserts a text form field where users can enter text, numbers, symbols, and spaces. You can also use text form fields to make calculations based on entries that users make in other form fields.
- Check Box Form Field inserts a check box in your document; users can either check the box or leave it unchecked.
- Drop-Down Form Field inserts a drop-down form field that gives a user a list of alternatives; the user is limited to choosing one of those alternatives.
- Form Field Options is used to specify the detailed settings for any form field after you've inserted it.
- Draw Table displays the Tables and Borders toolbar, which contains buttons the form designer can use to build tables more easily and activates the Draw Table tool.
- Insert Table inserts a table with a specific number of columns and rows, all of them the same height and width. This is the same Insert Table button that appears in the Standard toolbar.
- Insert Frame includes a free-floating box in the form that you can format with a precise size and location. (Insert Frame does not insert text boxes.)
- Form Field Shading toggles your form's text, check box, and drop-down fields between medium gray shading and no shading.
- Reset Form Fields clears all entries that have been added to a form.
- Protect Form enables you to protect all areas of a form from change, except for those that users are intended to fill in.

Tip from
Bill Camarda

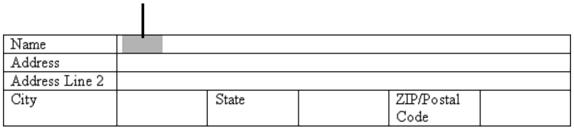
As you build your form, you'll often want to use Protect Form to toggle between protecting the form (which shows how it will look and act when users work with it) and unprotecting the form, so you can make changes to it.

INSERTING A TEXT FORM FIELD

When it comes to basic forms, by far the most common type of entry is plain text. Most forms, at a minimum, gather your name and address—and others may require everything from your e-mail address to your hat size. Text form fields are the workhorses of forms.

To enter a standard text form field, display the Forms toolbar, position your insertion point where you want the field to appear, and click the Text Form Field button. You've just inserted a text form field; it appears shaded in your document (see Figure 3).

Shaded Text Form Field inserted in document



Name	▒				
Address					
Address Line 2					
City		State		ZIP/Postal Code	

Figure 3
A text form field as it appears after being inserted in a document.

If form fields seem to disappear while you're trying to work with them, see "What to Do if Form Fields Disappear," in the "Troubleshooting" section of this chapter.

If you see codes such as { FORMDROPDOWN } where you expect to see elements such as drop-down boxes, see "What to Do if You See Field Codes Such as { FORMDROPDOWN }" in the "Troubleshooting" section of this chapter.

The generic 1/2" wide Text Form Field button you've just placed in your document enables users to insert any text, of any length. But you may want to restrict what users can enter here—and give them some help in entering the information you need.

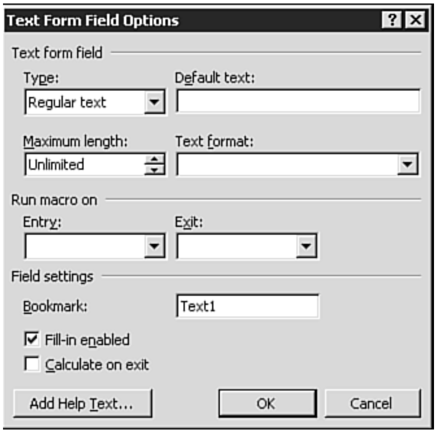
You control the options associated with a text form field through the Text Form Field Options dialog box, shown in Figure 4. This dialog box can be reached in two ways. You can select the field you want to edit and click the Form Field Options button on the Forms toolbar, or you can right-click on the field and click Properties from the shortcut menu that appears.

PLACING A NUMBER, DATE, OR TIME IN A TEXT FORM FIELD

The first aspect of the text form field you can control is whether it should contain text. You can use the Type drop-down list box in the Text Form Field Options dialog box to specify several alternatives, including

- **Regular Text**—The default setting; users can enter anything they want.
- **Number**—This setting restricts entry to numbers and number-related characters and punctuation (for example, $ and : characters). If a user enters a character other than a number, the form stores the number zero instead.
- **Date**—This setting restricts entry to a valid date or time. An error message appears if the user enters something else.

- **Current Date**—This setting inserts a { DATE } field. The field is updated when the document is first opened or created, and afterward in accordance with the traditional Word field updating rules (for example, you can select the field and press F9 to update it).
- **Current Time**—This setting inserts a { TIME } field.
- **Calculation**—This setting tells Word you want the field to perform a calculation. As you'll see later in this chapter, in the "Using Calculations in Text Form Fields" section, if you use this setting, you have to create the equation as well.



Type: Regular text

Maximum length: Unlimited

Bookmark: Text1

☑ Fill-in enabled

□ Calculate on exit

Figure 4
From the Text Form Field Options dialog box, you can control the behavior of a text form field.

Setting Default Information for Your Text Form Field

In many text form fields, users want to insert the same information most of the time. You can automatically provide this information, which then appears automatically in the form unless the user changes it.

To specify default information, display the Text Form Field Options dialog box and enter the information in the Default Text text box. You can enter text, numbers, symbols, or spaces, unless you've restricted the type of data that can be entered—for example, by choosing Current Date, Current Time, or Calculation.

The default information you specify will appear in the form as you typed it; the user can type over it to replace it with new information.

If your users are not familiar with Word forms, you may need to provide instructions that let the user know he or she can change this information.

Controlling the Length of User Input

Many forms are designed to restrict the number of characters a user may enter in a specific area. For example, if your users are entering Social Security numbers, you should limit the number of digits to nine; any more digits than that, and the input is incorrect. You can set the length of a text form field in the Maximum Length scroll box in the Text Form Field Options dialog box.

CONTROLLING CAPITALIZATION IN TEXT FORM FIELDS

Most of the text formatting you apply to form fields is done the same way as other text formatting: by manually applying it to characters or paragraphs, or by using styles. However, the Text Form Field Options dialog box does enable you to control the case to be used in regular text entries. Make sure that Regular Text is selected in the Type drop-down box; then choose Uppercase, Lowercase, First Capital, or Title Case from the Text Format drop-down box.

Note

Word 2002 provides two Text Format options designed for international environments. The first is Double byte, which allows users to enter characters from languages such as Chinese and Japanese whose character sets are far larger. The second is Single byte, which specifies that entries must be made in languages that use single-byte character sets, including English, Spanish, and most Western European languages.

USING NUMERIC FORMATS IN TEXT FORM FIELDS

As you've already learned, you can limit a text form field to one of several types of information. One of the most useful types of information available to you is Number.

Specifying that data be entered as a number takes you one small step toward ensuring data integrity. Nobody can fill in alphabetical characters, for example, in a field that requires a dollar amount.

After you choose Number, you can also specify the format in which the number appears. Make a choice from the Number Format combo box. Among the options Word provides are options to

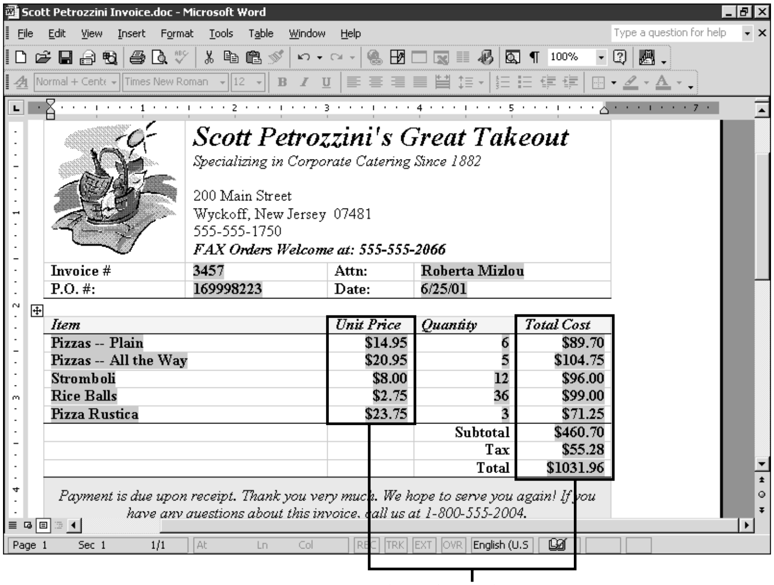
- Include decimal points; for example: `3.28`
- Add commas as thousands separators; for example: `12,516`
- Place negative numbers in parentheses; for example: `(132.14)`
- Use percentage symbols; for example: `38%`
- Use combinations of these symbols

Then, even if the user enters a number in a different format, Word automatically changes it to be consistent with all the other forms you're collecting.

Note

Some elements of the Text Form Field Options dialog box change their names depending on the type of information you choose to include in your text form field.

In the sample invoice shown in Figure 5, text form fields have been added to each table cell beneath Unit Price. These text form fields have been set to appear in dollars-and-cents format.



Scott Petrozzini Invoice.doc - Microsoft Word
File Edit View Insert Format Tools Table Window Help
Type a question for help
100%
Normal + Cent
Times New Roman
12
B I U
Page 1 Sec 1 1/1 At Ln Col REC TRK EXT OVR English (U.S

Scott Petrozzini's Great Takeout
Specializing in Corporate Catering Since 1882

200 Main Street
Wyckoff, New Jersey 07481
555-555-1750
FAX Orders Welcome at: 555-555-2066

Invoice #	3457	Attn:	Roberta Mizlou
P.O. #:	169998223	Date:	6/25/01

Item	*Unit Price*	*Quantity*	*Total Cost*
Pizzas -- Plain	$14.95	6	$89.70
Pizzas -- All the Way	$20.95	5	$104.75
Stromboli	$8.00	12	$96.00
Rice Balls	$2.75	36	$99.00
Pizza Rustica	$23.75	3	$71.25
		Subtotal	$460.70
		Tax	$55.28
		Total	$1031.96

Payment is due upon receipt. Thank you very much. We hope to serve you again! If you have any questions about this invoice, call us at 1-800-555-2004.

Text form fields formatted to appear as dollars and cents

Figure 5
The text form fields have been formatted to appear as dollars and cents.

Tip from
Bill Camarda

You can go beyond standard numeric formats by adding a "numeric picture" in the Number Format combo box. The numeric picture ###.#, for example, tells Word to round off any entry to tenths. Numeric pictures are explained in the "Customizing Numeric Formats to Your Specific Needs" section of Chapter 26, "Automating Your Documents with Field Codes."

USING DATES AND TIMES IN TEXT FORM FIELDS

You can also specify that text form fields include dates and/or times. You can allow users to enter the dates or times, following date and time formats you specify, or you can time stamp your form with the current date or time at the time the form is first opened by a user.

To allow users to enter a date, display the Text Form Field Options dialog box, and choose Date from the Type drop-down box. Choose a Date Format from the drop-down box; if you want to include space for Time, choose a date format that contains `h:mm`.

If you don't want your users to be able to enter a date, but rather want your form field to display the date when the form was opened for the first time, choose Current Date from the Type drop-down box. If you instead want to display the *time* the field was opened, choose Current Time from the Type drop-down box.

Caution

Remember that a user can still circumvent your date and time settings by resetting the system clock, thereby making it appear that a form was filled out sooner than it really was.

Caution

When you're creating or editing your form, get in the habit of protecting your document before you test your changes. Otherwise, selecting the form field and then entering some text overwrites any default value you have established for the field. To quickly lock and unlock your form, use the Protect Form button on the Forms toolbar.

After you test a form field, make sure to clear all entries you don't want to be included in the final form. You can do this by clicking the Reset Form Fields button.

Other features of the Text Form Field Options dialog box are covered later in the chapter. In particular, adding Help Text is covered in the "Adding Help to Your Forms" section, and running macros is covered in the "Running Macros from Form Fields" section.

USING CALCULATIONS IN TEXT FORM FIELDS

In Chapter 13, you learned that Word can perform simple calculations within tables, much as a spreadsheet program such as Excel can. Forms can especially benefit from this capability. For example, you can create invoices that calculate totals based on how many of each item someone orders, and how much each item costs.

Use the Text Form Field Options dialog box to build calculations. First, choose Calculation as the Type of data you want to insert. To the right of the Type drop-down box, the Expression text box appears. An equal sign is placed in the text box. (All formulas in Word begin with an equal sign.) You can now use any of Word's basic calculation techniques. The following tips may be helpful.

To specify values in table cells, you can specify the table cells. The first row of a table is row 1. The first column of a table is column A. The cell at the top left of a table, therefore, is cell A1. To add the contents of cells A1 and A2, enter

```
=A1+A2
```

Because every form field has a corresponding bookmark, you can also use bookmarks, such as

```
=Quantity1+Quantity2
```

You can also use any of the 18 functions Word provides. The simplest are SUM for addition and PRODUCT for multiplication. For example, to add the numbers in cells B1 and B2, enter

```
=SUM(B1,B2)
```

Or, to add all the numbers above your current cell, enter

```
=SUM(ABOVE)
```

To multiply the numbers in cells B1 and B2, enter

```
=PRODUCT(B1,B2)
```

USING CALCULATE ON EXIT TO AUTOMATE YOUR CALCULATIONS

If you've ever worked with a spreadsheet program such as Excel and then had to do calculations in a Word table, you've probably wished that Word could automatically recalculate formulas every time a value was changed. With the Calculate on Exit feature for form fields, Word can recalculate specific form fields whenever a user makes a change to a value he or she depends on. Calculate on Exit works when you click or tab away from the form field that contains this setting. (You don't have to wait until you exit Word or close the document.)

Because Calculate on Exit requires information from other fields in your form, setting this up is a bit more complicated than just inserting a field or two. To use Calculate on Exit

1. Click the Text Form Field button on the Forms toolbar to place text form fields where you want to enter your numbers. Then insert a final form field to use for the value you are calculating.
2. In all but the final form field (where the calculated value is to appear), first change the Type from Regular Text to Number in the Text Form Field Options dialog box. Next, check the Calculate on Exit check box. If you want, you can also change the Number Format. Note the name of each form field in the Bookmark text box.
3. In the text form field that is to display the total, first change the Type to Calculation. Next, put your formula in the Expression text box. You can refer to the other form fields through their bookmark names. A typical formula that, for example, might add up the total of three form fields would read "=SUM(TEXT1,TEXT2,TEXT3)".

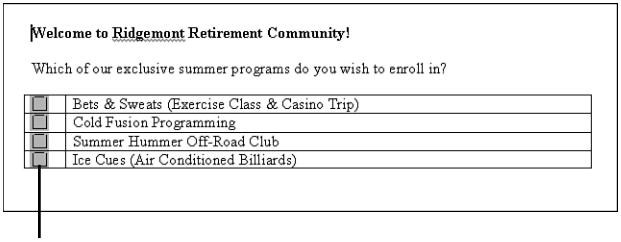
   It is very important that you do not select the Calculate on Exit option for this final form field box.
4. After you close the text form field dialog boxes, lock the form by clicking the Protect Form button on the Forms toolbar.

Now any time you enter a number into your form fields, the formula will automatically recalculate.

ADDING CHECK BOX FORM FIELDS

Check boxes are a handy way to enable users to select one or more options that are not mutually exclusive. For example, in Figure 6, check boxes indicate that a user can sign up for as many courses as he or she wishes.

Figure 6
An example of how check boxes can be used in a form.



Welcome to Ridgemont Retirement Community!

Which of our exclusive summer programs do you wish to enroll in?

□	Bets & Sweats (Exercise Class & Casino Trip)
□	Cold Fusion Programming
□	Summer Hummer Off-Road Club
□	Ice Cues (Air Conditioned Billiards)

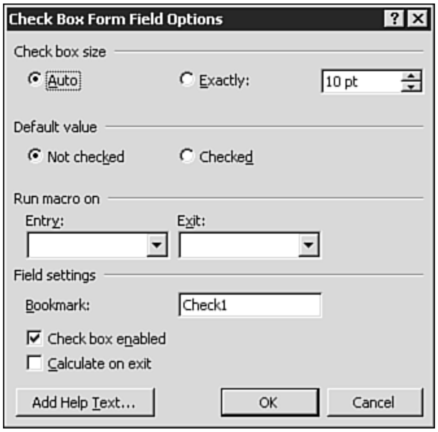
Check box form field

Check boxes are generally used when it doesn't matter how many of the items in a group your user can select. For example, a check box is the method of choice when you see the phrase "Check all that apply" in a survey or questionnaire.

To insert a check box, place your insertion point where you want the check box to appear and click the Check Box Form Field button on the Forms toolbar.

By default, Word displays boxes unchecked. If you want a box to appear checked by default, or to change other options associated with a check box, display the Check Box Form Field Options dialog box, shown in Figure 7.

Figure 7
Controlling the behavior of a check box form field.



Check Box Form Field Options

Check box size
● Auto ○ Exactly: 10 pt

Default value
● Not checked ○ Checked

Run macro on
Entry:
Exit:

Field settings
Bookmark: Check1
☑ Check box enabled
☐ Calculate on exit

Note

Opening the Check Box Form Field Options dialog box is the same as for doing so with a normal text box. Select the check box form field you just created and click Form Field Options on the Forms toolbar; or right-click on the form field and choose Properties from the shortcut menu.

To specify that a check box appear checked by default, choose Checked in the Default Value area. While you're here, you may want to consider some other settings as well.

Tip from
Bill Camarda

Whether you prefer a default setting of Checked or Not Checked, there may be times when you want to prevent users from changing the default. For instance, you might want to include a check box in your form to indicate that you plan to make an option available in the near future, but prevent the user from checking the box until the option becomes available.

To prevent the user from changing the setting you specify, clear the Check Box Enabled check box in the Field Settings area.

By default, Word keeps your check box the same size as the text that follows it; if that text changes size, so does your check box. The Check Box Form Field Options dialog box, however, enables you to change the size of the check box without changing the size of any surrounding text. For example, you could enlarge a box for emphasis.

To specify the precise size of a form field check box, choose the Exactly option button in the Check Box Size area. Then enter the new size in the spinner box.

Later, after you protect the form and make it available to users, they will see a square shaded box. To check it, they can click it once, or press either the spacebar or the X key. The same techniques uncheck a box that's already checked.

➔ For information about adding Help and macro functionality to check box form fields, **see** "Adding Help to Your Forms," on **p. 19**, and "Running Macros from Form Fields," on **p. 21**

Adding Drop-Down Form Fields

Often, you'll want to give users a specific set of options from which to choose—and prevent them from entering any other alternative. For example, you might create a form that asks your telephone customer service representatives to specify which product family a caller is calling about. Word's drop-down form fields enable you to do this.

Tip from
Bill Camarda

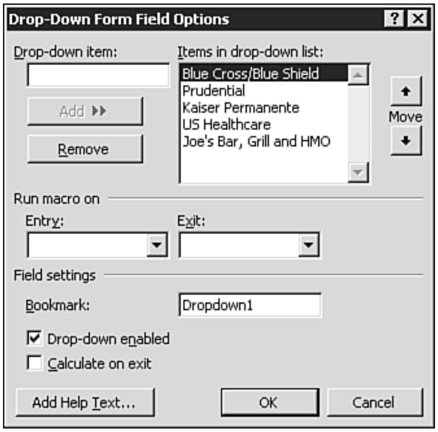
In other forms software, round radio buttons are sometimes used to indicate that only one choice is possible (for example a field that must be answered "Yes" or "No"). Word does not provide radio buttons for electronic forms;

however, drop-down fields are a good substitute in forms that will be filled out electronically.

Drop-down fields are *not* an adequate substitute in forms designed to be printed and hand-filled, since they generally don't show every option.

To add a drop-down form field to a form, display the Forms toolbar and click the Drop-Down Form Field button. If that's all you do, however, your form will contain a drop-down form field with no options. You can supply the options by displaying the Drop-Down Form Field Options dialog box (see Figure 8).

Figure 8
Creating options for a drop-down form field.


Drop-Down Form Field Options

Items in drop-down list:
- Blue Cross/Blue Shield
- Prudential
- Kaiser Permanente
- US Healthcare
- Joe's Bar, Grill and HMO

Bookmark: Dropdown1

☑ Drop-down enabled

☐ Calculate on exit

Note

To display the Drop-Down Form Field Options dialog box, select the drop-down form field you just created and click Form Field Options or right-click on the form field and choose Properties from the shortcut menu.

After the Drop-Down Form Field Options dialog box is open, you need to populate the drop-down list with items that will be available to your form's users. One by one, type the items in the Drop-Down Item box and click Add. Sometimes you won't think of, or enter, list choices in the order you want them to appear when a user accesses the drop down list. If necessary, select items you need to move and click the Move Up or Move Down arrows.

Word treats the first item in your list as your default choice—it's the one that appears selected when the user opens the form, as shown in the sample in Figure 9.

Suppose that you no longer need one of your options; perhaps you've discontinued a product or service. You can delete the option from your form by selecting it and choosing Remove.

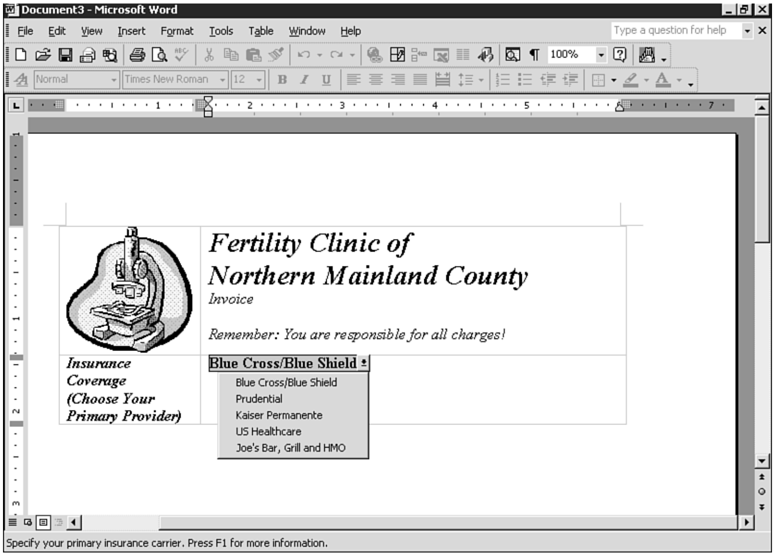


Figure 9
Sample use of a drop-down form field.

Document3 - Microsoft Word
File Edit View Insert Format Tools Table Window Help
Type a question for help
100%
Normal
Times New Roman
12

Fertility Clinic of
Northern Mainland County
Invoice

Remember: You are responsible for all charges!

Insurance Coverage (Choose Your Primary Provider)	**Blue Cross/Blue Shield**

- Blue Cross/Blue Shield
- Prudential
- Kaiser Permanente
- US Healthcare
- Joe's Bar, Grill and HMO

Specify your primary insurance carrier. Press F1 for more information.

As with check box form fields, there may be times when you want to specify choices but not allow users to access them. Perhaps you plan to make the choices available later, but for the time being users are stuck with your default option. To disable the drop-down list box while still displaying the default option, clear the Drop-Down Enabled check box.

➔ For information about adding Help and macro functionality to drop-down form fields, **see** “Adding Help to Your Forms,” on **p. 19**, and “Running Macros from Form Fields,” on **p. 21**.

Formatting Your Form Fields

You can format form fields just as you would any other text characters: select its contents and apply formatting using the Formatting toolbar, keyboard shortcuts, or menu commands. For example, you can make text and drop-down form fields bold, italic, or underline; change the font name or size; and so on. The field itself retains the same gray shading. You can toggle this gray shading on and off with the Form Field Shading button on the Forms toolbar.

Most of the formatting you add—especially font formatting—isn’t evident unless text appears in the field, either as a default setting or as entered by a user.

Note

The form must be unprotected for you to change the formatting of a form field.

Adding formatting such as boldface to text and drop-down fields is a good way to make it easier for users to see the information they (or others) have inserted in the form.

Caution

When applying font formatting in a form, make sure that you use fonts that will be available on all the computers using your form. Otherwise, Windows may substitute fonts that look unattractive or are difficult to read.

This isn't a problem if you use basic Windows fonts such as Arial or Times New Roman. However, if you want to use more interesting fonts other users might now have, then there is another option: Choose Tools, Options, Save and check the Embed TrueType Fonts check box. When enabled, this check box tells Word to actually store a copy of the font in the template.

Be aware that a template containing fonts increases the file's size, making it slower to open and close or send through e-mail. Also, some fonts cannot be embedded due to manufacturer restrictions. To discover whether a font contains embedding restrictions (and to learn more about the detailed characteristics of a font), download the ttfext add-in from Microsoft's Web site, at `http://www.microsoft.com/typography/property/property.htm`.

REMOVING A FORM FIELD FROM YOUR FORM

As you create and modify form templates, you'll likely find that some fields are no longer needed. To remove a form field from your document, first click on it to select it. The field turns a darker shade of gray than normal. Once selected, pressing Delete removes the form field; you can also click Cut on the Standard Toolbar, press Ctrl+X, or choose Edit, Cut.

Note

If your document is protected, you must first unprotect it. Choose Tools, Unprotect Document, or click the Protect Form button on the Forms toolbar.

Caution

While Cut works to remove an unwanted form field, you need to be more careful about using Copy and Paste to create multiple copies of a form field.

If calculations are involved (as in a column of form fields that each calculate a line on an invoice) you need to manually update each form field to make sure it is calculating the correct bookmarked form fields.

Also, if the form field you copy contains a bookmark, you need to manually create new bookmarks for all the copies.

Advanced Form Field Features

Word provides some powerful form field automation options that are available to you regardless of the types of form fields you're using. These include

- Adding help to your forms, including anything from a simple message in the status bar to a more detailed message that appears when the user presses F1. (Unfortunately, Word provides no way to add a ScreenTip to a form field—something your users might now be trained to expect.)
- Associating form fields with bookmarks.
- Running macros when users enter or exit a form field. This means that your form can help complete itself, based on the specific entries the user makes.

Adding Help to Your Forms

If you're in charge of helping people fill out their forms, you can cut down dramatically on the support you need to provide by adding built-in help to your online forms. Word's built-in help for forms can provide more detailed explanations than your form itself may have room for. You can use it to elaborate on the options you're offering, the information you want to collect, or how to use the form itself.

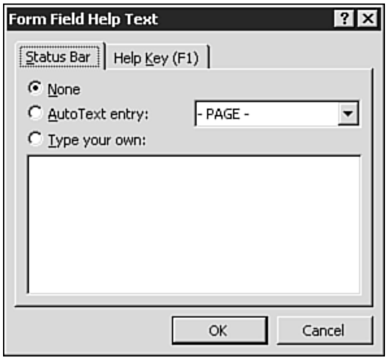
Tip from *Bill Camarda*

You should give at least basic help in the form itself, where the help is visible for people who don't know how to look for it. Type language such as the following directly onto the form:

To get help about any item, move to it with the mouse or the keyboard and press F1.

To add help text, create the type of form field you need, right-click on it, and choose Properties; its Form Field Options dialog box appears. Click the Add Help Text button. The dialog box shown in Figure 10 appears.

Figure 10
In the Form Field Help Text dialog box, you can specify where help comes from, where it appears, and what it says.



● None
○ AutoText entry: - PAGE -
○ Type your own:

You now have two choices to make: where your help message appears and where its contents come from. If you want help to appear in Word's status bar, click the Status Bar tab. If you want it to appear when the user presses F1, click the Help Key (F1) tab.

Note

You can create both kinds of help, by placing entries in each tab, as discussed in the next section, "Using Both Forms of Help Together."

No matter which tab you choose, you have the same two sources for your help text:

- An existing AutoText entry containing boilerplate text you've already created. (Choose the AutoText entry you want from the AutoText Entry drop-down box, which lists all AutoText entries that are available to the template you're working in.)
- New text (type it in the Type Your Own text box)

If you choose to use an AutoText entry, it's quite likely that this entry does not yet exist. To create an AutoText entry, type the text in your document; then choose Insert, AutoText, New. Enter a name for your AutoText entry, and choose OK.

Chances are, most of the time you will use Type Your Own rather than a pre-existing AutoText Entry—if for no other reason than that they are conveniently stored in this dialog box where they can be easily revised as needed. You might, however, use AutoText entries if you are creating a help entry you intend to use in multiple forms or multiple locations. This way, if you need to update the entry later, it will automatically be updated in all the locations where it appears.

Caution

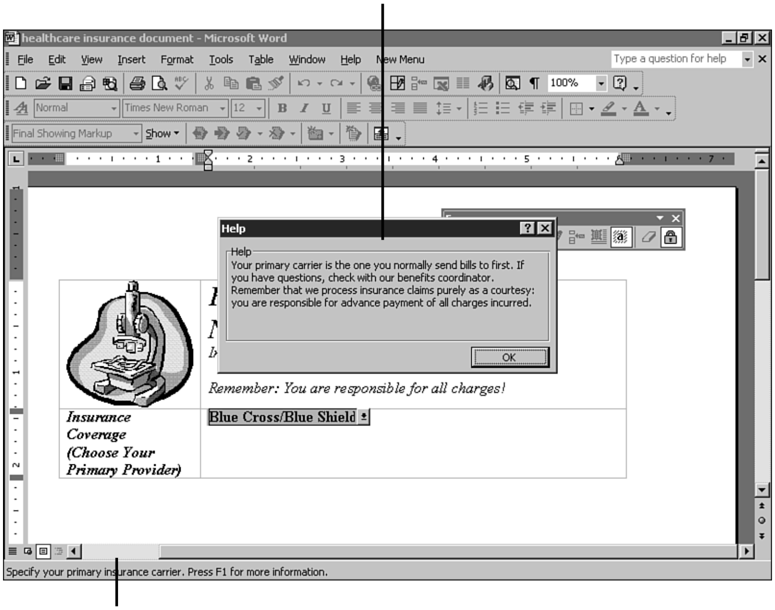
If you use an AutoText entry, be aware that neither the status bar nor these Help dialog boxes can contain graphics, even though graphics can be stored in AutoText entries.

When you finish creating help, click OK.

USING BOTH FORMS OF HELP TOGETHER

You don't have to choose between offering help in a dialog box or in the status bar. For example, you might provide abbreviated help in the status bar, ending the status bar message with "Press F1 for more help." You could then associate more extensive information with the F1 key, so that when the user presses it, Word displays the added help information in a special Help dialog box. (Status bar help is limited to 138 characters; help presented in a dialog box can be up to 255 characters in length.) Figure 11 shows how both kinds of help can complement each other in this fashion.

