



WordPerfect® X7

Going Further with
PDF Document Automation
by Gordon McComb

Going Further with PDF Document Automation

Gordon McComb

The Portable Document Format – most of us know it better as PDF – has become the defacto standard in law offices the world over. Just about everyone can open and read a PDF file. And being essentially a read-only format, second only to the printed page, what's displayed in the PDF screen is what the original author intended.

For years you've been able to publish PDF documents right from inside WordPerfect. But now with the enhanced PDF features in WordPerfect X7, you can automate more of the process using macros.

In the first article of this two-part series, I showed you the basics of PDF creation with WordPerfect X7, both using the built-in Publish to PDF feature (File->Publish to PDF), and a basic macro for generating a finished PDF file.

In this second part, we'll delve more deeply into the extended PDF automation capabilities of WordPerfect X7. You'll learn how to specify PDF style settings right from inside the macro, such as assigning the PDF version, or setting passwords and permissions. With just some simple macro commands, you'll learn how to make the entire PDF publishing process completely transparent to your users.

To best understand the concepts and techniques in this article, be sure to read the previous installment, *Getting Started with PDF Document Automation*.

The New and Improved Basic PDF Macro

WordPerfect macros for publishing PDF documents aren't complex. In fact, they're surprisingly simple, instead relying on WordPerfect to do all the heavy lifting. The macro can be as uncomplicated as this three-liner:

```
Application (WordPerfect; "WordPerfect"; Default!; "EN")
PDFStylesSetActive("MyStyle")
PdfDlg (?PathDocument + "MyFirstPDF.pdf")
```

I talked about this macro in the previous article, but here's a quick recap. The two critical commands are

- *PDFStylesSetActive* specifies the active ("current") style for publishing. Styles store all the settings you want when generating the document.
- *PdfDlg* creates a new PDF named *MyFirstPDF.pdf*, storing it in the WordPerfect documents directory.

This macro serves as a good template for creating more PDF automation solutions. One of the easiest changes is altering the *PdfDlg* command so that WordPerfect will automatically overwrite any previously existing PDF document of the same name, should one exist. Without this option, users are presented with a prompt that asks if they want to overwrite the file, assign a new name, or cancel.

To squelch the overwrite prompt, merely change the last line to:

```
PdfDlg (?PathDocument + "MyFirstPDF.pdf"; Options:ReplaceAlways!)
```

Now, when the macro runs, it'll always replace the *MyFirstPDF.pdf* document automatically, without prompting.

Setting PDF Styles From Inside a Macro

The main power of PDF publishing in WordPerfect X7 comes from manipulating the style options normally accessible from the Publish to PDF dialog box. These styles control the settings used to produce the PDF file.

Commonly used styles are:

- *PDF version.* Over the years new versions of PDF have supported new features of the Acrobat reader. For the widest document compatibility, set to Version 1.2, the oldest. But you should be aware that some companies and government agencies won't accept this version of PDF due to the wide variation of content allowed in them. You'll want to choose PDF/A-1a or PDF/A-1b formats instead.
- *Starting display.* PDF documents can be encoded to instruct the reader software to display the contents in full-screen mode. This is typically used in kiosks or other controlled environments. Nearly all PDF reader software will display a warning when opening a document in full-screen mode, as a security measure.
- *Optimize for Web.* This setting applies compression and other features to allow faster access from a Web page. The option has no bearing for files accessed from your computer desktop. Unless you're specifically producing PDFs for the Web leave this option off.
- *Generate log file.* WordPerfect can create a log of all your PDF publishing jobs. The main use of the log is as a troubleshooting aid. Along with controlling whether a log is created, you can specify where the log file – which is a standard text document – is stored.
- *Password for opening the document.* You can control who can open the document by specifying a password. All your PDF documents can have the same password, or with fancier macro coding, you can assign a different password to each user.
- *Password for accessing features* of the document, such as editing. PDF supports varying levels of access: you can control editing, printing, and copy/paste of text to only those users who provide the password you have specified.
- *Specify permissions for all users.* Separate or in combination with the permissions password above, you can specify how allowed users may access content in your PDF. You can allow/disallow printing, editing, and copying (for pasting text into another document).
- *Open PDF file after saving.* You may specify whether the document you just generated is automatically opened in your default PDF reader.

Creating a Testing Macro for Experimenting

You can see how the various PDF style settings work by using a test platform macro. The macro lets you fiddle with different settings, and you can see what each command does.

There are only a handful of macro commands related to PDF publishing, and just four that are really important that you must know about:

- *PDFStylesSave* – Creates a style set. Settings you modify in the macro are saved into this style.
- *PDFStylesSetActive* – Activates a specific style that's used when generating the PDF document. The style is usually the same one you created with *PDFStylesSave*.
- *PDFStyleSetSettingValue* – Sets a PDF style option. While it's just one macro command, with it you specify numerous style options by using different command parameters.
- *PdfDlg* – Generates the finished PDF file.

