

Chapter 2

Understanding What Macros Are, Where They Are, and How They Work

Links: The [Chapter Name](#), above, moves to the next chapter. All page header links go to the contents menu. Within the chapter, [Topic Titles](#) returns here. [Other Red Links](#) are to other locations in this paper. [Blue Links](#) are to web sources.

Top	Contents	Glossary	Index	Release Notes
Chapter 1 Additional Resources	Chapter 2 Understanding Macros	Chapter 3 First Things	Chapter 4 Controlling Macro Flow	Chapter 5 Using the Dialog Editor
Chapter 6 Math Routines	Chapter 7 Date Routines	Chapter 8 DialogShow/Callbacks	Chapter 9 Macro Examples	Chapter 10 Glossary
Main Topics In This Chapter				
General Definition	Storage Location of Macros	Usual Macro File Location	Shipping Macros	
How Macros Work		Playing Macros		

- GENERAL DEFINITION.** In simplest terms, a WordPerfect macro is WordPerfect document written in WordPerfect's macro language which, when compiled, i.e., translated into an object file (a hidden area of the macro that you never see), it does something. The "something" varies from performing very simple WordPerfect tasks which you could do in WordPerfect to enormously complex tasks which you could probably not do in WordPerfect in any other way. An example of a simple task might be to automate a signature block for a letter or a court document. An example of an enormously complex task is the mathematical computation of Oklahoma's Child Support Guidelines law based upon numerical data you enter into a dialog. If you are an Oklahoma lawyer, you know that performing such a task involves a blend of each parent's income, number of minor children, day care expense (and who's paying it), health insurance expense (and who's paying it), what statutory year is involved with the computation, whether or not "shared parenting" is involved, among other things. If you could summarize what a macro "is" with one word, the word would be "automation."

To "automate" tasks you would do in WordPerfect, you can write one or more WordPerfect macros. Such automation is particularly useful to perform repetitive tasks which you frequently do in WordPerfect. Because the list of things you can in WordPerfect is endless and is only limited by the perception of a particular user, the list of things you can do with macros is endless, as well. But, as a beginning point, just know this: Anything you can do in WordPerfect manually can be automated by one or more WordPerfect macros. And, as noted with the Child Support Computation example just mentioned, macros also give you the possibility of doing some things you could not do in WordPerfect without the use of macros – or, at least, if you could, it would be very very hard to accomplish without the use of macros.

- STORAGE LOCATION OF YOUR MACROS.** Since WordPerfect macro files are WordPerfect "files", they are located somewhere, usually on your hard disk. As a preliminary matter, it's good to give a little thought to disk organization – where do you plan to keep your macros? If you plan to keep them all in the same place, it's probably best to keep them all in your WordPerfect **default macro directory**, shown below. But, you might want to create one or more particular directories for particular kinds of things. For example, in my *Grande Macros* program for Oklahoma lawyers, which consists not only of numerous interconnected macros but interconnected regular WordPerfect documents also, it made sense to create a particular dedicated directory, C:\Grande5, in which to store all interrelated macro and document files.

The main thing is to have and know the location of where you keep your macros, even if it is simply an acceptance of the organization that WordPerfect has made for you in the first place. To edit a macro, you need to know where it's stored and you need to know its name – else you wouldn't know what file to open. If you make interrelated macros (macros which link to each other when they run), in my opinion it is best to use the full pathname of the target macro in the macro that runs it, particularly in post-WordPerfect 10 versions. If you use the full pathname of the target macro, you will never be harmed, but, you may be if you don't. If all your macros are in WordPerfect's default macro directory or default supplemental macro directory, then you can use the **?PathMacros** or **?PathMacrosSupplemental**

system variable as the pathname part of the file you want to open. If you store particular macros elsewhere, you can create a global variable which identifies that path, e.g., Global vPath="C:\Grande5\". Then, to identify a macro to run from a macro that is already running, the statement might be Run(vPath+"thatmac.wcm"). Note that if you use a specific path other than ?PathMacros or ?PathMacrosSupplemental in identifying part of the pathname, the literal pathname is not fluid and the macros and/or related documents must actually be in the specified directory – otherwise, target macros will fail. J. Dan Broadhead offers the following suggestions:

I have a suggestion that we usually made to macro users concerning the "Default" and the "Supplemental" macro locations. I think WP used to be set up this way, but somehow got changed along the way.

As you mentioned, I believe that, initially, WP is now configured so that the "Supplemental" macro directory is blank, and the "Default" macro directory points to the installation directory of the Shipping Macros.

I would suggest a change here. Move the "Default" installation directory location to the "Supplemental" location field. Then create your own personal macro directory someplace, like in "My Documents\My WP Macros" (or something), and put that location in the "Default" field.