Figure 11
Providing help to users of a form.



Help dialog box (appears when user presses F1)

Help in status bar

Help

Your primary carrier is the one you normally send bills to first. If you have questions, check with our benefits coordinator. Remember that we process insurance claims purely as a courtesy: you are responsible for advance payment of all charges incurred.

Remember: You are responsible for all charges!

Insurance Coverage (Choose Your Primary Provider)	Blue Cross/Blue Shield

Specify your primary insurance carrier. Press F1 for more information.

Some users provide simple instructions on using the form in the status bar and provide background about how to interpret the form's questions in the dialog box.

For example, in a travel reimbursement form, the status bar might say "Enter total airline ticket cost," whereas the F1 key might summarize the company's travel reimbursement policies.

Running Macros from Form Fields

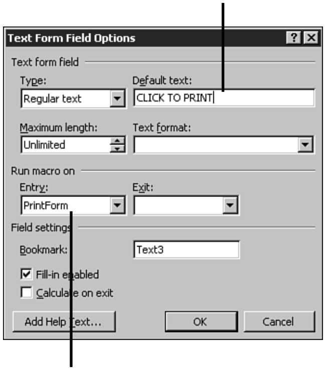
You can instruct Word to run a macro whenever a user enters or leaves a field. In either case, you can select from macros available in your current document or template.

Tip from
Bill Camarda

If you already have a macro you want to use, copy it to your form template using the Organizer. You can find more information about working with the Organizer in Chapter 11, on **p. 331**.

Figure 12 shows an example of how you might use this feature. In this example, a macro, PrintForm, has been recorded. PrintForm sends the form to a user's default printer. Whenever a user tabs to—or clicks in—the CLICK TO PRINT field, the file prints automatically.

Enter instructions or explanation



Text Form Field Options

Text form field

Type: Regular text

Default text: CLICK TO PRINT

Maximum length: Unlimited

Text format:

Run macro on

Entry: PrintForm

Exit:

Field settings

Bookmark: Text3

☑ Fill-in enabled

☐ Calculate on exit

Choose macro to run on Entry

Figure 12
Creating a form that includes a Print button.

Every form field is automatically assigned a bookmark name. This means that it is easy to create Word macros that check the current contents of a bookmark and, based on what they find there, place corresponding contents in other fields. For example, when a user inserts a name, the macro can automatically look up that name in a database and insert the corresponding company and address. Or, when the user checks a check box, the macro can enable other form fields that were previously grayed out.

After your macro is written, linking it to a form field is straightforward. First, double-click the form field to which you want to attach the macro, or right-click on the field and choose Properties. This displays the Properties dialog box associated with the form field.

Note

If double-clicking on the form field doesn't work, choose Tools, Unprotect Document to permit changes to form fields.

Second, in the Run Macro On section, choose between Entry or Exit (you can create different macros for each, if you want). Third, select the macro from those listed. Click OK, and you're finished. As always, make sure to lock the document using the Protect Form button on the Forms toolbar after you make your changes.

➔ For more information about recording macros that can be used in forms (or elsewhere), **see** "Recording and Running Visual Basic Macros," Chapter 28, on **p. 871**.

CONVERTING ELECTRONIC FORMS TO PRINTED FORMS

It's common to create forms that will be used both from within Word and in printed versions. Word delivers "What You See Is What You Get" formatting, so your forms appear in print exactly as they do onscreen. (There is one exception: The gray rectangles that mark form fields onscreen do not appear in print.)

Having said this, the art of creating an easy-to-use printed form varies slightly from the techniques you need for electronic forms. Keep these pointers in mind:

- When they're working in Word, users can enter large amounts of data even in a tiny text form field (as long as you haven't limited the maximum length of their entries). Of course, this isn't true in print. Make sure that you leave sufficient space in your design for users to enter all the text they need to enter.
- You need to remove drop-down fields (so that defaults are not printed in your form, preventing users from making other choices). Then you need to reformat your forms so that users can see a list of the choices from which you want them to select.
- You may need to enlarge Word's default check boxes to make them easily visible and usable in print.
- If you're designing a form that will be filled out on a typewriter, make the font size 12 or 10 points. The type elements for most typewriters are usually one of these two sizes, and it makes aligning the responses much easier.
- The Drawing toolbar enables you to create a variety of shapes perfect for forms: straight lines, arrows, boxes, circles, and numerous AutoShapes. Moreover, they can all be independently positioned and aligned.
- A table is an easy way to create a series of evenly spaced lines. You can turn off all but the bottom border and set the table height to be exactly a certain point size. The Forms toolbar has both Insert Table and Draw Table buttons.
- If you need to position an element of the form precisely, use a frame or a text box. To insert a frame, click the Insert Frame button on the Forms toolbar and drag the mouse pointer until you reach the approximate size you need. Then you can refine the size and position by right-clicking on the frame and choosing Format Frame from the shortcut menu.
- If your electronic form contains a time stamp, when you print it, the time stamp will represent the time of printing.
- Rather than use check box form fields, consider creating your check boxes through Format, Bullets and Numbering. Select the hollow square check box bullet. You can select any other symbol as a check box by clicking the Customize button on the Bullets and Numbering dialog box and then selecting the Bullet button. Click the Font button to change the bullet's size. Turn off your checklist by clicking the Bullet button on the Formatting toolbar.

Protecting an Entire Form

Word lets you control the changes that can be made to a form. When you protect a template, a user can't make any changes in documents based on that template, except where you've inserted form fields. In fact, a form doesn't behave like a form until you protect it.

When you protect a form, some form fields that provide specific information can't be changed either. In a text form field that makes a calculation, for example, the user can't override the calculation. And, as you've seen, unchecking the Fill-in Enabled check box in the Text Form Field Options dialog box also prevents a user from making changes in that field.

To protect a form, first open the form and then choose Tools, Protect Document. The dialog box shown in Figure 13 appears.

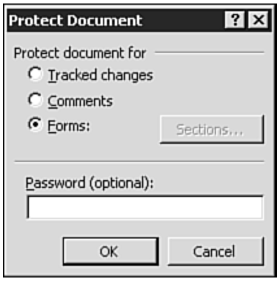


Figure 13
Protecting a form.

If you want, you can add a password. Including one probably makes sense if your form is to be used in a large organization where someone might feel like editing it inappropriately. (You don't want sabbaticals in Tahiti added to your benefits option form.)

Unlike protecting an entire document, form passwords don't encrypt the document. Users can still open a password-protected form template; they just can't unprotect and edit it. When a user does try to unprotect such a document, the Unprotect Document Password dialog box opens, as shown in Figure 14.

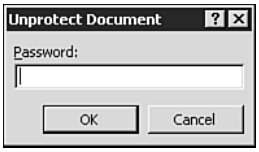


Figure 14
The Unprotect Document password dialog box.

As always when creating password names, choose a password you'll remember but nobody else can figure out. Don't write it down and leave it in an obvious location. And remember, after you create and confirm a password, you have no way to unprotect the document without the password.

To remove a password, first open the document (using its password). Then unprotect it using the Unprotect Document option on the Tools menu. Then protect it again by choosing Protect Document from the Tools menu. Choose Forms in the Protect Document dialog box. No password appears in the Password box. If you don't want the file to be protected by a password, click OK. When you save the file, it no longer requires a password. (If you want a new password, type it and confirm it. After you save the file, the new password goes into effect.)

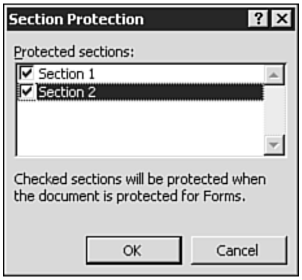
Protecting a Section of a Form

Word enables you to divide your document or template into multiple *sections* and protect only one section. This enables users to edit at will elsewhere in their form documents—but when they come to the section you've set up as a form, they're restricted to making changes in only the contents of form fields.

To protect only a section of your form document or template, start by separating your document into sections by choosing Insert, Break and choosing one of Word's section break types. (If you don't want extra page—or column—breaks in your form, choose Continuous.)

Then, to protect a specific section

1. Choose Tools, Protect Document.
2. Click the Forms button if it isn't already selected.
3. Click Sections. (If the Sections button is not available, either Forms is not selected or you don't have any section breaks in the document.) The Section Protection dialog box opens (see Figure 15).



☑ Section 1
☑ Section 2

Checked sections will be protected when the document is protected for Forms.

Figure 15
You can specify which sections of your form you want to protect.

4. Clear any sections you do not want to protect; check the sections you do want to protect.
5. Click OK when you're finished and click OK again to close the Protect Document dialog box.

Filling In Online Forms

To fill in an online form, create a new document based on the template that contains the form. Each form field is shaded in gray. The first field is shaded in deeper gray; that's where your insertion point is.

Unless you specified a short maximum length for your field, the gray area extends as you type. If the field is located in a text cell, the text simply wraps when you reach the end of a cell.

After you fill in a form field, press Tab or the down-arrow key: Both move you to the next form field in which you might make an entry. Word skips over form fields that it automatically calculates and fields in which you disable user input.

Table 1 shows Word's editing and navigation keys for editing forms. As you can see, some keys work a little differently in forms compared with other documents.

TABLE 1 WORD'S FORM EDITING COMMANDS

To Do This	**Use This Key or Combination**
Move to the next editable field	Tab or down arrow
Move to the preceding editable field	Shift+Tab or up arrow
Show the contents of a drop-down list	F4 or Alt+down arrow
Move up or down in a drop-down list	Up arrow or down arrow
Make a selection in a drop-down list	Enter
Mark or unmark a check box	Spacebar or X
Show help for a form	F1 (if you specified that Word display a dialog box to show help; otherwise, help appears in the status bar)
Insert a tab	Ctrl+Tab

Once you've filled out the form, you can save it, print it, or forward it to whoever is responsible for processing it—for example, by using File, Send To, Mail Recipient (as Attachment).

SAVING ONLY THE DATA IN A FORM

One of the key reasons for using a form is to collect data. Data is best stored in a database where it can be sorted, filtered, and output in various forms. Word gives you an easy method to extract the information from filled-out forms in a format that makes it easier to import into a database. You can save only the data in a form (as opposed to the entire contents of the form, including the skeleton framework surrounding the form fields). To save only the data

1. After your form has been filled out, click the Save button on the Standard toolbar.
2. In the Save As dialog box, choose Tools, Save Options.
3. In the Save (Options) dialog box, check the Save Data Only for Forms check box and click OK. The file type changes to Text Only.
4. Choose a filename and folder location for your file. Click Save.

Tip from
Bill Camarda

If you want your forms to only be saved as data only, regardless of who uses them or how often choose Tools, Options, click the Save tab and then enable the Save Data Only for Forms check box. This way users don't have to worry about selecting Save Data Only for Forms in the Save As dialog box.

When Word saves just the form data, it uses comma-delimited fields. You might be familiar with the concept of a comma-delimited field if you've worked with mail merges. The information from each field is placed in quotes and separated by commas. For example, one data file might look like this: "John","Johnson","123 Somter Street","Avery","SC","29678". The data appears in Word's default tab order for the form's fields, or, if you set the tab order using macros, it saves in the order you specified.

This file format is used for both text form fields and drop-down form fields; for drop-down fields, Word includes the item selected by the user. Information returned from a check box form field is handled slightly differently. A checked box shows up as a 1, whereas an unchecked box is a 0. Neither appears in quotes.

After you have saved your forms as data, the information can be imported into an existing database in a program such as Microsoft Access. Almost any database program can read comma-delimited fields saved in a text file. If you are comfortable writing Visual Basic for Applications code, you can write a macro to append the information in each form into one master file to make importing even easier.

➔ For more information on working with macros, **see** Chapter 28, "Recording and Running Visual Basic Macros," **p. 871**.

PRINTING ONLY THE DATA IN FORMS

Just as you might sometimes want to save only the data in forms, occasionally you might want to print only the data in a form. For instance, your form may be many pages long; you might want to have a quick printed record of your responses without printing the whole form. To print only the data, choose Tools, Options. On the Print tab, check the box marked Print Data Only for Forms.

Note

When you check Print Data Only for Forms, Word will only print the form's data until you uncheck this box.

This setting applies to the current form; if you save the form as a template, it also applies to new forms created based on this template.

Troubleshooting

What to Do if Word Won't Allow You to Edit an Existing Form

The original designer of the form has probably protected it from changes. To unprotect the form, choose Unprotect Document from the Tools menu. If your colleague has created the form with a password, you need to know the password to gain access.

The sections "Protecting an Entire Form" and "Protecting a Section of a Form," earlier in this chapter, cover protecting and unprotecting forms in more detail.

What to Do if Form Fields Disappear

Check to make sure that your forms are locked. If the form isn't locked, selecting a form field for entry is just like highlighting a word or a block of text; the next character you enter erases whatever is selected if you have Typing Replaces Selection enabled (Tools, Options, Edit). Always keep your Forms toolbar open and available when you are creating or editing a form. It's much easier to remember to click the Protect Form button than it is to choose Tools, Protect Document.

What To Do if the Tab Order in Your Form Fields is Incorrect

The easiest way of navigating a form when entering data is using the Tab key. However, this is only useful if the order Word uses when tabbing from field to field is a logical one. By default, tab order follows the placement of form fields on the page, starting with the form field closest to the top left of the page. From there, tabbing continues from left to right and down.

However, this may not be the best order for your form. For example, you may want users to answer all of a list of questions in a square area on the left of the form before they answer questions to its right. Forcing Word to go where you want requires you to manually build Visual Basic for Application macros, and incorporate them in each form field to run on exit. You can find detailed instructions for doing this in Microsoft's Knowledge Base article Q212378.

What to Do if You See Field Codes Such as { FORMDROPDOWN }

If you see field codes such as { FORMTEXT }, { FORMCHECKBOX }, or { FORMDROPDOWN } instead of the corresponding text boxes, check boxes, or drop-down boxes in your document, clear the Field Codes check box in the View tab of the Tools, Options dialog box.

How to Use Symbols in { FILLIN } Dialog Boxes

If you use the { FILLIN } field for prompting your users, you have to take extra steps to use certain characters in your prompt. For example, let's say that you wanted your prompt to say: "Type 'M' or 'F' in the box." If you entered that text directly in the Field dialog box, Word would see only the first pair of quotes, and your prompt would read "*Type*". You must preface the quote with a backslash character to have it appear correctly. In the Field dialog box, you would enter the following text in the Prompt text box:

```
"Type \"M\" or \"F\" in the box."
```

Notice that the backslash goes before each quote, not just each quote pair. Similarly, if you ever want to have a backslash appear in your Fill-In prompt, you must use an extra backslash before it. For example

```
"Name the file to be stored in c:\\Invoices\\1997."
```

WORD 2002 MULTILANGUAGE FEATURES

In this chapter

by Paul Colligan

Special Edition Using

QUE®

An Overview of Language Support in Word, Office, and Windows

If your work requires you to edit in multiple languages, or to share your documents with others located outside the United States and English-speaking Canada, you will appreciate the major improvements in multilanguage support that Microsoft introduced with Word 2002 and Office XP. Working in tandem with international support features built into current versions of Windows, you can do all the following:

- Enter text in foreign languages, including special characters such as ã and ç; characters from languages that use other alphabets, such as the Cyrillic alphabet; and characters from ideographic languages such as Chinese.
- Use Word's new Auto-Language Detect feature to automatically detect which language you are writing in and apply relevant formats and styles accordingly.
- Proofread in French and Spanish without purchasing any additional proofing tools, and proofread in 37 other languages by purchasing a single Microsoft add-on product.
- Display the Word and Office user interfaces and access Help in languages other than English.

In this chapter, you will learn how to customize your computer for international use by working with the features built into Windows, Office XP, and Word 2002, as well as powerful options such as Microsoft's Language Pack.

Most of the Language features detailed here are part of Office XP and Windows and are described interchangeably. Where most of the features are also available in the standalone version of Word 2002, a complete Office install should be considered if your project has considerable multi language requirements.

Setting Up Windows for International Environments

Some aspects of working with Office in an international environment are controlled in the Windows operating system; others are based on Office settings and components. The following sections walk you through several key elements that depend on Windows, including

- Windows Multilanguage Support
- Regional settings, including how Windows handles dates, times, and punctuation
- Keyboard layouts

Note

If you intend to work in languages that use non-Latin alphabets, you may need to install special fonts that support these alphabets. Issues related to fonts are covered later in this chapter, in the "Unicode and International Font Support in Office XP" section.

Caution

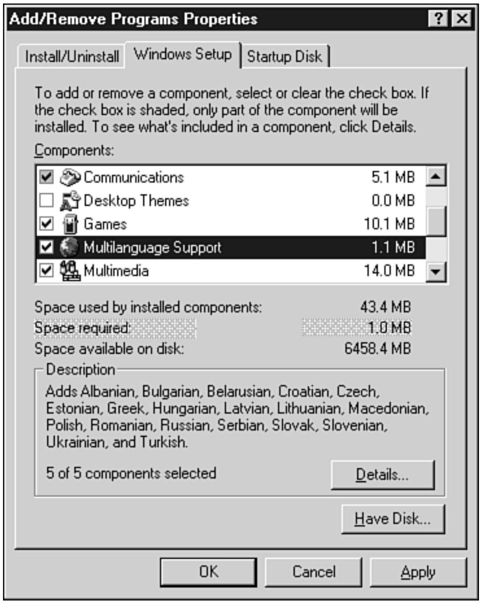
To complicate matters, each version of Windows offers different support for international users. In general, newer versions of Windows provide better support, with Windows 2000 delivering the best support options.

CONFIGURING MULTILANGUAGE SUPPORT IN WINDOWS MILLENNIUM EDITION (ME)

The United States/Canadian and British versions of Windows Millennium (Me) provide built-in support for English and many common Western European languages. However, if you want access to character sets from Eastern Europe, Greece, Turkey, and other locations, you must install the optional Multilanguage Support feature. First, insert the Windows CD-ROM or disks. Then, follow these steps:

1. Choose Start, Settings, Control Panel.
2. Double-click the Add/Remove Programs icon.
3. Click Windows Setup.
4. Check Multilanguage Support if it isn't already checked (see Figure 1).

Figure 1
Adding Multilanguage Support through Windows Me Setup.



To add or remove a component, select or clear the check box. If the check box is shaded, only part of the component will be installed. To see what's included in a component, click Details.

Components:

☑	Communications	5.1 MB
□	Desktop Themes	0.0 MB
☑	Games	10.1 MB
☑	Multilanguage Support	1.1 MB
☑	Multimedia	14.0 MB

Space used by installed components:	43.4 MB
Space required:	1.0 MB
Space available on disk:	6458.4 MB

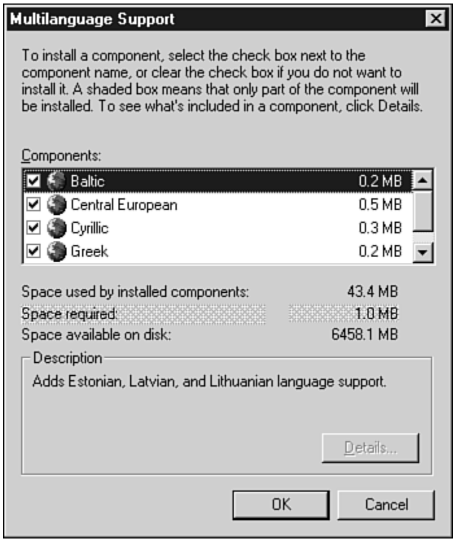
Description

Adds Albanian, Bulgarian, Belarusian, Croatian, Czech, Estonian, Greek, Hungarian, Latvian, Lithuanian, Macedonian, Polish, Romanian, Russian, Serbian, Slovak, Slovenian, Ukrainian, and Turkish.

5 of 5 components selected

5. With Multilanguage Support selected, click Details to specify the alphabets and languages you want to use (see Figure 2).

Figure 2
Choosing to add support for specified language families in Windows Me.



Multilanguage Support

To install a component, select the check box next to the component name, or clear the check box if you do not want to install it. A shaded box means that only part of the component will be installed. To see what's included in a component, click Details.

Components:

☑	Baltic	0.2 MB
☑	Central European	0.5 MB
☑	Cyrillic	0.3 MB
☑	Greek	0.2 MB

Space used by installed components:	43.4 MB
Space required:	1.0 MB
Space available on disk:	6458.1 MB

Description

Adds Estonian, Latvian, and Lithuanian language support.

6. Click OK twice.

Note

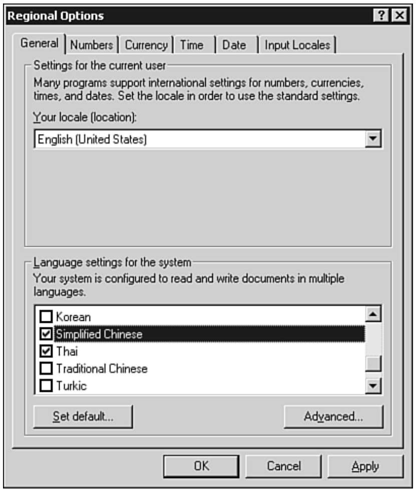
Windows Me Multilanguage Support provides support for the following languages that do not use the standard English/Western European character set: Albanian, Bulgarian, Belarussian, Croatian, Czech, Estonian, Greek, Hungarian, Latvian, Lithuanian, Macedonian, Polish, Romanian, Russian, Serbian, Slovak, Slovenian, Ukrainian, and Turkish.

Windows 98 Multilanguage Support does not include Croatian or Macedonian support.

CONTROLLING WINDOWS REGIONAL OPTIONS IN WINDOWS 2000

When you install Word 2002 (or most other Windows programs), the application looks to the Windows Regional Options to establish defaults for a variety of settings that vary from country to country. For instance, Regional Settings control number formats, currency formats, and how times and dates are displayed.

In Windows 2000, regional settings are controlled through Regional Options in the Windows Control Panel. To view it, choose Start, Settings, Control Panel, and double-click Regional Options (Regional Settings in Windows 98). Figure 3 shows the Regional Options dialog box in Windows 2000.



Settings for the current user

Many programs support international settings for numbers, currencies, times, and dates. Set the locale in order to use the standard settings.

Your locale (location): English (United States)

Language settings for the system

Your system is configured to read and write documents in multiple languages.

- ☐ Korean
- ☑ Simplified Chinese
- ☑ Thai
- ☐ Traditional Chinese
- ☐ Turkic

Figure 3
Choosing to add support for specified language families in Windows 2000.

In the United States, the default regional setting is English (United States). To change it, select a different regional setting from the drop-down box. When you make a change, Windows changes corresponding settings in the other tabs in this dialog box. For instance, if you choose German (Standard) and then display the Currency tab, you'll see that Windows has changed the standard currency format from $ to DM (Deutsche Marks).

You can also make changes individually on the other tabs, though you should be careful doing so: You can introduce inconsistencies that can cause your applications to behave in unexpected ways.

For instance, if you change regional number settings, you may encounter problems with Word nested fields that calculate based on the behavior of punctuation such as decimal points and commas. If you display the following nested field on a computer with U.S. settings, Word will add the values 3, 5, and 7.

```
{ =sum{3,5,7} }
```

If you display the same field on a computer whose number settings use the comma as a decimal symbol—such as a computer set to Portuguese (Brazilian)—the field will not work properly, and an error message will display. To solve the problem, you must either edit the field or change the regional settings.

Note

Windows may prompt you for the original CD-ROM or network installation source if you choose a Regional Setting you haven't already installed.

Later in this chapter, in the "Using Foreign Dates and Times in Your Documents" section, you will learn how to override the Windows settings for dates and times. Then, in the

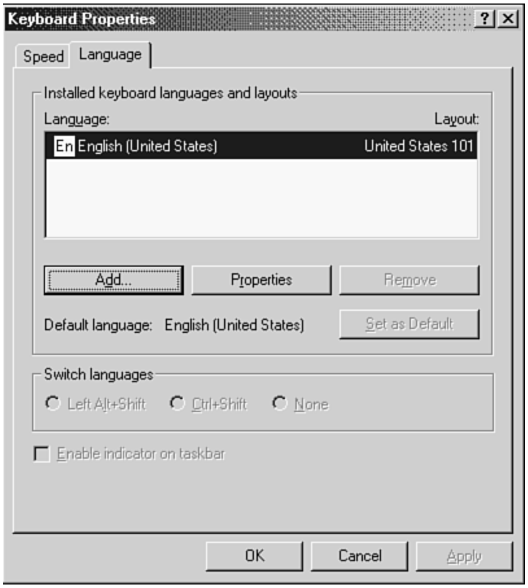
"Using Foreign Language Rules for Sorting" section, you will learn how to establish custom sorting settings based on the conventions associated with a specific alphabet.

Changing Your Keyboard Layout to Reflect a Different Language

If you are using a language that works with a different character set, you may need to change your keyboard layout to gain easy access to the characters that language uses. Have your Windows CD handy (or your network installation folder accessible); then follow these steps:

1. Choose Start, Settings, Control Panel.
2. Double-click the Keyboard icon.
3. In Windows Me, choose the Language tab (see Figure 4). In Windows 2000, this tab is named Input Locales.

Figure 4
Adding a new keyboard layout through the Keyboard Control Panel.



English (United States) — United States 101

Default language: English (United States)

4. Click Add. The Add Language dialog box opens, with a dingle drop down box with which you can choose from available languages.
5. Choose the language you want to add.
6. Click OK.
7. In the Language (or Input Locales) tab, select the language you want to use as your default keyboard setting and click Set as Default.
8. If you'd like an indicator to appear on the Windows the system tray showing which keyboard layout is currently in use, check the Enable Indicator on Taskbar check box.

9. Click OK to close the Keyboard Control Panel applet. Windows copies the files it needs to make the keyboard layout available.

By default, Windows provides the keyboard shortcut Left Alt+Shift to switch on-the-fly between keyboards. If you prefer, you can choose Ctrl+Shift as your keyboard shortcut, or choose None to enable no keyboard shortcut at all.

Whether or not you provide a keyboard shortcut for switching between keyboards, you can switch among them by clicking on the Language indicator in the system tray at the right side of the Windows taskbar (see Figure 5). All available keyboards are shown, as are any Input Method Editors (IMEs) you may have installed to type in Asian languages. (IMEs are covered later in this chapter, in the "Entering Asian Text with Input Method Editors" section.)

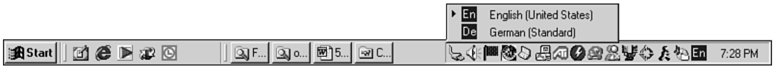


Figure 5
Switching between keyboard layouts in Windows Me.

Tip from
Bill Camarda

If you occasionally work with a variety of Western European languages, but you do not work enough with any one language to switch to that language's keyboard, consider installing and using the US-International Keyboard. This keyboard replaces many rarely used English-language characters with the most common foreign-language characters.

To do so, display the Language tab (or, in Windows 2000, the Input Locales tab) of the Keyboard Control Panel, as discussed earlier. Select the English (US) keyboard and click Properties. The Language Properties (or, in Windows 2000, the Input Locale Properties) dialog box opens. Choose US-International and click OK.

WORD AND OFFICE MULTILINGUAL FEATURES

Now that you've reviewed key Windows features that you may have to customize to work with foreign languages, the next several sections cover features and techniques specific to Word and Office.

TYPING TEXT IN A FOREIGN LANGUAGE

When it comes to entering and editing text in Word, languages fall into one of three categories:

- Western European languages that include characters that do not appear on the standard English (US) keyboard, but can easily be entered in several different ways

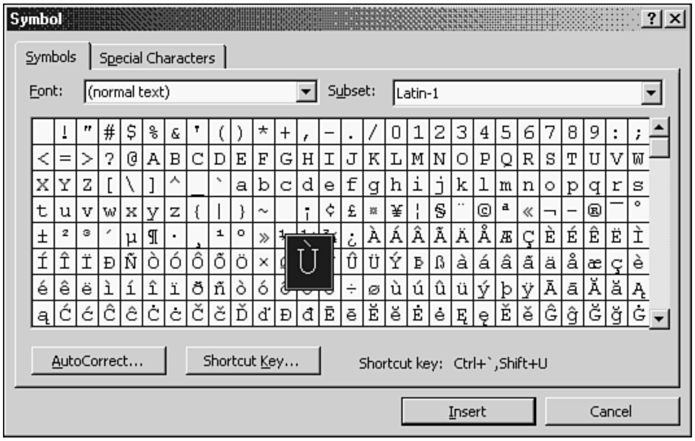
- Non-Western European languages, such as Greek, that require you to use a different keyboard layout to access the appropriate character sets
- Ideographic languages such as Chinese and Korean that require the use of a special piece of add-on software called an Input Method Editor (IME)

Each of these is covered next.

TYPING TEXT IN A WESTERN EUROPEAN LANGUAGE OTHER THAN ENGLISH

If you work primarily in English, but occasionally you must enter a word, phrase, sentence, or paragraph in another Western European language, Word provides several ways to insert the accented (diacritical) characters that you may encounter.

First, you can enter the characters through the Symbol dialog box, as shown in Figure 6. Choose Insert, Symbol; choose (normal text) from the Font drop-down box; click on the character you want; and click Insert.



Font: (normal text)

Subset: Latin-1

	!	"	#	$	%	&	'	(	)	*	+	,	-	.	/	0	1	2	3	4	5	6	7	8	9	:	;
<	=	>	?	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W
X	Y	Z	[	\	]	^	_	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s
t	u	v	w	x	y	z	{	|	}	~		¡	¢	£	¤	¥	¦	§	¨	©	ª	«	¬	-	®	¯	°
±	²	³	´	µ	¶	·	¸	¹	º	»	¼	½	¾	¿	À	Á	Â	Ã	Ä	Å	Æ	Ç	È	É	Ê	Ë	Ì
Í	Î	Ï	Ð	Ñ	Ò	Ó	Ô	Õ	Ö	×	Ø	[illegible]	[illegible]	Û	Ü	Ý	Þ	ß	à	á	â	ã	ä	å	æ	ç	è
é	ê	ë	ì	í	î	ï	ð	ñ	ò	ó	ô	õ	ö	÷	ø	ù	ú	û	ü	ý	þ	ÿ	Ā	ā	Ă	ă	Ą
ą	Ć	ć	Ĉ	ĉ	Ċ	ċ	Č	č	Ď	ď	Đ	đ	Ē	ē	Ĕ	ĕ	Ė	ė	Ę	ę	Ě	ě	Ĝ	ĝ	Ğ	ğ	Ġ

Shortcut key: Ctrl+`,Shift+U

Figure 6
You can enter characters in the Symbol dialog box.

