

How to Print Sample Fonts

WordPerfect Magazine

By Roger Gagon

Choosing a font for your document can be bewildering. With names like Bodoni, Garamond and Zapf Chancery, a font menu can seem as foreign as a French restaurant menu. If you're running WP 6.x in Graphics or Page Mode, the WYSBYGI ("what you see before you get it") window can help. But even this feature has its limitations.

What you really need are printed samples of all the fonts in the menu so you can see what you're getting before you select it. This month's column gives you a WP 5.1 and a WP 6.x macro that you can use to generate a complete list of font samples for your printer (see figure below). You can then print the list and refer to it as you're making font selections.

The macros are mercifully short. If you're using WP 6.x, you'll also learn how to utilize a little-known macro resource called LIBRARY.WPM in the macros you create.

A Ready-to-Use 6.x Macro

WordPerfect WP 6.x ships with a macro called ALLFONTS.WPM that you can use to print samples of your fonts. This macro doesn't print complete samples like those shown in the figure below, however. With ALLFONTS.WPM, the text used in the font sample is simply the name of the font. So, when names of the fonts only include five or six letters – like "Arial (TT)" – you aren't able to see a very complete sample, but at least you get an idea. Instead, use the included FONTLIST.WPM macro for both WP 5.1 and WP 6.x.

Using the Macros

Before using the macro, first make sure you have the correct printer selected. To check this, press Print (Shift-F7) and look at the (S) Select option. If you need to change the selected printer, press (S) Select, choose the desired printer, then press Exit (F7) until you return to the document screen. To run either macro, press Macro (Alt-F10), type "fontlist" and press (Enter). The macro begins to generate the sample font list.

In WP 5.1, a prompt at the bottom of the screen tells you which font is currently being added to the list. In WP 6.x a dialog box tells you which font is being added to the list and displays a percentage-processed counter to help keep you informed of the macro's total progress.

When the macro finishes, your cursor is returned to the top of the sample font list. To view the list in WP 5.1, press Print (Shift-F7), (6) View Document, or in WP 6.x press Print/Fax (Shift+F7), (7) Print Preview. Press Exit (F7) when finished.

Understanding the WP 5.1 Macro

You don't have to be a macro guru to understand how the macro works. If you're new to macro programming, follow the line-by-line explanation of this macro to pick up some macro basics.

Line 1 contains a {DISPLAY OFF} code that turns off the screen display so you don't see what the macro is doing while it is working. Line 2 first sets the justification to left, then sets 0.5" left and right margins.

Line 3 goes into the Print: Select Printer menu and assigns the name of the selected printer to a variable. Line 4 then exits back to the document screen and inserts a center code to center the first line.

Line 5 types the title for your font list, and the {VARIABLE}Printer~ command inserts the name of your printer it obtained a moment ago. After two hard returns are inserted to add spacing, line 6 goes into the Base Font menu and moves to the first font in the list.

Lines 7-20 form a large {WHILE} loop that repeats once for every font in the Base Font menu. The {SYSTEM}Entry~ command on line 8 obtains the name of the highlighted font, and the {ASSIGN} command assigns this font name to a variable called "Current." The number 1 at the end of this line selects the font.

Line 9 displays a prompt that includes the font name currently being added to the list. Lines 10-12 check to see if WordPerfect is asking for a point size in the Base Font menu (this happens for all scalable fonts). If so, the macro enters 12 for the point size.

Line 13 inserts the name of the font, followed by a colon and a space. Lines 14-16 type the rest of the text for the font sample. Lines 17-18 then block protect all three lines of the font sample so it can't be separated by a page break.

Line 19 returns to the Base Font menu and moves to the next font in the list. The {END WHILE} command on line 20 tells the macro to return to line 7 and repeat the loop for the next font. Each time the macro returns to line 7, the macro compares the new font name with the last font name it just inserted into your list.

As long as the new font name is different from the last one, the loop continues to repeat itself (the != on line 7 means "does not equal"). But as soon as the names are the same, the macro knows it's reached the bottom of the font list and the macro breaks out of the loop. At this point, the macro jumps to line 21, exits out of the Base Font menu and returns to the top of the document.

A Nifty WP 6.x Macro Tool

WordPerfect 6.x includes a macro called LIBRARY.WPM, which should be located in your macros directory. This macro is simply a library of commonly used macro subroutines that you can use in other macros. For example, if you want to add a percentage-processed dialog box similar to the one used in the FONTLIST.WPM macro, you don't have to reinvent the wheel and design this portion of your macro from scratch. Instead, you can have your macro borrow the ready-made macro code from LIBRARY.WPM.

If you're curious, you can open the LIBRARY.WPM macro into a document screen and read the explanations preceding each of the functions and procedures. To include these tools in other macros you create, however, you should first become familiar with the FUNCTION and PROCEDURE commands. (Both of these macro commands are described in the online Macro Help.) If you intend to use a function or procedure from LIBRARY.WPM in a macro, you need to first insert the USE("LIBRARY.WPM") command in your macro so the macro knows where to find these functions and procedures on your hard drive.

