

Automating
WordPerfect®
Templates

with Prompts, Bookmarks, and Macros

[For WordPerfect 8 - X9 and later versions]

by
Barry MacDonnell
Toolbox for WordPerfect
<https://wptoolbox.com>

© 2001-2018 by Barry MacDonnell.
All Rights Reserved.
Revised: August 15, 2018



Permission is granted to distribute
this document in its original unmodified
form and without charge or other
conditions to anyone.

Contents

Introduction.....	3
What is a <i>template</i> ?.....	4
Creating custom templates	5
Formatting custom templates	7
Automating custom templates.....	11
Using prompts and the Template Information dialog.....	11
Standard prompts	11
Address book prompts	12
Personal information prompts.....	13
Unprompted information	14
Creating or editing prompts	15
Using Prompt Builder.....	16
Using standard bookmarks and template macros	19
Template macros that insert unprompted material	20
Creating template macros	21
More complex template macros	26
Other template macro ideas.....	27
Load a letterhead logo from a file on disk	27
The template macro in "My Letterhead".....	28
Display “pick lists” for user input choices	28
“ <i>If...Then</i> ” template macros	29
Loading custom template documents	30
Use a <i>toolbar button</i> that points to the template.....	30
Use a <i>macro</i> to load a new template document	31
Create the macro first —.....	31
— then play it with a mouse click.....	32
Method A: Create a toolbar button to play the macro	32
Method B: Create a drop down menu to play macros	32
Appendix A - The template macro in “My letterhead”.....	35
Appendix B - Triggering a template macro	39
Appendix C - Did Windows hide some of your filename extensions?.....	42

Introduction

I began writing this paper in 2001 and it has been expanded several times over the years to include information related to WordPerfect's newer versions and to revise and clarify some existing information.

As a result, the paper is now 42 pages long, including several appendices. *Fear not!* Although it's a comprehensive reference on WordPerfect templates and how to automate them, most of the methods and techniques are more easily employed than it might seem from the amount of text used to describe them. As Roy "Iemoto" Lewis, a WordPerfect Universe administrator, once said:

"An explanation in words makes any activity seem more complex than it is in practice. To test this, write out instructions for boiling a kettle or putting a letter into an envelope. Or ride a bicycle."

Please take a little time now to scan it, or perhaps read the parts you are interested in at the moment. The time spent could save many hours later. (To help you navigate, blue ▲ symbols are clickable links.)

Barry MacDonnell
<https://wptoolbox.com>

What is a *template*?

In WordPerfect, a template is a document on your disk with a special filename extension, “.wpt,” where the “t” stands for “template”. Think of it as a stencil or model that you can use to make (or “spawn”) copies of the original.

Like other word processors, every document you create in WordPerfect is based on a template. Even new, empty documents with names like *Document1* are based on a special template, called the *default template*.^[1]

Templates—whether default or custom—define formatting and program options for a document such as margins, tab settings, styles, menus, toolbars, and keyboard definitions.^[2] Custom templates usually also include text, such as the “From:” and “To:” headings in a memo, or the “Sincerely yours” closing in a letter.

Most editions of WordPerfect ship with many predefined *static* templates, such as a simple, preformatted “To Do” list, as well as many *dynamic*, automated Projects^[3] that you can use to create personalized memos, faxes, résumés, and so forth, with little thought or effort.^[4] They are divided into groups, or *categories*, and can be seen when you click **FILE, NEW FROM PROJECT** (or **FILE, NEW** in WordPerfect 8).

[1] ▲ Tips:

- When you click on File, New from Project (or File, New in WordPerfect 8), the default template is listed as “Create a blank document” under the Custom WP Templates group. The actual filename on your disk is named something like WP12US.WPT, where “WP12” means WordPerfect 12 and “US” means the U.S. version. (If you can't find the default template in Windows Explorer, even though you are certain you know its filename, try this: Click on the Tools menu, select *Folder Options*, click the View tab, select *Show hidden files and folders*, and click OK. See <https://support.corel.com> and search for *Where are the WordPerfect® default templates?*. This document lists the standard installation locations for default templates for WordPerfect 9-X6, but later versions should be similar.)
- Unlike other templates, if the default template is renamed or deleted from the system WordPerfect will create a new (virgin) one the next time you start the program. Renaming it, and then editing the new version to copy “objects” such as toolbars and styles from the original, is often done by support staff if there could be problems with the default template. See, for example, the FAQ Forum at <http://www.wpuniverse.com>; look for a thread such as “Saving and Recovering Customisations.” Obviously, it is usually easier and faster to restore a template from a recent backup disk or tape.
- In some organizations the default template may have been customized and even relocated on another drive. To verify the name and location of your current default template, click Tools, Settings, Files, Template tab. This is where WordPerfect looks for the default template.
- To find out which template (default or custom) the currently open document is based on, click on Tools, Template Macro, Edit. You can see the template's filename in that dialog, or you can see it if you click the Location button in that dialog.

[2] ▲ You can automatically use a personalized keyboard or menu from another template so that the keyboard or menu will appear automatically when you open a new document. You can even have it appear only when you use a specific feature, such as a Header, Footnote, Outline, etc. See <https://wptoolbox.com/tips/Assoc.html>.

[3] ▲ A “Project” in WordPerfect is a dynamic template (the document changes based on your input) that includes the PerfectExpert panel on the side of your screen, and (in the Reveal Codes window) you often see [Named Region] codes, which are special bookmarks. On disk, a Project has two parts: a *.wpt file and a *.ast file. The latter is not modifiable by users, so be careful when modifying the former not to delete any codes in it.

The focus of this paper is on custom templates, not Projects. Custom templates require only a template file (*.wpt).

[4] ▲ While many predefined templates are installed during a Typical WordPerfect installation, more may be available using a Custom installation.

Once created and located in a template folder,^[5] templates are used only to create a *copy* or *image* of the document on your screen (i.e., in your computer’s temporary memory). The template itself is out of harm’s way for the casual user, making it more difficult to accidentally modify or delete. The document can then be edited, printed and saved without impacting the template.

Best of all, templates can be *automated*: You can insert prompts, bookmarks, and template macros in them to save many keystrokes and reduce the chance of input error.

Generally, you begin by creating a custom template, formatting it as desired. Then you automate it.


Creating custom templates

Creating a new template, such as a customized letterhead or fax cover sheet, is easier in practice than the following methods might seem at first.

Method A: The traditional method.

Advantages: The new custom template file will be displayed in the Project categories list along with other templates, and it can be easily found and edited with **FILE, NEW FROM PROJECT** (or **FILE, NEW** in WordPerfect 8). *Disadvantage*: This method takes an extra few steps, though the steps are typically easy and quick.

1. Click **FILE, NEW FROM PROJECT** (or **FILE, NEW** in WordPerfect 8). Under the “Create New” tab, select (i.e., highlight) “Custom WP Templates” or any other preferred group from the top drop list.
2. Click the **OPTIONS** button, then **CREATE WP TEMPLATE**. A new, blank template document should appear on screen, with the name *Template1* at the top of the window, and with the Template Property Bar displayed just below your other toolbars.^[6]
3. Type your text, format the document, and set any styles, toolbars, etc. Whatever you enter in this document and then save will become a template for future use.

 You can use an existing WordPerfect document as a basis for your new template by inserting it directly into the template. Simply place the cursor where you want the new file to be inserted and click **INSERT, FILE**. Choose your existing document and click **INSERT**. Answer “Yes” to any

[5] ▲ Normally, a template is stored in one of two special folders, the *default template folder* or the *additional template folder*. The actual locations of these folders can be found (and changed) in WordPerfect by clicking Tools, Settings, Files, Template.

[6] ▲ The Template Property Bar has the “Build Prompts” button on it. If it (or any property bar) does not show, click View, Toolbars, Property Bar to display it.

question about overwriting existing styles, which will overwrite defaults (such as font types) for the new template (not for all templates). Edit the newly inserted material as needed; for example, you might need to remove existing bookmarks, prompts, date codes, or text.

4. Click **FILE, SAVE**. The Save Template dialog appears.
 - a. In the Description field of the Save Template dialog, type a description (e.g., “My personal letterhead,” “Company invoice,” etc.). This appears in the New/PerfectExpert window when you create a new document.
 - b. In the Template Name field, type a filename for the project template, *without* a filename extension (e.g., “My personal letterhead”). The template file will be saved in a folder (subdirectory) that corresponds to the group, with a .WPT filename extension automatically added to whatever name you typed in the Template Name field.
 - c. From the bottom drop list, choose a template group/category (e.g., Custom WP Templates), then click **OK** to close the Save Template dialog.
 - d. Since the template file has just been saved, close the current window with **FILE, CLOSE**.^[7]



You can quickly load a new document based on this template as discussed in “Loading custom template documents” on page 30.

Method B: The non-traditional method.

Advantages: The new custom template can be very quickly created from an existing WordPerfect document file (.wpd) with a simple file renaming trick, and it can be placed anywhere on your computer (Desktop, My Documents folder, etc.).

Disadvantage: If you can’t remember its name or disk location, the template file might be difficult to find later since it is not displayed in the Project categories list. (But it can always be quickly used just like templates created with Method A, as noted below.)

1. [*Similar to Step 3 in Method A above.*] Type your text, format the document, and set any styles, toolbars, etc. Whatever you enter in this document will become a template for future use after completing the next steps.



You can use an existing WordPerfect document as a basis for your new template by inserting it directly into the template. Simply place the

[7] ▲ Unlike in WordPerfect 8, in WordPerfect 9 and later versions the filename you entered in the Template Name field will show up as the Display name in the PerfectExpert dialog (called the “New” dialog in WordPerfect 8) the next time you open it. But in any version you can change the Display name at any time: Simply right-click the project name in the dialog, click Project Properties, then change the Display name. Note that the Description is the text string that shows at the bottom left of the dialog, and is also modifiable with the Template Property Bar’s Description button when editing the template.

cursor where you want the new file to be inserted and click **INSERT, FILE**. Choose your existing document and click **INSERT**. Answer “Yes” to any question about overwriting existing styles, which will overwrite defaults (such as font types) for the new template (not for all templates). Edit the newly inserted material as needed; for example, you might need to remove existing bookmarks, prompts, date codes, or text.

2. *Save the file with a .WPT filename extension* (e.g., “MyLetterhead.wpt”) to any convenient folder—perhaps a folder set up for the purpose. *If the file already exists* on disk as a normal WordPerfect document (.wpd), you can use Windows Explorer or My Computer to find it, right-click on its name, and rename it directly in the file’s Properties dialog. *(If you cannot see the filename extension see the Appendix on page 42.)*
3. Whenever you double click on the template’s filename, a new WordPerfect document (.wpd) will be instantly “spawned” from it. The new document will be an exact duplicate of the template file, and the template itself will remain separate and unaffected when you edit or save the new document.



You can quickly load a new document based on this template as discussed in “Loading custom template documents” on page 30.

Formatting custom templates

As mentioned in the previous section, you can import (i.e., insert) an existing WordPerfect *document* as a basis for your new custom template, editing it as desired, then save your new template.

Optionally, you can edit an existing *template* by selecting the template’s name in **FILE, NEW FROM PROJECT** and clicking **OPTIONS, EDIT WP TEMPLATE** or by right-clicking the template name in the list. (If you created the custom template using Method B above, you will have to exit from WordPerfect, right-click on the file’s name in Windows, and choose Edit Template in WordPerfect in the Windows context menu that appears.)

When editing a template be sure to open the Reveal Codes window (**VIEW, REVEAL CODES**) to view the various bookmarks and formatting codes that may be present. Some bookmarks may be part of WordPerfect *prompts*, discussed in the next main section. You’ll want to be careful about changing or deleting them. ^[8]

As a safeguard, either import the existing template into a new, blank template with the

[8] ▲ Be especially careful not to delete any [Named Region] codes, which are used by PerfectExpert in some template Projects. Similarly, there may be style codes that are used by the template; these can be modified if required.

insert-file procedure in step 3 on page 5, or else immediately save the existing template with a different name. That way, you'll be modifying a copy of the original.