Second, you can use a keyboard shortcut. Table 1 presents the keyboard shortcuts for inserting a wide range of foreign-language characters.

TABLE 1 KEY COMBINATIONS FOR INSERTING FOREIGN CHARACTERS

To Insert	Press the Following Key Combination
à, è, ì, ò, ù, À, È Ì, Ò, Ù	Ctrl+` (accent grave) followed by the letter (A, E, I, O, or U, upper- or lowercase as needed)
á, é, í, ó, ú, y´, Á É, Í, Ó, Ú	Ctrl+' (apostrophe) followed by the letter (A, E, I, O, U, or Y, upper- or lowercase as needed)
â, ê, î, ô, û, Â, Î, Ô, Û	Ctrl+Shift+^ (caret); followed by the letter (A, E, I, O, or U, upper- or lowercase as needed)

Table 1 Continued

To Insert	Press the Following Key Combination
ã, ñ, õ, Ã, Ñ, Õ	Ctrl+Shift+~ (tilde); followed by the letter (A, N, or O, upper- or lowercase as needed)
ä, ë, ï, ö, ü, ÿ, Ä Ë, Ï, Ö, Ü, Ÿ	Ctrl+Shift+: (colon); followed by the letter (A, E, I, O, or U, upper- or lowercase as needed)
å, Å	Ctrl+Shift+@, followed by a or A
æ, Æ	Ctrl+Shift+&, followed by a or A
œ, Œ	Ctrl+Shift+&, followed by o or O
ç, Ç	Ctrl+, (comma), followed by c or C
ð, Ð	Ctrl+ ' (apostrophe), followed by d or D
ø, Ø	Ctrl+/, followed by o or O
¿	Alt+Ctrl+Shift+?
¡	Alt+Ctrl+Shift+!
ß	Ctrl+Shift+&, s

Every character in the standard English-language Windows character set has its own character code, a four-digit code that starts with 0. Some users prefer to insert Western European–language characters by entering these character codes.

To do so, click your insertion point where you want to enter the character. Make sure that the Num Lock key is toggled on. Then, hold down the Alt key and, on the numeric keypad, type the four-digit character code.

Table 2 lists selected character codes for the most common Western European characters. Note that a few useful U.S. characters, such as ÷, also appear in this portion of the Windows character set.

Table 2 Character Codes for Common Western European Characters

To Insert	Use This Code	To Insert	Use This Code
¿	0191	à	0224
À	0192	á	0225
Á	0193	â	0226
Â	0194	ã	0227
Ã	0195	ä	0228
Ä	0196	å	0229
Å	0197	æ	0230
Æ	0198	ç	0231

Table 2 Continued

To Insert	Use This Code	To Insert	Use This Code
Ç	0199	è	0232
È	0200	é	0233
É	0201	ê	0234
	0202	Ê	0235
Ë	0203	ì	0236
Ì	0204	í	0237
Í	0205	î	0238
Î	0206	ï	0239
Ï	0207	∂	0240
Ð	0208	ñ	0241
Ñ	0209	ò	0242
Ò	0210	ó	0243
Ó	0211	ô	0244
Ô	0212	õ	0245
Õ	0213	ö	0246
Ö	0214	÷	0247
×	0215	ø	0248
Ø	0216	ù	0249
Ù	0217	ú	0250
Ú	0218	û	0251
Û	0219	ü	0252
Ü	0220	ý	0253
Ý	0221	þ	0254
Þ	0222	ÿ	0255
ß	0223		

Tip from
Bill Camarda

If you use a specific foreign-language word or phrase often, store it as an AutoText entry (see Chapter 30, "Automating Your Documents with Field Codes").

If you use a specific foreign-language character often, consider recording a macro to enter it and then inserting that macro as a button on your Standard or Formatting toolbar (see Chapter 31, "Customizing Word").

TYPING TEXT IN A NON-WESTERN EUROPEAN LANGUAGE

Many European languages use characters that either do not appear in the standard Windows character set or are difficult to access if you are typing more than a few characters. These languages include

- Greek
- Central European languages, such as Hungarian and Polish (Windows Latin 2 Character Set)
- Slavic languages, such as Russian and Ukrainian (Windows Cyrillic Character Set)
- Turkish (Windows Latin 5 Character Set)
- Baltic languages such as Latvian and Lithuanian (Windows Latin 4 Character Set)

To enter text in these languages, first enable Multilanguage Support in Windows (see "Configuring Multilanguage Support in Windows Millennium (Me)," earlier in this chapter). Next, install the appropriate keyboard layout for the language you need to use, through the Keyboard Windows Control Panel (see "Changing Your Keyboard Layout to Reflect a Different Language," earlier in this chapter). Windows offers separate keyboard layout settings for nearly every country in these regions, from Albania to the Ukraine.

After you've done all this, you can switch to the appropriate keyboard in any of the following ways:

- If you've enabled automatic language detection and also enabled the language you want to use, Word will usually switch the keyboard for you automatically if it detects that you're using a language it has a keyboard available for. (To learn how to enable languages in Word and use automatic language detection, see "Having Word Assign Languages to Text Automatically," later in this chapter.)
- If you've set a keyboard shortcut for switching languages, such as Left+Alt+Shift, use that shortcut.
- If you've displayed the list of keyboards in the system tray at the right edge of the Windows taskbar, click on it and choose the keyboard you want from the list that appears.

After you select the appropriate keyboard, begin typing.

Tip from
Bill Camarda

If you intend to use one language extensively, you may want to purchase special key caps you can apply to your keyboard so that you can see the names of the keys you're typing.

If Word switches you to the wrong keyboard without being told, see "What to Do if Word Switches Your Keyboard Setting Incorrectly" in the "Troubleshooting" section of this chapter.

If Word reformats your text in a language you don't want, see "What to Do if Word Formats Text in the Wrong Language" in the "Troubleshooting" section of this chapter.

ENTERING ASIAN TEXT WITH INPUT METHOD EDITORS

Documents prepared in ideographic languages such as Chinese, Korean, and Japanese can be read in Word 2002 if you have appropriate fonts installed, such as the Asian fonts listed later in this chapter, in Table 3. However, text cannot be entered in these languages using standard methods.

If you need to write extensively in Korean, Japanese, or Chinese, the best solution is to configure Word for these languages using the techniques discussed later in this chapter.

However, many users need to enter text in these languages only occasionally. For them, Microsoft's Language Pack provides *Input Method Editors (IMEs)*—add-on software that converts keystrokes into ideographic characters.

IMEs typically contain an engine that converts keystrokes into phonetic and ideographic characters, as well as a dictionary of the most widely used ideographic words. As you type characters based on the sounds of the spoken language, the IME attempts to guess which ideographic character or characters you want. The IME also provides tools for manually choosing characters one at a time.

Note

Some third-party Input Method Editors are called "front-end processors."

INSTALLING AN INPUT METHOD EDITOR

Input Method Editors for Korean, Japanese, and Chinese are each installed separately. They do not install automatically as part of the Office or Language Pack setup processes. If you have the Office XP Language Pack, follow these steps to install the Input Method Editor of your choice:

1. Insert the appropriate Language Pack CD-ROM.
2. Accept the EULA (End User License Agreement).

Select the language for which you want to install international support.

You are now able to choose between the Input Method Editors installed on your system. Note that because they all come from a single executable, you are able to switch between then within Word. No resetting or restarting is required.

Formatting and Proofing Text in Multiple Languages

Just as text in Word can be formatted with fonts and styles, so too can it be formatted with a specific language. Formatting text in a specific language tells Word which of your installed proofing tools to use and also helps Word choose the right formatting for dates, times, and numbering.

There are two ways to format text in a specific language other than English:

- You can use Word 2002's automatic language detection feature to do it for you.
- You can do it manually, through the Tools, Language, Set Language dialog box.

Having Word Assign Languages to Text Automatically

Word 2002's automatic language detection feature can check your text as you write, identifying the language in which you're writing by comparing the words you type against the proofing tools you've installed. If Word determines that you are writing in a different language, it can format the text automatically for that language. The results, although not always perfect, are surprisingly good.

Automatic language detection can sometimes help you improve the accuracy of your foreign language documents, especially if you are less than fluent in the language. For instance, if you turn on this feature and enter a passage of text in French that is a direct quotation, Word converts the English language quotation marks to the correct French quotation marks: « ». This occurs even if you haven't installed French proofing tools.

To set up Word for automatic language detection, follow these steps:

1. Set up Windows with multilingual support (these steps are covered earlier in the chapter, in the sections beginning with "Setting Up Windows for International Environments").
2. Install the correct proofing tools for the languages you want to use. In default installations, the French and Spanish proofing tools are installed on first use.
3. Enable the languages you want Word to make available to you. These typically are the same languages for which you install proofing tools. Enabling languages is covered in the next section, "Enabling Languages in Office XP."
4. Turn on automatic language detection (covered later in this chapter, in the "Turning on Automatic Language Detection in Word" section).

Word's Automatic Language Detection feature works in 21 languages: Arabic, Chinese (Simplified and Traditional), Danish, Dutch, English, Finnish, French, German, Greek, Hebrew, Italian, Japanese, Korean, Norwegian, Polish, Portuguese, Russian, Spanish, Swedish, and Thai. In other words, automatic language detection does not work with some of the languages you can install support for, such as Turkish.

Automatic language detection works one sentence at a time, so it doesn't get tripped up by foreign language words or phrases you may use in an English language document, or by Greek characters you might use in a formula. In detecting languages, Word only considers languages that are

- Enabled for editing
- Set as the default for documents created with the template you're using
- Associated with the keyboard you are using

Enabling Languages in Office XP

To enable specific languages for use by Word and Office, you use a special utility program called Microsoft Office XP Language Settings. To access this utility, choose Start, Programs, Microsoft Office Tools, Microsoft Office Language Settings. The Microsoft Office Language Settings dialog box opens, displaying the Enabled Languages tab.

In the Show Controls and Enable Editing For scroll box, add check marks next to the language or languages for which you want to install additional support. The languages appear in the Enabled Languages box at the right. When you're finished, click OK. The next time you open Word, the language support is activated, and the language-specific styles and macros are automatically available.

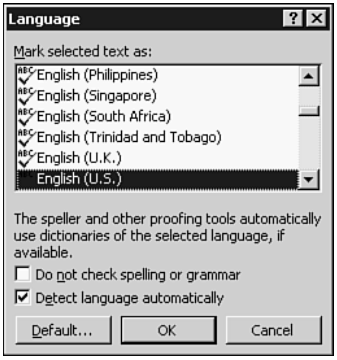
Caution

Not all features can be used in every version of Office. For example, as mentioned earlier, support for bidirectional languages, such as Hebrew and Arabic, is not provided in the United States/Canadian versions of Windows 95 and 98.

Turning on Automatic Language Detection in Word

After you've followed all the preceding steps, turning on automatic language detection is simple. First, open Word. Choose Tools, Language, Set Language. The Language dialog box opens (see Figure 7). Check the Detect Language Automatically check box and click OK.

Figure 7
The Language dialog box.



English (Philippines)
English (Singapore)
English (South Africa)
English (Trinidad and Tobago)
English (U.K.)
English (U.S.)

The speller and other proofing tools automatically use dictionaries of the selected language, if available.

☐ Do not check spelling or grammar

☑ Detect language automatically

Manually Assigning Text to a Language

If you do not want to set up automatic language detection, or if you do not have all the proofing tools you need to use it fully, you can still manually assign a language to your text. Doing so has two advantages:

- It tells Word not to bother proofing text in another language—sparing you the need to handle the spurious errors Word will find.
- When the document is moved to a computer that does have the appropriate tools, that computer can proof the foreign language text immediately.

To manually assign text to a language, first select the text. Then choose Tools, Language, Set Language. The Language dialog box opens (refer to Figure 29.10). In the Mark Selected Text As scroll box, choose the language you want Word to use and click OK.

Note

In the Mark Selected Text As list of languages, the languages you have installed proofing tools for are marked with a check mark icon and the letters ABC, similar to the spell check icon on the Standard toolbar.

If you format text in a language for which you have not installed proofing tools, the first time you attempt to proof text in that language Word may ask you to install the appropriate proofing tools. If you have the proofing tools, follow the steps to install them. Word then immediately spell checks the text using the language you selected. If grammar checking is also available for that language, it also runs the grammar check—unless you've disabled grammar checking.

Specifying "No Proofing" on Selected Text

Occasionally, you may edit in languages for which you do not ever expect to have proofing tools available or accessible. If so, you may want to specify that Word skip proofing all or part of your document. To do so, select the text you want to go unproofed. Next, choose Tools, Language, Set Language. The Language dialog box opens (see Figure 10). Check the Do Not Check Spelling Or Grammar check box and click OK.

Caution

If automatic language detection is turned on, Word automatically skips proofing for text blocks it cannot identify as being written in a recognizable language.

Tip from
Bill Camarda

Use the Do Not Check Spelling or Grammar option whenever you have long lists of product names, company names, or other text that Word would otherwise flag as potentially incorrect simply because it cannot find those words in its dictionary.

Changing Word's Default Language

Word automatically creates documents using whatever language you set as the default; in the United States, this language is English (US). You can change the default language if necessary, and Word adjusts its proofing tools as necessary.

Of course, you can set a foreign language as your default. You can also use any of the following variants on English, and Word then reflects these variants in its proofreading:

- English (Australian)
- English (Belize)
- English (Canadian)
- English (Caribbean)
- English (Ireland)
- English (Jamaica)
- English (New Zealand)
- English (Philippines)
- English (South Africa)
- English (Trinidad)
- English (UK)
- English (Zimbabwe)

To change the default language for all documents associated with the Normal template, follow these steps:

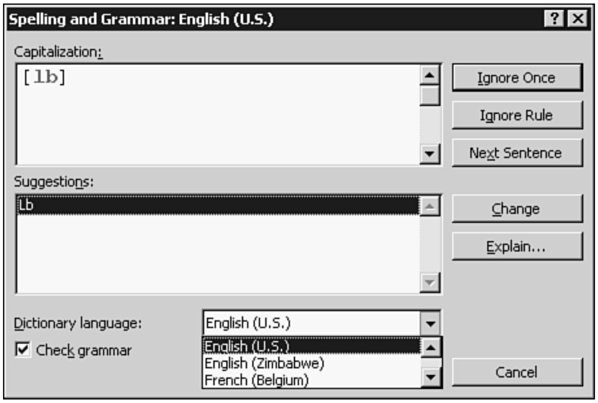
1. Choose Tools, Language, Set Language.
2. In the Language dialog box (refer to Figure 29.10), choose the language you want to use.
3. Click Default. A confirmation dialog box appears.
4. Click Yes to confirm that you want to change the default language.

Changing Proofing Languages During a Spell Check

Earlier in this chapter, you learned that you can format text with a specific language, and Word will automatically use that language's proofing tools, assuming that they are installed. Occasionally, however, you may not notice a change in language until you run a spell check—or you may not notice that Word's automatic language detection feature has made a mistake until then. To resolve this problem while you're in the process of running a spelling or grammar check, Word allows you to change the proofing language it is using.

To do so, press F7 or click the Spelling and Grammar button on the Standard toolbar to display the Spelling and Grammar dialog box. Then choose a different language from the Dictionary Language drop-down box (see Figure 8). The change takes effect immediately.

Figure 8
Manually changing the language you use for proofing.



Spelling and Grammar: English (U.S.)

Capitalization:
[1b]

Suggestions:
Lb

Dictionary language: English (U.S.)
English (U.S.)
English (Zimbabwe)
French (Belgium)

☑ Check grammar

If the Dictionary Language drop-down box does not appear, exit Word and use the Office Language Settings utility to enable the languages from which you want Word to choose. Doing so is covered earlier in this chapter, in the section “Having Word Assign Languages to Text Automatically.”

UNDERSTANDING THE MICROSOFT LANGUAGE PACK

In addition to the language support already covered in this chapter, Microsoft provides additional support through the Microsoft Office Language Pack.

To understand the Language Pack, it is important to understand a substantial change Microsoft made “under the hood” when they released Office 2000. This change has carried over into Office XP.

In Word and Office 97, Microsoft designed each international version separately, using separate code—27 versions in all. For Microsoft, each version required separate development and testing. For organizations implementing Word, this also meant more extensive testing—especially for custom *VBA* solutions. It also meant separate setup programs, more complex deployment and management challenges, and higher support costs.

In contrast, in Word and Office 2000, Microsoft created a single worldwide code base covering nearly all major European and Far East languages, as well as bidirectional languages where text can be entered from left-to-right and right-to-left, such as Arabic and Hebrew.

Layered on top of this code base is a modular user interface component that specifies which language is to be used in menus and dialog boxes, and handles variations in program behavior to meet the needs of different languages.

Users, in turn, can easily create documents using more than one language, without having to change the interface they are using or work with a different version of Word. In fact, for users who have proofing tools for multiple languages installed, Word automatically detects the language they are using at any given time and applies the correct proofing and AutoCorrect tools.

Note

The Language Pack does not substitute for the localized versions of Word and Office that Microsoft will continue to produce, based on this common code base. Rather, the Language Pack is intended for use in organizations that require either more extensive proofing coverage or multilingual user interfaces.

What's in the Language Pack?

The Language Pack includes the following:

- User interface components for many languages (though not all language components will work with standard United States/Canadian versions of Windows 98, or NT 4.0)
- Help files in several languages
- Customized templates and wizards in several languages
- Proofing tools in several languages, including some or all of the following for each language: spell checker, grammar checker, thesaurus, hyphenation dictionary, and *AutoCorrect* lists

Note

As mentioned earlier, Spanish and French proofing tools are already included in the standard version of Word 2002.

- Several TrueType fonts designed for international use (see Table 29.1, earlier in this chapter)
- A translator designed to translate text between simplified and traditional Chinese
- A Korean Hangul Hanja converter
- Input Method Editors, which enable users to build Chinese, Japanese, or Korean ideographs by using characters on a standard United States keyboard

Installing the Language Pack

The Language Pack can only be installed over the English-language version of Word or Office. To install most of the tools in the Language Pack, you must first run the Language Pack CD-ROM's Setup program. First, close any Office applications that are open. Next, insert the appropriate Language Pack CD-ROM, or connect to the network location containing the Language Pack files, and follow these steps:

1. The Windows installer should run automatically, displaying the first screen of the Setup Wizard (if it doesn't click Start, Run and locate the CD-ROM setup file to launch the wizard). This screen already contains the information you used when you first installed Word or Office.
2. Edit the information if necessary and click Next twice.
3. Specify the location where you want the Windows installer to install the Language Pack files; by default, this will be the folder you installed Office or Word in. If necessary, browse to and select a new location. When the location is correct, click Next.
4. Microsoft Language Pack now displays the languages available for you to install. You can choose to install multilingual user interfaces for only selected Office applications, or even for different applications in different languages.

Note

By default, Language Pack applications are set to install on first run, which means that the first time a user calls on them after you run this Setup routine, the files are installed from the CD-ROM or across the network. However, you can choose other options.

5. When you're finished choosing which applications and user interfaces to install, click Install Now. The Windows installer prepares the Office applications you selected to run the international user interfaces you selected.

Note

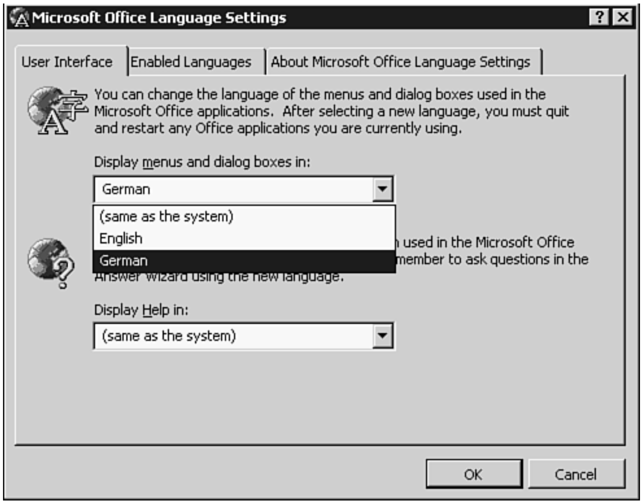
To get access to all of Office's international features, you may also need to run a maintenance setup to add International Support files that are part of Office Tools.

Changing Language Settings After You've Installed the Language Pack

After you've installed the Language Pack, you can add additional language support and switch user interfaces among the languages you've installed.

To work with the language settings, make sure that all Office applications are closed. Then, from the Windows desktop, choose Start, Programs, Microsoft Office Tools, Microsoft Office Language Settings. The Microsoft Office Language Settings dialog box opens, with the User Interface tab displayed.

To choose a different language for your menus and dialog boxes, select the language from the Display Menus and Dialog Boxes In drop-down box (see Figure 9).


Microsoft Office Language Settings
User Interface
Enabled Languages
About Microsoft Office Language Settings
OK
Cancel

You can change the language of the menus and dialog boxes used in the Microsoft Office applications. After selecting a new language, you must quit and restart any Office applications you are currently using.

Display menus and dialog boxes in: German

(same as the system)
English
German

n used in the Microsoft Office
member to ask questions in the
Answer wizard using the new language.

Display Help in: (same as the system)

Figure 9
Controlling in which language your menus and dialog boxes appear.

By default, Word displays Help in the same language as it displays dialog boxes. To read Help files in a different language, choose the language from the Display Help In drop-down box. When you're finished, click OK. Office reports that the changes will take effect the next time you start an Office application.

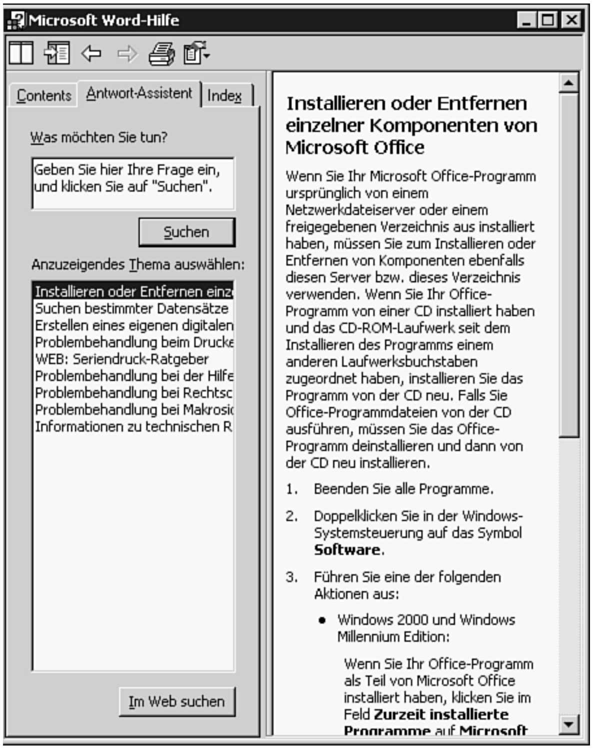
Tip from
Bill Camarda

If you're installing language support from the CD-ROM locally, keep the CD-ROM in your drive for a while: You may need it to add foreign-language dictionaries after you start editing.

Figure 10 shows Word displaying dialog boxes and Help in German.

Tip from
Bill Camarda

Rather than changing user interfaces, some organizations may choose to standardize on an English user interface and encourage users to switch among proofing tools at will.



Geben Sie hier Ihre Frage ein, und klicken Sie auf "Suchen".

- Installieren oder Entfernen einz
- Suchen bestimmter Datensätze
- Erstellen eines eigenen digitalen
- Problembehandlung beim Drucke
- WEB: Seriendruck-Ratgeber
- Problembehandlung bei der Hilfe
- Problembehandlung bei Rechtsc
- Problembehandlung bei Makrosic
- Informationen zu technischen R

Installieren oder Entfernen einzelner Komponenten von Microsoft Office

Wenn Sie Ihr Microsoft Office-Programm ursprünglich von einem Netzwerkdateiserver oder einem freigegebenen Verzeichnis aus installiert haben, müssen Sie zum Installieren oder Entfernen von Komponenten ebenfalls diesen Server bzw. dieses Verzeichnis verwenden. Wenn Sie Ihr Office-Programm von einer CD installiert haben und das CD-ROM-Laufwerk seit dem Installieren des Programms einem anderen Laufwerksbuchstaben zugeordnet haben, installieren Sie das Programm von der CD neu. Falls Sie Office-Programmdateien von der CD ausführen, müssen Sie das Office-Programm deinstallieren und dann von der CD neu installieren.

1. Beenden Sie alle Programme.
2. Doppelklicken Sie in der Windows-Systemsteuerung auf das Symbol **Software**.
3. Führen Sie eine der folgenden Aktionen aus:
   - Windows 2000 und Windows Millennium Edition:

     Wenn Sie Ihr Office-Programm als Teil von Microsoft Office installiert haben, klicken Sie im Feld **Zurzeit installierte Programme** auf **Microsoft**

Figure 10
Displaying menus, dialog boxes, and Help in German.

Using Foreign Dates and Times in Your Documents

In Windows Me and 2000, regional settings are controlled through the Regional Settings Control Panel. In addition to date and time settings, regional settings also control the way applications format numbers and currency values. However, after you install Word, you can change these defaults without changing them "Windows-wide."

➔ For more information about controlling date, time, and other settings with the Windows Regional Settings Control Panel, **see** "Controlling Windows Regional Options in Windows 2000," **p. 4.**

To change these settings only for Word, but not for the rest of Windows, follow these steps:

1. Enable the language you want to use, as discussed earlier in this chapter in the "Enabling Languages in Office XP" section.

Choose Insert, Date and Time.

3. From the Language drop-down box, choose the language containing the options you expect to use most.
4. From the Available Formats box, choose the setting you expect to use most.
5. Click Default.

6. Click Yes to confirm that you want to change the default.

Using Foreign Language Rules for Sorting

Different languages have different rules for sorting the items in a list. For example

- In French, diacritics such as ê are sorted right to left, not left to right as in English.
- In Norwegian and Swedish, characters with diacritic marks (such as Ä) follow Z at the end of the alphabet.
- In Spanish, CH is a separate letter of the alphabet between C and D; ñ is a separate letter of the alphabet between N and O.

When you install Word, default alphabetic sorting is set based on your regional settings in Windows. However, you can specify a different sorting language for any list, as follows.

First, select the paragraphs or table rows to be sorted and choose Table, Sort. Click Options; the Sort Options dialog box opens. In this dialog box, click the Sorting Language drop-down box, choose the language you want to use, and click OK twice.

Note

If you have installed proofing tools for a foreign language, and enabled them through the Microsoft Office Language Settings utility, when you format text in that language, Word applies that language's sorting rules automatically.

Microsoft Office's Language File Organization

Since Microsoft Office 2000, the multilingual orientation has led to changes in the way Microsoft Office installs on your computer—and, in turn, to some changes in the way it behaves as a result.

Microsoft Office stores proofing tools and user interface components for each language separately. When you install only the English-language user interface, as is the case by default, all Office files related to displaying the user interface, including Help files, localized templates and wizards, and default AutoCorrect lists, are stored in the \Program Files\Microsoft Office10\Office\1033 folder. Grammar-checking files are stored in the \Program Files\Common Files\Microsoft Shared\Proof\1033 folder.

The number 1033 is the locale ID Microsoft has assigned to the U.S. version of the English language. In the default installation, Microsoft also creates corresponding empty folders named 1034 and 1036 where Spanish and French files will be installed if you use them.

Other languages are placed in folders that use their assigned locale names when you install proofing tools or user interface components with either the Microsoft Office Proofing Tools Kit or the Language Pack. Table 3 shows the locale IDs associated with a wide variety of languages, sorted by number.

Table 3 Locale IDs Used in Microsoft Office Installation

Locale ID	Locale Name	Locale ID	Locale Name
1025	Arabic	1066	Vietnamese
1026	Bulgarian	1067	Armenian
1027	Catalan	1068	Azeri (Latin)
1028	Chinese (Taiwan)	1069	Basque
1029	Czech	1071	Macedonian
1030	Danish	1078	Afrikaans
1031	German	1079	Georgian
1032	Greek	1080	Faeroese
1033	English	1081	Hindi
1035	Finnish	1086	Malaysian
1036	French	1087	Kazakh
1037	Hebrew	1089	Swahili (Kenya)
1038	Hungarian	1091	Uzbek (Latin)
1039	Icelandic	1092	Tatar (Tatarstan)
1040	Italian	1093	Bengali
1041	Japanese	1094	Punjabi
1042	Korean	1095	Gujarati
1043	Dutch	1096	Oriya
1044	Norwegian	1097	Tamil
1045	Polish	1098	Telugu
1046	Portuguese	1099	Kannada
1048	Romanian	1100	Malayalam
1050	Croatian	1101	Assamese
1051	Slovak	1102	Marathi
1052	Albanian	1103	Sanskrit
1053	Swedish	1111	Konkani
1054	Thai	1112	Manipuri

Table 3 Continued

Locale ID	Locale Name	Locale ID	Locale Name
1055	Turkish	1113	Sindhi
1056	Urdu	1120	Kashmiri
1057	Indonesian	1121	Nepali
1058	Ukrainian	1122	Frisian
1059	Belarussian	2052	Chinese (PRC)
1059	Russian	2074	Serbian (Latin)
1060	Slovenian	2092	Azeri (Cyrillic)
1061	Estonian	2115	Uzbek (Cyrillic)
1062	Latvian	3082	Spanish
1063	Lithuanian	3098	Serbian (Cyrillic)
1065	Farsi		

If you install Office XP over an existing Office installation, your old templates remain, but Office XP's programs use copies of the files that are placed in these new numbered folders at installation. This can cause confusion: The Windows Find feature will find two copies of each template, and it's all too easy to browse to the wrong location in Windows Explorer when you want to copy, paste, rename, or remove templates.

