

Desktop Publishing with SAS® and WordPerfect®

Christopher Toppe, Ph.D.
Computer Sciences Corporation

Abstract

Let's face it: any report produced in the fixed space Courier font looks amateurish. And we all know that SAS output loves Courier and hates Times Roman or any other proportional, desktop publishing font. But that doesn't mean that you have to be the only one in your office who produces ugly reports. It is possible to create WordPerfect documents within SAS -- to imbed WordPerfect control codes for attributes like centering, bolding, underlining, and especially tabs. This allows you to go directly from SAS to WordPerfect, with all the desktop publishing capabilities of WordPerfect available for that SAS-created document.

The Secrets

Creating a WordPerfect document from SAS is really straight forward once you know the secrets. They are listed below.

- ▶ Start with a blank WordPerfect document

A blank WordPerfect document is not empty. It contains all sorts of useful information in the header portion of the file. This header establishes a file as a WordPerfect file and sets (for that file) attributes such as tab settings, margins, font style and size. It is into this blank-but-not-empty file that you will insert your SAS output.

- ▶ Use data step report generation to prepare your report

The process presented in this paper will not work with output generated by SAS procedures such as PROC PRINT or PROC TABULATE. You will have to generate your own report layout using FILE, PUT, and other SAS custom report generation statements.

But that's the whole point of wanting to go from SAS to WordPerfect. You know how you want the report to look; it's just that you can't get there with a PROC step. If nothing else, this procedure makes it easy to perform that one seemingly impossible task -- tab alignment.

You should already know how to prepare the totals and subtotals you need using any number of techniques. These should include data step processing using FIRST. and LAST. variables; PROC MEANS with CLASS and OUTPUT statements, combined with SET with a point; and PROC SQL. These provide techniques for data preparation and are best performed prior to report generation.

The primary reason to separate data preparation from report generation is that the process presented here makes it easy to revise your reports as specifications change over time. For example, changing the order of the columns can be easily accomplished because your report does *not* have column pointers imbedded in it (e.g. PUT @10 VAR1). Rather, your report contains WordPerfect codes for tabs, so changing the order of columns is as easy as changing the variable name that follows each tab character

- ▶ Store the WordPerfect attribute codes in easy-to-remember variables

I create SAS variables that have mnemonic names. For example the variable to turn on underlining is UON while the variable to turn it off is UOFF. Further, if you store these variables in Macro variables created in your AUTOEXEC.SAS file, they will always be available to you.

- ▶ Use PROC PRINTTO (PCs) or a binary copy to insert your SAS report into the blank WordPerfect file.

Remember that blank WordPerfect file mentioned

above? You need to put your SAS-generated output into that file without destroying the file. That is, you need to append your report to that header which identifies the file as a WordPerfect file. Once that is done, the attribute codes you inserted into the SAS file will be interpreted by WordPerfect as being WordPerfect control codes.

There are two ways to append the SAS-generated file to the blank WordPerfect file: PROC PRINTTO and binary copy. If you are using a platform such as Windows or UNIX, using PROC PRINTTO is the best way to join these files. A sample program using PROC PRINTTO is shown below:

```
FILENAME WP "C:\BLANK.WP";

PROC PRINTTO PRINT=WP;

DATA _NULL_;
FILE PRINT N=PS NOTITLES;
--REPORT CODE--;
RUN;

PROC PRINTO;
RUN;
```

Note that the first PROC PRINTTO opens and holds the file referenced by the fileref WP. The FILE PRINT statement appends the results from your report code to that file. The last PROC PRINTTO closes and releases the file. After your job finishes, you can open that file in WordPerfect as you would any existing WordPerfect file.

If you are working on a mainframe or other platform which requires you to transfer the SAS-generated report file to the system where you have WordPerfect, then you cannot use PROC PRINTTO. You will have to use the binary copy process.

Create the file on the mainframe system and transfer it to the PC using whatever file transfer utility you normally use. Append that file to the blank WordPerfect using one of the following DOS

commands:

```
COPY BLANK.WP+SAS.OUT /B
COPY BLANK.WP+SAS.OUT NEW.WP /B
```

Note that the first file named in the COPY command is the WordPerfect file and the second one is the SAS output file. The 'B' option indicates that the copy is a binary copy, necessary to preserve the HEX codes.