Here are a few tips to help you format your new template.

- **Hide certain format codes.**
 - Double-click the [Open Style: DocumentStyle] code in Reveal Codes at the top of the template being edited, or use **FILE, DOCUMENT, CURRENT DOCUMENT STYLE**. Either way, this brings up the Styles Editor for the template's initial, or default, document style. As the code's name indicates, it is an "open" style; hence, anything added or changed in the Styles Editor will take effect from the top of the document onward, until replaced or discontinued by some other format code.^[9]
 - In the Styles Editor, you can change the font and font size, and add other format codes (e.g., paragraph spacing, margin settings, etc.) from the Styles Editor's top menu and toolbar.
 - By placing format codes inside the Open Style, they will be relatively immune from editing by casual users. This can (potentially) prevent their accidental or deliberate removal.^[10]
- **Some formatting cannot be done in the Styles Editor using its menu or toolbar.** However, it might be possible to "trick" WordPerfect by first creating the codes in a template document's body text area, then cutting these codes to the clipboard (**CTRL+X**) and pasting them (**CTRL+V**) into the Editor's Contents field.

For example, suppose you decide to change the margins on page 1 of the template to accommodate your first page letterhead formatting. You could set up a *delay* on page 1 for restoring margins back to your preferred defaults for normal body text until page 2 (if there is a page 2). Cutting and pasting the [Delay] code (that is produced in the process) from the body text area into the Open Style would protect this code from accidental deletion when typing the letter. Users would simply see the margins automatically return to the normal

[9] ▲ Some WordPerfect users never use styles, preferring to control formatting in the text itself, where all format codes are clearly visible. Since WordPerfect is a "stream oriented" program, where codes take effect until discontinued, replaced or removed by another code, this makes some sense. You can see (in Reveal Codes) just what is happening with formatting by looking for various On and Off codes for a particular format to verify where a style begins and ends, or to check problems with overlapping or competing codes.

Styles, however, are like containers: They can hold text as well as format codes (and even other styles), and the codes inside them are relatively immune from global operations (such as Find and Replace) and casual user editing. They are easy to create (or edit) and apply, and may be suited for a variety of tasks. They can give a consistent look to the document, and have the advantage of being globally changed simply by editing any instance of the style.

In fact, WordPerfect ships with several paragraph styles (e.g., Heading 1 - Heading 5, outline styles, etc.) Even some things we tend to think of in WordPerfect as separate features are in fact a type of style: Headers, footers, footnotes, endnotes, watermarks, comments, outlines, etc. -- all are built-in styles! For more information on creating or modifying styles, see <https://wptoolbox.com/tips/Outlines.html>.

[10] ▲ You cannot password protect a template (.WPT). Only regular documents (.WPD) can be password protected when saving them.

defaults, if and when a second page is created during typing. Here's how to do it.

- Step 1. Close the Styles Editor if it is open. Put the cursor at the top of page 1, then click **FORMAT, PAGE, DELAY CODES**. Accept the "1" in the "Number of pages..." field and click **OK**.
- Step 2. In the Define Delayed Codes window that appears, click **FORMAT, MARGINS** (or click the **PAGE SETUP** or **PAGE SIZE** button).
- Step 3.
 - *WordPerfect 9 and earlier* - In the Page Setup dialog that appears, set new margins or accept the default of 1" for all margins, then click **OK**. Go to Step 4 below.
 - *WordPerfect 10 and later* - If you have not changed any margins on page 1 but you want to ensure that margins are always set back to the 1" default on any second or subsequent page, follow the instructions in the next paragraph. If you have changed any margins to a new value on page 1, you should not need these instructions (but please read them before going to Step 4 below).
 - In the Page Setup dialog that appears, change any of the margin settings a small amount, then click **OK**. This will insert all four margin codes in the Reveal Codes window of the Define Delay Codes screen. Then change the margin settings back to your default (1") or to some other setting. This extra step of changing margins a small amount in Define Delay Codes, then resetting them to the desired amount, is a workaround for a small (uncorrected) bug in WordPerfect 10/11/12/X3. In WordPerfect 9 and earlier versions the program would insert the four margin codes -- with a default 1" setting -- by simply clicking **OK** in Page Setup. If you just click **OK** in WP10/11/12/X3 to try and insert default margin settings, nothing happens: no codes are inserted.
- Step 4. Click **CLOSE** to exit from the Define Delay Codes window. In Reveal Codes, you should see a [Delay] code on page 1. In Reveal Codes, select *just* the new [Delay] code, cut it to the clipboard with **CTRL+X**. Then double-click the initial [Open Style: DocumentStyle] code at the top of the document to open the Styles Editor; paste the [Delay] code into the Contents field with **CTRL+V**, then click **OK**. (Just remember to look inside the *template's* Open Style code if you ever need to revise or delete this [Delay] code!)

- ***Don't use the FILE, DOCUMENT, DEFAULT FONT option!***

While you can easily set up a new font and/or font size for your template in the Styles Editor, be aware that any codes for fonts and font sizes that are entered in the Editor's Contents field will take precedence over any font and font sizes you might enter with **FILE, DOCUMENT, DEFAULT FONT**, either during template creation or later, after the template is saved to disk. The Styles

Editor's codes are "downstream" from any changes made with **FILE, DOCUMENT, DEFAULT FONT**, so it is best to avoid using this menu selection. Instead, make changes directly in the Styles Editor. ^[11] ^[12]

- ***Suppress some items on a page, and delay activating others until another page.*** ^[13]

For example, suppose you are writing a book and you need to create a "Chapter" template with a page number at the *bottom* of the first page and remaining page numbers at the *top* of the following pages. Here's how to do it.

- **Step 1.** Create a *header* (either A or B) that will appear at the top of *subsequent* pages in the chapter: Put your cursor at the very top of the template and click **INSERT, HEADER/FOOTER**. Choose the header type (A or B), and when the header area appears (the cursor will be in it), type any desired text and then insert a page number code with **FORMAT, PAGE, INSERT PAGE NUMBER**. Justify the text and number as desired, then click outside the header area to return to the main document.
- **Step 2.** *Suppress* this new header (and thus the page number displayed inside it) with **FORMAT, PAGE, SUPPRESS, HEADER** (choose either A or B). This keeps the header from appearing on the first (i.e., current) page of the chapter, but it will appear at the top of subsequent pages since it is suppressed just on the first page.
- **Step 3.** Create a *footer* (A or B) with a page number code in it (as above). Then "discontinue" it on following pages: Click **FORMAT, PAGE, DELAY CODES, 1, OK**; click the **HEADER/FOOTER** button, then choose **FOOTER** (A or B); then click **DISCONTINUE, CLOSE**. The page number will appear at the bottom of the first page of the chapter but it will *not* appear at the bottom of subsequent pages, since it will have been "turned off" on the next page and following pages.

Note that headers, footers, and watermarks continue until replaced at some point by a new version of the header, footer, or watermark (which is what happens here in a multi-chapter document) or continue until discontinued (in which case, they simply stop at that point). The latter is often done with a Delay code to ensure the header, etc., appears on a given page but does not appear after some (delayed) number of pages.

[11] ▲ This is also true of existing regular documents (.WPD) where you might want to change their "default" font. If you use File, Document, Default Font and the font for that document doesn't change, double-click its Open Style and check to see if there are codes in the Contents field. Also check if there are font codes in the body text area. These are even further "downstream" and will take precedence over codes in the Contents field or set with File, Document, Default Font.

[12] ▲ In addition, according to Charles Rossiter (Corel C_Tech) the font codes entered in the Styles Editor will apply to all printers on your system, while those entered with either File, Document, Default font or File, Font will apply to the current printer only -- even if the Settings button, "Set face and point size as default for all documents," is enabled (this setting is available in WordPerfect 9 and later versions).

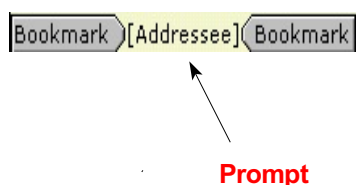
[13] ▲ You can also do this with a macro. For more details, see <https://wptoolbox.com/library/SupDelay.html>.

Automating custom templates

When you load a new template-based document such as a fax cover sheet, did you ever wonder how WordPerfect displays a fill-in form and then inserts the information into the document? Even if you don't see the form, how does WordPerfect enter your name, address, and telephone number on some documents? The answers lie in the use of *prompts*.

Using prompts and the Template Information dialog

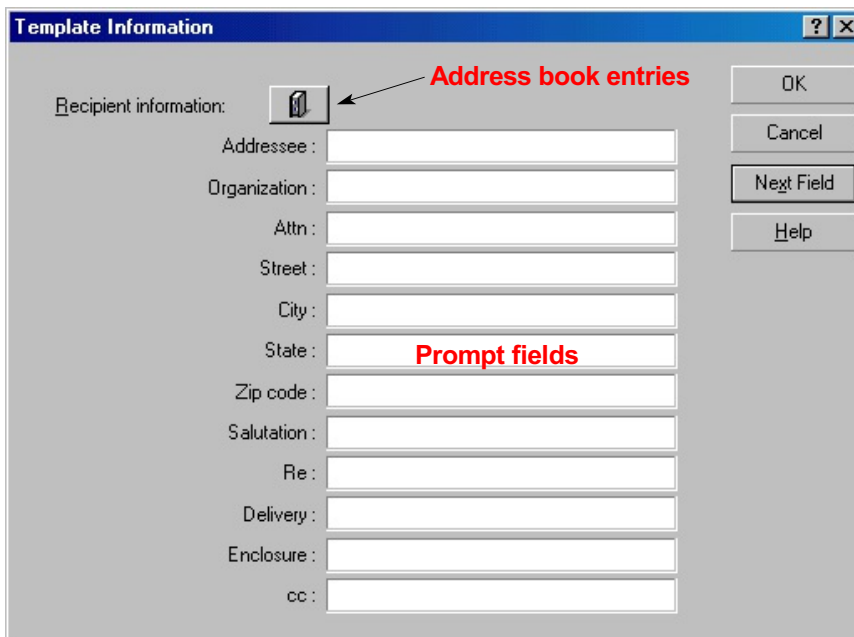
In a nutshell, prompts are special *bracketed text strings* in templates that are surrounded by *paired bookmarks* (we'll discuss single bookmarks later). If you open Reveal Codes (**VIEW, REVEAL CODES**) in a prompted template, they look something like this:



You can create and name these prompts yourself, using WordPerfect's *Prompt Builder* (discussed below), which does almost all the work for you. Later, when you load a new document based on the template, WordPerfect will use these bookmarks to locate the places in the new document where you want certain information to appear (e.g., a name or address). It then replaces the text prompts between the bookmarks with the necessary information.

Standard prompts

Some prompts, such as the standard prompt above, are replaced only after you fill in some *prompted information* in a form, known as the Template Information dialog. The Template Information dialog (created by Prompt Builder) pops up whenever you load a new document based on a template that has prompts in it. For a letterhead, it might look something like this:



WordPerfect uses this type of “fill-in screen” for information that you will change every time you use the template, such as the name of a letter’s recipient. Once you enter the information in the Template Information dialog’s prompt fields, the information is automatically inserted in the correct locations in the template.

Address book prompts

In addition to this simple template automation feature, you can *link* address book fields to these Template Information prompts with a button in Prompt Builder. This creates an address book prompt, which looks just like a standard square-bracketed prompt in the document. However, when the Template Information dialog pops up you can click the Recipient Information button at the top of the dialog and choose an entry from your address book. ^[14] WordPerfect will insert that entry into the proper Template Information prompt fields. When you dismiss the dialog, it will insert this address book information in the document. ^[15]

[14] ▲ Of course, the address book must have been installed and have some names entered in it for this to work. And if you haven’t linked any fields in the Template Information dialog to your address book, no Recipient Information button will appear at the top of the dialog. See “Using Prompt Builder” on page 16 for more information.