Using the Euro Symbol

Starting on January 1, 1999, many nations in Europe began to phase in a shared currency, called the *Euro*.

In addition to presenting complex accounting issues for businesses working with European currency, the Euro presents a simple challenge for anyone working in Word. The Euro character didn't exist until just a few years ago. Therefore, unless you purchased your computer system recently, it probably wasn't built into the fonts on your computer or those built into your printer.

Windows Me and Windows 2000 both contain built-in support for the Euro, including updated versions of Times New Roman, Arial, and Courier New. To add the same Euro support to Windows NT 4.0, install either Service Pack 4 or the NT 4.0 Euro Product Update, which is available at `www.microsoft.com/windows/euro.asp`. When you install Office XP, an updated version of the Tahoma font that includes the Euro symbol is installed. However, not all fonts installed or updated by Office XP include the Euro symbol.

Note

You can get Euro-updated versions of Comic Sans MS and Andale (formerly called Monotype.com) at `www.microsoft.com/typography/fontpack/default.htm`. You can also get Euro-updated versions of Trebuchet MS and Georgia at `http://windowsupdate.microsoft.com/`. If you use Adobe fonts, you can get free fonts consisting of only the Euro character at `www.adobe.com/type/eurofont.html`.

If you have a cross-platform environment, with some users working on Macintosh computers running Word 98, visit `developer.apple.com/technotes/tn/tn1140.html` for details on Apple's approach to supporting the Euro.

To enter the Euro character

1. Click your insertion point in a location formatted with a font that contains the Euro character.
2. Press NumLock on the numeric keypad.
3. Press and hold down the Alt key while you type `0128` on the numeric keypad.
4. Release the Alt key. The Euro character appears.

Tip from
Bill Camarda

Or, choose Insert, Symbol; display the Symbols tab; choose a font that contains the Euro character from the Font drop-down list box; and select the Euro character from the Symbol dialog box.

As mentioned earlier, many printers sold before 1998 do not include the Euro symbol in their built-in ("resident") fonts. In particular, Hewlett-Packard began phasing the Euro symbol into its printers only in Fall 1998.

If you need to print the Euro symbol on an older printer, you may have to adjust your printer driver so that it downloads TrueType fonts rather than using the printer's resident fonts. Often, the setting is called Print TrueType as Graphics.

Caution

Unfortunately, printing TrueType fonts as graphics can slow down your printing considerably.

Note

If you own a Hewlett-Packard printer, you can learn more about hardware and software upgrade options for printing the Euro at `www.hp.com/cposupport/euro/indexes/euro.html`.

Tip from
Bill Camarda

If you only need to use the Euro symbol on rare occasions, you can retrieve an image of the Euro from a Web site, such as the official European Commission site (`http://europa.eu.int/`), and insert it into your document as you would insert any other image.

UNICODE AND INTERNATIONAL FONT SUPPORT IN OFFICE XP

Office XP supports Unicode, a standard character set that includes all commonly used characters in virtually all the world's major languages, including Asian languages. This makes it much easier to create documents that are entirely, or partially, in languages other than English.

Office XP comes with one complete Unicode font, Arial Unicode MS (Arialuni.ttf), which covers all languages that Unicode supports. It also comes with several additional fonts that support large subsets of the Unicode character set. These fonts are listed in Table 4.

TABLE 4 UNICODE AND MULTILINGUAL FONTS PROVIDED WITH OFFICE XP

Font	Filename	Languages Supported
Batang	Batang.ttf	European, Korean
BatangChe	BatangCh.ttf	European, Korean
MingLiU	Mingliu.ttf	English, Traditional and Simplified Chinese, Japanese
MS Mincho	Msmincho.ttf	European, Japanese
MS UI Gothic	Msuigoth.ttf	European, Japanese
SimSun	Simsun.ttf	English, Traditional and Simplified Chinese

Note

One disadvantage of standardizing on a true Unicode font such as Arial Unicode MS is its enormous size: more than 17 megabytes, compared with 100–200KB for most other TrueType fonts. Even the international fonts listed in Table 29.4 are much larger than typical TrueType fonts.

Working with fonts this large can cause system instabilities and reduced performance, especially on slower systems with less memory.

Support for Unicode is one reason Word 2002, 2000, and 97 files are significantly larger than Word 6/95 files.

When you install Office XP, Office also checks your current fonts and adds or updates many of them with big fonts that contain multiple character sets for use in international environments—while remaining relatively compact in file size. Office XP makes sure that the following fonts are updated:

- Arial, Arial Black, Arial Bold, and Arial Narrow
- Bookman Old Style
- Courier New
- Garamond
- Impact
- Tahoma (the font used by Office menus)
- Times New Roman
- Trebuchet (adds Central and Eastern European character sets)
- Verdana

Office XP's Unicode and international fonts enable you to use the same font to display documents in many languages, rather than reformat text in a specific language font. This means that you won't find yourself inadvertently transforming foreign language text into gibberish when you change fonts.

Note

Both Windows Me and Windows 2000 can work with Unicode. However, 2000 includes stronger support for Unicode and is especially preferred if you plan to work with Asian text.

⚠ *If Word does not print Unicode symbols correctly, see "What to Do if Your Printer Fails to Print Unicode Characters Properly" in the "Troubleshooting" section of this chapter.*

TROUBLESHOOTING

WHAT TO DO IF WORD CANNOT FIND A PROOFING FILE YOU NEED

If you attempt to proofread text in a language for which you have installed proofing tools, and Word reports that it cannot find the appropriate files, check to make sure that you have not installed an older version of the proofing tools. You can install the new version over the older versions without any problems.

WHAT TO DO IF YOUR PRINTER FAILS TO PRINT UNICODE CHARACTERS PROPERLY

Many printer drivers do not support Unicode properly. These include the following:

- Canon BubbleJet BJ-C600, 4000, 4200, 4500, 4550, v 3.40
- Canon Multipass 2500 3.40
- Epson Color 500

- Epson Stylus Color, v 2.x
- Epson Stylus Pro / XL v. 2.11BE
- HP Color LaserJet 5 PCL, printer driver version F 1.300
- HP DeskJet 1600C, printer driver version 4.20
- HP LaserJet 4 PCL, printer driver version 3.78
- HP LaserJet 6P, standard printer driver
- Okidata 4 laser printer

The best solution is to see whether your printer manufacturer has a newer printer driver that supports Unicode. Failing that, you may have to edit the Windows Registry to reset your printer driver so that it uses ANSI character layout functions rather than Unicode character layout functions. Detailed instructions for doing so may be found in the Microsoft Knowledge Base article Q159418. Knowledge Base articles can be found online at `http://support.microsoft.com`.

What to Do if Word Switches Your Keyboard Setting Incorrectly

If automatic language detection is turned on, Word might incorrectly determine which language you are writing in and switch your keyboard to the wrong language automatically. To switch your keyboard back, choose the correct keyboard from the keyboard indicator on the system tray, or use the appropriate keyboard shortcut (for example, Left Alt+Shift). To make sure that this never happens again, choose Tools, AutoCorrect; click the AutoCorrect tab; and clear the Correct Keyboard Setting check box.

What to Do if Word Formats Text in the Wrong Language

Word's new automatic language detection feature isn't perfect. It can especially encounter problems if you write sentences in multiple languages in the same paragraph or short document, or if you have enabled two similar languages.

If the problem only occurs occasionally, you can reformat the text manually by selecting it; choosing Tools, Language, Set Language; choosing the appropriate language; and clearing the Detect Language Automatically check box.

If the problem occurs repeatedly, try determining which language is causing the confusion—the one Word uses to reformat text in an undesired way. Then, if you rarely use that language, disable it through Start, Programs, Microsoft Office Tools, Microsoft Office Language Settings.

INTEGRATING WITH MICROSOFT OFFICE XP

In this chapter

by Bill Camarda

Special Edition **Using**

que®

WORD AND OFFICE: MORE TIGHTLY INTEGRATED THAN EVER

One of Word's most important strengths is its tight integration with Microsoft Excel, Access, PowerPoint, Publisher, and the rest of Office XP. These programs extend Word's power, and knowing how to use them with Word will make you much more productive. In Office XP, Office's main applications are more tightly integrated than ever. In this chapter, you'll learn to leverage all of Microsoft Office to make Word an even more powerful tool.

Note

To work with Excel, PowerPoint, Outlook, Publisher, or Access, you must first install them, either as part of Microsoft Office or as standalone applications.

INTEGRATING EXCEL AND WORD

Word 2002 can perform a surprising number of calculations all by itself, as you learned in the "Calculating with Tables" section in Chapter 13, "Tables: Word's All-Purpose Solution for Structuring Information." However, it's not a dedicated spreadsheet program like Microsoft Excel.

Luckily, if you've installed Excel, you can call on it whenever you need extra number-crunching power. You can insert Excel spreadsheets or charts when you need the extra number-crunching power. Or perhaps you just want to take advantage of work already completed in one program so that you can avoid redoing it in another.

This integration between Word and Excel is a two-way street. Excel offers tremendous mathematical prowess, but it's obviously more limited than Word when it comes to creating and formatting complex documents. You can use Word to present your Excel data in a format that communicates the information more effectively than Excel could alone.

Moreover, the connection between Word and Excel is as lively as you want it to be. If, for example, you link your quarterly report to sales data kept in three different files (gathered by three different sales representatives) on your company's network, every time you open or print the report, your numbers are updated. Just as importantly, you can also set the links to not update automatically so that you can always generate an accurate archival record of a project as it existed at a given point in time.

You can use several techniques to bring Excel data into Word; none are overly complicated, and some are downright simple. In the next section, you'll leverage Excel's mathematical capabilities in Word by integrating a new Excel worksheet into a Word document.

INSERTING A NEW EXCEL WORKSHEET IN A WORD DOCUMENT

In Chapter 13, you learned how you can use Word to add numbers in a table or insert a field that can calculate a formula anywhere in a document, using basic arithmetic operations and functions. You may well find, however, that you need to perform calculations beyond Word's capability. Fortunately, you can tap Excel's powerful capabilities without leaving Word.

Note

Of course, you need to know how to use Excel to take advantage of its features. But, as an Office application, Excel shares Word's interface–somewhat smoothing the learning curve.

INSERTING AN EXCEL WORKSHEET OF A SPECIFIC SIZE

If you need to create a new set of data that is reasonably compact, Word enables you to insert a blank Excel worksheet of specific proportions. The higher your screen resolution, the more rows and columns you can insert.

To insert a worksheet using the toolbar button, click in your document where you want the worksheet to appear and click the Insert Microsoft Excel Worksheet button on the toolbar. A grid appears (see Figure 1). Drag the mouse pointer down and across the grid to define the size of your worksheet. The worksheet appears in your Word document.

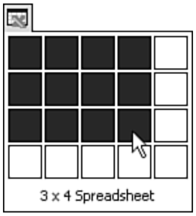


Figure 1
Dragging the Insert Excel Worksheet grid to insert a worksheet with specific proportions.

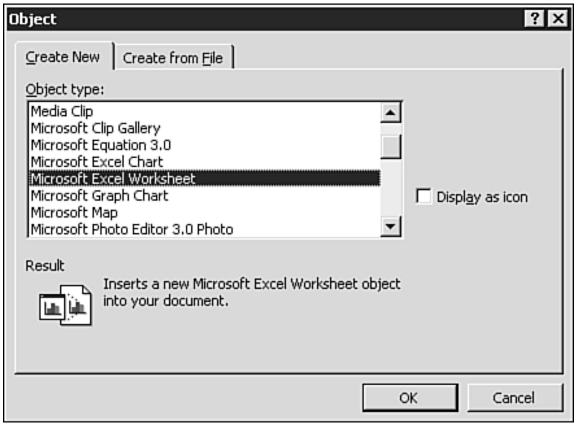
INSERTING AN EXCEL WORKSHEET THROUGH MENU COMMANDS

Another way to insert a new Excel worksheet is through menu commands. Choose Insert, Object to open the Object dialog box (see Figure 2). From the Create New tab, scroll down the Object Type list until you see Microsoft Excel Worksheet. Double-click it to insert the worksheet and close the dialog box.

The default worksheet is 7 columns wide by 10 rows high. Because this procedure can only insert a sheet of the default size, you may have to adjust the size manually afterward, by dragging on the sizing handles at the corner of the worksheet object.

Note

You can use the Object dialog box to create an object corresponding to any Windows program that supports Object Linking and Embedding (OLE), a Microsoft standard that allows you to create objects with one application and link or embed them in another. Simply choose the type of object you want to create in the Object Type scroll box.



Inserts a new Microsoft Excel Worksheet object into your document.

Figure 2
Inserting a new Excel worksheet through the Object dialog box.

Inserting a new worksheet through the Object dialog box gives you one option you don't have if you use the Insert Microsoft Excel Worksheet toolbar button. You can choose to insert the worksheet as an icon, rather than display the data itself in your document (see Figure 3). To do so, check the Display as Icon check box.

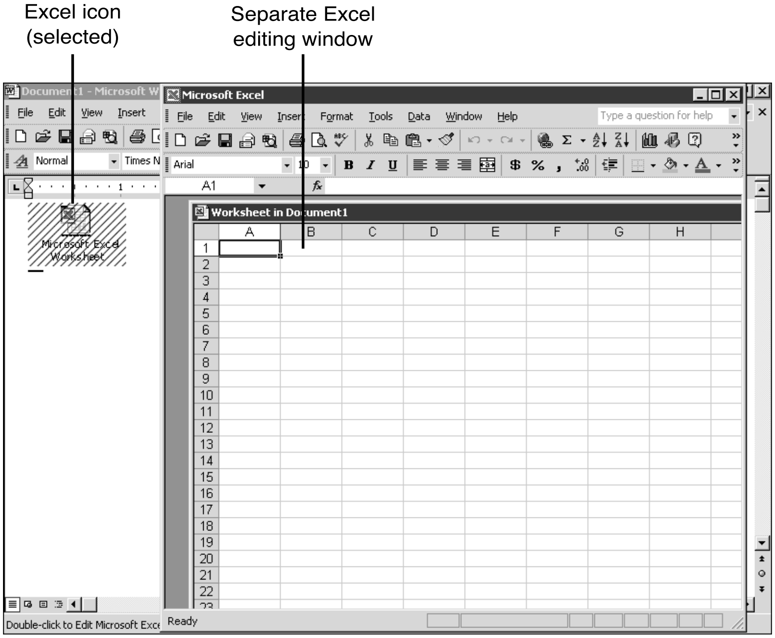


Figure 3
Displaying Excel data as an icon in your Word document.

After you insert the icon, you can double-click on it to edit the Excel worksheet in a separate Excel window.

When would you display a worksheet (or any other object) as an icon? When you won't need to print it, and when you're running Word and Excel on a relatively slow computer. You still have access to the live data, but Word runs a bit faster, because it needs to display only an icon representing the worksheet except when you are actually working with the data.

Tip from
Bill Camarda

If you choose to display worksheet cells in your document, and later decide you prefer to display an icon—or vice versa—you can easily swap between the two options. Right-click on the object to display the shortcut menu and choose Worksheet Object, Convert. The Convert dialog box appears. Check or clear the Display as Icon check box and click OK.

If you open, edit, and save an Excel worksheet in Word and then cannot reopen it in Excel, see "What to Do if You Can't Open a Worksheet in Excel After You Edit and Save It in Word," in the "Troubleshooting" section of this chapter.

Importing Excel Objects

In the previous sections, you learned how to insert a blank Excel worksheet. But it's equally likely that you'll already have an Excel workbook containing the data you want to include. Many users prefer to create their data in Excel before inserting the data into a Word document.

If you already have an Excel worksheet, Microsoft gives you plenty of options for incorporating it into a Word document. For example, you can

- Import a copy of an entire existing Excel workbook
- Import a linked version of an existing Excel workbook
- Insert a range of cells from Excel as a table in Word
- Insert a range of cells from Excel into Word and retain all the Excel formatting
- Insert a range of cells as a link from Excel to Word

Note

A word about terminology: An Excel file is called a *workbook*. Workbooks consist of one or more *worksheets*—individual "pages" of the spreadsheet that can be displayed separately.

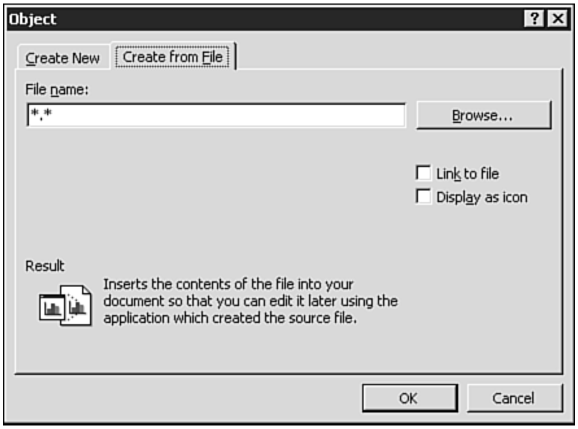
When you insert a blank worksheet into your Word document, Excel creates a workbook consisting of a single worksheet. If you insert an existing workbook, you need to pay attention to whether you're inserting one worksheet or several. Later in this section, you'll learn how to control which parts of a workbook you insert.

When you insert a workbook with multiple sheets, only the active cells in the first sheet are visible, and only those cells print. However, if you double-click on the workbook to edit it, you can choose to display the active cells from a different sheet by clicking on that sheet's tab at the bottom of the Excel window.

Importing an Entire Workbook

To import an entire workbook that already exists, follow these steps:

1. Choose Insert, Object.
2. Click the Create from File tab (see Figure 4).



File name: *.*

□ Link to file
□ Display as icon

Result

Inserts the contents of the file into your document so that you can edit it later using the application which created the source file.

Figure 4
Creating an object from an existing file.

3. Unless you know the exact filename and path, click Browse. The Browse dialog box appears, which looks like a standard Open File dialog box. (If you know the filename and path, you can simply enter it in the File Name text box, in place of the *.* characters that are present when you display this tab.)
4. Browse to and select your file.
5. Click Insert.
6. Click OK.

Tip from
Bill Camarda

You've already learned how to embed a Microsoft Excel object; you can do the same thing with any kind of object, including a PowerPoint presentation, a Visio graphic, or any other file created in an OLE-compatible application–from Microsoft or anyone else.

Choosing How to Import Your Workbook

Word gives you three options for how to import a workbook as an object:

- By default, Word embeds the workbook's contents in your document. You can edit them in Excel, but there is no connection to the original file, and if the original file changes, the change is not reflected in the Word document.

- Check the Link to File check box in the Create from File tab. This not only inserts the contents of the Excel worksheet, it also establishes a link to the source file so that updates to the source file can be reflected in the Word document.
- Check the Display as Icon check box in the Create from File tab. You've already learned that this option displays an icon in place of the worksheet cells; you can double-click the icon to open the worksheet in a separate editing window.

After you confirm your choice by clicking OK in the Object dialog box, a copy of the entire Excel workbook is inserted in your Word document as an object.

Linking to the Source File: Advantages and Disadvantages

When you choose Link to File, you're no longer working with a copy of the original file; you're working with the actual file. When you are working in Word and double-click the Excel object to edit it, Excel opens the workbook in another window, and any changes you make are incorporated into the source file. Likewise, with a linked object, any changes you make in the source file are reflected in the linked version.

Inserting a workbook as a linked file is both a blessing and a curse. The good news is that all your updates are centralized, and you don't have to worry about making changes in both Excel and Word. On the other hand, sometimes you want to lock in your report data after a certain point. Luckily, Word 2002 lets you have it both ways.

Caution

Linking worksheets into a document creates a path to the spreadsheets on your local/networked drive. If you send a file containing links to a colleague via e-mail or on a floppy disk, you need to send the linked files as well. If the linked files are in a different folder, you might need to place all the linked files in the same folder as the original file and edit the links to match for your colleague to adequately use the links.

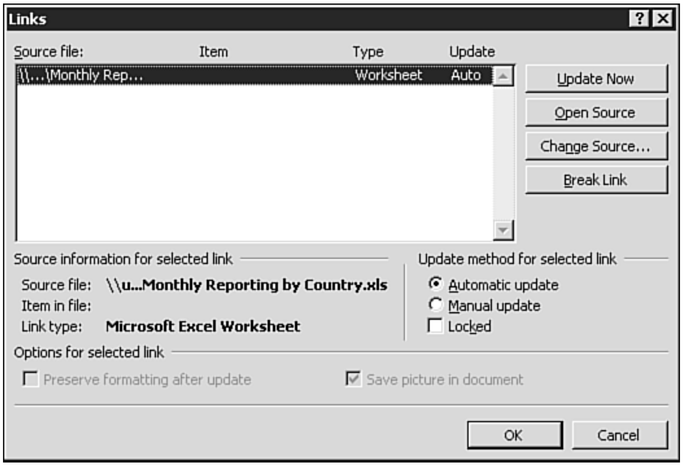
If you cannot create a link to a worksheet you recently created, see "What to Do if You Cannot Link Cells from an Excel Worksheet," in the "Troubleshooting" section of this chapter.

Modifying an Object's Links

After you have embedded a linked object into your Word document, the Edit, Links option becomes active. Choosing this menu option opens the Links dialog box, as shown in Figure 5. (You can also display the Links dialog box by right-clicking on the linked worksheet to display the shortcut menu and choosing Linked Worksheet Object, Edit Link.)

From the Links dialog box, you can choose to update your link automatically, to update it manually, or to completely lock the link.

If the Manual Update option is chosen, you must select the object (or the entire document) and press F9 or the Update Now button from the Links dialog box. Locking the link deactivates the Update Now button and prevents any updates from occurring until the link is unlocked. Your original Excel file can still be edited, but the Word file cannot be updated to reflect them.



Source file:	Item	Type	Update
\\...\Monthly Rep...		Worksheet	Auto

Source information for selected link

Source file: \\u...Monthly Reporting by Country.xls
Item in file:
Link type: Microsoft Excel Worksheet

Update method for selected link

● Automatic update
○ Manual update
□ Locked

Options for selected link

□ Preserve formatting after update
☑ Save picture in document

Figure 5
You can modify a linked object's status at any time through the Links dialog box.

RESIZING A WORKSHEET TO FIT YOUR WORD DOCUMENT

Often, after you insert an Excel worksheet, you'll discover that it is larger than your Word page. If this occurs, Word displays as much of your worksheet as it can, up to the edge of the page. Remaining cells beyond the edge are cut off and not displayed.

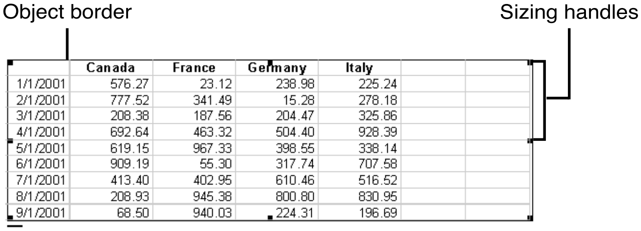
In some cases, making page setup adjustments can solve the problem. For example, to accommodate a worksheet that's too wide, you can change your page to Landscape mode. To change the page to Landscape, click on the worksheet object once to select it and choose File, Page Setup. In the Margins tab, select the Landscape option button. Choose Selected Text in the Apply To drop-down box and then click OK.

If Page Setup changes are insufficient, Word gives you two ways to resize a worksheet object, which are covered in the next two sections.

➔ For more information on sections and Landscape mode, **see** Chapter 5, "More Day-to-Day Productivity Tools," **p. 147**.

RESIZING A WORKSHEET WITHOUT CHANGING THE NUMBER OF CELLS DISPLAYED

The first approach to resizing a worksheet retains the same number of cells but shrinks or stretches the contents of each cell, changing font sizes if necessary. This approach is especially helpful if you need to make minor sizing adjustments and cannot change the number of cells that appear in your document. To resize the worksheet, click on the worksheet object once to select it—black sizing handles appear (see Figure 6). Click and drag a sizing handle to the proportions you want and then release the mouse button.



	Canada	France	Germany	Italy		
1/1/2001	576.27	23.12	238.98	225.24		
2/1/2001	777.52	341.49	15.28	278.18		
3/1/2001	208.38	187.56	204.47	325.86		
4/1/2001	692.64	463.32	504.40	928.39		
5/1/2001	619.15	967.33	398.55	338.14		
6/1/2001	909.19	55.30	317.74	707.58		
7/1/2001	413.40	402.95	610.46	516.52		
8/1/2001	208.93	945.38	800.80	830.95		
9/1/2001	68.50	940.03	224.31	196.69		

Figure 6
When you click on an Excel object, sizing handles appear within solid line borders.

Caution

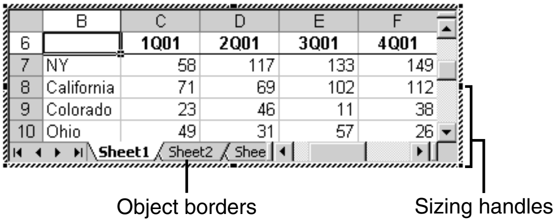
Note that when you single-click an Excel worksheet in a Word document and drag its borders to change its shape, the existing cells are stretched or squeezed. If you need to shrink the worksheet extensively, the cells may become too small to be read comfortably.

CHANGING THE NUMBER OF CELLS DISPLAYED IN YOUR DOCUMENT

Sometimes you may need to change the number of cells that appear in your Word document. For example, you may have to show fewer cells to make the information fit. Conversely, your worksheet may have changed and you need to show more cells. Perhaps you've added several new products and you need to display several additional rows of data about them in your Word document.

To adjust both the size of your worksheet object and the number of cells shown in it, follow these steps:

1. Double-click on the worksheet to edit it. Excels menus and tools appear, and the worksheet is surrounded by diagonal cross-hatching (see Figure 7).



	B	C	D	E	F
6		1Q01	2Q01	3Q01	4Q01
7	NY	58	117	133	149
8	California	71	69	102	112
9	Colorado	23	46	11	38
10	Ohio	49	31	57	26

Figure 7
When you double-click on an Excel object, sizing handles appear within diagonal cross-hatch borders.

2. Drag one of the sizing handles inward or outward to adjust the proportions of the worksheet.
3. Click outside the worksheet: The new proportions appear in your Word document.

Note

Note that when you single-click an Excel worksheet in a Word document, and drag its borders to change its shape, the existing cells are stretched or squeezed. However, when you double-click an Excel worksheet to edit its contents, you can change the number of rows and columns displayed, or the height and width of individual rows and columns.

INSERTING A RANGE OF CELLS

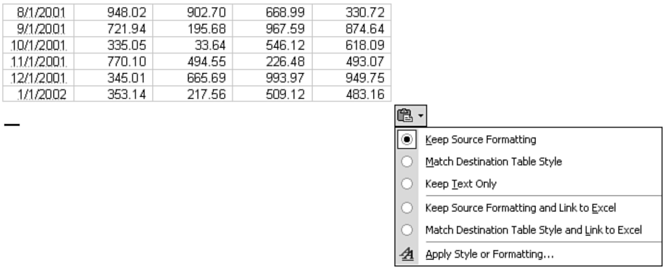
Most of the time, you won't need to insert an entire Excel workbook into your Word document, just a specific range of cells. The process for doing this is one you're familiar with: Cut and Paste. As noted previously in the discussion of inserting a workbook, you can achieve three different results when you insert a range of Excel cells: They can appear as a Word table, as an independent Excel object, or as a linked Excel object. The following sections cover each of these alternatives.

PASTING CELLS AS A WORD TABLE

The simplest method of transferring data is to select and copy it from Excel and paste it into Word by clicking the Paste button on the Standard toolbar or pressing Ctrl+V. Cutting and pasting is a good alternative for simple, fixed data where there is little or no chance of the former formulas needing to be recalculated.

Your pasted entry is converted into a Word table. By default, numbers are right-justified, and formulas become values. Most formatting is retained, with the exception of spanned columns, which can be simulated by selecting the cells in question and choosing Table, Merge Cells.

You can use Word 2002's new Paste Options button to control how information from an Excel spreadsheet is pasted into your document. Click the Paste Options button (see Figure 8) and choose the option you prefer. For example, if you are pasting Excel cells into an existing table that uses a Word table style, you can match the Excel cells to the formatting in the table style by choosing Match Destination Table Style and Link to Excel. If you want none of the formatting to import, you can choose Keep Text Only.



8/1/2001	948.02	902.70	668.99	330.72
9/1/2001	721.94	195.68	967.59	874.64
10/1/2001	335.05	33.64	546.12	618.09
11/1/2001	770.10	494.55	226.48	493.07
12/1/2001	345.01	665.69	993.97	949.75
1/1/2002	353.14	217.56	509.12	483.16

● Keep Source Formatting
○ Match Destination Table Style
○ Keep Text Only
○ Keep Source Formatting and Link to Excel
○ Match Destination Table Style and Link to Excel
Apply Style or Formatting...

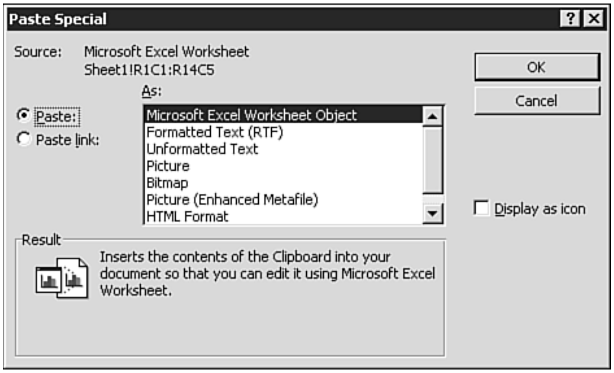
Figure 8
Using Paste Options to control how Excel cells are pasted into a Word document.