Here's why. It's a good idea to keep "your" macros separate from the WP macros. If they are stored together, then when you go to edit your macros, you might accidentally (or on purpose) edit and change one of the standard shipping/included macros that came with WP, and you might "break" it.

If your macros are kept separate, then when you open a file open or save dialog for your macros, then your own personal macro folder is displayed containing only your own macro files, and not the standard WP macros.

Every time you play a macro, WP will first look in the "Default" location, and then the "Supplemental" location till it finds the macro, so it can still find the standard macros.

All your macros will now normally be stored in your new personal macro folder, and if this is in the Windows "My Documents" folder someplace, then you can be sure that if you backup the "My Documents" folder on your computer, that you will get copies of all your macros as well. The idea behind the "My Documents" folder, is that all the things on your computer that you change, are all stored in this one place and you don't need to hunt all over your system for the files that you have changed.

If you want to change a shipping macro, then copy it from the standard installation directory (now pointed to by the "Supplemental" field) to your personal macro folder (the "Default" field), and make changes to the copy. Now when you run the shipping macro, it will first find and run your personally modified version instead, which is also in your personal macro folder someplace in "My Documents", and it will also get backed up when you save your "My Documents" folder.

If you mess up the shipping macro, it is only a copy, and you can delete the copy to get back to the original shipping version.

And for all the other file locations in WP that support both "Default" and "Supplemental" fields, I would do the same thing, pointing the "Supplemental" fields to the standard installation locations normally found in the "Default" fields, and pointing the "Default" fields to someplace in the "My Documents" folder. And do the same for PR and QP as well.

- **USUAL MACRO FILE LOCATION.** Unless you have changed the WordPerfect default macro directory, the default macro directory was established when you installed any Corel WordPerfect version. Those directories are:

WordPerfect 6.1	C:\Office\WpWin\Macros
WordPerfect 7.0	C:\Corel\Office7\Macros\WpWin
WordPerfect 8.0	C:\Corel\Suite8\Macros\WpWin

WordPerfect 9.0	WordPerfect 9 varies. If you had an early release, through Service Pack 2, it will be in your C:\Program Files\ directory shown below, but if you have the later versions, Service Pack 3 or 4, it will be in the C:\Windows\ directory shown below: Early versions (lower than release 9.0.0.738): C:\Program Files\Core\WordPerfect Office 2000\Macros\WpWin Later versions (release 9.0.0.738 and higher): C:\Windows\Application Data\Core\PerfectScript\9\WordPerfect
WordPerfect 10.0	C:\Documents and Settings\Owner\Application Data\Core\PerfectScript\10\WordPerfect C:\Windows\Application Data\Core\PerfectScript\10\WordPerfect (supplemental)
WordPerfect 11.0	C:\Documents and Settings\Owner\Application Data\Core\PerfectScript\11\WordPerfect C:\Program Files\WordPerfect Office 11\Macros\wpwin (supplemental)
WordPerfect 12.0	C:\Documents and Settings\Owner\Application Data\Core\PerfectScript\12\WordPerfect C:\Program Files\WordPerfect Office 12\Macros\wpwin (supplemental)

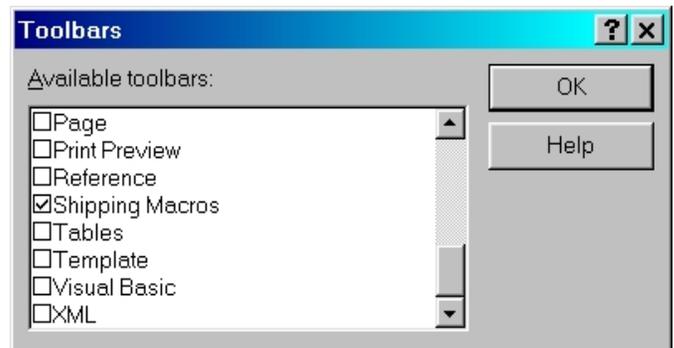
To know what your "release" version of WordPerfect is, click Help|About WordPerfect... and the specific release number will be stated.

To know (or change) your default macro directories in Wp8.0 and higher, select Tools | Settings... . In WordPerfect 6.1 or 7.0, select Edit | Preferences... . The Settings Dialog opens. Click on the Files icon which opens the File Settings dialog. Click the Merge/Macro tab. You will see two default macro directories, the Default and the Supplemental Macro directories.

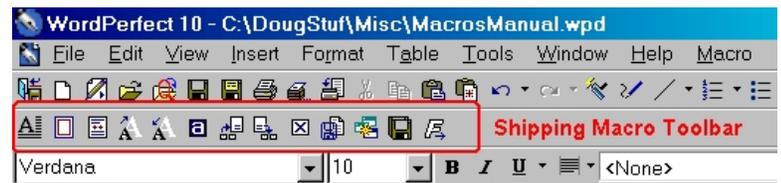
Subject to what was said in "J. Dan Broadhead's Tip", above, it is OK to store all your macros in the default macro directory/directories (although that will "clutter up" the original directory's contents with your own macros) or in some other directory you specify (which would probably be better). For example, the macros in my Grande Macros 6.0 program are all stored in C:\GrandeMacros.