[15] ▲ Note that if you click on Cancel from the Template Information dialog, you’ll get a document that contains the visible text placeholders (e.g., “[Addressee],” “[Organization],” etc.) and their associated bookmarks. If you simply want a new document without the prompts (i.e., the text placeholders), immediately click OK instead. This deletes the prompts (but not the bookmarks).

Personal information prompts

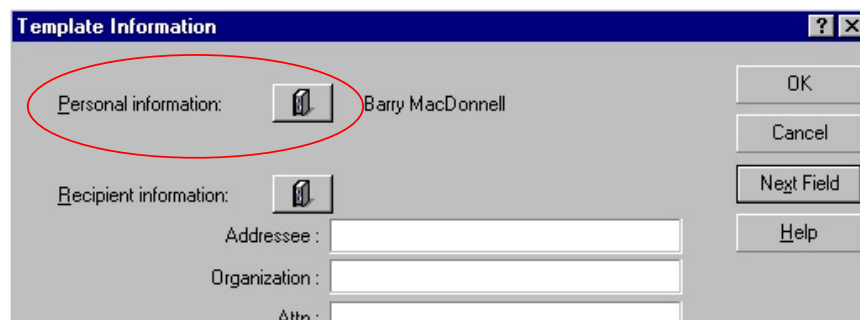
Finally, some prompts — called *personal information* prompts — are replaced *automatically* with personal information from your address book.^[16] These prompts are linked to fields in the current personal information entry in your address book (i.e., your own name and other personal data).

How can we distinguish personal information prompts from standard and address book prompts in the template? Easy. Personal information prompts appear in <angle brackets> in the template document, like this:

Bookmark <Organization> Bookmark

Note that personal information fields associated with these prompts do not appear in the Template Information dialog box. WordPerfect uses these angle-bracketed prompts "behind the scenes" to insert your name, phone number, etc., into a template document.

However, if you use personal information prompts, Prompt Builder will create a Personal Information label and button above the Recipient Information area in the Template Information dialog, like this:



[16] ▲ The first time you open a predefined template that requires it, WordPerfect will ask you for your "personal information." This will be stored in your address book just like any other address. Generally, this is a one-time procedure. To modify it later, click File, New from Project (or File, New in WordPerfect 8), then click Options, Personal Information. You'll see a message about the current setting; click OK to open the address book, then select another address to be the default. (You may want to use the Options button again to Refresh Projects.)

You can have only one such default address entry for your Personal Information, and if you change it (with File, New from Project [or File, New in WordPerfect 8] and then use Refresh Projects) the change will affect all new documents that use personal information prompts. For this reason, those who need different names on a document (such as a letterhead) may be better off creating copies of the template *without* any personal information prompts (i.e., type personal information directly on the template itself without using prompts). Each template can be invoked from a toolbar button or menu (which is explained later in this paper).

This allows you to select the address book entry that you wish to use for personal information in the template.^[17] Note that this choice is “sticky”: it will remain in effect until you select another entry.

Using the steps outlined in the Prompt Builder section on page 16, you can create either type of prompt and paste it into position in a template. As soon as you finish doing this for all the prompts you need, WordPerfect will automatically create the Template Information dialog box and save it as part of the template. Later, this box will pop up whenever you load a new document based on that template. You just fill in the necessary fields and WordPerfect does the rest.

You can prompt yourself for different things. For example, you might create a custom letterhead with prompted information that includes the recipient’s name, address, and salutation, as well as optional delivery method, attention (Attn:), reference (RE:), enclosure, and courtesy copy (cc:) lines.

As mentioned above, you can either type this material into the Template Information dialog’s prompt fields or you can have WordPerfect automatically insert it in the dialog from your address book. The latter is accomplished by clicking the Recipient Information’s address book icon in the dialog, selecting a person’s name from the address book, and clicking **INSERT**. Once you dismiss the Template Information dialog, WordPerfect will insert all of the information from the prompt fields into your document.

Note that you can add, edit, or delete prompts from a template at any time. Both standard and address book prompts can be deleted using Prompt Builder’s **DELETE** button. Personal Information prompts can be deleted in the template itself, taking care to delete the bracketing bookmarks around the prompts. (You should do this with Prompt Builder on screen so that when you click **OK**, Prompt Builder will regenerate all remaining prompts. Also see the cautions on page 15.)

Unprompted information

Unprompted information—that is, information you do not have to type into the Template Information dialog—can come from three sources:

- Personal Information (your own name, address, etc.), inserted automatically by the program. You first must paste *personal information prompts* with their paired bookmarks into the template with Prompt Builder, using the instructions below.

- and/or -

[17] ▲ Two points to consider: (a) Be sure that all potential Personal Information entries contain data in all related fields (e.g., phone number or Zip code) used in the template. Otherwise, some information may not be inserted where expected. (b) If you work in a busy office where you are constantly changing the personal information selection, it may be easier to create several “hard coded” letterhead templates where the personal information is typed directly on the templates, and not use Prompt Builder’s Personal button to insert these fields, as explained on page 17. Multiple templates can be easily accessed with individual toolbar buttons or from a menu, and may save you some time and reduce possible error. Alternatively, you could code a template macro to automatically display one or more lists of personal information; the choices from the lists would be inserted in the document by the macro. See “Example #2” on page 28 for more information.

- Information inserted automatically by a *template macro*, using *single bookmarks* instead of prompts. You first must insert the bookmarks into your template, then create the template macro, using the instructions below. For example, a letterhead template might contain a “startup” template macro whose job is to locate a specific bookmark, named “[CreationDate].” When it finds it, it enters the current date at that location as a text string (which will remain unchanged and always show the document’s creation date). The macro then might go on to find other bookmarks on the page to insert text labels (if required because a Template Information dialog was used to get and insert certain data) such as "Attn:," "Re:," and so forth.

- and/or -

- Information displayed by a *code*, such as a date code (which will always show the current date) or a page number code. You insert codes like these in the template when you create or edit it, using **INSERT, DATE/TIME** or **FORMAT, PAGE, INSERT PAGE NUMBER, PAGE** (or **TOTAL PAGES**), **INSERT**.

In summary, prompted information (e.g., your recipient’s name and the subject matter of your letter) and unprompted information (e.g., your own name and the date of the letter) are inserted at specific locations automatically, using a prompt (either square-bracketed or angle-bracketed), a bookmark-with-template-macro, or a code.

Creating or editing prompts

In order to get prompted information from the user, or automatically insert your own name and organization, you use WordPerfect’s Prompt Builder feature from the Template Property Bar as explained on page 16. It will do the job of creating pairs of bookmark codes surrounding text prompts (see above for examples as seen in Reveal Codes). You then insert or “paste” the prompts at the desired locations in the template.

Note that Prompt Builder creates a *one-choice-to-one-location* or a *one-choice-to-many-locations* relationship between the Template Information dialog’s fields and the document locations where the data in those fields will be inserted. For example, if the Template Information dialog asks for an addressee’s name, it can be automatically inserted into the document at one location or in several locations (e.g., at the top of page 1, in a header, in a text box, etc.).

Though more complex to create than a Template Information dialog, it is possible to create a *many-choices-to-one-location* or *many-choices-to-many-locations* relationship between data input (such as a custom dialog’s static or drop-down list of many data items) and data insertion locations with a template macro. This will be discussed later, along with sample macro code.

Important cautions (and some tips)

When you use Prompt Builder, you will be asked to give each bookmark pair a name. WordPerfect also internally *numbers* the pair, which comes in handy later if you need to create a template macro to search for bookmarks that are not part of your

personal information or that require special processing, such as deleting empty address or reference lines. Using a bookmark's number is more specific in a search than using a bookmark's name, especially if you've used the same prompt (and thus, the same bookmark name) in more than one location.

Therefore, *if you add or delete prompts*^[18] in a template using Prompt Builder, or use the same prompt more than once in different locations (such as using the Addressee of a letter in both the address block and a header), WordPerfect will assign new numbers to the bookmarks surrounding the text placeholders—and not always in numerical order in the document. In most templates this is not a problem and you can simply ignore this process.

However, *if there already is a template macro in the template* that plays when the template is accessed, you should make modifications to prompts with caution. For example, part of the author's "start up" template macro in his letterhead template searches for certain prompts by their bookmark number, not their name. Other authors may have written and stored macros in templates that may act in a similar manner.

Putting the cursor in front of a paired bookmark code in Reveal Codes will display the bookmark's number inside the code itself. You may have to identify numbered bookmarks and then edit any "startup" template macro to make sure it searches properly for the renumbered bookmarks.^[19]

If you move prompts, be sure to move any related, adjacent bookmarks (single marks, if present, and both paired marks) as a block. Otherwise, some information may not appear exactly where intended.

Finally, you are limited to 64 prompts per template.^[20]

Using Prompt Builder

1. Create a new *custom* template, or open an existing one: Click **FILE, NEW FROM PROJECT** (or **FILE, NEW** in WordPerfect 8), then click the **OPTIONS** button and choose either **CREATE WP TEMPLATE** or **EDIT WP TEMPLATE** (to open an existing *custom* template).^[21]

[18] ▲ The best way to delete a prompt is with Prompt Builder, and not by editing the template and deleting the text string that is the actual prompt. Using Prompt Builder, you can delete all instances of a prompt at once, and delete their bookmarks, too.

[19] ▲ When the template is being edited, click Tools, Template Macro, Edit. Examine any macros to determine if they use the BookmarkFind() command. A number in the parenthesis of this command means it is designed to search for the bookmarks by number, not by name.

[20] ▲ The original release of WordPerfect 9 (version 9.0.0.588 as shown in Help, About) will not display more than twelve prompts in the Template Information dialog. This was corrected in Service Pack 2 for WordPerfect 9. See Corel's Support page at <http://www.corel.com> for WordPerfect service pack downloads. Note that WordPerfect 9's Service Pack 4 (SP4) was a complete rewrite of the program and was never available as a download; moreover, it is no longer available from Corel. However, if you obtain SP4 from a friend or colleague you can legally use it—if you are a legal owner of WordPerfect 9. You will need to completely uninstall any previous version of WordPerfect 9 first, so be sure to read the Release Notes for SP4.

[21] ▲ There is a simpler way (*but with some drawbacks*) to create a custom template: Just rename a regular, saved document (.wpd) with a template filename extension (.wpt). For more information see <https://wptoolbox.com/tips/Templates.html#custom>.

2. Enter or modify the text in the template, such as the letterhead, and format the document as desired. Leave the cursor in the body text area of the document—not in a header, footer, etc.
3. Click on **BUILD PROMPTS** on the Template Property Bar. The Prompt Builder dialog opens. ^[22]
4. Click **ADD** to create a prompt. The Add Template Prompt dialog appears. The prompts created in this small dialog will eventually appear in the Template Information fill-in window that appears when you load the new document.
 - a. Type “Addressee” (or whatever prompt name you want to use, without quote marks) in the Prompt field. This will eventually produce what we have been calling a “square-bracketed” prompt, like this: [Addressee]. This type of prompt will be used later by WordPerfect to insert information that changes each time you load a document based on the template, such as the name of a letter’s recipient.

Optional: You can link the prompt to an address book field so that the information stored in that field is automatically inserted later with a mouse click in the Template Information dialog window. In the Link to Address Book field drop list, select “Display Name,” “Name,” “Organization,” or other required field. If you select <none>, no link is created and the information will have to be entered manually in the Template Information dialog when you load a new letter.
 - b. Click **OK**. You should now be back in the Prompt Builder dialog.
5. Repeat step #4 and add the other prompts you need. ^[23]
6. You can move the prompt up or down in the list with the appropriate button. This also moves the associated field’s position in the Template Information dialog.
7. Next, you need to *paste* each of the prompts into the appropriate locations on the template's body text area by positioning the cursor in the template where the prompt is to go. (You can use returns, tabs, spaces, or advance codes to position the cursor.) Once positioned, click the **PASTE** button in the Add Template Prompt dialog. This dialog will remain on screen (and can be dragged out of the way) while you paste all prompts in the body text area. You can add a text label such as “Attn: ” or “cc: “ in front of a prompt. ^[24]
8. *Optional:* To paste a personal information prompt, such as your name or address, click the **PERSONAL** button. In the paste procedure the prompt will appear

[22] ▲ For the Prompt Builder to work, you must have these two Corel files available: PROMPTS.WCM (in your default or supplemental *macros* folder) and _AUTOTMP.WPX (in your default or additional *templates* folder; in Windows XP this typically is the ..\Custom WP Templates\ folder; also, note the leading “_” in the second file’s name).