In the first COPY example, the SAS.OUT file is appended to the BLANK.WP file. In the second example, a new file is created, NEW.WP, which is the combination of the two files.

Selected Codes

Now that you know how to insert attribute codes into WordPerfect from SAS, all you need to know are the codes. Here, then, are some of the key codes you will need to become an expert desktop publisher using SAS and WordPerfect.

► BOLD

```
ON      "C30CC3"X;
OFF     "C40CC4"X;
```

► UNDERLINE

```
ON      "C30EC3"X;
OFF     "C40EC4"X;
```

► CENTERING

```
"C1E0CB000000000000"X;
```

► TAB

```
"C1000000000000C1"X;
```

► DECIMAL TAB

```
"C142D419781E4C00C1"X;
```

► RIGHT JUSTIFY (FLUSH RIGHT)



Note that the attribute codes are all HEX codes. Also note that the letter oh (O) does not occur in any of those codes; what looks like an oh is the number zero (0).

Systems

This paper was developed using WordPerfect 5.1 under DOS and SAS version 6.08 under OS/2. It has also been run successfully on a Sun SparcStation using SAS version 6.09 and WordPerfect 5.1 (UNIX version) under SUN OS 4.1.2. For those of you using MVS or CMS, the procedure has been implemented successfully using the binary copy process in place of PROC PRINTO.

About the Author...

Chris Toppe is a decision support systems specialist and Director of Educational Services at Computer Science Corporation's Commercial and Civil Business Center in Rockville, MD. In the DSS area, Dr. Toppe advises on decision strategies in data warehouse and EIS projects. In the education area, he manages the delivery of SAS and other application development training to CSC's commercial and government clients. He teaches base SAS, Intermediate SAS Programming, PROC TABULATE, PROC SQL, and Report Writing. He has been using SAS for nineteen years.

Chris Toppe
1301 Piccard Drive
Rockville, MD 20850
(301) 258 - 5308 (Voice)
(301) 258 - 6878 (Fax)
Christopher_Toppe/SED/CSC.CSC@CSCMAIL.CSC.COM (e-mail)
CTOPPE@CSC.COM (SAS-L)

Caution

WordPerfect changed their attribute HEX codes in version 6, so the codes presented in this paper will not work. Version 6 codes will be released when I have them available.

Conclusion

By sending your SAS output to WordPerfect, you will enter the era of desktop published, professional-looking report generation. As a matter of fact, not only does this process produce better looking reports, it also makes the process of report layout infinitely easier. I encourage you to give it a try.

On the following pages are selected slides from the actual presentation.

SAS is a registered trademark of the SAS Institute, Cary, NC
WordPerfect is a registered trademark of WordPerfect Corporation

The SECRETS

Start with a blank WordPerfect document

Imbed WordPerfect control codes in SAS
output

Use PROC PRINTTO to append SAS
output to the blank file

or

-----presentation page 1-----

Create an output file from SAS and append
it to the blank WP file using the binary copy
command:

COPY /B BLANK.WP SAS.OUT

-----presentation page 2-----

A Blank WordPerfect Document is not empty

It contains codes in a "Header" that
WordPerfect uses to recognize a file as a
WordPerfect file.

This file stores information such as:

WordPerfect version and creation date
Type of printer
Fonts

Without this "Header" information on a file,
WordPerfect will not interpret the file's
contents correctly.

-----presentation page 3-----

SELECTED WordPerfect HEX CODES

Bold ON C30CC3

Bold OFF C40CC4

Underline ON C30EC3

Underline OFF C40EC4

Center
C1E0CB000000000000

Tab
C100000000000000C1

Decimal Tab
C142D419781E4C00C1

-----presentation page 4-----

Right Justify C160C1
WordPerfect Codes in SAS

1. Create variables that hold the WP codes in HEX

2. To write HEX codes in SAS:

- put the code in quotes
- followed by an X

For example, to turn BOLD on and off:

```
PUT "C30CC3"X
    "This will be in bold"
    "C40CC4"X;
```