- **SHIPPING MACROS.** Some macros, called "Shipping Macros", are installed during your installation of Corel WordPerfect (any version), unless you chose not to install them during the installation process. They would be installed in the "default" macro directory for the version of WordPerfect you are running.

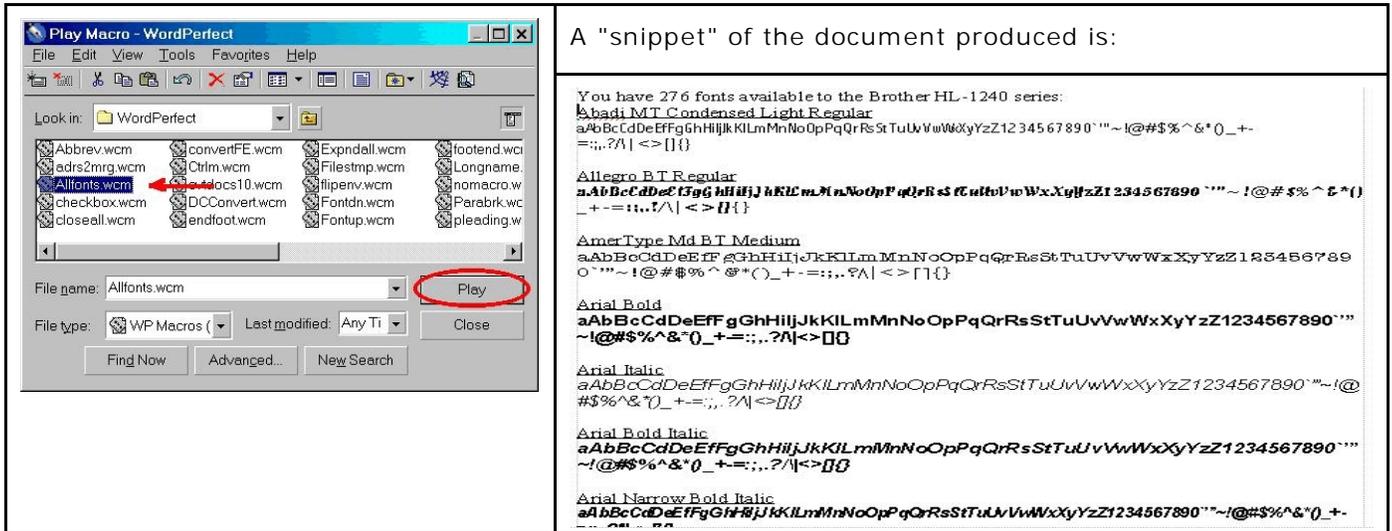
WordPerfect 9.0 and higher even have a special "Icon Toolbar" that you can use to play the Shipping Macros. To turn on this special toolbar, click View|Toolbars... and select Shipping Macros from the list, as shown here. The WordPerfect 10 toolbar looks like this:



Explanation/discussion about Shipping Macros is beyond the scope of this manual, even though I'll discuss a couple on the next page. To see what they can do, open the Shipping Macros Toolbar in WordPerfect 9.0 or higher, as shown above. Hover your mouse over an icon in the toolbar for a brief description of what the macro represented by the icon does, and, if you're interested, click it!



A shipping macro common to all WordPerfect versions is the macro file, "Allfonts.wcm." It creates a WordPerfect document which shows you ALL fonts installed on your computer, for your information. To run this macro, press Alt+F10 or click Tools, Macro..., Play, and from the Play Macro dialog select the Allfonts.wcm macro, shown below in much reduced size, just readable enough to make the point:



The "real" and "full" document produced by this macro includes each installed font, together with a three-line presentation of each font. Why run this macro? To see what you have – chances are good you have installed fonts that you weren't even aware of. You can print the document for later reference.

Other Shipping Macros usually include: endfoot.wcm (converts endnotes to footnotes), footend.wcm (converts footnotes to endnotes), filestmp.wcm (inserts the filename in a footer on that page); reverse.wcm (which created this white-on-black text box; and several others.

- **HOW MACROS WORK.** When a macro "runs", a WordPerfect macro generally works the way WordPerfect does, top to bottom, responding to various codes as they are encountered (e.g., formatting) and in a lineal fashion. However, many twists and turns can occur depending on how the macro document is constructed. But, unless a macro is "told" otherwise (by including specific macro commands which do the "telling"), a macro starts at its top and progresses "downward" through the macro code until the macro is done. As each macro command is encountered, it is executed.