[23] ▲ As mentioned on page 16, the original release of WordPerfect 9 will not display more than twelve prompts in the Template Information dialog.

[24] ▲ The author’s letterhead template uses a template macro, reproduced at the end of this paper, to remove such text labels if no related information was inserted at that location. Such “clean up” tasks are one of the major benefits of using a template macro.

surrounded by <angle brackets>. As mentioned, this will not create a fill-in field in the Template Information dialog since there is no need for one: This type of prompt will be replaced automatically in the document with your personal information.

9. Close Prompt Builder with **OK**. WordPerfect then immediately and automatically “generates” the prompts.
10. During the paste process, you can add some of the same prompts to other locations in the template (creating the *one-choice-to-many-locations* relationship mentioned above). For example, if you created prompts named “Addressee” and “RE” you could paste them in the *both* the body text area of a letter and into (for example) Header A. Further, if you suppress Header A on the first page with **FORMAT, PAGE, SUPPRESS**, the addressee and reference information will show up only on the second and subsequent pages.

To do this, first create Header A by clicking **INSERT, HEADER/FOOTER, HEADER A, CREATE**. When you are ready to paste the prompts in Header A, click **INSERT, HEADER/FOOTER, HEADER A, EDIT**. The cursor should now be inside Header A.

But wait—where’s the Prompt Builder button? Since you are now inside a header, not in the body text area of the template, the Header Property Bar is displayed. This is as it should be. However, you need to toggle on the Template Property Bar: Right-click anywhere on the Property Bar area, and choose **SETTINGS**. Under the Toolbars tab, check “Template” then click **CLOSE**. Now you can add the prompts. Click the **PROMPT BUILDER** button and **PASTE** the prompt into the header. Close the Header A window and return to the template document window.

11. Save the template. (See instructions under **FILE, SAVE** on page 6.) When you open a new document based on that template, the Template Information dialog will appear so that you can input information (which is then plugged into the prompt areas of the document, as explained on page 11).
12. If you update the template at some later date, be sure to update the prompts, too. Click **FILE, NEW FROM PROJECT** (or **FILE, NEW** in WordPerfect 8), then click the **OPTIONS** button and choose **EDIT WP TEMPLATE**. Click on the **BUILD PROMPTS** toolbar button and click **OK**. Save changes to the file and close the template.

Using standard bookmarks and template macros

Some material can be inserted in a document *without* using Prompt Builder or the Template Information dialog.^[25] To do this you use one of WordPerfect's standard features to create *bookmarks* at the locations in the template where you want the material to appear. Then you create a *template macro* that is “triggered” to play when you open a new document based on the template (see the next section). The macro will find the bookmarks and insert the unprompted material.^[26]

Here's how to create these bookmarks:

1. If the template is not already on screen, open it for editing by clicking **FILE, NEW FROM PROJECT** (or **FILE, NEW** in WordPerfect 8). In the “Create New” tab, locate the template in the list and right-click its name; in the context menu that appears, click **EDIT WP TEMPLATE**. Alternatively, click **OPTIONS**, then **EDIT WP TEMPLATE**.
2. In the template, place your cursor where your first unprompted information will go—for example, the location where you want the current date to appear as a text string. (If you prefer, the date can be set flush right by first clicking **FORMAT, LINE, FLUSH RIGHT**.) With the cursor now in the correct location, click **TOOLS, BOOKMARK, CREATE** and create a bookmark named “Date” (without quotes). Do *not* check the box, “Selected Bookmark”: You want to set a simple location marker and not select anything in the process.

Notice that in Reveal Codes the bookmark will appear at the cursor location as a *single* (not *paired*^[27]) bookmark. Verify the location for this bookmark.

[25] ▲ You could actually do both: You could create a template macro (see page 20) that displays the Template Information dialog and which, after it is finished inserting the material entered by the user in the Template Information dialog into the document, could display a *custom* dialog—say, one with a pick list of names or other items the user can choose from—which then inserts the information in specific locations in the document.

[26] ▲ An slightly related feature is using a special bookmark called a *QuickMark* to find a location in a document. If you automatically set one of these in a *normal* document (i.e., a .wpd file, not a .wpt template file) with Tools, Bookmark, “Set QuickMark on file save,” and “Go to QuickMark on file open,” then WordPerfect will search for this code and place the cursor there when you open that same document again. [N.B.: This setting does not work properly in templates, only in normal documents. For templates, use a template macro to find one or more standard bookmarks, as described above.]

However, be aware that if you use a template macro associated with the Post Open trigger (triggers are discussed on page 22), the template macro's operation will take precedence over any Go to QuickMark process, and it probably will leave your cursor at a location other than your last typing position. Hence, if you use the automatic QuickMark placement method, you should include a QuickmarkFind() command at the end of the startup template macro, so that the last thing the macro does is locate the QuickMark.

[27] ▲ Paired bookmark codes are created in two ways: (1) by using Prompt Builder, and (2) by first selecting some text before using Tools, Bookmark, Create. In Reveal Codes, paired codes are wedge-shaped and are either automatically numbered by Prompt Builder or, if created by the user when selecting text, they are named when created. Because they “bracket” text, if you—or a macro—search for and select the bookmark pair, *any text between the paired codes can be deleted or replaced*. This is how the DoFiller function (see footnote 30) works when you finish filling out the Template Information dialog: it replaces the bookmarked Prompts with the data from the Template Information fields.

Ordinary (single) bookmarks, created by the user, are rectangular and are not numbered by WordPerfect. They are identified by name only. They act as “locators” so that you can find a spot in the document (just like using a real bookmark in a book), or so that a macro can find them and insert something at their location.

3. Repeat this process for other bookmarks you want to set for the positioning of unprompted material, giving each a different name.^[28]
4. Save the template with **FILE, SAVE**.

Template macros that insert unprompted material

Now that bookmarks for unprompted material exist in the template, you can locate each of them with a simple search procedure and do something at the bookmark's location. This is accomplished by creating a *template macro* that will be played ("triggered") automatically when a specific event occurs, such as loading a new document based on the template, printing the document, etc.

A template macro is simply a macro stored in the template itself, rather than existing as a separate file on your computer. It can find bookmarks, insert or replace information, conduct a spell check, and even pop up a message for the user. Anything you can do with a regular macro can be included in a template macro, which you can then have WordPerfect play when a specific trigger event occurs. (Triggers are discussed on page 22.)^[29]

There are several ways to create a template macro. It can be recorded, or it can be written from scratch, or it can be created with a combination of these procedures. It also can be copied from an existing macro on disk directly into the template using the Copy/Remove Object button on the Template Property Bar (select "Macros on Disk" as the "Object Type").

Let's create two very simple template macros to illustrate this process. One will find a bookmark in the document and insert the date (as text characters, not as a dynamic date code), and the other will display a simple reminder message to the user before printing the document. We'll record the first macro, and write the second to illustrate both procedures.

Important note about using template macros with Prompt Builder

Many people will stop the process of automating a template after using Prompt Builder (see page 17) since all they want is a "fill in" form when the document loads. You might, however, want to have a fill-in form get information from the user *and then*

[28] ▲ User-created bookmarks, unlike bookmarked Prompts, must be uniquely named. You cannot use one of these bookmarks in several document locations, since they are not internally numbered by WordPerfect in the way that bookmarks created with Prompt Builder are numbered. However, you could name some of them similarly if they perform similar functions (e.g., Date1, Date2, etc.).

[29] ▲ If you plan to share documents containing template macros with other WordPerfect users, take note of this:

If a template containing a template macro is set to be triggered by some event, note that *the template macro will only trigger in a document (.WPD) if the template (.WPT) is present, either in the original template folder or in the default (parent) template folder*. If WordPerfect cannot locate the template, the template macro will not play even though it is embedded in the document itself, since the trigger specification is stored in the template, not in the resulting document.

This can cause problems when sharing such automated documents with other WordPerfect users. Make sure they also have the template that spawned the document and it is located in the same relative folder as the original. [This might explain why template macros fail to automatically play in some documents if the original template has been renamed, relocated, or removed. See also Footnote 34 for another reason this might happen.]

have a template macro do some other things such as remove blank address lines. These are two separate things.

If you want to use *both* the pop-up Template Information fill-in dialog *and* a template macro to insert additional data or perform various other tasks, you will need to add the *TemplateFill()* command to the template macro, typically as the first command, and replace the existing association with the Post New trigger. [See footnote^[30] below.] You can add the *TemplateFill()* command to the macro anytime by editing the template macro using the instructions below.

Creating template macros

Example: Record a template macro that will insert the date (as text) when loading a new document^[31]

1. If the template is not on screen, open it for editing by clicking **FILE, NEW FROM PROJECT** (or **FILE, NEW** in WordPerfect 8). In the “Create New” tab, locate the template in the list and right-click its name. In the context menu that appears, click on **EDIT WP TEMPLATE**.
2. If you have not already created a bookmark named “Date” (or something similar), do so now using the procedure on page 19.

Note that if you want to make the template macro insert the date (as text) in several different locations, you must create several bookmarks, one for each location (e.g., Date2, Date3, etc.).^[32]

[30] ▲ In this case of such “dual automation” techniques — using fill-in forms plus template macros — there is a small issue to deal with. Something called the “<dofiller>” function, whose job is to automatically display the Template Information dialog in new documents and optionally insert personal information (among other things), is normally activated by Prompt Builder and then automatically associated with the Post New trigger in the template. (Triggers are discussed on page 22.) The macro-like <dofiller> feature thus occupies the single “slot” used by the Post New trigger in the template’s Associate dialog. You normally couldn’t use this slot to play your template macro, too!

However, in effect *TemplateFill()* is the macro command equivalent of <dofiller>. So, if you want to use a “startup” template macro — that is, one *associated* with Post New — *and since you can have only one item associated with a given trigger event* (normally, this is <dofiller> for the Post New trigger) — you can use *TemplateFill()* to do the job of <dofiller> instead. Here are the basic steps:

Use Prompt Builder (see page 16) to create and insert bookmarks/prompts and automatically build the Template Information dialog. Then create the template macro. You can place the *TemplateFill()* command anywhere in the template macro where you want the Template Information dialog to appear in the automation sequence, but most often it is used as the first command. Then complete the rest of the macro (see page 21). Once you save the template macro in the template, you can associate it with the Post New trigger *by first removing any <dofiller> command in the Associate dialog’s field and inserting the name of the template macro instead*. [N.B.: Be sure to use *TemplateFill()* only once in any given template.]

[31] ▲ If you are editing a template that already has one or more [Date] codes in it, the date will be displayed by the [Date] code. The date is “dynamic” and changes to the date the document is opened. (You can delete these codes if you don’t need them.) Compare this to the “static” date we are using in this example; some users always want static dates inserted—perhaps just at print time—and do not want dynamic dates.

For more on this topic, along with some macro snippets you can use in any template, see “Change all date codes (dynamic dates) to text (static dates)” at <https://wptoolbox.com/library/DateRepl.html>.

[32] ▲ Unlike Prompt Builder, which can use the same bookmarked prompt it generates in several locations (because they are internally numbered differently), user-created bookmarks must be uniquely named.

The next steps will use the bookmark(s) to insert the current date (as text) in the document.