Pasting Cells as a Worksheet Object

If you think it's even remotely possible that you'll need to update your numbers and recalculate your formulas, it's best to paste the Excel data as an object. The process is basically the same as regular cutting and pasting, with one little twist: You use Edit, Paste Special instead of the Paste command. Follow these steps to insert Excel information as an object:

1. From Excel, select the range of cells you want to insert.
2. Click the Copy button on the Standard toolbar.
3. Switch to your Word document.
4. Place the insertion point where you want the data to appear.
5. Choose Edit, Paste Special. The Paste Special dialog box opens, as shown in Figure 9.

Figure 9
Paste Special enables you to maintain all your Excel data's formulas and formatting.



Source: Microsoft Excel Worksheet
Sheet1!R1C1:R14C5

As: Microsoft Excel Worksheet Object

● Paste:
○ Paste link:

□ Display as icon

Inserts the contents of the Clipboard into your document so that you can edit it using Microsoft Excel Worksheet.

6. From the As list, select Microsoft Excel Worksheet Object.
7. If you want to establish a link to your source document, click the Paste Link option button; otherwise, leave the Paste option button selected.
8. If you want to display the worksheet as an icon, check the Display as Icon check box.
9. Click OK to insert the object.

Tip from
Bill Camarda

You can also create a link using Paste Options. Paste the Excel cells into Word; then click the Paste Options button at the bottom right of the pasted cells and choose either Keep Source Formatting and Link to Excel or Match Destination Table Style and Link to Excel.

Inserting Worksheet Cells as a Word Hyperlink

If you click Paste Link in the Paste Special dialog box, Word displays a new set of options for the kind of links you can create. One of these is a Word hyperlink.

If you paste the worksheet cells as a Word hyperlink, the worksheet cells appear in your document at your insertion point, with blue underline formatting. When you press Ctrl and click on any of these cells, you're taken directly to the worksheet in Excel.

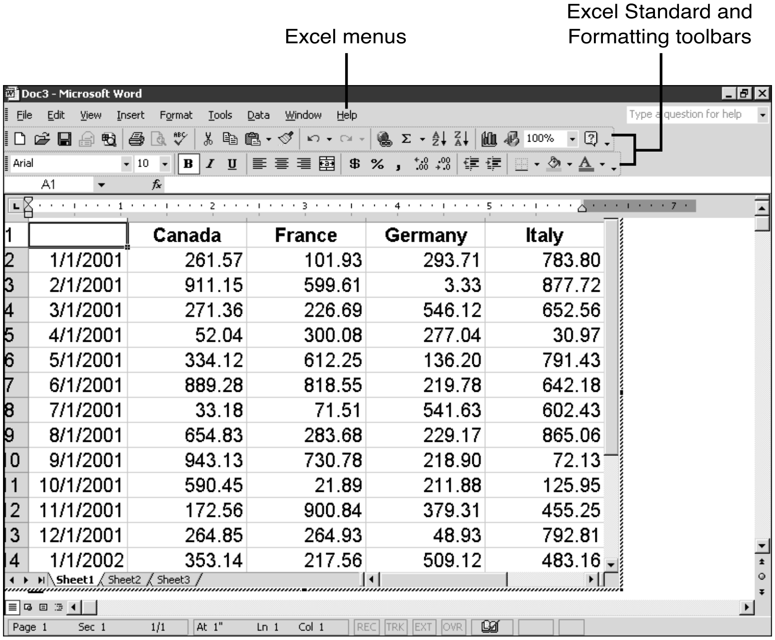
Note

To paste a spreadsheet into a document as a hyperlink, you can also select Paste as Hyperlink from the Edit menu.

Working in a Worksheet You've Inserted

Unless you insert your worksheet as an icon, the worksheet cells appear and are selected after you insert it. While the worksheet is selected, the standard Word menus and toolbars change to Excel menus and toolbars, as shown in Figure 10.

Figure 10
Word displays Excel's menus, toolbars, and other controls, and embeds a row-and-column worksheet within your document.



Excel menus

Excel Standard and Formatting toolbars

1		Canada	France	Germany	Italy
2	1/1/2001	261.57	101.93	293.71	783.80
3	2/1/2001	911.15	599.61	3.33	877.72
4	3/1/2001	271.36	226.69	546.12	652.56
5	4/1/2001	52.04	300.08	277.04	30.97
6	5/1/2001	334.12	612.25	136.20	791.43
7	6/1/2001	889.28	818.55	219.78	642.18
8	7/1/2001	33.18	71.51	541.63	602.43
9	8/1/2001	654.83	283.68	229.17	865.06
10	9/1/2001	943.13	730.78	218.90	72.13
11	10/1/2001	590.45	21.89	211.88	125.95
12	11/1/2001	172.56	900.84	379.31	455.25
13	12/1/2001	264.85	264.93	48.93	792.81
14	1/1/2002	353.14	217.56	509.12	483.16

After you insert your worksheet, you'll notice the typical Excel layout with alphabetic column headings and numeric row headings. You can now enter and format your data and formulas in the worksheet's rows and columns as you would if you were working in Excel. You can also resize the columns just as you can in Excel—by moving your mouse pointer over the boundary of a column and then clicking and dragging the boundary to a new width.

Note

For all practical purposes, you are working in Excel: You just happen to be looking at it through a narrow portal that Word has provided.

Tip from
Bill Camarda

Excel 2002 supports multiple worksheets accessed by clicking one of the sheet tabs at the bottom of the Excel window.

When you finish editing inside the worksheet, click anywhere outside it, and your menus and toolbars return to their usual appearance, displaying Word commands. The borders of each cell are displayed in light, nonprinting gray. You can work inside the worksheet again at any time by double-clicking on it, or by right-clicking and choosing Worksheet Object, Edit from the shortcut menu.

If you've inserted the Excel worksheet cells with a link to an Excel workbook elsewhere, the changes you make to the worksheet cells are saved to the original workbook when you save your Word document. If you've inserted a new Excel workbook, the changes are saved only as part of the Word document, not as part of a separate workbook.

DELETING AN EXCEL WORKSHEET FROM A WORD DOCUMENT

To delete an Excel worksheet from your Word document, first make sure that you're in Word editing mode (if the Excel layout is active, click outside the worksheet into the regular document). Then select the object by clicking it once and press Delete.

If you have inserted a workbook with several worksheets, you can delete a single worksheet from within Word. Double-click on the worksheet to select it for editing and right-click on the worksheet tab you want to delete. Choose Delete from the Excel shortcut menu that appears.

CREATING CHARTS IN EXCEL

In Chapter 17, "Using Graphs to Make Sense of Your Data—Visually," you learned how to create charts in Microsoft Word, using the Microsoft Graph applet. However, you may have already built your charts in Excel. Word makes it easy to insert these charts into Word documents:

1. Starting in Excel, click on the chart to select it.
2. In Excel, click the Copy button on the Standard toolbar to copy the chart to the Clipboard.
3. Switch to Word and place your insertion point where you want the chart to appear.
4. If you don't need a link to the original chart, click the Paste button on the Standard toolbar. Otherwise, choose Edit, Paste Special.

5. If you want a link, click the Paste Link button.
6. Click OK.

In the Paste Special dialog box, Word gives you two options for the type of chart object you can create:

- **Microsoft Excel Chart Object**—This is the default setting, and it enables you to double-click on the chart to edit it in Excel.
- **Picture (Enhanced Metafile)**—With this option selected, your chart is inserted as a bitmap graphic that can be edited using tools such as Microsoft Paint or Microsoft Photo Editor.

WORKING WITH ACCESS AND WORD

Increasingly, much of an office's day-to-day business involves keeping track of data. Take an overdue invoice notice, for example. Although the letter you send to your debtor is typically a word processing document, the key bits of information incorporated in it may well be stored in a database: the business name and address, invoice numbers, amount due, and so forth. Word's tight integration with Microsoft Access enables you to produce an unlimited number of reports, letters, labels, and other documents based on the same data source.

In Chapter 18, "Using Mail Merge Effectively," (see the section titled "Using an Access Database as a Data Source"), you walked step by step through creating a mail merge using a Microsoft Access database as a data source. This may be the most common scenario for using Access data in Word, but it's far from the only one. For example, you may want to

- Incorporate specific, filtered elements of a database in a document you're creating, such as a discussion of new customers or sales opportunities
- Create a report based entirely (or largely) on Access data but utilize Word's more sophisticated formatting capabilities

Aside from mail merge, there are three ways to retrieve Access data for use in Word:

- You can use Word's Database toolbar to specify which Access data you want and manage it after you insert it. This technique often makes sense when you're incorporating Access data into a Word document that already exists.
- You can use Access's Publish It with MS Word feature to build an RTF (Rich Text Format) file that can be edited and formatted in Word. If you're at least reasonably familiar with Access, this is the fastest way to create a Word document containing large amounts of Access data.
- You can use Access's standard data export tools, which enable you to specify the name, placement, and type of file Access creates when it exports data.

The following sections walk you through each of these techniques. But first, the next section provides some background about databases and how Access works with Word.

➔ For more information on running mail merges, **see** "An Overview of Word's Mail Merge," **p. 558**.

SOME IMPORTANT POINTS ABOUT ACCESS DATABASES

Remember that a database is made up of many records, each of which represents one business, person, or transaction. The records, in turn, are composed of a number of different fields. Each field represents one unique aspect of a record, such as a first name, last name, street address, ZIP Code, and so on. Databases are often represented as tables with each column representing a different field and each row a different record. The first row of the table is reserved for the field names and is called the *header row*.

In Access, you can enter field names of up to 64 characters with spaces and most special characters. However, if you're planning on using your database in Word, it's best to limit your field names to 20 or fewer characters and avoid spaces and any special characters other than the underscore. Otherwise, when you link your Word document with your Access database while setting up a mail merge, Word automatically truncates the field names to 20 characters and alters any spaces or special characters to the underscore character.

Word inserts your Access data just as it appears in Access. So if your AmountDue field isn't formatted to show dollars and cents in Access, it won't show up that way in Word. The same is true of date fields and text fields.

INTEGRATING MICROSOFT ACCESS DATA INTO AN EXISTING WORD DOCUMENT

Imagine that you're writing a report to management that describes all the new sales opportunities your division has generated in the past 30 days. You've already written plenty of glowing prose about your sales team's hard work. Now it's time to get down to cases: Which companies represent the largest sales potential? The data is stored in Access. You want to include it in your Word report, in the form of a Word table. In this section, you'll learn how to make that happen.

Tip from
Bill Camarda

If possible, familiarize yourself ahead of time with the specific Access database file you'll be using. For example, you need to know in which database tables the information you're seeking is stored, and which fields exist in those tables.

Follow these steps to import data into Word from Access:

1. Choose View, Toolbars, Database to display the Database toolbar (see Figure 11).
2. Click the Insert Database button. The Database dialog box opens (see Figure 12).
3. Click Get Data. The Open Data Source dialog box opens, which looks like a standard Open File dialog box.

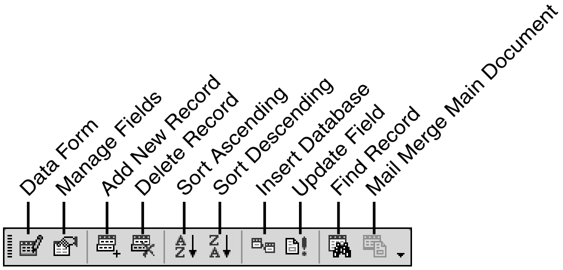


Figure 11
The Database toolbar enables you to import and control database data.

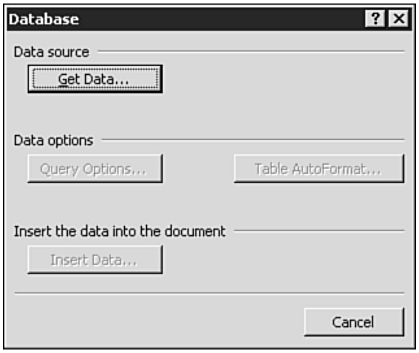


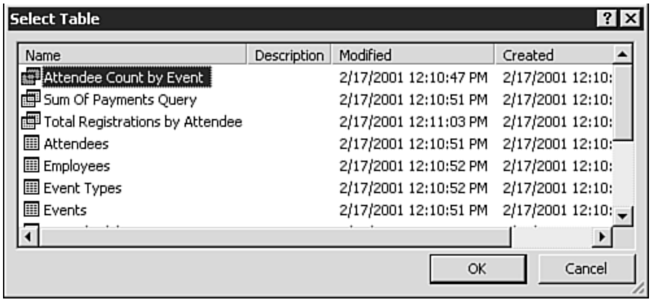
Figure 12
The first step in integrating Access data into a Word document is to click Get Data and retrieve the data you need.

Tip from *Bill Camarda*

If you want to insert another type of data source, such as an Excel worksheet, choose that option instead.

If your data is actually stored in an SQL Server 2000 database accessed through a Microsoft Access front end, choose Access Projects from the Files of Type drop-down box.

4. Browse to and select the database you want to use.
5. Click Open. The Select Table dialog box opens (see Figure 13).
6. Choose the table that contains the database fields you want to use. (Or, if a predefined query exists within Access that generates the data you want, click the Query Options tab and choose the query instead.)
7. Click OK.



Name	Description	Modified	Created
Attendee Count by Event		2/17/2001 12:10:47 PM	2/17/2001 12:10:
Sum Of Payments Query		2/17/2001 12:10:51 PM	2/17/2001 12:10:
Total Registrations by Attendee		2/17/2001 12:11:03 PM	2/17/2001 12:10:
Attendees		2/17/2001 12:10:51 PM	2/17/2001 12:10:
Employees		2/17/2001 12:10:52 PM	2/17/2001 12:10:
Event Types		2/17/2001 12:10:52 PM	2/17/2001 12:10:
Events		2/17/2001 12:10:51 PM	2/17/2001 12:10:

Figure 13
Choosing a database table or preexisting query.

You've now connected your Access database to your Word document, but you haven't actually inserted any data yet. At this point, you have a choice:

- If you want to insert all the data stored in the database table you selected (or associated with the query you selected)—or specific consecutive numbered records from that table or query—, you can click Insert Data in the Database dialog box and work from there. (See "Inserting Data Through the Insert Data Dialog Box," later in this chapter).
- If you want to create a more complex query, sort the information placed in your document, or choose which fields to include, click Query Options in the Database dialog box. The Query Options dialog box opens, offering options for specifying exactly which data to include in your Word document. Query Options is covered in the next section.

Creating Query Options

As covered in the previous sections, you manage the process of importing data from Access through the Database dialog box. After you've selected the file that contains your data, you specify which data you want to use by clicking Query Options in the Database dialog box.

Most of Word's database Query Options are covered at length in Chapter 20, in the section titled "Choosing the Records to Merge" Briefly, however

- In the Filter Records tab (see Figure 14), you can filter which data appears, based on any field in the database table you've chosen. You can also use comparisons such as "Equal to" and "Is Not Blank" to refine the data further. Finally, you can establish six different criteria to widen or narrow the "data net" you're casting.
- In the Sort Records tab (see Figure 15), you can choose which field or fields to sort by. You can choose up to three sort fields and specify whether each of them sorts in ascending or descending order.
- Finally, in the Select Fields tab—which is not available in mail merge—you can choose which fields from your data source are placed in your document. This tab is shown in Figure 16.

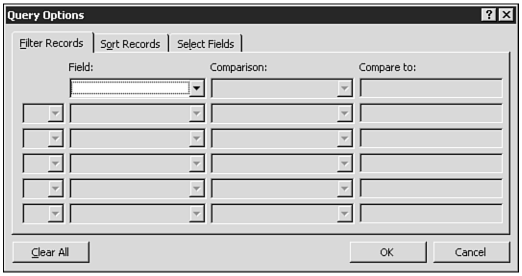


Figure 14
The Filter Records tab of the Query Options dialog box.

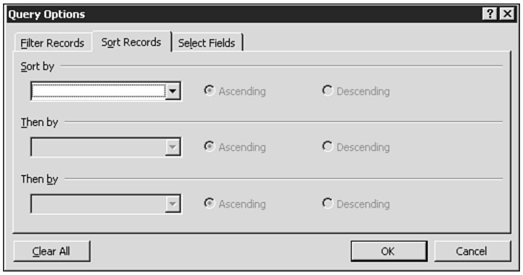
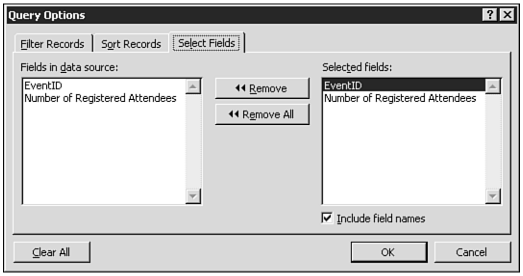


Figure 15
The Sort Records tab enables you to specify a sort order for your data, based on up to three fields.



Fields in data source:
EventID
Number of Registered Attendees

Selected fields:
EventID
Number of Registered Attendees

☑ Include field names

Figure 16
The Select Fields tab enables you to control which fields of data are placed in your Word document.

By default, every field in the data source is selected. To remove one field, highlight it in the Selected Fields scroll box and click Remove. To remove all fields, click Remove All. You can then add fields back one at a time. In the Fields in Data Source scroll box, highlight the field you want to add. The Remove and Remove All buttons are replaced by Select and Select All buttons. Click Select.

If you don't want a header row containing the field names, clear the Include Field Names check box. When you're finished, click OK.

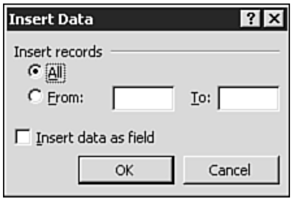
AutoFormatting Your Data Before You Insert It

In Chapter 13, you learned about Word's tools for AutoFormatting a table after you place it in your document. Because you might want to AutoFormat the information in the database you import, the Database dialog box (refer to Figure 13) enables you to choose Table AutoFormat before you place the information in your document.

After you complete the Get Data step and set Query Options (if any), click Table AutoFormat. The Table AutoFormat dialog box appears. Choose the Formats and other settings you want, and click OK to return to the Database dialog box.

INSERTING DATA THROUGH THE INSERT DATA DIALOG BOX

Now you're ready to insert the data. In the Database dialog box (refer to Figure 12), click Insert Data. The Insert Data dialog box opens (see Figure 17). Here, you have one last chance to refine your data. By default, Word searches the entire database for records that fit the query options you've already set. If you prefer to search only a specific range of numbered records, enter the range in the From and To text boxes.



Insert Data

Insert records

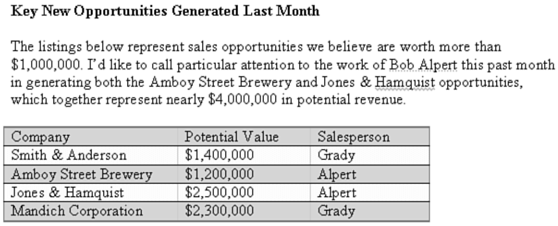
● All

○ From: To:

□ Insert data as field

Figure 17
In the Insert Data dialog box, you can specify a range of records, or specify that data be inserted as an updatable field.

Finally, if you want the data to be inserted as a { DATABASE } field that can be updated based on the source database file, check the Insert Data as Field check box. When you're finished, click OK. The information appears in your document (see Figure 18).



Key New Opportunities Generated Last Month

The listings below represent sales opportunities we believe are worth more than $1,000,000. I'd like to call particular attention to the work of Bob Alpert this past month in generating both the Amboy Street Brewery and Jones & Hamquist opportunities, which together represent nearly $4,000,000 in potential revenue.

Company	Potential Value	Salesperson
Smith & Anderson	$1,400,000	Grady
Amboy Street Brewery	$1,200,000	Alpert
Jones & Hamquist	$2,500,000	Alpert
Mandich Corporation	$2,300,000	Grady

Figure 18
Formatted database data inserted in a Word document.

➔ For more information about working with fields, **see** Chapter 26, "Automating Your Documents with Field Codes," **p. 793**.

Tip from
Bill Camarda

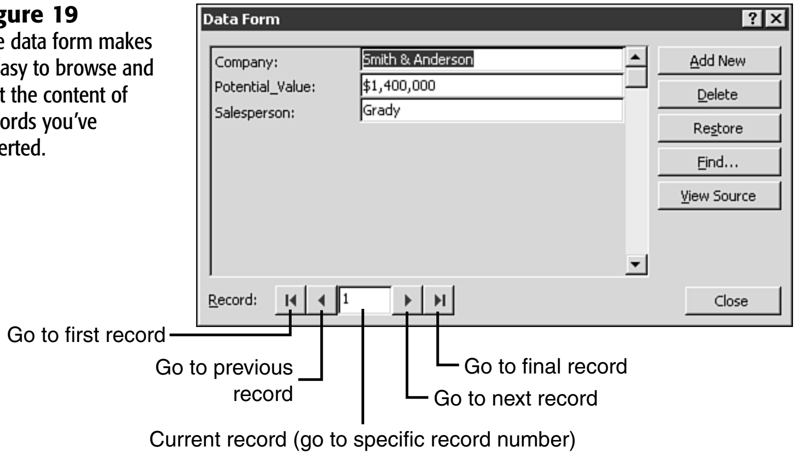
If you've inserted a { DATABASE } field linked to a database file, you can update the database information at any time by clicking inside the data in your document and either pressing F9 or clicking the Update Field button on the Database toolbar.

Using the Database Toolbar to Modify Data Placed in Your Document

After you've inserted database data in your document, the Database toolbar gives you tools for managing it. The following sections cover each of the tools.

Viewing the Data in a Form

To display the data through a data form, which makes it easier for you to move among records and edit them, click Data Form (refer to Figure 11). Figure 19 shows the navigation tools available in a data form.



Data Form

Company: Smith & Anderson
Potential_Value: $1,400,000
Salesperson: Grady

Record: 1

Go to first record

Go to previous record

Current record (go to specific record number)

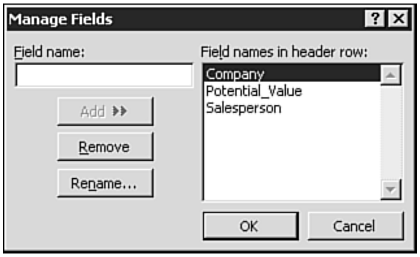
Go to final record

Go to next record

Figure 19
The data form makes it easy to browse and edit the content of records you've inserted.

Managing Fields

To add, rename, or delete fields displayed in the table, click Manage Fields (refer to Figure 11). The Manage Fields dialog box opens (see Figure 20).



Manage Fields

Field name:

Field names in header row:
Company
Potential_Value
Salesperson

Figure 20
In the Manage Fields dialog box, you can add, rename, or remove fields from the database data in your document.

From here, you can

- Add a new field by typing a new Field Name and clicking Add
- Remove a field by selecting it in the Field Names In Header Row scrollbox and clicking Remove.
- Rename a field by selecting it in the Field Names In Header Row scrollbox and clicking Rename.

Adding, Deleting, and Sorting Records

To add a blank line where you can enter an additional database record, click Add New Record. This adds a new column to the Word table Access has entered in your document.

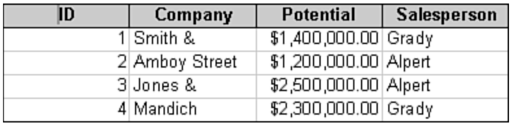
To remove an existing record, click inside it and click Delete Record.

To sort the records (excluding the header row), click in the column you want to sort by; then click Sort Ascending. To sort in reverse order, click Sort Descending.

Publishing Access Data to Word Via Office Links

Microsoft Access contains a powerful shortcut for exporting data to Word: the Publish It with MS Word feature, one of a family of "Office Links" features. To publish database information using this feature, open the Access database you want to use. You can publish to Word directly from an Access table, query, form, or report.

To publish to Word, select or display the item from which you want to publish. Then choose Tools, Office Links, Publish It with MS Word. Access creates an RTF file containing a table of database information and immediately opens the RTF file as a table in a separate Word document window (see Figure 21). You can then save the RTF file as a Word document; copy the data into any other Word document; link or embed the Word file; or edit and format your file as needed.



ID	Company	Potential	Salesperson
1	Smith &	$1,400,000.00	Grady
2	Amboy Street	$1,200,000.00	Alpert
3	Jones &	$2,500,000.00	Alpert
4	Mandich	$2,300,000.00	Grady

Figure 21
An Access table, published to Word.

Keep the following points in mind:

- If you publish to Word from a report or form, the appearance of the Word table depends on the formatting you used in the Access report or form. You may find this unsatisfactory. If so, use Table AutoFormat or Word's manual formatting techniques to reformat the data after it appears in Word.

- As shown in Figure 22, a published Access table may include a column for the primary key, which you may not want. You can, however, manually delete this, by selecting the column and clicking Cut or pressing Ctrl+X.
- If you publish from an Access report, the information isn't stored in a Word table at all; rather, Access uses tabs to separate columns. You may want to immediately select the data and choose Table, Convert Text to Table to reformat the information in a Word table, which is almost always easier to manage.
- If you publish from a query or a report, you can refine the information you select using Access query and reporting tools, prior to inserting it in Word.
- Regardless of which Access source you use, you may need to adjust column widths and alignments after your data is displayed in Word.
- Access automatically names the RTF file after the table, report, form, or query on which it was based, unless it finds an RTF file in the same directory that already uses this name. In that case, you're given a chance to rename the file.
- The published data does not retain any link to Access, so it is not automatically updated if the database is updated. Often, the best way to update information you've published to Word is to simply delete it and republish it using Access's Publish It with MS Word feature again.

GETTING MORE CONTROL OVER ACCESS DATA EXPORT

Access's one-click Publish It with MS Word feature is handy, but you may need more control over how you export Access data to Word. If so, open Access and select the form, report, table, or query you want to export. Then, from within Access, choose File, Export. The Export To dialog box, which looks similar to a normal Save As dialog box, appears.

From this dialog box you can

- Specify a File Name
- Browse to the folder where you want the file to appear
- Specify a file type (Save as Type); possibilities include Rich Text Format (RTF), Text Files, or HTML Documents

When you're finished, click Save. You now have a file with the name you selected, in the format and location you selected, containing all the information from the table, form, report, or query. You can open or insert this file in Word and edit it there as you want.

USING POWERPOINT WITH WORD

Over the years, PowerPoint has developed into the multimedia center of Office. Not only can you use it to incorporate text and images in a slide format, but you can also add animation, sound, and movies to teach, explain, and persuade. Because both Word and

PowerPoint are Office family members, you have a strong connection that enables you to share information in both directions.

Consider just a few of the ways you can use PowerPoint and Word together:

- You can draft your outlines in Word and then import them into PowerPoint.
- You can send presentation information from PowerPoint to Word, including outlines, notes, and handouts.
- You can take quick notes and mark action items in PowerPoint while you are making a presentation in front of an audience and then publish them to Word for editing and refinement.
- You can embed slides or an entire presentation into a Word document for inclusion in a report.

In the following section, you'll learn how you can use an outline from Word as the basis for a PowerPoint multimedia presentation.

USING WORD OUTLINES IN POWERPOINT

Often, the impetus for a PowerPoint presentation is a report drafted in Word. If you use Heading styles or Outline levels in Word, you can build your presentation with much less effort. The key to "no-brainer" Word-to-PowerPoint conversions is setting up your Word document using heading styles or outline levels.

Each paragraph or title, formatted with Heading 1 style (or Outline Level 1), becomes the title of a new slide, and each paragraph or subtitle, formatted with the Heading 2 style (or Outline Level 2), becomes the first level of text, and so on.

To export a finished Word outline to PowerPoint, open the document and choose File, Send To, Microsoft PowerPoint. You'll see a brief progress bar and then the presentation appears in PowerPoint, as shown in Figure 22.

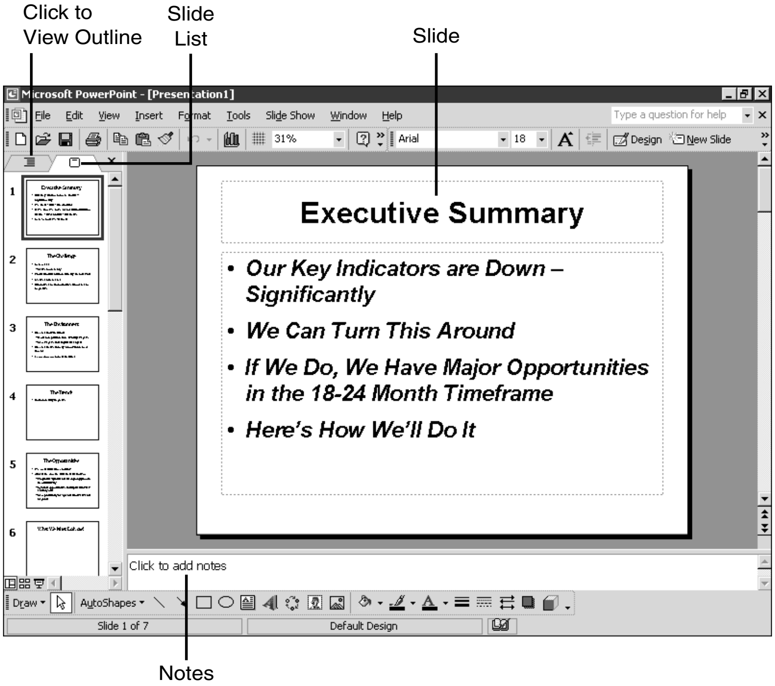
Tip from
Bill Camarda

If you haven't used heading styles or outline levels in Word, but you have used tabs to indicate headings and subheadings, save your document as a text file. Then, in PowerPoint, choose File, Open. Change the file type to All Outlines, select your text file, and choose Open. Each first non-indented paragraph becomes a slide title on a new slide; paragraphs with one indent become first-level text, and so forth.

EXPORTING POWERPOINT FILES TO WORD

The slides are the flashy part of a PowerPoint presentation, but there's also a lot of supporting material: an outline, speaker notes, and handouts, to name a few. PowerPoint makes it easy for you to export all this information and more to Word, where you can further modify it or incorporate it into an existing document. Moreover, through PowerPoint's Meeting Minder utility, you can take notes during your presentation and even assign tasks or actions; later, you can export these items to Word as well.

Figure 22
Build your presentation's "talking points" by sending your Word outline to a PowerPoint presentation.