Here's a testing platform macro you can start with. For your convenience, this macro is also available as a separate download from the Corel Web site – I've named it *PDFTestingPlatform.wcm*. Unzip the file, and place into your WordPerfect macros folder (shown in Tools->Macro->Play).

```
Application (WordPerfect; "WordPerfect"; Default!; "EN")

// 1. Create and activate a new style
StyleName := "MyNewStyle"
PDFStylesSave(StyleName)
PDFStylesSetActive(StyleName)

// 2. Modify some settings in the newly created style
PDFStylesSetSettingValue(StyleName; Setting:PDFVersion!; SettingValue:PDFVersion12!;)
PDFStylesSetSettingValue (""; Setting:OptimizeForWeb!; SettingValue:False!)
PDFStylesSetSettingValue (""; Setting:GenerateLogFile!; SettingValue:False!)
PDFStylesSetSettingValue (""; Setting:OpenPDFAfterSaving!; SettingValue:True!)

// 3. Save the changes you made to the style
PDFStylesSave(StyleName)

// 4. Publish to PDF
Success := PdfDlg (?PathDocument + "MySecondPDF.pdf"; Options:ReplaceAlways!)
```

Here's what the macro does:

1. In step #1, a new PDF publishing style is defined. I've called it *MyNewStyle* (note the quotes around the style name – very important!) Because this style is referenced several times in the macro, I've stored the name in a variable (*StyleName*). The variable is then used in place of the actual style name throughout the macro.
2. In step 2, I've specified four commonly used options. The first specifies the lowest PDF version, version 1.2; this maximizes compatibility with virtually all PDF reader software on the planet. Both the *OptimizeforWeb!* and *GenerateLogFile!* options are turned off, but I've turned on the option for *OpenPDFAfterSaving!*.
3. In step #3, all the changes to the style are stored.
4. And finally in step #4, the PDF file is generated. The resulting file is *MySecondPDF.pdf*, stored in WordPerfect's documents folder. The *ReplaceAlways!* option is set to Always, which means WordPerfect should overwrite any previous file of the same name.

Tip: Notice the exclamation symbols after the option names in some of the commands? These symbols tell a WordPerfect macro that they are command options. If you type out your macros, be sure to include the ! character (called a "bang" in programming parlance), or else the macro won't work.

Finding Various PDF Style Options to Use

One of the best ways to learn about macro programming in WordPerfect is to turn on macro recording, and manually go through the steps you want to duplicate.

Alas, this method doesn't work with the PDF publishing tools. All the various commands and command options are not stored when the macro is recorded. All you get is the *PdfDlg* command.

All is not lost. You can self-discover the various PDF programming options using the Commands button, one of the buttons shown on the Macro Toolbar when you are editing a macro.

1. Open a macro for editing. The test platform macro shown above is a good one to work from.
2. Click the Commands button in the Macro Toolbar.
3. If not already set to display WordPerfect macro commands, set the Command Type to *WordPerfect*.
4. Scroll through the list to the PDF commands; they're all listed alphabetically.
5. Click on the *PDFStylesSetSettingValue* command, then click on *Settings* in the Parameters listbox. These are the top-level style settings you can specify.
6. Click on *SettingValue* in the *Parameters* listbox. These are the individual options you can specify for each setting. All the options are shown together, but you must be careful how each *Setting* and *SettingValue* are paired. Most are self-explanatory, but be prepared for some experimenting. Example, you use the *PDFVersion12!* *SettingValue* only with the *PDFVersion!* setting.

Best PDF Style Housekeeping

As you develop your WordPerfect PDF automation solutions, you'll want to be mindful of the styles you create, either manually – using the Settings button in the Publish to PDF dialog box – or in a macro.

Be neat and tidy by getting rid of styles you no longer need. It's easy: Choose File->Publish to PDF, then click on the Settings button. Review the styles that have been stored in the PDF Styles drop-down list. For any you don't want, select it, then click on the - (minus) button next to the list. Note that you cannot delete a built-in PDF style, only the ones you've created.

Tip: Get into the habit of clearing out old styles as you develop your PDF macros. Previous settings may remain unless you explicitly overwrite them. If you find some style settings appear to be ignored, delete the style and create a new version of it in your macro.

With these core basics of automating PDF documents with WordPerfect X7 under your belt, you can now apply them to creating even more robust solutions.

Combine the macro commands you've read about with your document assembly systems. Instead of printing out finished documents, publish them automatically to PDF. Code you provide in your macros can customize each document for each client, all with little or no user intervention. You save time, and end up with consistent PDF documents you can trust.