3. Click **TOOLS, TEMPLATE MACRO, RECORD**. The Record Template Macro dialog box appears. Give the macro a name such as “Insert current date.” (Whatever name you give it will be used in step 9 below.) Click **RECORD**.
4. Click **TOOLS, BOOKMARK**, then select the bookmark named “Date” (or whatever name you have given it). Click **GO TO** to position the cursor at the bookmark's location in the document.
5. Insert the current date with **INSERT, DATE/TIME**. Select the date/time format from the list. Be sure that the box at the bottom, “Keep the inserted date current,” is *not* checked. For this exercise, you’ll want the date to remain static, so that if you open the document later, the date will reflect the document creation date, not the (then) current date. Click **INSERT** to insert the date.
6. Repeat steps 4 and 5 if you have additional static dates to insert, and their locations were previously bookmarked in step 2.
7. Click the **STOP** button on the Macro Recording toolbar. (It is the leftmost button on that toolbar, with a small black square on it.) The template macro is now stored in the template.
8. Since the macro added text date(s) while you were recording it, remove these text date(s) from the template itself—but do not remove the bookmark(s) that mark the date location(s). You can usually do this by undoing the changes with **EDIT, UNDO** or **<CTRL+Z>**. Or, use Reveal Codes and carefully remove just the text date(s) characters, but leave the codes alone.
9. To check things out—and give you some experience in modifying a template macro—we’ll now edit the macro you just created. Click **TOOLS, TEMPLATE MACRO, EDIT**. Select the macro’s name in the bottom window, and click **EDIT**. A new window appears with the macro code in it. It should look something like this (if you had only one date to insert):

```
Application (WordPerfect; "WordPerfect"; Default!; "EN")
BookmarkFind (Name: "Date")
DateText ()
```

If there are any other commands other than one Application command, one or more BookmarkFind commands, and one or more DateText commands, it is safe to delete them. Then click the **SAVE & COMPILE** button on the Macro Toolbar, then click **OPTIONS**, then click **CLOSE MACRO**. (If there are no extraneous commands in the code, simply click **OPTIONS**, then **CLOSE MACRO**.)

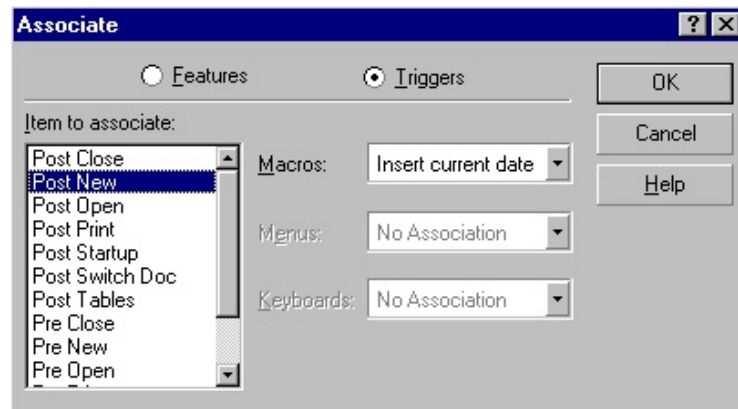
10. You should now be back in the template document.

Now we’ll “**trigger**” this template macro to play automatically when you load a document based on the template.^[33] Click the **ASSOCIATE** button on the Template

[33] ▲ Note that ordinarily only *template macros* can be associated with a trigger. Macros on disk that are not already part of that template cannot be associated with template triggers, but they can be copied into the template (continued...)

Property Bar. Next, click the **TRIGGERS** radio button, then select “**Post New**” in the left window. Finally, in the Macros drop list, choose the name of the template macro (“**Insert current date**”).

The dialog should look something like this for our example:



Click **OK** to close the Associate dialog.

11. Save the template with **FILE, SAVE**.
12. Test the operation of the macro by loading a new document based on the template. The date(s) should appear in the proper location(s) as soon as the document loads. ^[34]

Example: Write a template macro that will display a message before printing the document

Here, we’ll use two techniques to write (rather than record) a template macro. One might be called the “separate file” technique because you write the macro’s code in a separate file, save it, and then copy the macro from your disk into the template. The other might be called the “same file” technique because you write the macro’s code directly in the template itself.

[33](...continued)

first with Copy/Object on the Template Property Bar, and then associated with a trigger. *However*, if a disk macro already has been included on a toolbar, or on a menu, or associated with a keystroke *in that template*, then it will be available to associate with a trigger in that template.

The two most common triggers used with simple template macros are Post New (i.e., play on opening a new document based on the template) and Pre Print (i.e., play at print time). Each trigger can have one (and only one) template macro associated with it, and only *one macro per template* can have the (optional) TemplateFill() command in it, and even then it should be used only once in that template.

For more information on triggers, see the WordPerfect help file topic, “Triggering macros from templates,” or Appendix B (page 39).

[34] ▲ If the template macro does not run when loading a new document, template macros may have been turned off by a user. According to the Corel Support Database (<https://supportcorel.com>), “When a template is opened which contains a PerfectScript® template macro, a dialog is presented asking if the user wishes to disable the macros in the document. If you answered enable the “Do not show this message again” option and clicked Yes, you will be unable to use any template macros within WordPerfect. . .” See the above Corel site and search for the solution (or see [here](#)).

The first technique — because it is done in a separate file — may be easier to test and debug. The second technique may be useful for simple macros that contain only a few lines.

The “separate file” technique

1. Click **FILE, NEW FROM PROJECT** (or **FILE, NEW** in WordPerfect 8). Display the Macro Toolbar with **TOOLS, MACRO, MACRO TOOLBAR**. Among other things, this toolbar turns off typographical (“curly”) quotes, which the macro compiler doesn’t like.
2. Carefully type the following command at the top of the window (note the two semicolons):^[35]

```
MessageBox(;"Reminder";"Print two copies of this document.")
```
3. Click **SAVE & COMPILE**, giving the macro a name, such as “Print message.” Close the macro with **FILE, CLOSE**.
4. Return to editing your template. On the Template Property Bar, click **COPY/REMOVE OBJECT**. In the dialog that appears, click the down arrow in the Object Type window and select “Macros on Disk.” In the Source field, click the browse button and select the macro you just wrote and saved to disk. Click the **COPY>>** button to copy the macro to the template, then click **CLOSE**.
5. You should now be back in the template document. Now we’ll “trigger” this macro to play automatically immediately after you choose **FILE, PRINT...** (but before the print job starts). Click the **ASSOCIATE** button on the Template Property Bar. Next, click the **TRIGGERS** radio button, then select “Pre Print” in the left window. In the Macros drop list, choose the name of the macro (“Print message”) and click **OK**.
6. Save the template with **FILE, SAVE**. Test the operation of the macro by loading a new document based on the template, typing some text into it, and then printing it with **FILE, PRINT**. The message should appear as soon as you attempt to print the document.^[36]

The “same file” technique

An easy way to write (or paste) a template macro’s code directly into the template is to record an “empty” or trivial template macro first, such as by starting the template macro recording function, typing a space character, then stopping the recording. This will let you store an otherwise empty template macro. Then edit the template macro to remove any unwanted material, and write (or paste) the new macro commands into the template macro window. Here’s how.

[35] ▲ Note that there is not Application() command in this example. While WordPerfect inserts this command every time you record a macro, often it is not needed in a template macro since the program “knows” you are running a WordPerfect macro and which language code is required.

[36] ▲ See footnote 34.

1. While editing the template document, click **TOOLS, TEMPLATE MACRO, RECORD**. Enter a macro name, such as “Print message,” then click **RECORD**. Type a space with the **SPACEBAR**, then click on the Macro Toolbar’s **STOP** button (the leftmost button on the Macro Toolbar). You've just created a macro that types a single space—pretty useless, but it accomplished the task of storing the “Print message” macro in the template.
2. Click on **TOOLS, TEMPLATE MACRO, EDIT**. Double-click the macro's name (i.e., “Print message”) in the lower window. The macro editing window appears, with these two lines of code in it:

```
Application (WordPerfect; "WordPerfect"; Default!; "EN")
Type (Text: " ")
```

Delete the Type(Text: “ ”) command from the macro,^[37]

3. Type the same command(s) as you typed in step 2 of the separate file technique above; again, note the use of two semicolons:

```
MessageBox( ; "Reminder" ; "Print two copies of this document.")
```

Optionally, you could first copy some relevant macro code from another source to the Windows clipboard, then paste it into the template macro window with **EDIT, PASTE SPECIAL, UNFORMATTED TEXT**. This is a useful technique with lengthy macro code.

4. Click **SAVE & COMPILE**, then **FILE, CLOSE**.
5. You should now be back in the template document. As before, we’ll “trigger” this macro to play automatically when you attempt to print a document based on the template. Click the **ASSOCIATE** button on the Template Property Bar. Next, click the **TRIGGERS** radio button, then select “Pre Print” in the left window. In the Macros drop list, choose the name of the macro (“Print message”) and click **OK**.
6. Save the template with **FILE, SAVE**. Test the operation of the macro by loading a new document based on the template, typing some text into it, and then printing it with **FILE, PRINT**. The message should appear as soon as you attempt to print the document.^[38]

[37] ▲ You can delete the Application() command, too, but there is no harm in leaving it alone, and it might help if you have several Corel applications open.

[38] ▲ See footnote 34.

More complex template macros

Let's assume that you want to create a template macro to remove blank (empty) lines from the address block of a letter. For example, you might have a line in the middle of the address block that directs the letter to a particular person or department—the so-called “attention” line. Let's further assume that you want to automatically add a text label (“Attn:”) in front of any information inserted on that line. (The information will come from the fill-in Template Information dialog, created when you created the prompts for the address block.)

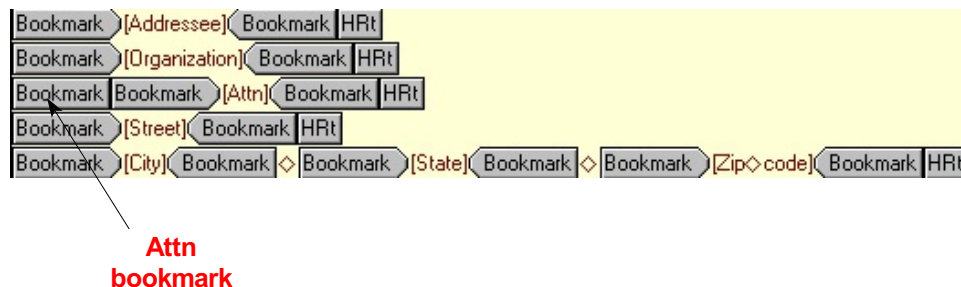
In other words, if you insert attention information in the Template Information dialog, your letter's address block might look like this:

The ABC Corporation
Attn: Accounting Department
123 Main Street
Anytown, Oregon 97001

If you *don't* insert any attention information, you want the address block to look like this (no empty lines, no stray labels):

The ABC Corporation
123 Main Street
Anytown, Oregon 97001

The template macro example code below assumes that you have previously placed a *single* bookmark named “Attn” (without quote marks) at the *beginning* of the appropriate line of the address block in your letter template, probably on the line after the Organization's name. You created this bookmark according to the instructions on page 19, and in Reveal Codes your template's address block might look something like this:



The idea is to get rid of the third line if it isn't used and close up all lines in the block, or if it is used, to add a text label to the line. Here's an example of some code to do the job: ^[39]

[39] ▲ As mentioned in the note on page 20, if you are using the Template Information dialog to gather and insert prompted information, which probably is the case in this example, you need to place a TemplateFill() command at the top of the macro so that the Template Information dialog will display. You also need to trigger the macro (see page 22) so that it plays automatically when you open the document.

```

BookmarkFind ("Attn") // Find the bookmark named Attn ...
SelectLineEnd () // ...select everything to the end of the line
vAttn = ?SelectedText // ...and store the selected text in a variable
If (vAttn = "") // If no text was stored (i.e., the variable is empty) ...
    PosLineVeryBeg () // ... go to the beginning of the line before all codes
    SelectLineDown () // ... drop down a line
    SelectLineVeryBegin () // ... select the whole thing
    DeleteCharNext () // ... then delete the selection
Else // Otherwise, if something was stored ...
    SelectOff () // ... turn select mode off
    BookmarkFind ("Attn") // ... find the bookmark
    Type ("Attn: ") // ... and add the text label
EndIf

```

The first three lines of the code simply locate, select, and store any text on the attention line in a variable (vAttn). Depending on whether anything was stored, the If...Endif statements will act accordingly. That is, if nothing was stored (i.e., no information was inserted on the line), the first part of the If statement is true and it will delete everything on the line, including the hard return (HRt) code at the end of the line. The resulting address lines will close up and no empty attention line will appear. If something was stored (i.e., information was inserted), the second part of the If statement (“Else . . .”) is true and it will type the “Attn: ” label in front of the information.