Executive Summary

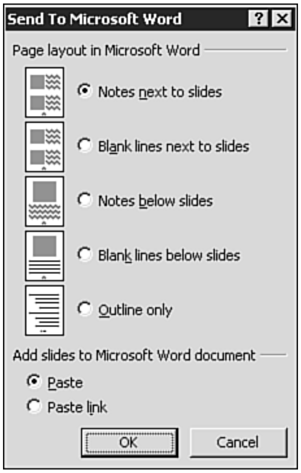
- Our Key Indicators are Down – Significantly
- We Can Turn This Around
- If We Do, We Have Major Opportunities in the 18-24 Month Timeframe
- Here's How We'll Do It

When giving a presentation, it's useful to have a hard copy of your talk that tells you what slide comes next and what to say about it. After you've created your slide show, you can export it to Word so that you can further develop your speaker's notes. From PowerPoint's main menu, choose File, Send To, Microsoft Word. This opens the Send to Microsoft Word dialog box shown in Figure 23.

The Send to Microsoft Word dialog box comes with various layout options that not only govern how your notes will look but also what you are actually exporting. You can choose from the following options:

- **Notes Next to Slides**—With this option selected, Word creates a 3-column table showing the slide number, a small image of the slide, and speaker notes. Three slides appear per page.
- **Blank Lines Next to Slides**—This option creates a 3-column table showing slide number, a small image of the slide, and a series of underscored lines for speaker notes. Three slides appear per page.

- **Notes Below Slides**—When this option is selected, Word puts each slide on its own page with a slide number and the speaker text.
- **Blank Lines Below Slides**—This option gives you each slide on its own page with a slide number and a series of underscored lines for speaker text.
- **Outline Only**—This option gives you just the outline for the presentation without a picture of the slide. The text of the presentation appears in the form of a Word outline, with each slide's main heading corresponding to a first-level heading in Word.



Send To Microsoft Word

Page layout in Microsoft Word

● Notes next to slides
○ Blank lines next to slides
○ Notes below slides
○ Blank lines below slides
○ Outline only

Add slides to Microsoft Word document

● Paste
○ Paste link

Figure 23
You can send your entire slide presentation along with any speaker notes to Word for further editing through the Send to Microsoft Word dialog box.

You'll notice that Paste and Paste Link options are available for all but the Outline Only selection. As with all embedded objects in Word, choosing Paste Link causes the document to be updated whenever changes are made to the source material, in this case a slide or presentation.

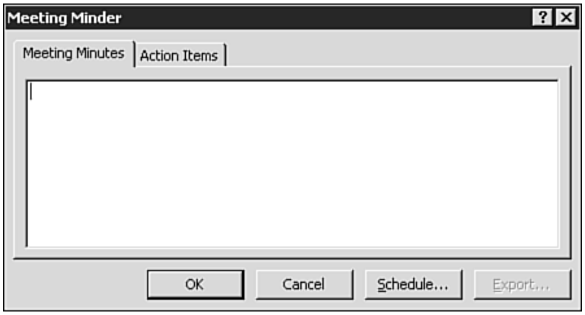
After you make your selections about how you want the PowerPoint presentation exported to Word, click OK. PowerPoint converts the file, and Word opens, displaying the file in the format you requested.

Exporting Meeting Minutes and Action Items

Meeting Minder is a PowerPoint utility that enables a presenter to take notes or keep minutes during a presentation. To use it, right-click on a slide during a presentation and select Meeting Minder from the shortcut menu (see Figure 24).

The Meeting Minder dialog box has two tabs: Meeting Minutes and Action Items. Meeting Minutes is an open-ended text area; Action Items are task-oriented to-do items. After you enter text in either tab, the Export button becomes available; click it. In the Meeting Minder Export dialog box that appears next, make sure that Send Meeting Minutes and Action Items to Microsoft Word is checked and click Export Now.

Figure 24
You can enter meeting minutes and action items in the Meeting Minder dialog box and then export them to Word.



Note

If your workgroup uses Microsoft Outlook and Microsoft Exchange, you can also export action items directly to the task lists of the individuals you want to perform them. In the Meeting Minder Export dialog box, click Post Action Items to Microsoft Outlook and choose Export Now.

Tip from
Bill Camarda

It's best to export these items only once at the end of the session because Word creates a new document each time the information is sent.

Embedding a New PowerPoint Slide in a Word Document

PowerPoint comes with many templates and designs to give your message a professional edge. Word gives you full access to them: You can include slides in your documents at will, even if you have no intention of creating a full-fledged slide show. To insert a single PowerPoint slide as a graphic in a Word document, follow these steps:

1. In Word, position your insertion point where you want the slide inserted.
2. Choose Insert, Object to open the Object dialog box.
3. From the Create New tab, choose Microsoft PowerPoint Slide. The default slide appears as an object in your Word document.

In Word, whenever you double-click the slide, PowerPoint's menus and toolbars appear; you can now edit the slide using PowerPoint's menus and toolbars, as shown in Figure 25.

To quickly format the slide, right-click on it and choose Slide Design (see Figure 26). In the Slide Design side pane, choose the template you want to apply. (If you want larger previews, right-click on a design template and choose Show Large Previews.)

When you're satisfied, click Apply.

Figure 25
Word displays PowerPoint's menus and toolbars whenever you double-click on a slide embedded in your document.

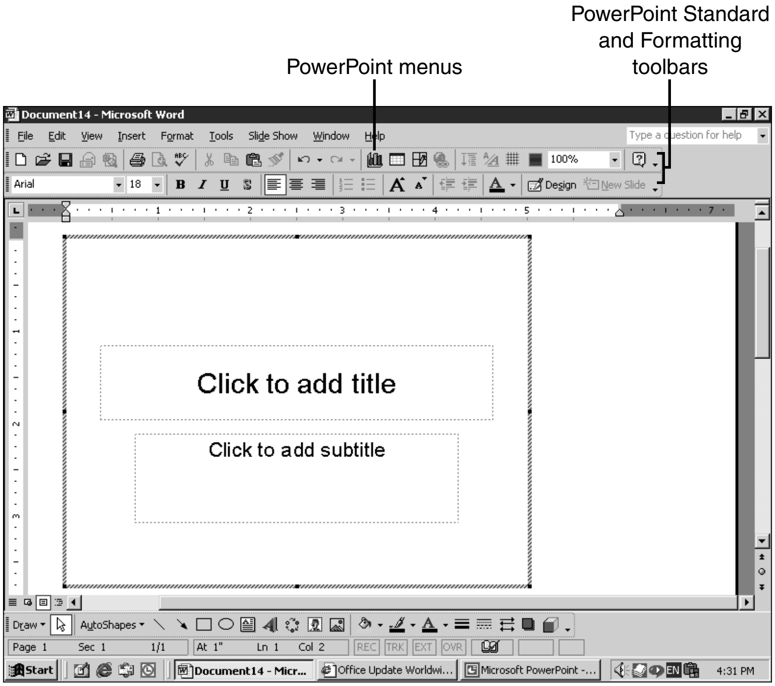
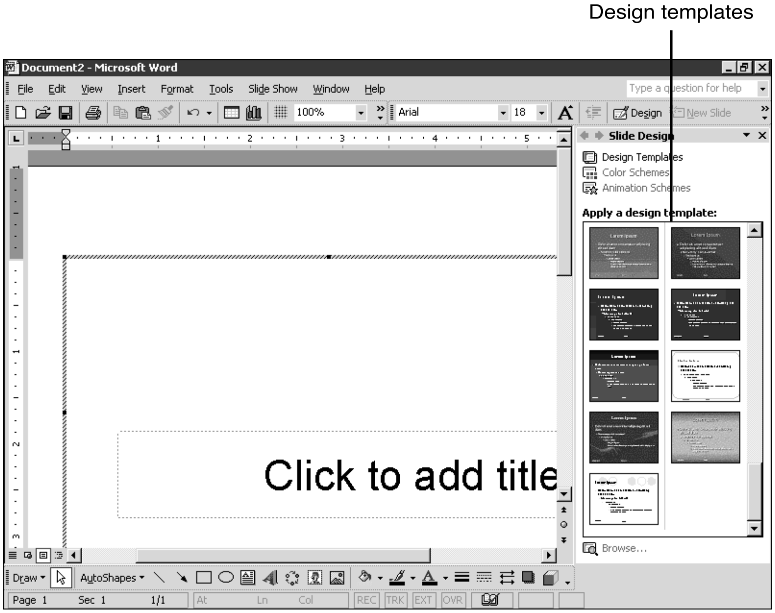

PowerPoint menus
PowerPoint Standard and Formatting toolbars
Document14 - Microsoft Word
File Edit View Insert Format Tools Slide Show Window Help
Type a question for help
100%
Arial 18
Design New Slide
Click to add title
Click to add subtitle
Draw AutoShapes
Page 1 Sec 1 1/1 At 1" Ln 1 Col 2 REC TRK EXT OVR
Start
Document14 - Micr...
Office Update Worldwi...
Microsoft PowerPoint -...
4:31 PM

Figure 26
Choosing a design template with which to format the slide.



You can now enter text in the slide. Editing text in PowerPoint slides is much like editing text in Word text boxes. You simply click next to a heading or bullet. A box that resembles a Word text box appears; start typing. When you're finished working in the slide, click outside it, and Word's menus reappear. You can work on the slide again by double-clicking it.

EMBEDDING AN EXISTING PRESENTATION IN A WORD DOCUMENT

You can also embed an entire PowerPoint slide presentation in a Word document. This works well for online training, manuals, and other documents that would benefit from a multimedia component.

Choose Insert, Object and click the Create from File tab. Next, browse to your existing presentation, select it, click Insert, and click OK. Word normally displays the first slide of the presentation in the document. To run the presentation, double-click the slide object.

Note

Of course, you can also incorporate any existing PowerPoint slide by using the Copy and Paste features. As with Excel and other objects, you can place a plain copy of your slide in your document by clicking the Paste button on the Standard toolbar.

You also can insert a linked copy of the slide by choosing Edit, Paste Special; selecting Microsoft PowerPoint Slide Object from the list; and then selecting the Paste Link option before you click OK. Now, whenever you update your PowerPoint slide, your Word document is also updated.

Tip from
Bill Camarda

If you want to copy an entire PowerPoint slide into your Word document, make sure that you're in PowerPoint's Slide Sorter view (choose View, Slide Sorter).

If you're in Slide view (rather than Slide Sorter view), only the selected portion of your slide will be copied and pasted. You will lose formatting and text stored in the slide master (in other words, content that is applied to every slide in the presentation, not just a single slide).

USING OUTLOOK WITH WORD

Microsoft Outlook 2002, Microsoft Office's "personal information manager," is included in every version of Microsoft Office XP. Outlook 2002 integrates with Word in three key ways that can make you more productive:

- If you use Outlook as your e-mail client software, you can edit your e-mail with Word. This gives you access to Word's extensive formatting and proofing capabilities. You can also send a Word file as an attachment, using the File, Send To, Mail Recipient (as Attachment) command. Using Word as your e-mail editor is covered in detail in Chapter 7, "Using Word as an E-mail Editor."

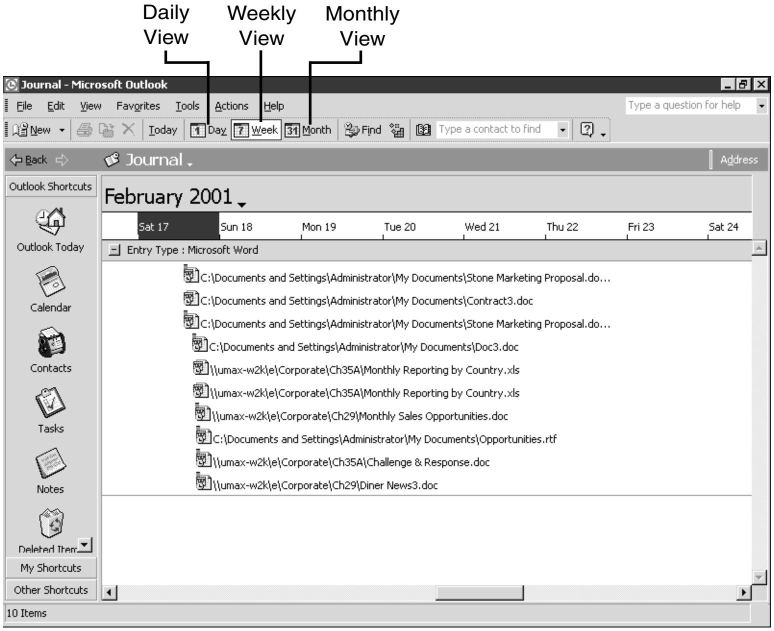
- You can track your contacts in Outlook and use your contact information in a Word mail merge (This is covered in detail in Chapter 21, in the section "Using an Outlook Address Book as a Data Source.)
- You can track your Word documents through Outlook's Journal feature, making it easy to find out what files you worked on when, and for how long you worked on them. This is covered in the following section.

Tracking Word Progress in the Outlook Journal

Did you ever wonder how much time you spent on a particular document over a series of days? Or maybe you're looking for a particular file you worked on sometime last Wednesday in the afternoon, but now you can't find it or remember its name. Outlook's Journal feature can keep track of all your work in Word (and Excel and PowerPoint), recording exactly what you worked on, when you worked on it, and for how long.

→ To learn how to turn on Outlook's Journal tracking feature, **see** "Controlling Whether Outlook Tracks Your Work" **p. 31**.

To see which entries, if any, are currently in your Journal, click the Journal folder in the Outlook bar on the left side of the screen. If the Journal folder doesn't appear, choose Journal from the list of options in your current view. You'll see a timeline of days at the top of the Journal window and filenames at the bottom (see Figure 27).



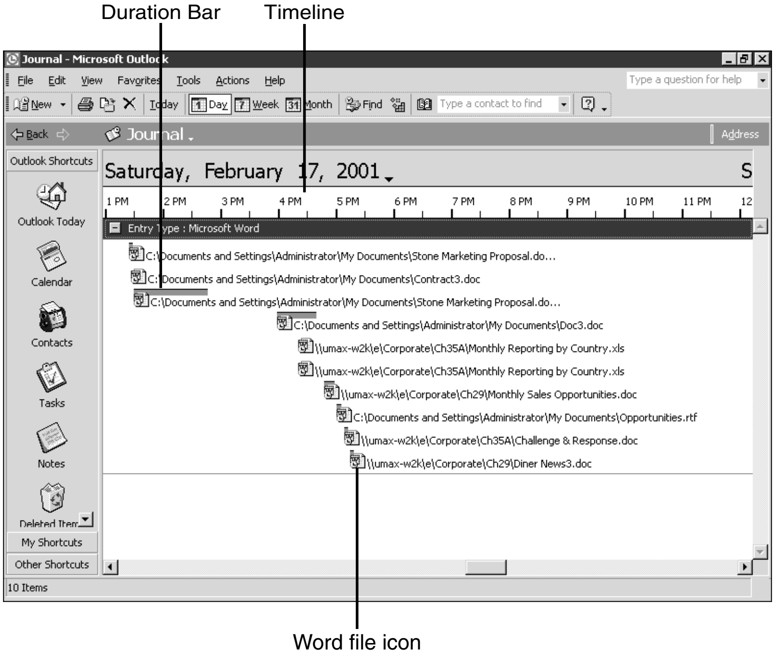
Entry Type : Microsoft Word

C:\Documents and Settings\Administrator\My Documents\Stone Marketing Proposal.do...
C:\Documents and Settings\Administrator\My Documents\Contract3.doc
C:\Documents and Settings\Administrator\My Documents\Stone Marketing Proposal.do...
C:\Documents and Settings\Administrator\My Documents\Doc3.doc
\\umax-w2k\e\Corporate\Ch35A\Monthly Reporting by Country.xls
\\umax-w2k\e\Corporate\Ch35A\Monthly Reporting by Country.xls
\\umax-w2k\e\Corporate\Ch29\Monthly Sales Opportunities.doc
C:\Documents and Settings\Administrator\My Documents\Opportunities.rtf
\\umax-w2k\e\Corporate\Ch35A\Challenge & Response.doc
\\umax-w2k\e\Corporate\Ch29\Diner News3.doc

Figure 27
Word files tracked by the Outlook Journal.

You can choose between a daily, weekly, or monthly timeline by clicking the appropriate toolbar button. Figure 28 shows a daily view. Each icon has a Duration bar above it to show when and for how long the document was open. By default, if you double-click the icon, the journal entry associated with the file opens. This journal entry contains information about the file, including exactly how long it was opened. It also contains a shortcut you can double-click to open the file itself.

Figure 28
The Journal shows how long you worked on a document and when.


Duration Bar
Timeline
Journal - Microsoft Outlook
File Edit View Favorites Tools Actions Help
Type a question for help
New Today Day Week Month Find
Type a contact to find
Back Journal Address
Outlook Shortcuts
Outlook Today
Calendar
Contacts
Tasks
Notes
Deleted Items
My Shortcuts
Other Shortcuts
Saturday, February 17, 2001
1 PM 2 PM 3 PM 4 PM 5 PM 6 PM 7 PM 8 PM 9 PM 10 PM 11 PM 12
Entry Type : Microsoft Word
10 Items
Word file icon

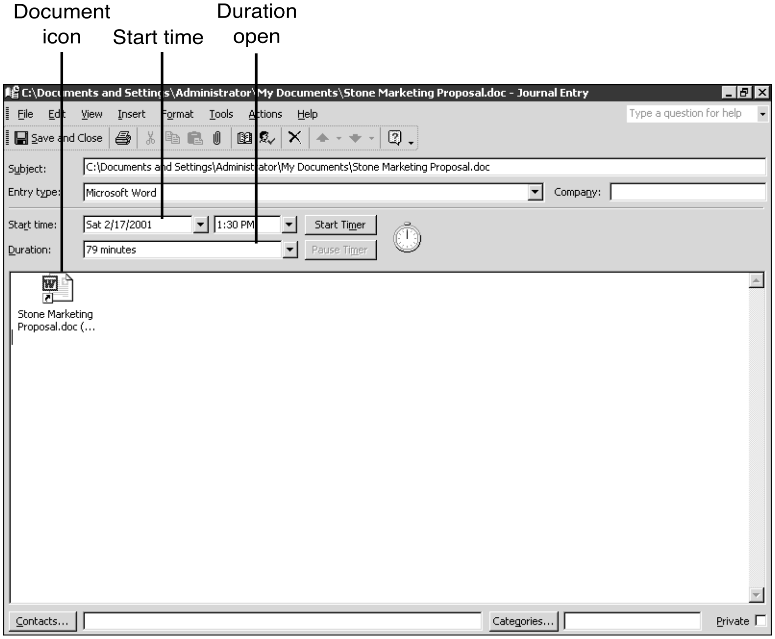
C:\Documents and Settings\Administrator\My Documents\Stone Marketing Proposal.do...
C:\Documents and Settings\Administrator\My Documents\Contract3.doc
C:\Documents and Settings\Administrator\My Documents\Stone Marketing Proposal.do...
C:\Documents and Settings\Administrator\My Documents\Doc3.doc
\\umax-w2k\e\Corporate\Ch35A\Monthly Reporting by Country.xls
\\umax-w2k\e\Corporate\Ch35A\Monthly Reporting by Country.xls
\\umax-w2k\e\Corporate\Ch29\Monthly Sales Opportunities.doc
C:\Documents and Settings\Administrator\My Documents\Opportunities.rtf
\\umax-w2k\e\Corporate\Ch35A\Challenge & Response.doc
\\umax-w2k\e\Corporate\Ch29\Diner News3.doc

You can also right-click the journal entry for a list of options. For example, Open Item Referred To opens the file the journal entry refers to, without the extra step of displaying the journal entry.

Working with the Journal Entry

An Outlook journal entry contains much more than a filename and date. To work with the journal entry, right-click it. The shortcut menu appears. Choose Open Journal Entry. The file's journal entry appears (see Figure 29).

Start Time shows when you started working with the file on that day. Duration shows how long the file was open. You can assign the file to a contact by clicking Contacts and selecting a name from your Outlook contact list. When you finish reviewing or editing the journal entry, click Save and Close.



C:\Documents and Settings\Administrator\My Documents\Stone Marketing Proposal.doc - Journal Entry

Subject: C:\Documents and Settings\Administrator\My Documents\Stone Marketing Proposal.doc

Entry type: Microsoft Word

Company:

Start time: Sat 2/17/2001 1:30 PM

Duration: 79 minutes

Stone Marketing Proposal.doc (...

Figure 29
A Word file's journal entry in Outlook XP.

Tip from
Bill Camarda

You can open the Word file directly from within Outlook.

From the Journal window, right-click on the journal entry and choose Open Item Referred To from the shortcut menu. The Opening Mail Attachment dialog box appears; click Open It and click OK.

From within the Journal Entry dialog box, double-click on the file icon. The Opening Mail Attachment dialog box appears; click Open It and click OK.

Note

To link a document to a specific contact, enter the contact name in the document's property keywords.

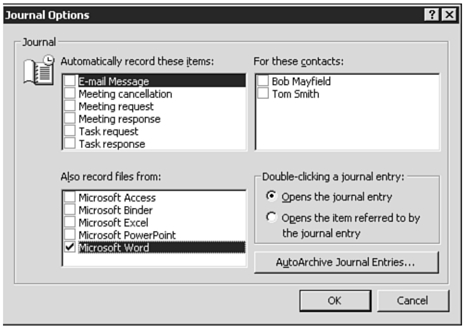
Controlling Whether Outlook Tracks Your Work

By default, the Journal tracking feature in Outlook is turned off. To turn it on, or to make sure that it is still turned off, follow these steps:

1. Open Outlook.
2. Choose Tools, Options.

3. If the Preferences tab isn't already open, click Preferences to display it.
4. Click Journal Options. The Journal Options dialog box appears (see Figure 30).

Figure 30
Changing the way Outlook tracks your work in Word.



Journal Options

Journal

Automatically record these items:
- □ E-mail Message
- □ Meeting cancellation
- □ Meeting request
- □ Meeting response
- □ Task request
- □ Task response

For these contacts:
- □ Bob Mayfield
- □ Tom Smith

Also record files from:
- □ Microsoft Access
- □ Microsoft Binder
- □ Microsoft Excel
- □ Microsoft PowerPoint
- ☑ Microsoft Word

Double-clicking a journal entry:
- ● Opens the journal entry
- ○ Opens the item referred to by the journal entry

5. In the Also Record Files From box, check or clear the check boxes next to Microsoft Word and any other Office program you want Outlook to track or to stop tracking.
6. Click OK.

After you turn on tracking, Journal tracks your work whether you open the Outlook application or not.

Integrating Word with Microsoft Publisher

If you own Office XP Small Business Edition, you also own the surprisingly powerful Microsoft Publisher, a desktop publishing program that goes far beyond Word's desktop publishing capabilities. Publisher makes it easy to build a wide range of highly visual, attractive documents.

For Word users, one of Publisher's most attractive features is its intimate integration with Word. You can use Word as your editing tool at the same time you take advantage of Publisher's desktop publishing capabilities.

When you first run Publisher, it displays a side pane at left that provides several approaches to building a new publication (see Figure 31).

The power of Publisher is its capability to simplify the creation of new publications based on existing designs—so that you don't have to create a design from scratch.

If you want to view all the publications of a specific type, choose By Publication Type from the Start From a Design drop-down box. Publisher displays a list of categories: Newsletters, Web Sites, Brochures, and so forth. Choose a category, and Publisher displays a list of Quick Publications designs at right. To build a publication based on one of these designs, click on the design you want to use.

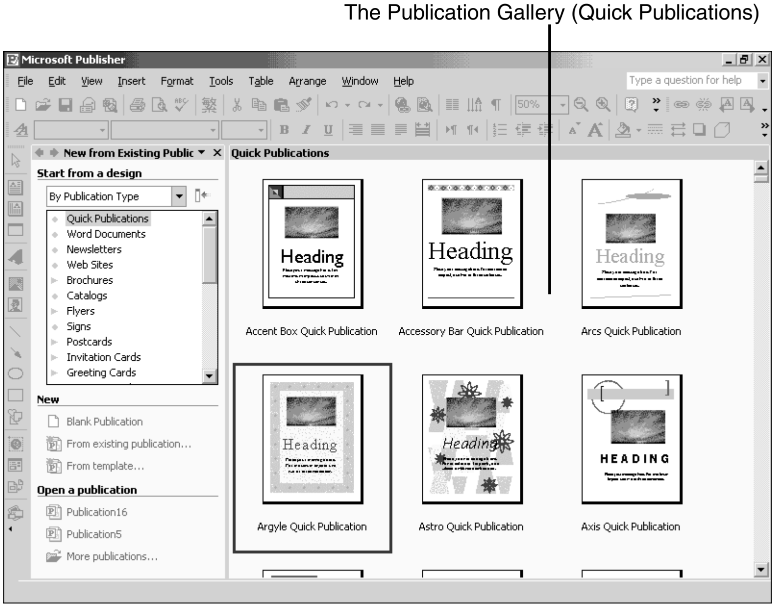


Figure 31
Microsoft Publisher displays several options for creating a new publication.

EDITING PUBLISHER TEXT IN WORD

When you finish running the wizard to create either a Publication by Wizard or a Publication by Design, a document appears, containing boilerplate text—much like the text that Word inserts when you run Word's Newsletter or Brochure Wizard.

To edit this (or any other) Publisher text using Microsoft Word, right-click in the story you want to edit. From the shortcut menu, choose Change Text, Edit Story in Microsoft Word. Word opens with the story displayed. You can now edit the story using Word's editing and formatting tools. Publisher can read nearly all text and paragraph formatting you apply in Word and vice versa. This includes boldface, italic, and underlining; styles; indents; text alignment; and bullets and numbering. There are a few exceptions, however. For instance,

- You can't insert images in Word and see them when you return to Publisher.
- Borders you apply to paragraphs in Word do not appear in Publisher.
- Publisher's built-in color schemes include some colors that may not appear accurately in Word.
- Word's animated text effects do not appear.

When you're finished editing your Publisher story in Word, choose File, Close & Return to Publication.

Inserting a Word File in a Publisher Document You've Already Created

If you want to insert a Word file in a Publisher document, click in the text frame where you want the Word document to go. Next, choose Insert, Text File. The Insert Text File dialog box opens. Browse to and select the file you want. (Publisher directly imports Word 2002, Word 2000, Word 97, Word 95, Word 6, and Rich Text Format files, as well as Word 2001, Word 98, Word 6, and RTF files created on the Macintosh.)

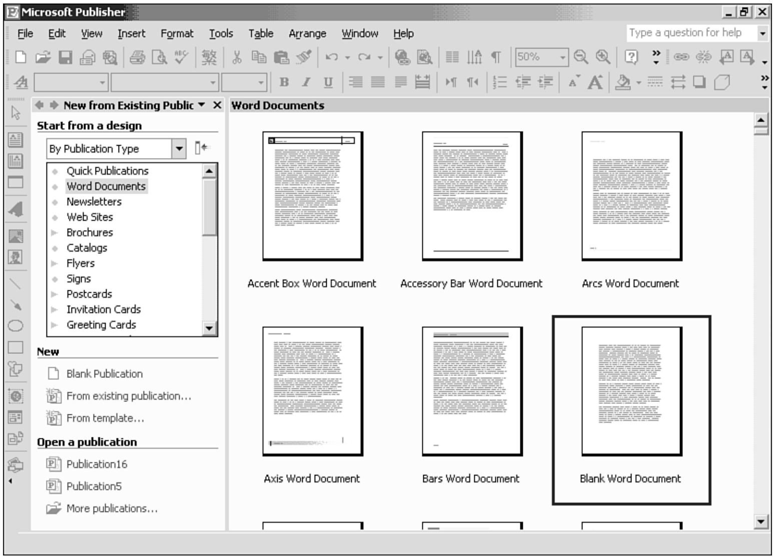
After you select the file, click OK to insert it.

Creating a New Publisher Document Based on a Designed Word Document

In some cases, you may organize and write a document in Word and add some design elements there, as well. For example, you might format your Word document in three columns. Later, you may decide to use Publisher to finish the job, adding formatting and other elements that may be easier to create there than in Word. Publisher 2002 makes this easy to do.

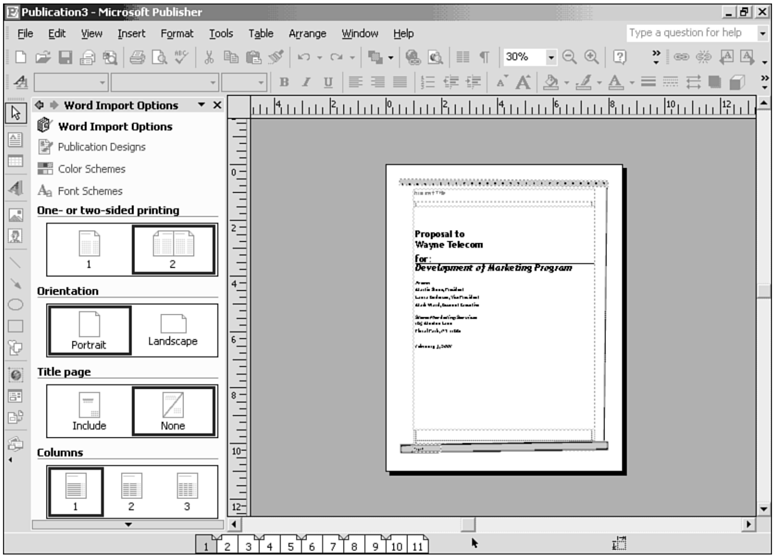
With the New Publication side pane open, choose By Publication Type from the Start From a Design drop-down box. Then choose Word Documents from the list of categories (see Figure 32).

Figure 32
Publisher's design options for importing Word documents.



Publisher displays a list of designs that can add headers, footers, or page borders to the Word document you're already planning to import. Click on a design from the Document Gallery to start working with it. If you prefer not to add Publisher's headers, footers, or borders, double-click on Blank Word Document.

Publisher displays the Import Word Document dialog box. Browse to and select the Word document you want to import. Publisher creates a new Publisher document that contains the contents of the Word file. At left, the Word Import Options pane appears (see Figure 33).