-----presentation page 5-----

HINT: Store the WP HEX codes as variables:

```
BON = "C30CC3"X; /*BOLD ON*/
```

```
BOFF = "C40CC4"X; /*BOLD OFF*/
```

```
PUT BON 'This is bold' BOFF 'and this is
not';
```

-----presentation page 6-----

Example WordPerfect Codes SAS

```
BON = "C30CC3"X; /*BOLD
ON*/
BOFF = "C40CC4"X; /*BOLD
OFF*/
UON = "C30EC3"X;
/*UNDERLINE ON*/
UOFF = "C40EC4"X;
/*UNDERLINE OFF*/
CON =
"C1E0CB00000000000000"X; /*CENTER*/
TAB =
"C10000000000000000C1"X; /*TAB*/
DTAB =
"C142D419781E4C00C1"X; /*DECIMAL
TAB*/
RON = "C160C1"X; /*RIGHT
JUSTIFY*/
```

-----presentation page 7-----

```
PUT
BON "BOLD" BOFF /
UON "UNDERLINE" UOFF /
BON UON "BOLD AND UNDERLINE"
UOFF BOFF /
CON "CENTERED BY 'CON'" /
TAB "TAB USING 'TAB'" /
RUN;
```

-----presentation page 8-----

THE WordPerfect DOCUMENT

BOLD

UNDERLINE

BOLD AND UNDERLINE

CENTERED BY 'CON'

TAB USING 'TAB'

-----presentation page 9-----

Storing the WP HEX Codes

You can have the WordPerfect HEX codes always available if you define them as macro variables in your AUTOEXEC.SAS file.

1. Edit (or create) your AUTOEXEC.SAS file
2. Add the WP HEX Codes as macro variables:

%LET BON = "C30CC3"X;

%LET BOFF = "C40CC4"X;

3. Execute AUTOEXEC.SAS when you start SAS

Example:

SAS -AUTOEXEC C:SAS\AUTOEXEC.SAS

USING PROC PRINTTO

-----presentation page 10-----

You can use PROC PRINTTO to send your report directly to the WordPerfect document. In this example, the report will be appended to the file referenced by the FILEREF WP, the blank document:

FILENAME WP "C:\BLANK.WP";

PROC PRINTTO PRINT = WP;

DATA _NULL_;

FILE PRINT N=PS NOTITLES;

REPORT CODE;

RUN;

PROC PRINTTO;

NOTE1: You must issue the final PROC PRINTTO to release the file.

-----presentation page 11-----

NOTE2: This will 'ruin' your BLANK.WP file; it will no longer be blank after this program is run. Therefore, it is a good idea make a copy of BLANK.WP before you run this job.

-----presentation page 12-----

Special Considerations

1. The codes for Bold and Underline insert a blank space before the string to be bolded or underlined. You can balance that blank by adding a blank to the end of the string in SAS, if necessary.
2. WordPerfect reads the SAS new page code, and it works as intended as long as the SAS page break comes before a default WP page break. You may want to hard code your SAS page breaks.

-----presentation page 13-----

- 3.. If you are going to bold and center a line, put the center code inside the bold-on and bold-off codes.
4. Remember that the results of your SAS job is a WordPerfect document. Bring it into WordPerfect by using RETRIEVE (SHFT-F10), not by using DOS TEXT-RETRIEVE (CNTL-F5, 2).

-----presentation page 14-----

STEPS TO FOLLOW

1. Dummy a document in WordPerfect:
 - Lay out your rows and columns;
 - Set up your tabs;
 - Save this as your blank document;
2. Build the SAS output file using the WP HEX Codes instead of column pointers;
3. Run the SAS job against a subset of your data, being sure that you use enough data to cross a page boundry;
4. Open the resulting output file into WordPerfect as a WordPerfect document.
5. Review the resulting layout, noting any changes to your SAS program that are required. Also, this is a good time to experiment with fonts, tab stops, margins,

-----presentation page 15-----

and other WordPerfect controlled features;

6. Revise the SAS program and run a final test;
7. Repeat steps 4 - 6 as necessary, remembering that each iteration requires a new BLANK.WP document;
8. Run the final job.

-----presentation page 16-----