Other template macro ideas

Example #1 - Load a letterhead logo from a file on disk

This is a simpler version of Example #2 (which automates many aspects of a business letter), but it is one that performs a common task.

Many firms find that the names or phone numbers can change on the letterhead all too often. If, however, the letterhead logo (i.e., the text and graphics for your firm’s first-page letterhead stationery) is stored in an ordinary, separate (.WPD) file on disk – usually in the same folder as other templates – it can be kept up-to-date easily since it is in just one location, and it can be automatically inserted in a new document when you create a new document based on its template. Here’s how:

Open a new, blank file (**FILE, NEW**) and create the letterhead (only), positioned at the top of the page, perhaps in a centered text box that could include a graphic logo. Then save the file to disk.

Edit the existing (main) letterhead template (**FILE, NEW FROM PROJECT, <NAME OF TEMPLATE>, OPTIONS, EDIT WP TEMPLATE**) and record a simple template macro to insert the letterhead logo (see page 20 for creating template macros). For example, if the letterhead logo file is in the folder shown below, you could edit the template and then record a small template macro to insert the letterhead logo file directly into the template with **INSERT, FILE**. The macro will look something like this, depending on the file name and version of WordPerfect:

```

FileInsert (Filename: "C:\Documents and Settings\Administrator\Application
Data\Core\PerfectExpert\13\EN\Custom WP Templates\Letterhead.wpd"; AutoDetect: No!)

```

This command in the template macro should "point" to the actual path and filename of the file that contains the letterhead material.

Finally, in the letterhead template, trigger the template macro with the **POST NEW** trigger, then save the template. (See page 22 for more on triggering a template macro.)

Example #2 - The template macro in "My Letterhead"

The code snippet on page 27 is part of the startup template macro in the author's "My letterhead" template. (See Appendix A, page 35ff. The entire macro is an example of a multi-purpose macro. Feel free to use the code for your own purposes.

The macro's operation is simple: When a new document is opened based on that letterhead template, the template macro will immediately display a Template Information dialog, and when the user is done filling in the dialog with information specific to the letter and clicks **OK**, it will insert that information. Then it will automatically insert the current date and other material, and finish up by performing some cleanup tasks.

Note that for any of the code segments to function properly, you must have used the same prompts as the author's letterhead template. Otherwise, you'll need to modify your own template macro code so it can find the various bookmarks you've set in the document.

Example #3 - Display "pick lists" for user input choices

Suppose you want to have a template macro display a custom dialog (either when the document opens or at print time) with one or more lists containing several names or other items per list. Unlike in our previous bookmark example, you want the user to pick an item from a *list of several items* and have the macro insert the same item(s) of information in one or more places in a document, such as a standard company form or contract. This is a *many-choices-to-one-location* or *many-choices-to-many-locations* relationship between input data sources and the locations in the document where the data should appear.

You might also want the user to press radio buttons or enable check boxes to instruct WordPerfect to do something, such as insert an external file or change to a different page definition to select a different printer tray. Or you might want to present a blank field to allow the user to type a message that will be inserted at a specific location in the document.

This will require a custom dialog^[40] and some code that will either find single bookmarks (with BookmarkFind) or find paired bookmarks (with BookmarkBlock) surrounding text markers similar to those created with Prompt Builder, and then insert

[40] ▲ A custom dialog can be written using the Dialog Editor on the macro toolbar, or with standard PerfectScript commands such as DialogDefine, DialogShow, etc. Creating dialogs is beyond the scope of this paper, but a good starting point is Doug Loudenback's *A Common Person's Macro Manual*, downloadable in a ZIP file at <https://wptoolbox.com/otherauthors.html#dougloudenback>.

the information from the custom dialog. Or the code could search for “text placeholders” (similar in function to bookmarks but created with unique, plain text characters such as “<<CaseID>>”) and replace them with the information from the custom dialog. ^[41]

The custom dialog creation commands can be coded into the macro *before* the TemplateFill() command or *after* it, depending on when you want the dialog to be displayed to the user.

Example #4 - “If...Then” template macros

If you have placed paired prompts in the template with Prompt Builder, and if data from the Template Information dialog has been inserted at those bookmarked locations, the data can be used by a start-up template macro to do things based on the particular information inserted. In other words, it is possible to use *conditional* or *branching* logic in the template macro, based on particular information the user enters when the document first loads. ^[42]

For example, suppose a template has a paired bookmark in it, internally numbered by WordPerfect as “12” as part of a series of bookmarks. Further suppose the prompt (i.e., text placeholder) between the paired codes for that bookmark has already been replaced with some information from the Template Information dialog (e.g., a city name such as Chicago). Before the template macro exits, the information then can be found, selected, ^[43] stored, and used by the template macro with an If() or Switch() statement, something like this:

```
BookmarkBlock ("12")           // (finds and selects the paired bookmark and its text)
vSel := ?SelectedText          // (stores the selected text)
SelectOff ()                   // (turns off select mode [if desired])
If (ToUpper (vSel = "CHICAGO")) // (sets up the logic statement)
    // ...then do something here... // (macro does something or branches to another line)
Endif
```

[41] ▲ As implied by the next example (“If...Then” template macros), in addition to—or as a substitute for—*replacing* items inside paired bookmarks with other text, you could have a macro *remove* certain blocks of text based on user choices in a pop-up dialog. This custom dialog could be displayed by a template macro or even by a regular macro that is played manually in the resulting document at any later time.

However, you would not create these paired bookmarks with Prompt Builder which, as mentioned, internally numbers each bookmark. Instead, in the template itself, you would simply select the text that you might (or might not) want removed with a macro, then use the manual method to create a regular bookmark by clicking Tools, Bookmark, Create. Give the bookmark a unique name (e.g., “bm#1”), make sure the “Selected Bookmark” checkbox is enabled, and click OK. The resulting bookmark can then be located by the macro and the text deleted with two commands, BookmarkBlock(“bm#1”) and SelectDelete.

[Be sure to use OnError() before each BookmarkBlock() command to direct the flow of macro processing to a Label, so that if such a bookmark paired code is not found the macro will jump to that Label. (Failure of a macro to find a Bookmark generates an Error condition, not a NotFound condition.) This is especially true if you plan to use an ordinary user-playable file macro to find and delete text this way, since other editing in a document could cause accidental removal of one or more of these paired bookmark codes. Also be sure you don’t have any other bookmarks or other required format codes *between* the paired bookmark codes, or the macro might remove them, too. Using Reveal Codes during editing, and thoroughly testing the macro, are especially important here.]

[42] ▲ This probably will work more reliably if the user is instructed to enter the text carefully, or use only letters or numbers such as a city name or a Case ID number.

[43] ▲ Once found and selected with BookmarkBlock, the selected text between the bookmark codes can be further processed. For example, it could be converted to UPPER case with the macro command, ConvertCaseUppercase.

After processing or branching, the paired bookmark codes and the information between them can be deleted at that location, if desired:

```
BookmarkBlock ("12")           // (finds and selects the paired bookmark and its text)
SelectDelete ()                // (delete the blocked text)
BookmarkDelete("12")          // (delete the bookmark codes)
```

Loading custom template documents

The standard method of loading a new, empty document based on a custom (i.e., non-default) template is to use **FILE, NEW FROM PROJECT** (or **FILE, NEW** in WordPerfect 8), then select the category and name of the Project, then click **CREATE**.

However, there are easier and faster ways, especially if you frequently use the same template-generated custom documents. One method uses a new toolbar button, and the other uses a macro (which you can record yourself).

Use a toolbar button that points to the template

This method uses a toolbar button on your default or custom toolbar that directly “points” to a template.^[44] You need only click the toolbar button and the document loads.

Note, however, that this works best with only one version of WordPerfect installed on your system. The method uses the most recently associated WordPerfect filename extension for templates in Windows Explorer. Therefore, the method loads *that* version of WordPerfect (e.g., WordPerfect 12) along with the document—even if you start from an earlier version (e.g., WordPerfect 8). This might be confusing, and can also take up system resources on some systems.

But if you have only one version of WordPerfect installed, the following procedure is easy to use and the result is an icon that loads a document in a jiffy:

1. Right click on the Toolbar you want to use, and select **EDIT** from the context menu.
2. Click on the **PROGRAMS** tab (not the Macros tab), and click the **ADD PROGRAM** button.
3. In the Open File window, change the File type (in the lower left corner) to “WP Templates (.wpt).”
4. In the browse window, locate the desired template on your hard drive, select it, and click **OPEN**. A new icon will appear on your toolbar.

[44] ▲ You can create a custom toolbar for just such purposes: Click Tools, Settings, then double-click Customize. Under the Toolbars tab, click Create. (When created, be sure to toggle it “on” back in the Available toolbars window of the Customize Settings dialog.) The new toolbar will be saved as part of your default template, but it can be copied to other templates using the Copy/Remove... button on the target template’s Template Property Bar.

Note that you can also create a desktop shortcut to “point” to the template file (.WPT). Clicking the desktop shortcut will open WordPerfect and load a new document based on that template. However, unlike in WordPerfect 9 and earlier versions, in WordPerfect 10, 11, and 12 (at least) any startup template macros (i.e., a template macro associated with the Post New trigger) will not automatically play *unless WordPerfect is already open*.

5. Right-click the icon, and select **CUSTOMIZE**. Change the Button text, Quick Tip, and Image, as desired.
6. Click **OK** to save your changes, and click **OK** to back out of all dialogs and close the Toolbar Editor.

Use a *macro* to load a new template document

Another method is to use a very simple “loader” macro that can be played from the toolbar or a menu item. The macro selects the template and loads a new document based on it. (There are no differences in terms of usage or speed of execution between this method and the preceding one.)

If you need access to only one or two frequently used templates, you might want to create just one or two buttons on a Toolbar to play these loader macros. On the other hand, if you have several templates—such as letters, envelopes, and invoice forms—it may be better to create a drop-down menu on the top menu bar, and populate it with items that play the various macros. Both techniques are discussed below.

Create the macro first —

1. *In WordPerfect 9 and later:* ^[45] Open a blank document with **FILE, NEW**.
2. Click **TOOLS, MACRO, RECORD**. Give the macro a name (e.g., *Load my letterhead*), then click **RECORD**. The Macro Property Bar appears just above the main document area.
3. Click **FILE, NEW FROM PROJECT**. Select the template or Project you want to use, then click the **CREATE** button. The document will load on screen. (If a Template Information dialog appears, you would normally fill in its fields; but for the procedure here, click **CANCEL** to remove it from the screen. At this point you only want to load a new document based on the template and not do anything else that might get recorded.)
4. Stop the macro recording by clicking the far left button (■) on the Macro Property Bar (its Tool Tip says “Stop macro play or record”).
5. Close the document on screen with **FILE, CLOSE** (i.e., don’t save it).
6. Test the macro with **TOOLS, MACRO, PLAY**. The macro should immediately load the document in the usual manner.
7. Repeat the previous steps to create a differently named macro for each template that you want to access this way.