Proposal to
Wayne Telecom
for:
Development of Marketing Program

Figure 33
Adjusting settings associated with the Word document you imported.

Here, you can specify adjustments to any of three settings that existed in your previous Word document:

- One- or two-sided printing
- Orientation: Portrait or Landscape
- Columns

You can also add a title page by clicking the Include button in the Title Page area of your document. If you do, Publisher adds an attractive, consistent title page as page 1, containing information that you provided the first time you ran Publisher, such as your name and the name of your company (see Figure 34).

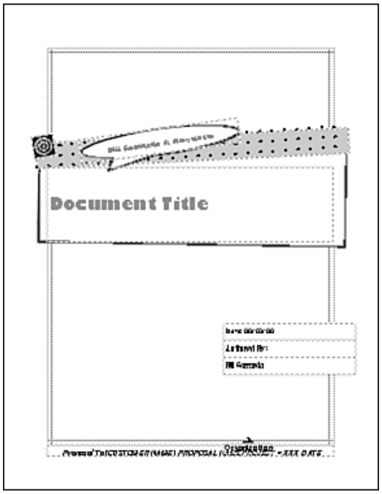


Figure 34
Adding a title page to a Publisher document you built from a Word document.

Using Microsoft Equation Editor 3.1

Microsoft Equation Editor 3.1 is useful for creating extended mathematical expressions. A variety of mathematical symbols and templates simplify the process of constructing the most complex formulas. Begin by choosing Insert, Object and selecting Microsoft Equation from the option list. A working area, surrounded by a dotted box, opens on your screen, as does the Equation toolbar shown in Figure 35.

Numbers, variables, and ordinary mathematical operators (such as =, +, –) can be typed directly from the keyboard. To enter an operator or symbol not available from the keyboard, click the appropriate button on the top row of the Equation toolbar and select from one of the options that appear in the drop-down menu.

Complex expressions such as fractions, square roots, or integrals can be added by clicking the appropriate button on the bottom row of the Equation toolbar and choosing one of the drop-down options. Depending on the choice, an expression has one or more outlined boxes for inserting numbers or other symbols.

The relative spacing for the formula is handled through the Format, Spacing menu option. This opens the Spacing dialog box (see Figure 36), where you can control the line spacing, matrix row and column spacing, superscript height, subscript depth, and the limit height between symbols.

Figure 35
Use the Equation Editor for inserting complex formulas in your Word document.

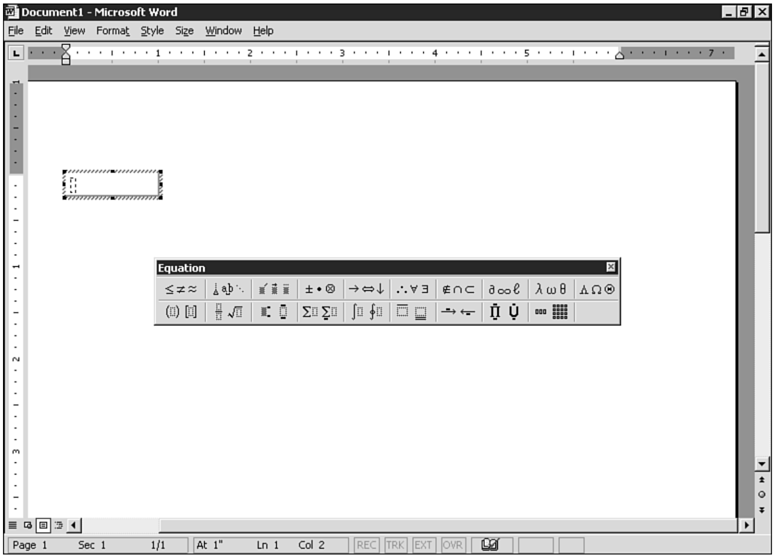
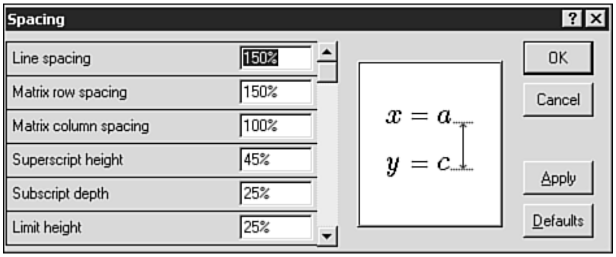

Document1 - Microsoft Word
File Edit View Format Style Size Window Help
Equation
Page 1 Sec 1 1/1 At 1" Ln 1 Col 2 REC TRK EXT OVR

Figure 36
You can control a formula's overall spacing through the Spacing dialog box.



Line spacing	150%
Matrix row spacing	150%
Matrix column spacing	100%
Superscript height	45%
Subscript depth	25%
Limit height	25%

Using Microsoft Office Document Imaging

Office XP comes with a new applet, Microsoft Office Document Imaging, which provides basic scanning and Optical Character Recognition (OCR) features that make it easy to use Word for editing documents you scanned from print sources or received as a fax and stored in TIFF format. Using Microsoft Office Document Imaging, you can perform these and other tasks:

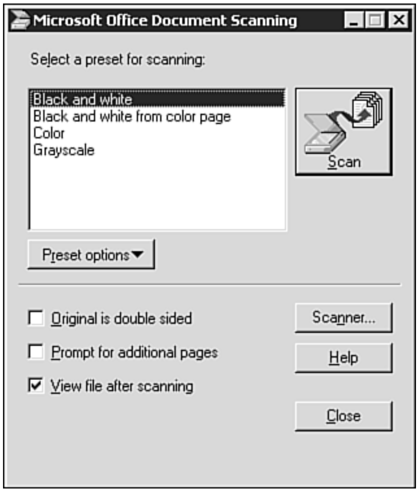
- Read and search for text in a scanned document or received fax
- Export some or all of a scanned or faxed document's text to Word
- Archive text from a scanned or faxed document
- Rearrange pages in a scanned or faxed document

Microsoft Office Document Imaging contains two components that work together: a simple Scanning applet that allows you to scan any document, whether or not Word (or any other Office application) is open; and an Imaging applet that allows you to work with—and run OCR on—any document you scan or receive by fax. We'll cover both components in the following sections.

Scanning a Document Using Microsoft Office Document Scanning

To scan a document using Microsoft Office Document Scanning, choose Start, Programs, Microsoft Office Tools, Microsoft Office Document Scanning (see Figure 37).

Figure 37
The Microsoft Office Document Scanning applet.



Microsoft Office Document Scanning

Select a preset for scanning:

- Black and white
- Black and white from color page
- Color
- Grayscale

□ Original is double sided

□ Prompt for additional pages

☑ View file after scanning

By default, Microsoft Office Document Scanning assumes that you are scanning a black-and-white document. If you are scanning a different type of document, choose it in the Select a Preset for Scanning box.

Note

Microsoft Office Document Scanning's four presets are designed to do a reasonable job with a wide variety of documents. However, if you find that the results are not quite what you hoped for, you can adjust the program's settings.

To do so, click Preset Options. To adjust an existing present, choose Edit Existing Preset. To create a new preset, choose Create New Preset.

Note

If you are connected to more than one scanner, click Scanner to choose the one you want to use.

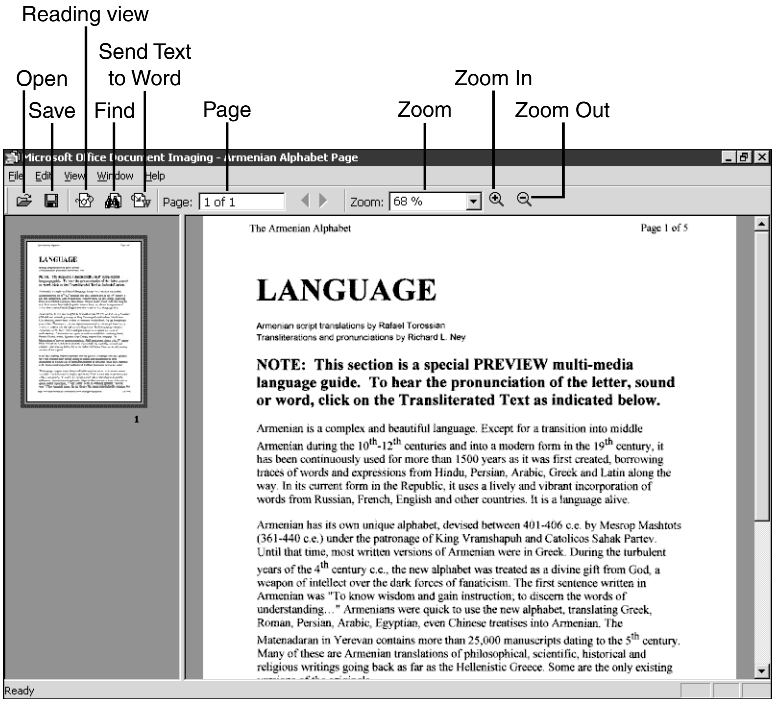
If you are scanning a document with multiple pages, check the Prompt for Additional Pages check box; Word then prompts you when it's ready for a new page. (If you don't check this box, Word ends the scan and displays the scanned page.)

If you are using a scanner with an Automatic Document Feeder (ADF) that can read both sides of a page, check the Original is Double Sided box. Make sure that you insert pages in your document feeder so that the fronts of pages are scanned first. If you do, Microsoft Office Document Scanning collates the finished scanned pages in correct order.

When you've established the settings you need, click Scan. Microsoft Office Document Scanning will scan your document.

WORKING WITH A SCANNED DOCUMENT USING MICROSOFT OFFICE DOCUMENT IMAGING

When Microsoft Office Document Scanning has completed scanning your document, it displays the finished scan in Microsoft Office Document Imaging (see Figure 38). Here, you can work with the scan you've created.



The Armenian Alphabet

Page 1 of 5

LANGUAGE

Armenian script translations by Rafael Torossian
Transliterations and pronunciations by Richard L. Ney

NOTE: This section is a special PREVIEW multi-media language guide. To hear the pronunciation of the letter, sound or word, click on the Transliterated Text as indicated below.

Armenian is a complex and beautiful language. Except for a transition into middle Armenian during the 10th-12th centuries and into a modern form in the 19th century, it has been continuously used for more than 1500 years as it was first created, borrowing traces of words and expressions from Hindu, Persian, Arabic, Greek and Latin along the way. In its current form in the Republic, it uses a lively and vibrant incorporation of words from Russian, French, English and other countries. It is a language alive.

Armenian has its own unique alphabet, devised between 401-406 c.e. by Mesrop Mashtots (361-440 c.e.) under the patronage of King Vramshapuh and Catolicos Sahak Partev. Until that time, most written versions of Armenian were in Greek. During the turbulent years of the 4th century c.e., the new alphabet was treated as a divine gift from God, a weapon of intellect over the dark forces of fanaticism. The first sentence written in Armenian was "To know wisdom and gain instruction; to discern the words of understanding…" Armenians were quick to use the new alphabet, translating Greek, Roman, Persian, Arabic, Egyptian, even Chinese treatises into Armenian. The Matenadaran in Yerevan contains more than 25,000 manuscripts dating to the 5th century. Many of these are Armenian translations of philosophical, scientific, historical and religious writings going back as far as the Hellenistic Greece. Some are the only existing

Figure 38
The Microsoft Office Document Imaging applet.

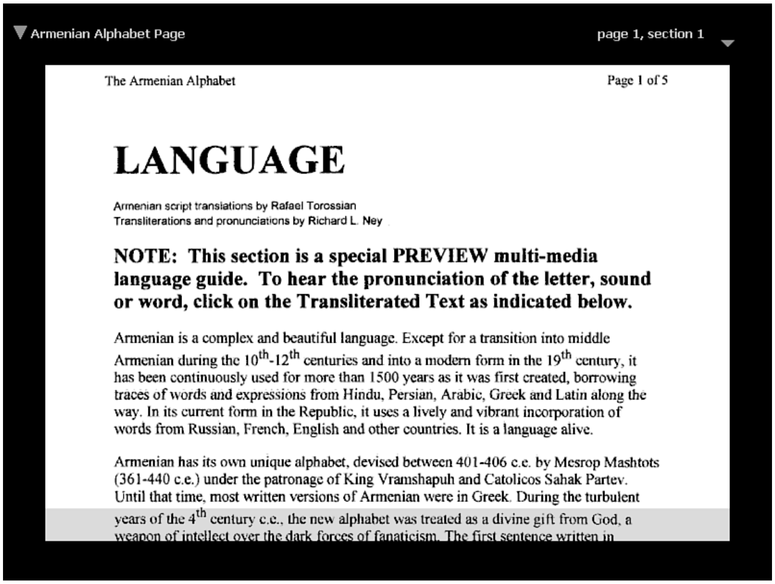
Tip from
Bill Camarda

Sometimes you may want to open Microsoft Office Document Imaging without first creating a scan. For example, you may want to run optical character resolution on a fax you received through Microsoft Windows 2000 Fax Services.

To run Microsoft Office Document Imaging on its own, choose Start, Programs, Microsoft Office Tools, Microsoft Office Document Imaging.

To view a scanned document more closely, click the Reading View button, or choose View, Reading View. The document is displayed in full size against a black background, making it easier to read (see Figure 39). When you're finished with Reading view, click Esc.

Figure 39
Viewing a scanned document in Reading view.



The Armenian Alphabet Page 1 of 5

LANGUAGE

Armenian script translations by Rafael Torossian
Transliterations and pronunciations by Richard L. Ney

NOTE: This section is a special PREVIEW multi-media language guide. To hear the pronunciation of the letter, sound or word, click on the Transliterated Text as indicated below.

Armenian is a complex and beautiful language. Except for a transition into middle Armenian during the 10th-12th centuries and into a modern form in the 19th century, it has been continuously used for more than 1500 years as it was first created, borrowing traces of words and expressions from Hindu, Persian, Arabic, Greek and Latin along the way. In its current form in the Republic, it uses a lively and vibrant incorporation of words from Russian, French, English and other countries. It is a language alive.

Armenian has its own unique alphabet, devised between 401-406 c.e. by Mesrop Mashtots (361-440 c.e.) under the patronage of King Vramshapuh and Catolicos Sahak Partev. Until that time, most written versions of Armenian were in Greek. During the turbulent years of the 4th century c.e., the new alphabet was treated as a divine gift from God, a weapon of intellect over the dark forces of fanaticism. The first sentence written in

To save a scanned document, click the Save button. The document is saved as a TIFF graphics file that can be opened with a wide range of graphics software. A filename is assigned based on the first few words of text on the first page of the document.

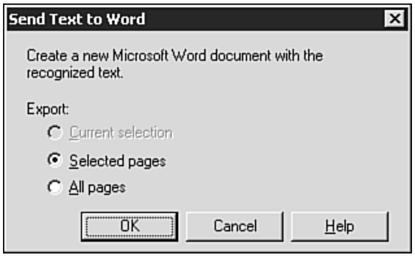
To find text within a scanned document, click the Find button and enter the text in the Find box. Microsoft Office Document Imaging searches for the first instance of the text and displays it in reverse type, with a red rectangle around it.

Caution

Because optical character recognition is not perfect, especially on printed documents of poor quality, such as second-generation photocopies or newspaper clippings, there is no guarantee that Find will locate every reference to specific text in a scanned document.

You can easily copy text from your scanned document into a Word document that can be edited. If you want to copy an entire file, click Send Text to Word, choose All Pages from the Send Text to Word dialog box (see Figure 40), and choose OK. To copy only selected text, first select the text; then click Send Text to Word and choose OK.

Figure 40
Choosing which text to send to Word.



Send Text to Word

Create a new Microsoft Word document with the recognized text.

Export:

○ Current selection

● Selected pages

○ All pages

Word opens, displaying a new, unsaved and unnamed file containing the text from the scanned document.

TROUBLESHOOTING

WHAT TO DO IF YOU CANNOT LINK CELLS FROM AN EXCEL WORKSHEET

Make sure that you've actually saved the Excel worksheet; if the worksheet doesn't have a filename yet, the link won't work properly. After you've done so, create the link again, specifying the filename you just created.

WHAT TO DO IF YOU CAN'T OPEN A WORKSHEET IN EXCEL AFTER YOU EDIT AND SAVE IT IN WORD

Word enables you to open an Excel worksheet directly, by choosing it in the Open dialog box and clicking Open. If you do this, the cells are placed in your document in the form of a Word table. Because Word cannot save in Excel format, when you resave the file, it is converted to a Word file, and Excel cannot read it. If you've made changes, you can copy them manually into Excel, though you'll have to reformat them. If you need to make changes that will be readable in Excel, use the techniques described elsewhere in this chapter to create a link to an Excel document, rather than opening it directly.

WHAT TO DO IF PARTS OF A POWERPOINT SLIDE DON'T APPEAR IN WORD

If you want to copy an entire PowerPoint slide into your Word document, make sure that you're in PowerPoint's Slide Sorter view (choose View, Slide Sorter). If you're in Slide view (rather than Slide Sorter view), only the selected portion of your slide will be copied and pasted.

This can cause problems, because you'll lose formatting stored in the slide master (in other words, most formatting that is applied to the entire presentation, not just a single slide).

What to Do if You Can't Send a Word Outline to PowerPoint

If you use File, Send To, PowerPoint, and Word displays an error message instead of opening PowerPoint, it's possible that you've recently installed the PowerPoint Viewer to view older PowerPoint files (perhaps files stored on the Internet or your intranet). You now need to reregister PowerPoint XP as the application you want to use with PowerPoint files. To do so, quit all Office programs and choose Start, Run. Then enter the following text in the Open box:

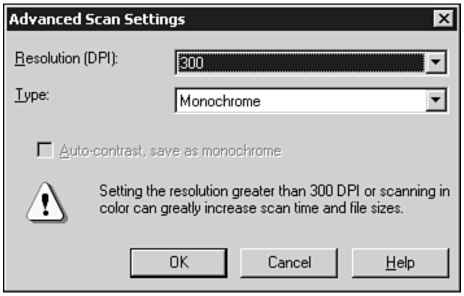
```
powerpnt /regserver
```

Click OK. If this does not work, run a maintenance install of Microsoft Office XP or Microsoft PowerPoint XP.

What to Do to Improve the Accuracy of Optical Character Recognition

The Microsoft Office Document Scanning applet is optimized to deliver the best possible optical character recognition (OCR) for most scanned documents, and offers relatively few options for tweaking. However, there are a few things you can do if you're not satisfied with the accuracy of the OCR files generated from your documents.

- **Use the Best Preset**—The software's default setting is Black and White, for documents with black text on white paper. If your documents are colored—for example, magazine articles or text printed on a color inkjet—make sure to switch to Black And White From Color Page.
- **Increase Resolution**—By default, Office scans at 300 dpi, which is usually the best tradeoff between file size, speed, and OCR accuracy. However, most scanners support higher resolutions. To increase the resolution, display the Microsoft Office Document Scanning dialog box. Click Preset Options, and choose Edit Selected Preset from the drop-down list. Display the General Tab, and click Advanced. In the Advanced Scan Settings dialog box (see Figure 41), choose a resolution higher than 300.


Advanced Scan Settings

Resolution (DPI): 300

Type: Monochrome

☐ Auto-contrast, save as monochrome

Setting the resolution greater than 300 DPI or scanning in color can greatly increase scan time and file sizes.

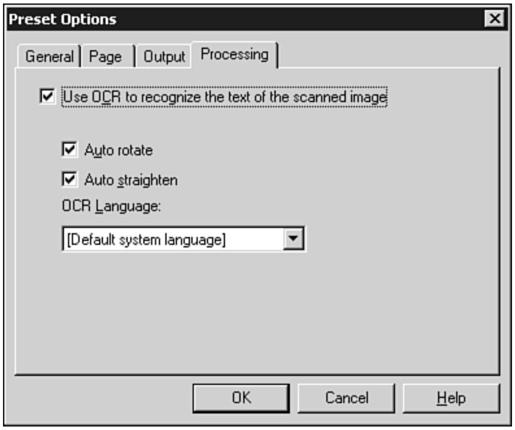
Figure 41
Changing scanning resolution in Microsoft Office Document Scanning.

Caution

Remember to switch back to 300 when you're finished scanning your current document; otherwise, future scans will be unnecessarily slow, and will create unnecessarily large files.

- **Make Sure Pages Are Being Optimized for OCR**—In the Preset Options dialog box, choose the Processing tab (see Figure 42). Make sure the Auto Rotate and Auto Straighten checkboxes are checked.

Figure 42
Checking Processing options in Microsoft Office Document Scanning.



☑ Use OCR to recognize the text of the scanned image

☑ Auto rotate

☑ Auto straighten

OCR Language:

[Default system language]

Note

If you're scanning text written in a language other than your system's default language, also choose the OCR Language from the drop-down box on this tab.

BONUS WORD CHAPTER

Managing Word More Effectively

In this chapter

by Bill Camarda

Special Edition Using

que®

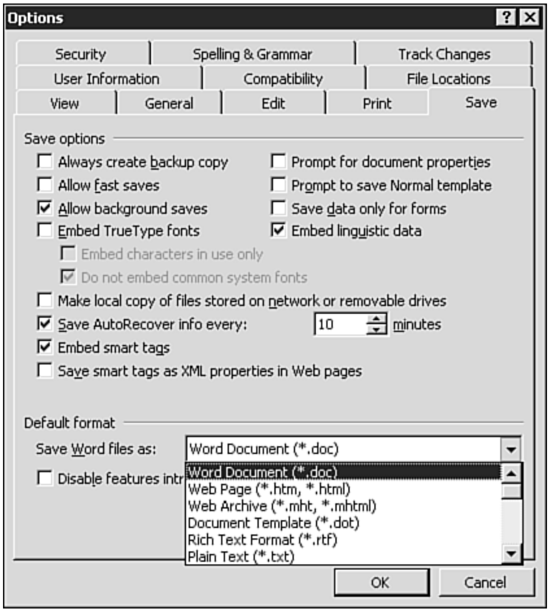
The Word 2002 File Format

In Word 2002, Microsoft has retained the same binary (DOC) file format used in Word 2000 and Word 97 for Windows, as well as Word 2001 and Word 98 for the Macintosh. However, files saved in Word 2002 support features not available in previous versions of Word, and these features are lost when you open and resave the file using older versions of Word, even those using the same format.

Setting a Default File Format

If not everyone in your organization has upgraded to Word 2002 yet, you can specify that your users save Word files in an older format until you have completed the migration. You can also specify HTML as your standard format, or a foreign-language Word format, or an older WordPerfect format such as WordPerfect 5.1 for DOS.

To specify a format other than Word 2002/Word 2000/Word 97 as your default save format, choose Tools, Options and select the Save tab (see Figure 1). Choose the format you want to use from the Save Word Files As drop-down box and click OK.



Save options

□ Always create backup copy
□ Allow fast saves
☑ Allow background saves
□ Embed TrueType fonts
□ Embed characters in use only
☑ Do not embed common system fonts
□ Prompt for document properties
□ Prompt to save Normal template
□ Save data only for forms
☑ Embed linguistic data
□ Make local copy of files stored on network or removable drives
☑ Save AutoRecover info every: 10 minutes
☑ Embed smart tags
□ Save smart tags as XML properties in Web pages

Default format

Save Word files as: Word Document (*.doc)

□ Disable features intr

Word Document (*.doc)
Web Page (*.htm, *.html)
Web Archive (*.mht, *.mhtml)
Document Template (*.dot)
Rich Text Format (*.rtf)
Plain Text (*.txt)

Figure 1
Choosing a different format as your default save format.

If you are gradually upgrading from Word 97 or Word 2000, you may want to prevent users from saving files containing Word 2002 features that cannot be used in these older versions of Word. To do so, display the Save tab, check the Disable Features Not Supported After check box, and choose the newest version of Word you want your saved files to reflect.

Note

The default format you set here only affects new files: when you resave an existing file, it is saved in its existing format.

WORKING WITH COLLEAGUES USING OTHER WORD PROCESSING PROGRAMS

Millions of users are still working with Word 6 and Word 95, and have no intention of upgrading anytime soon—not to mention all those still working with earlier versions of Word and other word processing programs. Word 6 and Word 95 share a file format different from the one used in Word 2002 and Word 97. Moreover, it's possible that Word 2002 users will lose features they care about when saving back to the Word 6/95 format or other word processing formats.

If your company contains a mix of Word 2002/2000/97 and Word 6/95 users—a common situation—you have two options:

- You can give Word 6/95 users a filter that enables them to open Word 2002/2000/97 files. The files can be read intact; however, if they edit and resave the files with the same name and location, the newer formatting is lost. This converter, MSWRD832.CNV, can be downloaded through Microsoft's Office Web site, at `http://officeupdate.microsoft.com`.
- You can specify Word 6/95 as your default file format, using the procedure described earlier, in the "Setting a Default File Format" section. Then, Word 6/95 users will not need a special filter to open the files, but Word 2002/97 users will lose the newer features when they save files to the older format.

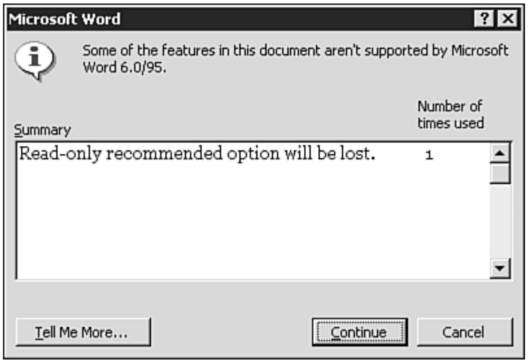
Caution

If you need to password-protect documents–or use protection for tracked changes, comments, or forms–these documents lose their protection when saved back to Word 6/95 format (even though Word 6 and 95 do have their own password protection features).

When you try to save a Word 2002 document in a different file format, a dialog box may appear (see Figure 2). The dialog box specifies which features will be lost or substituted if the file is saved in the different format.

If the user chooses Tell Me More, Word's Help system opens, displaying detailed information about what happens when you save a Word 2002 file to the chosen format.

If the user chooses Continue, Word saves the file to the old format. Word 2002 (and Word 97) features disappear or are converted immediately.



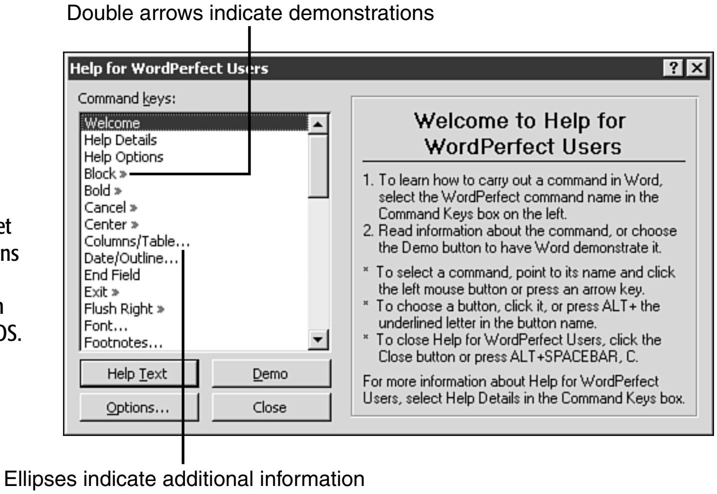
Some of the features in this document aren't supported by Microsoft Word 6.0/95.

Read-only recommended option will be lost. 1

Figure 2
When you try to save a Word 2002 document in a different format, you may see this warning message when you save a file for the first time.

CONVERTING FROM WORDPERFECT

If your company is converting from WordPerfect—especially WordPerfect 5.1 for DOS, the word processing software that dominated the market several years ago—Word comes with several features designed to make the transition easier. To access these features, choose Help, WordPerfect Help. The Help for WordPerfect Users dialog box opens (see Figure 3).



Welcome to Help for WordPerfect Users

1. To learn how to carry out a command in Word, select the WordPerfect command name in the Command Keys box on the left.
2. Read information about the command, or choose the Demo button to have Word demonstrate it.

* To select a command, point to its name and click the left mouse button or press an arrow key.
* To choose a button, click it, or press ALT+ the underlined letter in the button name.
* To close Help for WordPerfect Users, click the Close button or press ALT+SPACEBAR, C.

For more information about Help for WordPerfect Users, select Help Details in the Command Keys box.

Double arrows indicate demonstrations

Ellipses indicate additional information

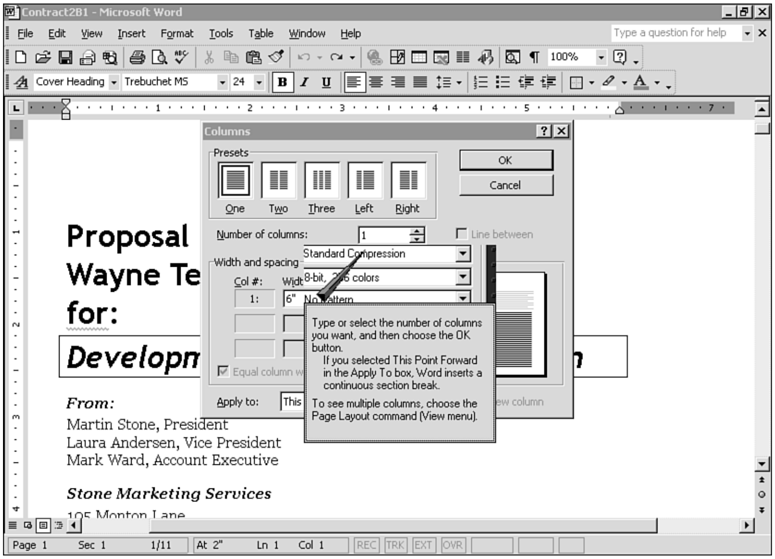
Figure 3
With the Help for WordPerfect Users dialog box open, people making the transition from WordPerfect can get detailed explanations of how Word features compare with WordPerfect for DOS.

The Command Keys scroll box includes a list of WordPerfect for DOS command keys. When you select a command key, Word displays a description of how to accomplish the same task in Word 2002.