The FONTLIST.WPM macro calls two procedures from LIBRARY.WPM: PercentInit (line 13) and Percent (line 14). Although the PercentInit and Percent procedures aren't called until lines 13-14, it's a good practice to insert the USE("LIBRARY.WPM") command near the beginning of the macro if you'll be needing it later.

I'll get back to lines 13-14 in a minute. First, take a look through the WP 6.x macro from the beginning.

Understanding the WP 6.x Macro

Like the WP 5.1 macro, the WP 6.x macro begins with a DISPLAY(OFF!) command on line 1 that turns off the screen display. The second line of the macro sets the justification to left, then sets 0.5" left and right margins and finally inserts a center code to center the first line.

The DLGINPUT(On!) command on line 3 can be a little confusing, so I'll explain it briefly. Unlike WP 5.1, the WP 6.x macro language records only the end result of your keystrokes, not the keystrokes themselves. This can pose a problem when you need to obtain a piece of information (such as the name of the selected printer) from a WordPerfect menu or dialog box. How do you get to the information you need if you can't record the keystrokes to take you into the dialog box where the information is located?

The answer is to use DLGINPUT (On!). This command allows your WP 6.x macros to behave like WP 5.1 macros while you're in a dialog box, typing the needed keystrokes until you've accomplished what you need in that dialog box.

For example, the PrintDlg command on line 3 goes into the Print/Fax dialog box, then the Type("S") command chooses the Select button on this menu to take you to the Select Printer dialog box. (This same principle is used on line 6 of the macro to choose the (1) Font option in the Font dialog box.) Remember, this is only possible if you first include the DLGINPUT (On!) command in your macro to allow for dialog input.

Once inside the Select Printer dialog box, line 4 assigns the name of the selected printer to a variable called Printer, then the two ExitDlg commands return to the document screen.

Line 5 types the title for your font list, which includes the name of the selected printer it obtained a moment ago. After inserting two hard returns to add spacing, line 6 lists the fonts in the Font dialog box and moves to the first font in the list. The FontNum:=?List command assigns the total number of fonts in this list to variable FontNum.

Lines 7-10 assign all of the font names to an array (a series of variables), then the two ExitDlg commands on line 11 exit out of the dialog boxes and return to the document screen.

Lines 12-23 form a large FORNEXT loop that repeats once for each font in your font list. Line 13 calls the PercentInit procedure in the LIBRARY.WPM macro (mentioned earlier) to initialize the percentage-processed dialog box. The Font[Current] parameter on line 13 is used to display the name of the current font just above the percent processed bar. The number 54 simply indicates the width of the dialog box.

The PercentInit and Percent commands must be used together, so line 14 calls the Percent procedure in LIBRARY.WPM. Note the two parameters – Current and FontNum – used with the Percent procedure on line 14. Each time the FORNEXT loop executes, variable Current passes the number of the current iteration from the FORNEXT loop (see line 12) to the Percent procedure so it can update the percent processed counter. The second parameter – FontNum – indicates the total number of repetitions that will be used (in this case, the total number of fonts in the font menu).

Line 15 selects the actual font, then line 16 inserts the name of the font, followed by a colon and a space. Lines 17-19 type the rest of the font sample. Lines 20-21 then block-protect all three lines of the font sample so it can't be separated by a page break. Line 22 inserts two hard returns to add spacing.

The ENDFOR command on line 23 tells the macro to return to line 12 and repeat the loop for the next font. This FORNEXT loop continues to repeat itself until the total number of fonts stored in variable FontNum (line 12) is reached. When this happens, the macro breaks out of the loop and skips to line 24, where the cursor is returned to the top of the document and the macro quits.

```

Font list for printer [your printer name]

BrushScript-WP (Type 1): abcdefghijklmnopqrstuvwxyz
ABCDEFGHIJKLMN O PQRSTU VWXYZ
1234567890,.;'?"@#$%^&*()-_+=\|{}|<>

Century-WP (Type 1): abcdefghijklmnopqrstuvwxyz
ABCDEFGHIJKLMN O PQRSTU VWXYZ
1234567890,.;'?"@#$%^&*()-_+=\|{}|<>

ChelmsfordBook-WP (Type 1): abcdefghijklmnopqrstuvwxyz
ABCDEFGHIJKLMN O PQRSTU VWXYZ
1234567890,.;'?"@#$%^&*()-_+=\|{}|<>

CommercialScript-WP (Type 1): abcdefghijklmnopqrstuvwxyz
ABCDEFGHIJKLMN O PQRSTU VWXYZ
1234567890,.;'?"@#$%^&*()-_+=\|{}|<>

CooperBlack-WP (Type 1): abcdefghijklmnopqrstuvwxyz
ABCDEFGHIJKLMN O PQRSTU VWXYZ
1234567890,.;'?"@#$%^&*()-_+=\|{}|<>

Courier: abcdefghijklmnopqrstuvwxyz
ABCDEFGHIJKLMN O PQRSTU VWXYZ
1234567890,.;'?"@#$%^&*()-_+=\|{}|<>

Courier 10 Bold (Speedo): abcdefghijklmnopqrstuvwxyz
ABCDEFGHIJKLMN O PQRSTU VWXYZ
1234567890,.;'?"@#$%^&*()-_+=\|{}|<>

```