[45] ▲ **IMPORTANT:** In some versions of WordPerfect (e.g., version 8) this method of recording a loader macro might not work to produce a macro. If that happens — or you simply want a faster method of creating such macros — you can write a simple one-line macro containing the `TemplateSelect()` command. Be sure to use the full drive\path\<template name>, and enclose everything in double quotes. For example:

```
TemplateSelect ("C:\Documents and Settings\Administrator\Templates\My Letterhead.wpt")
```

— then play it with a mouse click

As mentioned above, at this point you can either create a toolbar button to play each macro,^[46] or you can create a drop down menu on the top menu bar with selections that play your macros. Either or both of these methods can be used to automate daily tasks by playing macros.

Method A: Create a toolbar button to play the macro

1. Right-click anywhere on the toolbar, then click **EDIT**. The Toolbar Editor appears.
2. Click the **MACROS** tab, then click the **ADD MACRO** button. From the Select Macro window, choose the loader macro you created above, then click Select. Usually, you can answer the “Save macro with full path?” question with “No” since WordPerfect will automatically look for the macro in one of the two folders specified in Tools, Environment, Files, Merge/Macro.
3. A small gray “cassette” icon appears on your toolbar. Right-click on it and choose **CUSTOMIZE**. Give the button a name in the **BUTTON TEXT** field and a **QUICK TIP** to display when you pass your mouse over it. Click on the Image **EDIT** button, then click Clear to remove the default icon image. Then use the tools to create your own custom icon.^[47] Click **OK** all the way back to the document window. Note that whenever the Toolbar Editor is on screen you can move a button by dragging it, or delete it by dragging it from the toolbar.

Method B: Create a drop down menu to play macros

Alternatively, you can create a drop down menu with selections that play your loader macros. This method can be used to play *any* macro on your disk.

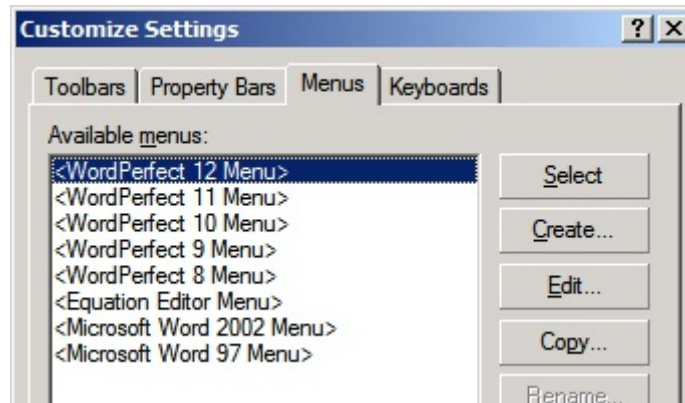
This is a similar method to the previous one, since it also uses a loader macro to load the template document. However, instead of toolbar buttons, you create a menu. Each item on the menu plays a different loader macro.

[46] ▲ See footnote 44.

[47] ▲ If you are “artistically challenged” and want to have something more than a picture of an audiocassette on the toolbar button that plays your macros (i.e., the default icon that appears whenever you add any new macro), here’s one way to do it. [Note: This applies to WordPerfect X6 and earlier. The WPX7 toolbar editor has been replaced with a graphic image import button.] Since WordPerfect is full of features that you can add to a toolbar with the Toolbar Editor’s Features tab, these already have pre-drawn icons associated with them. You can copy any of their icon images to your new macro’s toolbar button and modify it if desired. Here’s how:

Scroll through the features, select one of them and add it (temporarily) to the toolbar, then edit this temporary icon (right-click it, then click Customize, Edit) and click Copy to copy the image. Click OK back to the Toolbar Editor, and edit your new macro’s icon. Clear the gray audiocassette image, and paste the new image into it, making any desired changes, then click OK to get back to the Toolbar Editor. Finally, with the Toolbar Editor on screen, right-click the temporary icon and select Delete.

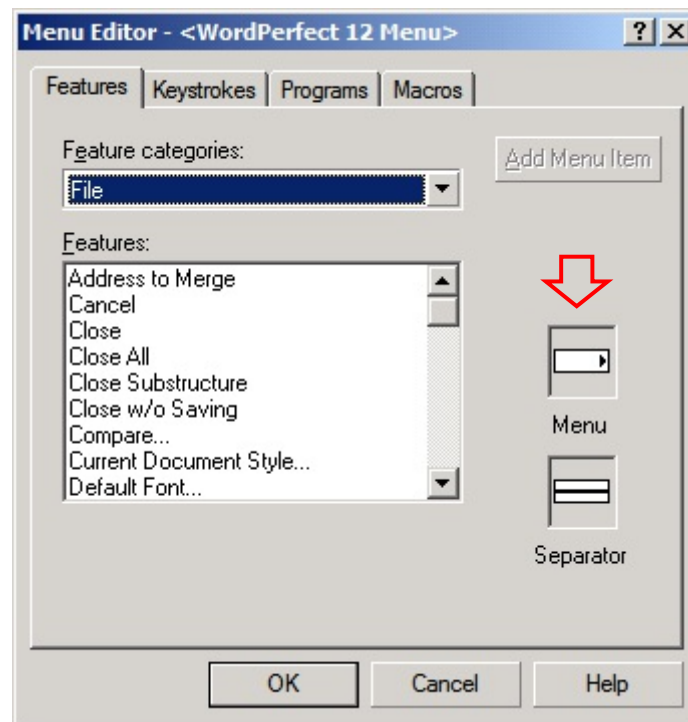
1. Right-click on the top menu (the one with **FILE, EDIT, VIEW**, etc.). Click on **SETTINGS** from the context menu. In the Customize Settings window that pops up, click on




the **MENUS** tab and choose menu you want to use in the Available menus list.

Generally this is the <WordPerfect X Menu> (where “X” is the WordPerfect program version number that is loaded), unless you have previously created a customized menu—in which case, click on that one. Click **EDIT** to bring up the Menu Editor.

2. In the Menu Editor window you’ll see an indented box on the right labeled “Menu” with an icon inside it:



Drag the Menu icon from this box up to WordPerfect's top menu, to the right side of other menu items.

3. Put your cursor over this new menu item (which is simply labeled "Menu"), and when it is selected—it turns dark blue—double click it. This will allow you to change the name in the Menu Item field to "Letterheads," "Stationery," or something equally useful, and add a floating description (Quick Tip). You can type an ampersand (&) immediately before any letter in the name that you want to use as a mnemonic, which will select the macro if you use the **ALT+<LETTER>** key combination. Click **OK** when you are done entering the name.

4. While the Menu Editor window is still open, click on the **MACROS** tab at the top, and then click the **ADD MACROS** button. Select the macro that loads your letterhead or other desired template. Click **SELECT** to close the Select Macro window, and click anywhere in a blank area of the Menu Editor window to make it active again. You'll notice the macro's name was placed on the top menu line, but *next to* your new menu item. You'll want to put it *under* the new item, and change its name:
5. Place your cursor over the new macro's name on the top menu until it turns blue (i.e., it's now selected), then *drag it* with your left mouse button over to the Letterhead menu item — *then drag it down* to the blank (gray) area that pops up just under the new menu item and *drop it there*. (Sort of "go west, then go south" with your mouse, while holding down the left button.) This is where the first menu selection will appear.
 - a. *Note:* For additional items you can drop the name either below or above any existing menu items. A heavy horizontal line will appear to guide your placement. You also can add a separator line between menu items by dragging and dropping it from the Menu Editor window.
6. Rename the menu selection by highlighting it with your mouse. Double click the macro's name and rename it in the pop up box. You can type an ampersand (&) immediately before any letter in the name that you want to use as a mnemonic and that will select the macro if you use the **ALT+<LETTER>** combination.
7. Click **OK** when you are done to close the Menu Editor.
8. Click **CLOSE** in the Customize Settings window.

□ □ □

Appendix A - The template macro in “My letterhead”^[48]

The example template macro code below came from an automated letterhead template. The template macro is triggered to play immediately at start up, and it accomplishes several things:

1. It (optionally) pops up a message box. [The message can be changed by editing the `MessageBox()` command.]
2. It displays the Template Information dialog on start up to get prompted information from the user.
3. It finds several single (i.e., not paired) bookmarks in the template and inserts information adjacent to them, or deletes the lines they are on if no information was input.
4. It (optionally) adds a path and filename “stamp” to the bottom of the document.
5. It deletes any blank lines in the address block.
6. Finally, it places the cursor at a specific spot in the document at the point where the user normally wants to begin typing.

While this macro may appear complex at first, much of the code is simple and repetitive. The lines preceded by double slash marks are comment lines and are ignored by the macro.

Note the two forms of the `BookmarkFind()` commands, which search for various bookmarks. (The template contains twelve bookmarks, two of them used again in Header A.) One form of the command finds bookmarks by their *names* and the other by the *numbers* assigned to them by Prompt Builder.^[49]

Also see *Example #3: Display “pick lists” for user input choices* (page 28) and *Example #4: “If...Then” template macros* (page 29).

[48] ▲ The letterhead template is available as LETTERHD.ZIP at <https://wptoolbox.com/library.html>

[49] ▲ For how WordPerfect internally numbers bookmarks, how you can find the bookmarks' numbers, and some cautions about these bookmarks, see page 15.

```

//  STARTUP macro for the "My letterhead.wpt" template -

//  Note: If you have two different language versions of WP on your computer, you will need to add a semicolon
//  and "EN" (or whatever language code you require) to the end of the parameters in the Application command
//  below. Also, if you use a N.A. English version of WordPerfect 7, insert a semicolon and "US" - like this:
//  Application (WordPerfect; "WordPerfect"; Default!; "US")

Application (WordPerfect; "WordPerfect"; Default!)
OnError(End@) // If an error occurs, the macro jumps to the end and exits

//  The next command is optional. Remove any leading comment marks ("//") to turn the command on. (The
//  code NTOC(0F90Ah) enters a hard return in the dialog box.) It displays a message to remind users to enter
//  the recipient's name in the address book -

// MessageBox("Reminder";"Add new recipients to the address book with the"+NTOC(0F90Ah)+"Recipient
// Information button in the next dialog box.")

//  The next command, TemplateFill(), is needed to display the Template Information dialog when the
//  document first loads -

TemplateFill ()

//  The next code segments find several single, manually entered bookmarks (Date, DateInTable, Attn,
//  Delivery, Re, ReInTable, Enclosure, cc, and Filename) and insert information adjacent to them (or delete the
//  lines if no information was input)

BookmarkFind ("Date") // Find the bookmark named Date ...
DateText () // Insert the date as a static text string, not as a code
BookmarkFind ("DateInTable") // Find the bookmark named DateInTable ...
DateText () // Insert the date as a static text string, not as a code

BookmarkFind ("Attn") // Find the bookmark named Attn ...
SelectLineEnd () // ...select everything to the end of the line
vAttn = ?SelectedText // ...store the selected text in a variable

If (vAttn = "") // If no text was stored (variable is empty)...
  PosLineVeryBeg () // ... go to the beginning of the line
  SelectLineDown () // ... drop down a line
  SelectLineVeryBegin () // ... select the whole thing
  DeleteCharNext () // ... then delete the selection
Else // Otherwise -
  SelectOff () // Turn select mode off
  BookmarkFind ("Attn") // Find the bookmark
  Type ("Attn: ") // Add a text label
EndIf

BookmarkFind ("Delivery") // Find the bookmark named Delivery
SelectLineEnd ()
If (?SelectedText = "")
  PosLineVeryBeg ()
  SelectLineDown ()
  SelectLineVeryBegin ()
  DeleteCharNext ()
EndIf
SelectOff ()

BookmarkFind ("Re") // Find the bookmark named Re
SelectLineEnd ()
vRe = ?SelectedText
If (vRe = "")
  PosLineVeryBeg ()
  SelectLineDown ()
  SelectLineVeryBegin ()
  DeleteCharNext ()