Where Word offers additional help for WordPerfect command key submenus, the command key is followed by an ellipsis. You can view the additional information (for all commands except the first three—Welcome, Help Details, and Help Options) by clicking Help Text. After you've drilled down to the bottom level of information, you can click Help Text again, and Word displays the help information on top of your editing window so that you can view step-by-step instructions at the same time you perform the task.

If Word has a demo available, and you want to watch the task performed, click Demo (or Demo Now). Word displays your current document and performs the action at your current insertion point. In some cases (see Figure 4), Word opens a dialog box and shows you what to do next.

Figure 4
Word opens the appropriate dialog box and tells you exactly where to insert information.



Type or select the number of columns you want, and then choose the OK button.
If you selected This Point Forward in the Apply To box, Word inserts a continuous section break.
To see multiple columns, choose the Page Layout command (View menu).

Proposal
Wayne Te
for:
Developm

From:
Martin Stone, President
Laura Andersen, Vice President
Mark Ward, Account Executive

Stone Marketing Services
105 Monton Lane

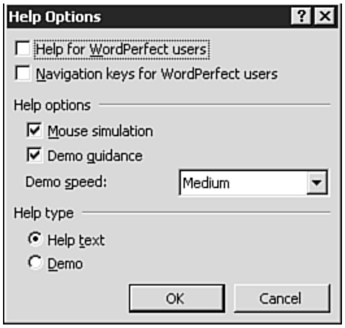
Tip from
Bill Camarda

If WordPerfect Help performs an action you didn't intend, click Undo.

SETTING WORDPERFECT HELP OPTIONS

Word gives you options about how to use WordPerfect Help, based on how far along you and your users have come in transitioning from WordPerfect and DOS. To use these options, choose Help, WordPerfect Help and choose Options. The Help Options dialog box appears (see Figure 6).

Word displays or demonstrates instructions whenever you press a WordPerfect for DOS key combination. If you check the Navigation Keys for WordPerfect Users check box, Word changes the way its Page Up, Page Down, Home, End, and Esc keys behave to match the way they work in WordPerfect for DOS.



Help Options

□ Help for WordPerfect users

□ Navigation keys for WordPerfect users

Help options

☑ Mouse simulation

☑ Demo guidance

Demo speed: Medium

Help type

● Help text

○ Demo

OK Cancel

Figure 5
You can control exactly how WordPerfect Help behaves.

Depending on the speed of your computer, you may find Word's feature demonstrations run too quickly—or too slowly. You can change their speed by choosing Fast, Medium, or Slow from the Demo Speed drop-down box.

Note

If you or your users miss WordPerfect's Reveal Codes feature, remember that even though Word doesn't have equivalent formatting codes, you can get detailed information about the formatting associated with any text. Use Word 2002's new Reveal Formatting feature.

Click on or select the text you're interested in; and choose Format, Reveal Formatting. Word displays detailed formatting information in the task pane at the right.

➔ For more information about Reveal Formatting, **see** Chapter 4, "Quick and Effective Formatting Techniques," **p. 107**.

Displaying Word 2002 Files on Computers Without Any Version of Word

Some of your computers may not have any version of Word installed at all. For example, individuals primarily responsible for data processing may not use Word. Investing in Word or Office software licenses for their workstations would be expensive and unnecessary. Moreover, these individuals may be using older workstations, such as 80486-based systems, which cannot run Word 2002.

As time passes, you may discover that these people can benefit from access to existing Word documents. For example, your corporate intranet may include manuals or sales guides written in Word. Giving customer service representatives access to these resources may help them solve customer problems more effectively. For situations such as this, you have two options:

- You can use Word 2002 to publish the data in HTML format that can be read by anyone using a recent Web browser. (For more information on using HTML in Word 2002, see Chapters 18 and 19.)
- You can provide the Microsoft Word Viewer, a freeware application that can be copied and distributed. This small program, available at `http://officeupdate.microsoft.com`, enables any Windows user to view and print any Word 2002 document. It supports many, though not all, Word 2002 features. For example, it supports Print Layout view, Outline view, Web Layout view, Document Map, zooming, headers, footers, footnotes, comments, and hyperlinks—but not toolbars.

You can't edit text in Word Viewer. However, you can copy the text into other applications through the Windows Clipboard.

You can think of Word Viewer as Microsoft's answer to Adobe Acrobat Reader. It has one major advantage compared with Acrobat Reader: Anyone who owns Word can author documents that can be read with Word Viewer.

It also has a major disadvantage, which limits its value as an Internet solution for distributing Word documents: It works only in Windows environments (not Macintosh or Unix).

Tip from
Bill Camarda

You can use Word Viewer as a helper application for viewing Word documents downloaded from the Internet. Word Viewer can also make it a little easier to work with customer and vendor organizations that have standardized on a different word processing platform.

Two versions of Word Viewer are available:

- **Viewer for Word 2002/2000/97 on PCs Using Windows 95/98/Me or Windows NT/2000**—Displays Word 2002, Word 2000, and Word 97 files on any system running a 32-bit version of Windows.
- **Viewer for Word 2002/2000/97 on PCs Using Windows 3.x**—Displays Word 2002, Word 2000, and Word 97 files on Windows 3.x systems.

Each of these viewers runs on virtually any computer that meets the minimum hardware requirements of the version of Windows it is already running.

Caution

Although Word Viewer can coexist with Word on the same computer, it works best on computers where Word isn't installed. Even though the Word Viewer Setup program is designed to ask whether Word or Word Viewer should be the default for opening Word files, you may sometimes find that the wrong application loads if they are both installed.

BATCH FILE CONVERSIONS

If you are upgrading your entire organization to Word 2002 at the same time, you may find it convenient to convert your collection of Word 6/95 files to Word 2002 format all at the same time. However, that's not the only time you may want to convert many files at once. For example

- You might be migrating from WordPerfect and want to convert your WordPerfect files for use in Word 2002.
- You might be migrating from Macintosh to Windows and want to convert older Macintosh Word files for use in Word 2002.
- You might be upgrading a set of documents created in an earlier version of Word.

Word 2002 provides a batch conversion utility, the Conversion Wizard, for purposes such as these. With the Conversion Wizard, you can create Word 2002 documents from files in the following formats:

- Lotus 1-2-3
- Microsoft Excel
- Microsoft Works 4.0 or 5.0 for Windows
- MS-DOS Text with Layout
- Outlook Address Book
- Personal Address Book
- Rich Text Format (RTF)
- Schedule+ Contacts
- Text
- Text with Layout
- Windows Write
- HTML
- Encoded Text
- Word 6.0 & 95 for Windows & Macintosh
- Word 4.0–5.1 for Macintosh

- WordPerfect 5.x
- WordPerfect 6.x

In addition, Word can attempt to extract the valid text from any file, using the Recover Text from Any File option.

Similarly, you can convert documents from Word 2002 to the following formats:

- Microsoft Works 4.0
- MS-DOS Text, MS-DOS Text with Line Breaks, or MS-DOS Text with Layout
- Rich Text Format (RTF)
- Text Only, Text with Layout, or Text with Line Breaks
- Word 2.x for Windows
- Word Template
- HTML
- Encoded Text
- Word 4.0 for Macintosh
- Word 5.0 for Macintosh
- Word 5.1 for Macintosh
- Word 6.0/95 for Windows & Macintosh
- WordPerfect 5.0 (including Secondary Files)
- WordPerfect 5.1 for DOS
- WordPerfect 5.1 or 5.2 Secondary Files
- WordPerfect 5.x for Windows

The Conversion Wizard can convert all files in a specific folder. Before you use it, either place all the files you want to convert in the same folder or, if you want to convert files in their current folders, list the folders you want to convert and run the Conversion Wizard separately in each folder.

Caution

Test your file conversions on a few sample documents before performing them *en masse* and putting the resulting documents into production.

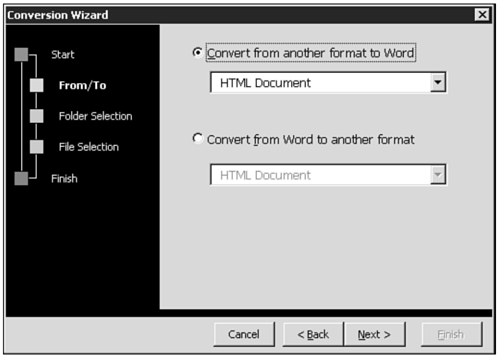
In rare instances, if you find the converted documents unusable, you might consider trying a third-party filter, such as those included in Conversions Plus for Windows. (To obtain a copy of Conversions Plus for Windows, contact DataViz, 1-800-733-0036, `www.dataviz.com`.)

o run the Conversion Wizard, choose File, New; click General Templates on the New Document side pane; select Batch Conversion Wizard from the Other Documents tab; and click OK.

After you've displayed the Conversion Wizard, follow these steps:

1. Click Next. The From/To screen appears (see Figure 6).

Figure 6
On the From/To screen of the Conversion Wizard, you specify to (or from) which format you want to convert your file.

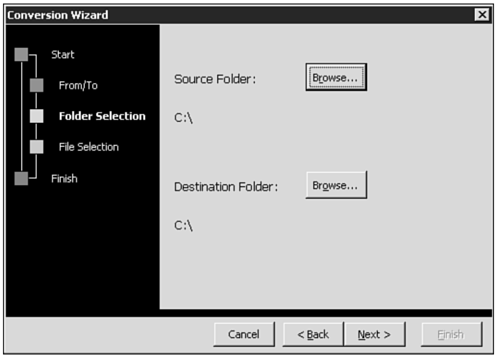


● Convert from another format to Word
HTML Document

○ Convert from Word to another format
HTML Document

2. If you want to convert files stored in another format into Word 2002 files, click Convert from Another Format to Word; then choose the format from the highlighted drop-down box. If you want to convert Word 2002 files into another format, click Convert from Word to Another Format. A drop-down box becomes active beneath this button. From this drop-down box, select the format you want.
3. Click Next to move to the next window.
4. From the Folder Selection window (see Figure 7), click Browse (next to Source Folder) to specify the folder where your existing files are.

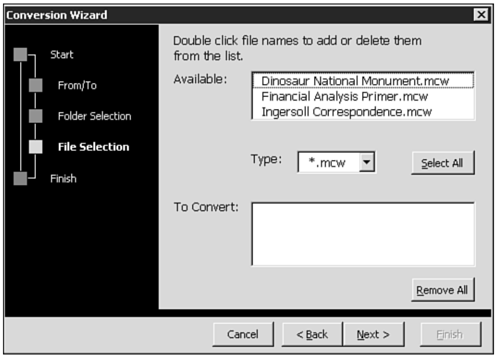
Figure 7
From the Folder Selection dialog box, you can choose both a source and a destination folder.



Source Folder:
C:\

Destination Folder:
C:\

5. The Browse for Folder dialog box appears. Select a folder and click OK.
6. Click the Browse button located next to Destination Folder.
7. In the Browse to Folder dialog box, select a destination folder and click OK.
8. Click Next to display the File Selection window (see Figure 8). From here, you can select the specific files you want to convert.



Double click file names to add or delete them from the list.

Available: Dinosaur National Monument.mcw, Financial Analysis Primer.mcw, Ingersoll Correspondence.mcw

Type: *.mcw

To Convert:

Figure 8
In the File Selection window, you can choose to convert all or only some files in a folder.

Note

If you choose the same folder for input and output, and you've chosen a conversion where the file extension does not change–such as converting Word 6 to Word 2002 files–the old files are overwritten in the new format. Be careful to make sure that this is what you want to happen before you run the conversion.

9. To convert all the files in the folder you selected, click Select All. All the files now appear in the To Convert box.

Tip from
Bill Camarda

If no files appear in the Available box, click the arrow next to the Type drop-down box to choose a different file extension.

10. To prevent a file from being converted, double-click it in the To Convert box.
11. When you finish selecting files, choose Next.
12. Click Finish to perform the file conversions. A progress bar appears onscreen showing how far along in the conversion process the Conversion Wizard is. When the process finishes, the Conversion Wizard offers you an opportunity to run another conversion. If you click Yes, the wizard runs again.

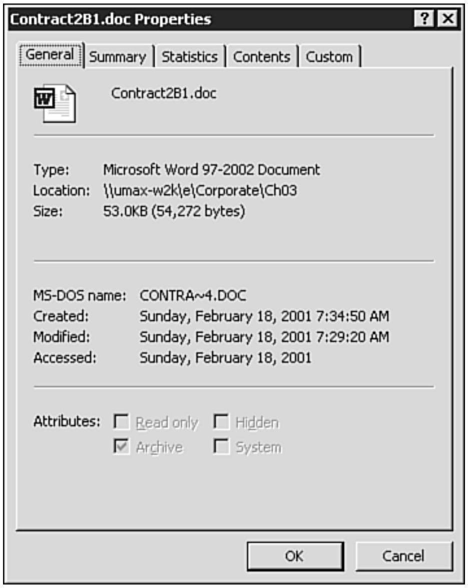
Note

Don't assume that the entire process will run unattended: Check in every few minutes. The conversion process may occasionally stop to report an error message or, in some cases, to attempt to load a Web page included in a file being converted.

Using Document Properties to Simplify Document Management

In Chapter 3, "Basic Document Creation, Storage, and Retrieval," you learned about Word's powerful capabilities for finding files, which are accessible by choosing Tools, Find from within the Open dialog box. You may recall that one of the ways Word can search for files is by document property. This section takes a closer look at document properties and shows you how to use them to your advantage—whether you're responsible for one desktop or many.

To work with document properties, choose File, Properties. The Properties dialog box opens (see Figure 9). It contains five tabs:



Contract2B1.doc

Type: Microsoft Word 97-2002 Document
Location: \\umax-w2k\e\Corporate\Ch03
Size: 53.0KB (54,272 bytes)

MS-DOS name: CONTRA~4.DOC
Created: Sunday, February 18, 2001 7:34:50 AM
Modified: Sunday, February 18, 2001 7:29:20 AM
Accessed: Sunday, February 18, 2001

Attributes: ☐ Read only ☐ Hidden
☑ Archive ☐ System

Figure 9
You can review a document's properties (and edit many of them) by displaying the Properties dialog box.

- **General**—This tab includes information Word automatically stores about every document, including when it was created, how large it is, and where it is stored.
- **Summary**—This tab includes the document's title, comments, author, keywords, and other important document information.

- **Statistics**—This tab includes information Word compiles about the size and contents of a document, as well as how long it has been open and how many times it has been saved.
- **Contents**—This tab includes major parts of a document file.
- **Custom**—Options available on this tab enable you to create your own document properties or choose from 27 optional document properties Word can provide.

General Information Stored in the Properties Dialog Box

Display the General tab of the Properties dialog box when you want to know basic information about your Word document, including its current size and its MS-DOS “short” (8.3) filename, as well powerful as when the file was created, last modified, and last accessed.

Most of the items in the General tab are self-evident, but a few are worth calling to your attention:

- **Type**—Ordinarily, this is a Microsoft Word 97-2002 (XP) document.
- **Location**—This displays the complete file path, whether on a local or networked computer.
- **Modified**—This tells you when a file was last saved so that you can tell whether specific edits are likely to be reflected.
- **MS-DOS Name**—This is the old-fashioned 8.3 short filename stored with every file in Windows 98 and Windows NT—the name that stays with your file even if you send it to a Windows 3.x system or an older Macintosh or Linux/Unix system that doesn’t support long filenames.

No general information is stored with a file until you save it for the first time.

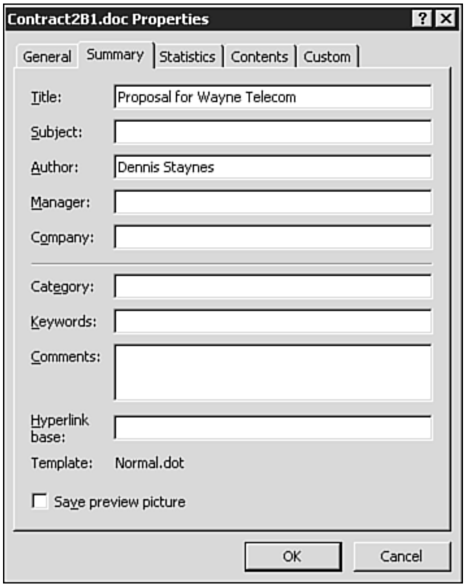
Working with Summary Information

Chances are, the Properties tab you’ll use most is the Summary tab (see Figure 10). Here, you insert editable information about your file—including the categories by which you’re most likely to search for it in Word’s Tools, Find dialog box, such as Author, Comments, or Keywords.

If you’ve set up Word properly, much of this information can be entered for you automatically. Word enters a title based on the first line of text in your document—commonly a document’s title. Word enters the Author based on the name stored in the User Information tab of the Tools, Options dialog box. If you ever run AutoSummarize, Word automatically copies the list of keywords it generates into the Keywords box. Finally, you can include Manager, Company, Category, and other information along with templates so that they are automatically included in every document built with those templates.

One more item on this tab is worth pointing out: Save Preview Picture. If you check this check box, the next time you look at the Properties dialog box, Word copies its current headings into the Contents tab of the Properties dialog box. If you check this box, Word displays a thumbnail of the first page in the Preview pane of the File, Open dialog box, reflecting formatting. If you do not check the box, you can scroll through the text of the entire document in the Open dialog box's Preview pane.

Figure 10
For most people, the Summary tab is the workhorse of the Properties dialog box.



Contract2B1.doc Properties

Title: Proposal for Wayne Telecom

Author: Dennis Staynes

Template: Normal.dot

□ Save preview picture

Tip from
Bill Camarda

If nothing else, you might add your company name to the Summary tab of the Properties dialog box in Normal.dot. This provides an added measure of security by showing who owns the computer on which your documents were created. It's easy to remove this information–but many Word users don't even realize it's there.

Note

The name specified in the Author box doesn't change if you forward the file to a colleague for editing on his or her computer. However, the Last Saved By name in the Statistics tab does change when your colleague saves the file.

If you are responsible for the documents created by an entire workgroup or organization, consider requiring (or at least actively encouraging) users to include Summary information along with their document. One way to do this is to have Word display the Summary tab of the Properties dialog box whenever a user saves a file for the first time. To do this, choose Tools, Options and select the Save tab. Check the Prompt for Document Properties check box and click OK.

Some Word users do not want to store this information with their documents, for privacy and security reasons. To remove it, along with other document information that can be traced to a specific user or computer, follow these steps:

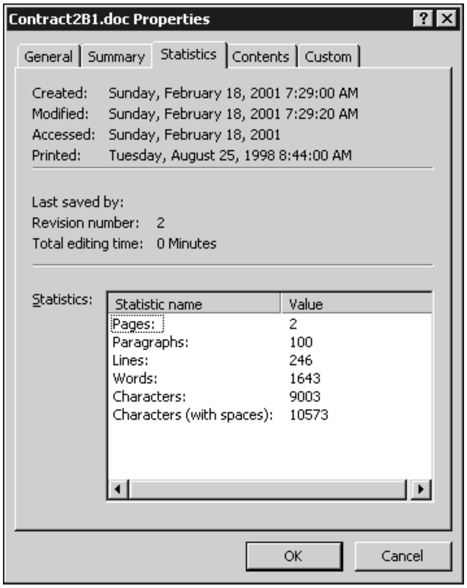
Choose Tools, Options, Security; then check the Remove Personal Information from This File on Save check box and click OK.

➔ For more information on this and other Word 2002 file security and privacy options, **see** Chapter 29, "Word Document Privacy and Security Options," **p. 891**.

Understanding and Using Document Statistics

The Statistics tab (see Figure 11) compiles several useful statistics about your document. You can see when your document was created, modified, and accessed—the same information you already saw in the General tab. Here, however, you can also see when your document was last printed—and who saved it last, even if it was saved by someone other than the author.

Figure 11
The Statistics tab gives you a quick look at the most important statistics associated with your document.



Created: Sunday, February 18, 2001 7:29:00 AM
Modified: Sunday, February 18, 2001 7:29:20 AM
Accessed: Sunday, February 18, 2001
Printed: Tuesday, August 25, 1998 8:44:00 AM

Last saved by:
Revision number: 2
Total editing time: 0 Minutes

Statistics:

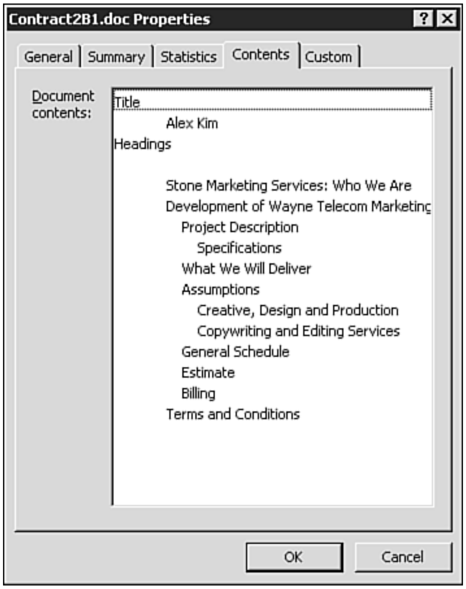
Statistic name	Value
Pages:	2
Paragraphs:	100
Lines:	246
Words:	1643
Characters:	9003
Characters (with spaces):	10573

Revision Number tells you how many times you've saved the file. Because you may save a file every few minutes, it's easy to generate hundreds of revisions. If you want a true draft number, consider writing a Visual Basic macro that requests the current draft number and stores it in another Properties box, such as Comments. You can name the macro AutoClose so that it runs whenever you save a file associated with the template where the macro is stored.

The Statistics tab also reports Total Editing Time, which is actually the amount of time the document has been open. Of course, several other documents, or for that matter several different applications, may be open at the same time. Word doesn't care; it assumes that you're editing a document whenever it's open.

Using Contents Information

As you saw earlier, if you check the Save Preview Picture check box in the Summary tab, the next time you look at the Properties dialog box, Word stores its headings in the Contents tab. You can see an example in Figure 12.



Title
    Alex Kim
Headings

    Stone Marketing Services: Who We Are
    Development of Wayne Telecom Marketinc
        Project Description
            Specifications
        What We Will Deliver
        Assumptions
            Creative, Design and Production
            Copywriting and Editing Services
        General Schedule
        Estimate
        Billing
    Terms and Conditions

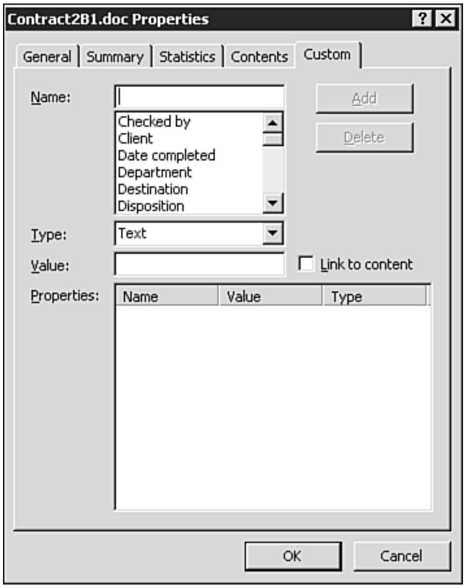
Figure 12
The Contents tab shows the document's title and can also display all the headings in your document.

Note

Unfortunately, several document elements you might reasonably expect to find here are not included. For example, Contents does not report on multiple document versions stored in the same file, macro modules stored with a file, or embedded graphics.

Creating Custom Properties

If the properties you've seen so far aren't enough, Word provides 27 more custom properties you can assign at will—or you can create your own. To work with custom properties, choose File, Properties and display the Custom tab (see Figure 13). Follow these steps to add a new category:



Name:

Checked by
Client
Date completed
Department
Destination
Disposition

Type: Text

Value:

Properties:

Name	Value	Type

Figure 13
If you need a property that Word doesn't provide, you can add it in the Custom tab of the Properties dialog box.

1. Choose a name for your custom property in the Name scroll box, or type a name of your own.
2. In the Type box, specify the kind of information you want your property to contain: text, a date, a number, or a Yes/No choice.
3. In the Value text box, enter the value with which you want your custom property to start. For example, if you've created a Date Completed property, you might enter the date 2/1/2001. The Value you insert must be in a format compatible with the Type you've just chosen.
4. When you're finished, click Add. The custom property, its type, and its value now appear in the Properties scroll box.

After the property has been created, you can modify it by selecting it in the Properties scroll box, entering a new Value or Type, and choosing Modify. Or you can delete it by selecting it and choosing Delete.

Note

Because custom properties require a Value, you can't create an empty custom property. If you want to include a custom property in your document or template, insert a "dummy" value that users can later replace with a real one.

CREATING CUSTOM PROPERTIES THAT UPDATE THEMSELVES

You can create a custom property that knows how to update itself, based on changing information in your document.

Imagine, for example, that every division in your company sends you a monthly executive summary reporting new sales activity. Your reporting template includes a table; that table contains a cell listing the total value of all new customer sales made that month. You might want to search all those documents, quickly identifying the divisions where sales exceeded $10,000,000. However, you have dozens of divisions and you would rather not open each file individually to see the value stored in the table: You only want to know who has exceeded a certain threshold so that you can learn why.

To accomplish this, you first create a bookmark associated with the text in your document that you want to attach to your property:

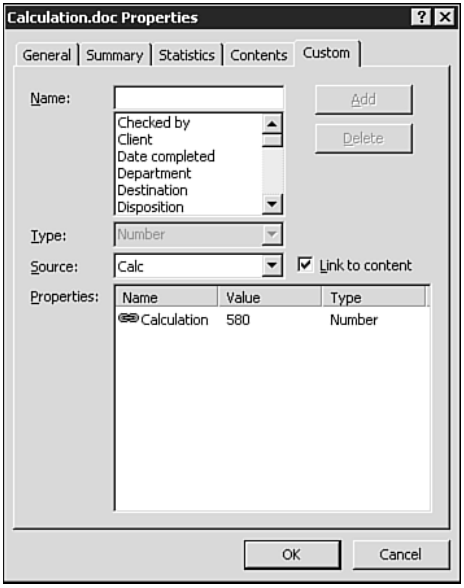
1. In your template, select the formula field that contains the calculation of total profits.
2. Choose Insert, Bookmark.
3. Type a bookmark name and click Add.

Next, create a custom property based on that bookmark:

1. Choose File, Properties.
2. Choose the Custom tab.
3. Check the Link to Content check box. (This box is grayed out unless your document contains at least one bookmark.) Notice that the Value box has turned into a drop-down box named Source.
4. In the Name text box, enter or select a name for your new custom property.
5. In the Source drop-down box, choose the bookmark to which you want to link your custom property.
6. Click Add. The new custom property appears in the Properties list. A Link icon appears next to it, indicating that this custom property is linked to a bookmark (see Figure 14).

USING FIELDS TO DISPLAY PROPERTIES IN YOUR DOCUMENT

Many of the elements stored in the Properties dialog box can be placed in your document automatically, using fields. For example, you might want to create a cover sheet that prints with each of your documents. Table 1 lists fields that use information stored in the Properties dialog box. Of course, any time you update your fields, changes in the corresponding Properties boxes are reflected in them.



Name:

Checked by
Client
Date completed
Department
Destination
Disposition

Type: Number

Source: Calc

☑ Link to content

Properties:

Name	Value	Type
Calculation	580	Number

Figure 14
You can create a custom property linked to bookmarked text in your document.

TABLE 1 FIELDS THAT USE INFORMATION STORED IN THE PROPERTIES DIALOG BOX

Field	What It Displays	Notes
Author	Author data from Summary tab	
Comments	Comments data from Summary tab	
CreateDate	File creation date from General tab	
DocProperty Select	Information from any property you choose	Can also work with custom properties after you create them. properties the property in Insert, Field, Options.
EditTime	Editing time from Statistics tab	
FileName	Filename	\p switch adds full pathname.
FileSize	File size	\k switch specifies kilobytes. m switch specifies megabytes.
Info	Information from any summary of your choice in Insert, Field, Options.	Select the property
Keywords	Keywords data from Summary tab	
LastSavedBy	Last saved by data from Statistics tab	
NumChars	Number of characters in document	
NumPages	Number of pages in document	
NumWords	Number of words in document	

Table 1 Continued

Field	What It Displays	Notes
PrintDate	Print date from Statistics tab	
SaveDate	Last saved date from General tab	
Template	Template currently attached to document	\p switch adds full pathname of template.
Title	Title data from Summary tab	

➔ For more information about working with fields, **see** Chapter 26, "Automating Your Documents with Field Codes," **p. 793**.

Troubleshooting

Solving Line and Page Break Problems Associated with File Conversions

Sometimes, line and page break problems are introduced after files are converted to Word 2002 format from other word processing programs and other platforms, such as the Macintosh. These may be associated with font substitutions Word makes when it cannot find the fonts originally used to create the document.

The best solution, if possible, is to install fonts that match those used on the system that created the original document. If this is not possible, you may sometimes be able to improve your results by changing the fonts Word substitutes. Doing so is covered in detail in "Using Font Substitution," in Chapter 31, "Customizing Word."

Understanding Problems Associated with WordPerfect Conversions

Like most complex file converters, Word's WordPerfect import and export converters are not perfect.

Some inherent differences in the way that Word and WordPerfect structure documents make it difficult for Microsoft (or anyone else) to implement a perfect converter. For example, WordPerfect places style definitions in a prefix for every document file, even if users don't specify styles. Word's conversion filter picks up these style definitions, sometimes using them in troublesome ways.

For example, WordPerfect style definitions that have the same name as Word's existing built-in style definitions override the Word definitions—so different documents with the same styles will look different, depending on where they were created.

The following resources may be helpful in understanding and troubleshooting WordPerfect conversions to Word:

- *Microsoft Office 2000 Deployment and Administration* (Que, ISBN: 0-7897-1931-2), Chapter 18, "Migrating From or Coexisting with Legacy Applications"
- Microsystems White Paper, "Making the Change from WordPerfect to Word in a Legal Environment," downloadable at `www.microsystems.com/expert.htm`. (This page contains several excellent documents troubleshooting unusual or buggy Word document behavior.)
- Microsoft's Word Conversion Resource Center, currently located at `support.microsoft.com/support/Office`.