But, you will want to include commands which vary this "lineal" progression to fit your needs. You are the traffic cop – you tell the macro what to do, where to go, etc., by including various "traffic signals", so to speak, in the macro itself. See Chapter 4.

Think of a macro (for a moment) as though it were a "city." A city has predefined locations (street addresses). In macro vernacular, each such "street address" (location) is a "Label." A label is a "place" in the macro, that place being defined by you when you use the "label" statement. In conjunction with "Go" and/or "Call" commands, you can tell a macro to "go" to a particular label and execute commands at that location, shifting the "lineal" flow to a different location. Or you can "call" the commands contained at a label, instructing your macro to execute the commands within that called label *before proceeding* in the lineal flow. This is developed more fully in Chapter 4, Controlling Macro Flow, below, but, for now, understand that you can tell a macro to interrupt its "lineal" movement either temporarily or permanently. As I said, "you are the traffic cop" and you have the unfettered ability to design the macro any which way you want! So, while "structure" is indeed of great importance in writing macros, it might be helpful to know that *you are the authority* which determines what that structure is – WordPerfect's macro language does not place constraints upon you in that process, you are the person who imposes whatever constraints you might choose upon your particular macro creation. As the "creator", you call all the shots, even though you must abide the macro language's "rules" in doing so.

The bottom line: YOU determine "how a macro works." There is no "preordained" way. While you must conform to the very precise macro language's writing requirements, you can design a macro to do whatever your needs, interests and imagination, can conceive.

- **PLAYING MACROS.** Different methods exist. The main methods are described here.

! Using the Menu. Press Alt+F10 or click Tools| Macros | Play.... Either method opens the "Play Macro" dialog. Unless you've been using a different macro directory during a WordPerfect session, the **default Macro Directory's** contents will be shown. Find the macro you want, select it, and click the "Play" button.

! Using the Icon Toolbar. If you include a macro on a WordPerfect Icon Toolbar, all you need to do is click the icon to run that macro. You can edit any WordPerfect Icon Toolbar (the bar at the top of WordPerfect with all the little pictures in it) or you can make your own, e.g., a toolbar with nothing BUT macros on it. This manual doesn't show you how to edit a Icon Toolbar, but it's not hard to do. For exact visual instructions if you are internet ready, go here: www.doglouddenback.com/wp/toolbar/index.htm

! Using Your Remapped Keyboard. You can "map" your WordPerfect keyboard so that (for example) Alt+S or Ctrl+Alt+S or Alt+Shift+Ctrl+S) where "S" is the letter you want to use) plays a macro. Since some "letters" in combination with the Alt, Ctrl, Shift keys are already used for other purposes, they will be unavailable unless you map your keyboard – and here's a nice tip from Barry MacDonnell: <http://home.earthlink.net/~wptoolbox/Tips/Assign.html>.

For macros, features, programs, and keystrokes: You can use "reserved" keys such as Alt+T (which normally accesses the Tools menu). But if you do, the macro (or feature or program or keystrokes) will play instead of the Tools menu opening. However, you can still get at the reserved keys: If you press Alt, release it, and then press T (i.e., press the keys in sequence, not simultaneously), the Tools menu will open.

This manual doesn't show you how to edit a Keyboard to include a macro, but it's not hard. For exact visual instructions if you are internet ready, go here: www.doglouddenback.com/wp/keyboard/index.htm.

! Using Shortcut Keystrokes. When naming a macro, you can use simultaneous keystroke combinations, such as Ctrl+1, Ctrl+Sft+1 as the macro's name (e.g., to play "Ctrl1.wcm" or "CtrlSft1.wcm"). To play such a macro, you press the same keystrokes simultaneously. See **Using Shortcut Names**, described in Chapter 3, below. Unlike in the good ol' Wp5.1 DOS days, when naming the macro you cannot use your "Alt" key to name the macro (e.g., Alt+G), even though you can remap your keyboard to use available Alt+keystroke as mentioned above.

! Using Startup or Desktop Macro Links. If you want, you can make desktop (or other) shortcuts which link to a specific macro. Theoretically, if such a link is clicked, WordPerfect should then open and immediately run the macro. However, in practice, that usage does not always work satisfactorily – particularly if the macro is large and complex. In such instances, it can cause WordPerfect to lock up. My recommendation is to NOT do that UNLESS the macro is short and sweet. That said, using Desktop Links to "[filename].wpd" doesn't have the lockup problem. So, should you find the need, you could make a WordPerfect document WHICH contains links to macros in the document. The Desktop Link could be to THAT document, and the macros linked within THAT document could then be safely run. This is just an idea for you to consider. I do it with my Oklahoma Lawyer macro program and it works well.

NOTES