```

```

Else
  SelectOff ()
  BookmarkFind ("Re")
  AttributeAppearanceOn (Bold!) // Make the "Re:" label bold
  Type ("Re: ")
  AttributeAppearanceOff (Bold!)
EndIf

BookmarkFind ("ReInTable") // Find the bookmark named ReInTable
If (vRe <> "")
  AttributeAppearanceOn (Bold!)
  Type ("Re: ")
  AttributeAppearanceOff (Bold!)
EndIf

BookmarkFind ("Enclosure") // Find the bookmark named Enclosure
SelectLineEnd ()
If (?SelectedText = "")
  PosLineVeryBeg ()
  SelectLineDown ()
  SelectLineVeryBegin ()
  DeleteCharNext ()
Else
  SelectOff ()
  BookmarkFind ("Enclosure")
  Type ("Enc: ")
EndIf

BookmarkFind ("cc") // Find the bookmark named cc
SelectLineEnd ()
If (?SelectedText = "")
  PosLineVeryBeg ()
  SelectLineDown ()
  SelectLineVeryBegin ()
  DeleteCharNext ()
Else
  SelectOff ()
  BookmarkFind ("cc")
  Type ("cc: ")
EndIf

// Delete or comment out ("///") the next 5 lines if you do NOT want the filename to appear at the bottom of the
// letter, or if you use the another template macro to produce a file stamp (i.e., use one or the other method, but
// not both). Note that the file would have to be saved before any filename will display in the document.

//BookmarkFind ("Filename") // The bookmark named Filename is found at the very end of the letter, after the
// closing
//Font ("Arial Regular") // Set the file stamp's font
//AttributeRelativeSize (Fine!) // Set the file stamp's relative size to Fine
//InsertFilenameWithPath () // Insert the file name with the full path
//AttributeRelativeSize (NormalSize!) // Return the size to normal

// The next code segments find paired bookmarks (by their numbers) surrounding prompts, and will remove
// any blank lines in the addressee block – assuming the bookmarks in the address block lines are
// numbered "2" through "8" – by removing both the line and the HRt at the end of the line -

BookmarkFind ("2") // This is the bookmark next to Addressee's name
PosLineVeryBeg ()
SelectLineEnd ()
If (?SelectedText = "") // If no name, delete the entire line if empty -
  PosLineVeryBeg ()
  SelectLineDown ()
  SelectLineVeryBegin ()
  DeleteCharNext ()
EndIf

```

```

BookmarkFind ("3") // This is the bookmark next to Organization
PosLineVeryBeg ()
SelectLineEnd ()
If (?SelectedText = "") // If no Organization, delete line if empty -
    PosLineVeryBeg ()
    SelectLineDown ()
    SelectLineVeryBegin ()
    DeleteCharNext ()
EndIf

// (Bookmark 4 is for Attn, and is processed above)

BookmarkFind ("5") // This is the bookmark next to Street
PosLineVeryBeg ()
SelectLineEnd ()
If (?SelectedText = "") // If no street, delete line if empty -
    PosLineVeryBeg ()
    SelectLineDown ()
    SelectLineVeryBegin ()
    DeleteCharNext ()
EndIf

BookmarkFind ("6") // This is the bookmark next to City
PosLineVeryBeg ()
SelectLineEnd ()
If (?SelectedText = "") // If no City, State, or Zip, delete line if empty -
    PosLineVeryBeg ()
    SelectLineDown ()
    SelectLineVeryBegin ()
    DeleteCharNext ()
EndIf

// Here's some code that is activated if you place the Zip code prompt on a separate line below the rest of the
// address block. If the Zip code prompt is placed on the same line as the City and State prompts (the usual
// case), and some City (or State) information is input, it won't delete anything since that line will not be
// "empty" -

BookmarkFind ("8") // This is the bookmark next to the Zip code
PosLineVeryBeg ()
SelectLineEnd ()
If (?SelectedText = "") // If no Zip, delete line if empty -
    PosLineVeryBeg ()
    SelectLineDown ()
    SelectLineVeryBegin ()
    DeleteCharNext ()
EndIf

// Turn off Select Mode, if on -
SelectOff()

// Go to the location in the body text where you want to start typing the letter -
BookmarkFind ("Text") // This is the bookmark named Text

Label(End@)
Quit
// End of macro code

```

Appendix B - Triggering a template macro

[From: "Trigger" a macro from inside a template to play automatically at specific times (events), such as when you open a new document or before you print a document" at <https://wptoolbox.com/tips/Triggers.html>.]

“Trigger” a macro from inside a template to play automatically at specific times (events), such as when you open a new document or before you print a document.

WordPerfect 9's online Help (F1, Index tab, “Triggering macros from templates”) says this:

You can play a macro as you use a template. To do this, record a template macro, then associate the macro with a trigger to start the macro.

For example, if you use a template to create letters, you can set a PRE PRINT trigger in the template that runs a spell-check macro. The macro then lets you spell-check your letters just before they print.

You can use the macro triggers in the following table:

TRIGGER	DESCRIPTION
POST CLOSE	Runs in the active window after you close a window.
POST NEW	Runs after you open a window by clicking File, New.
POST OPEN	Runs after you open a window by clicking File, Open.
POST PRINT	Runs after you send a print job to the printer.
POST STARTUP	Runs when you start WordPerfect. The macro must be associated with the default template.
POST SWITCH DOC	Runs in the active window after you switch from another window.
POST TABLES	Runs after you create a table structure in a document, but before you enter data in the table.
PRE CLOSE	Runs before you close the active window by clicking File, Close.
PRE NEW	Runs in the active window before you open another window by clicking File, New.
PRE OPEN	Runs in the active window before you open another window by clicking File, Open.
PRE PRINT	Runs after you click File, Print, but before the print job is generated.
PRE SWITCH DOC	Runs in the active window before you switch to another window.
PRE TABLES	Runs after you insert a table by clicking Insert, Table, but before the table structure is created in the current window.

Note that you can have only *one* macro associated with a particular trigger.

The two most common triggers used with simple template macros are **Post New** (play on creating a *new* document based on the template) and **Pre Print** (play at *print* time).

In addition to recording a template macro (i.e., recorded while editing the template) you can use a *previously recorded macro* stored on your disk. Here's how.

Assuming the macro is located in your default macros folder as shown in WordPerfect's **TOOLS, SETTINGS FILES, MERGE/MACRO**. Then:

- Edit the template on which the document will be based by (usually) clicking on **FILE, NEW FROM PROJECT** -- then right-clicking the template's name and choosing **EDIT WP TEMPLATE**. (The **default template** on which all new documents are based is (blandly) named "Create a new document.") The template document should now be on screen. [This is the point at which you could record a template macro with **TOOLS, TEMPLATE MACRO, RECORD**. However, here we are going to use a macro already stored on disk.]
- On the Template Property Bar, click **COPY/REMOVE OBJECT**. In the dialog that appears, click the **DOWN ARROW** in the Object Type window and select "Macros on Disk." In the Source field, click the **BROWSE** button and select the macro by name. Click the **COPY>>** button to copy the macro to the template, then click **CLOSE**.
- You should now be back in the template document. Now we'll “trigger” this macro to play automatically. For example, we can make it play immediately after you choose **FILE, PRINT**. (but before the print job starts). Here's how:
- Click the **ASSOCIATE** button on the Template Property Bar. Next, click the **TRIGGERS** radio button, then select "Pre Print" in the left window. In the Macros drop list, choose the name of the macro and click **OK**.
- Save the template with **FILE, SAVE**. Test the operation of the macro by loading a new document based on the template, typing some text into it, and then printing it with **FILE, PRINT**.

Notes

- *Older* documents that were based on the template (or some other template) will not automatically play the macro—the macro didn't exist inside those templates at the time the documents were created—so you may also want to assign the macro to a toolbar **button** or **keystroke** combination. You can then manually play it as needed.
- Ordinarily, only *template macros* can be associated with a trigger. Macros on disk that are not already part of that template cannot be called (i.e., started) by a template trigger, but they can be copied into the template first and then associated with a trigger, as explained above. *However*, if a disk macro already has been included on a toolbar (or menu, or associated with a keystroke) *in that template*, then it will be available to associate with a trigger.
- If you want to use the Post New trigger and notice something already associated with it called “<dofiller>” here's an explanation (admittedly a bit convoluted):

WordPerfect's DoFiller feature is normally activated by Prompt Builder and then associated with the Post New trigger in the template. DoFiller automatically fills in fields in a template that are linked to your Personal Information. But if you use a “startup” template macro associated with Post New—and since you can have only one

template macro associated with a trigger—you will need the `TemplateFill()` command to activate DoFiller as well as display the Template Information dialog, if required, to get prompted information from the user. You can place the `TemplateFill()` command anywhere in the “startup” template macro where you want the Template Information dialog to appear, but most often it is used as the first command. [Think of `TemplateFill()` as the macro command equivalent of the DoFiller feature.] . . .

- If you plan to *share* documents containing template macros with other WordPerfect users, take note of this:

If a template containing a template macro is set to be triggered by some event (such as opening a new document based on that template), note that the template macro will only trigger in a document (.WPD) if the template (.WPT) is present, either in the original template folder or in the default (parent) template folder. If WordPerfect cannot locate the template, the template macro will not play even though it is embedded in the document itself, since the trigger specification is stored in the template, not in the resulting document.

This can cause problems when sharing such automated documents with other WordPerfect users. Make sure they also have the template that spawned the document and it is located in the same relative folder as the original. [This might explain why template macros fail to automatically play in some documents if the original template has been renamed, relocated, or removed.]

- A corollary of the previous note: If you change or add a template macro in a template, all documents that were previously created with that particular template should -- if the template is present -- be able to run the revised (or new) template macro if they are set up to be triggered (or if they are manually played through the menu).
- If a template macro does not run automatically when loading a new document or WordPerfect Project, or when some other expected trigger event occurs, template macros may have been turned off by a user. According to the Corel (Support) Knowledge Base,

“This issue can occur if template macros have been disabled. When opening one of [these automated] files, you will get a dialog box entitled ‘PerfectScript’, explaining that this document has a macro, and that macros can contain malicious code. If the ‘Do not show this message again’ box is checked, and ‘Yes’ is selected, that prompt will not come up again, and the macros will not run.” For solutions, see [here](#).

- The trigger associations for a given template can also be determined with a macro. See Klaus Pfeiffer's macro posted (and downloadable) on WordPerfect Universe [here](#).

Appendix C - Did Windows hide some of your filename extensions?

There are several references in this paper to file name extension such as **.wpd** and **wpt**.

If (for example) you cannot find a WordPerfect template on disk with Windows Explorer or My Computer (or just Computer in recent versions), it might be because of a default setting in Windows itself that prevents you from seeing certain files. Here's what to do.

Windows XP:

1. Open Windows Explorer (click the Windows **START** button, then click on **PROGRAMS**, **ACCESSORIES**, **WINDOWS EXPLORER**, or just right-click on the Windows **START** button to open Windows Explorer).
2. Click on the **TOOLS** menu; select "Folder Options..."; click on the **VIEW** tab; then -
(1) select (check) "Show hidden files and folders";
(2) *deselect (uncheck)* the "Hide..." options; then
click **OK**. You should now be able to find the file.

Windows Vista and Windows 7:

1. Open Windows Explorer or Computer (click the Windows **START** button then click on either **DOCUMENTS** or **COMPUTER**; or just right-click on the Windows **START** button).
2. If you don't see a **FILE/VIEW..HELP** menu on that dialog, press the **ALT** key (to display the menu), then click **TOOLS, FOLDER OPTIONS**; on the **VIEW** tab, *uncheck* the box that says "Hide extensions for known file types"; then
click **OK**. You should now be able to find the file.

Windows 10 (and possibly Windows 8):

1. Open Settings (a gear-shaped icon on the **START** menu, and on the **NOTIFICATIONS** screen).
2. At the top of the Settings window, type either "Folder Options" or "File Explorer Options" in the search field. In the drop list that appears, choose **FILE EXPLORER OPTIONS**. *Uncheck* the box that says "Hide extensions for known file types"; then
click **OK**. You should now be able to find the file.

□ □ □