Paper 064-30

Mainframe SAS® for the 21st Century

Kryn Krautheim, National Center for Health Statistics, Research Triangle Park, NC

ABSTRACT

Versions 8 and 9 of the SAS System have included many enhancements that are typically enjoyed by users of the PC platform. While working on a major overhaul of one of our reporting systems, we discovered many features of SAS that worked very well on the Mainframe, but were not widely known. This paper illustrates a few 'modern' tools that SAS Programmers can use in the Mainframe environment to move their systems forward into 21st Century techniques and applications. We will take a quick look at generating PDF files, using the SAS FTP access method, and generating e.mails, all from the z/OS (Mainframe) environment.

OUTPUT DELIVERY SYSTEM

One of the most significant developments in Version 7 of SAS was the inclusion of the Output Delivery System, or ODS. Greatly enhanced in Versions 8 and 9, ODS allows users to create customized versions of output in a variety of destinations (including HTML, RTF, XML, and PDF) instead of being limited to traditional line-listing printouts. For flexibility and convenience, we have converted many of our reports to PDF format.

Generating default (unenhanced) PDF files is fairly simple with ODS, but there can be a lot of extra work involved in making minor changes to the default PDF generator. SAS includes a number of 'styles' for generating ODS output. Styles are collections of parameters that tell SAS how to format output by specifying fonts, colors, margins, and many other attributes. You can also modify existing styles to suit your needs, but that is beyond the scope of this paper. A copy of one of our customized styles is included as an appendix.

Simple ODS output can be accomplished by 'wrapping' your existing SAS program with a few lines of code. Before your procedures, you will need to code at least 3 statements:

1 FILENAME, which creates a file reference name (fileref) and allocates space for your PDF file on the Mainframe.
2 ODS LISTING CLOSE, which closes the default LISTING destination. It is recommended to keep open only the specific destinations you are sending output to.
3 ODS PDF, which opens the PDF destination, and provides SAS with additional details about the file that you are generating.

There are other ODS statements and SAS options that you may find useful to include in this initialization section.

Here is a sample of some code used to initiate a PDF job:

```
1      FILENAME OUTPDF "KJK4.FDWCTNAT" DISP=NEW UNIT=TEMP RECFM=VB LRECL=259;

2      ODS LISTING CLOSE;

3      OPTIONS ORIENTATION=LANDSCAPE;
4      ODS NOPTITLE;
5      ODS PATH SASHELP.TMPLMST (READ) NEWTEMP.TEMPLAT (READ);

6      ODS ESCAPECHAR='^';
7      ODS PDF FILE=OUTPDF STYLE=STYLES.TEST1 NOTOC;
```

Code Explanation:

Line 1 These are the recommended Data Control Block (DCB) parameters from SAS when defining a PDF file on the Mainframe. On this particular system, UNIT=TEMP will automatically delete the file after 3 to 8 days, depending on the file size.
Line 2 Before generating output with ODS, you should close the default LISTING destination.
Line 3 ODS output uses portrait layout unless otherwise specified.

Line 4 This optional statement suppresses the printing of the default procedure name in your output.

Line 5 This statement is only necessary if you have defined your own ODS Style templates, and refers to your own Template store; if you are using one of the built-in styles, you can omit it.

Line 6 This optional statement defines a special character not present in your dataset that can allow you to add very specific in-line formatting inside of some of your procedure code.

Line 7 This is the primary statement that initiates the generation of your PDF file, and refers to the FILENAME statement (Line 1) specified earlier. The STYLE parameter refers to the desired ODS style you want used for your output. This example refers to a user-defined ODS template called TEST1. The NOTOC option suppresses the automatic generation of the PDF Table of Contents.

Remember that only statements 1, 2, and 7 are required to generate your ODS code.

Following these statements, any output generated from SAS Procedures will be directed to the ODS destination. After your last output procedure, you can then close your ODS destination with just two lines of code:

```
8       ODS PDF CLOSE;
9       ODS LISTING;
```

Line 8 This statement is issued after the final data step boundary (make sure to include a RUN statement after your final procedure, before this statement) to end the generation of your PDF file.

Line 9 This statement re-opens the default destination, and is good practice to include after closing another ODS destination.

The capability to generate PDF files was put into production for the Mainframe in Version 8.1, although the files generated were quite large. Improved compression for PDF files was introduced in Version 9.0, which was used for the code in this paper.

SAS FTP ACCESS METHOD

Once you have created your ODS output on the Mainframe, you have a number of options to move that file somewhere where it can be easily accessed by you and your users.

Within your mainframe terminal emulation software, there may be an option to directly transfer files, or you may have a standalone software package (such as WS_FTP) that can be used. If you have worked with UNIX System Services on the Mainframe, and have access to a web-served directory on the system, you can actually create PDF files on the Mainframe that can be directly browsed by Internet Explorer, without having to transfer the files.

We opted to use the SAS built-in method of transferring files via FTP. The process is fairly simple, and is accomplished through the FILENAME statement. To use this method, you will need access to an active FTP server. Here is a sample FILENAME statement:

```
FILENAME OUTFTP FTP
  "TESTFILE.TXT"
  CD="/home/kjk4/public_html"
  HOST=”111.111.111.111”
  USER="kjk4"
  PASS=”password”
  RECFM=S
  DEBUG;
```

While the syntax for the FTP Filename statement is quite different from typical FILENAME statements on the Mainframe, the parameters that are coded are fairly standard.

After the FILENAME statement, code a fileref that you will refer to later. In this example, OUTFTP is the fileref. FTP is then coded to indicate to SAS that you will be using File Transfer Protocol with this file. Next, list the name of the file that you will be creating on the remote host. The naming conventions of the *destination* should be used when specifying the filename.

In this example, we are transferring data to a Linux server, so the options that are shown here may or may not apply to other destinations you may use. The SAS Language Reference documentation for “FILENAME, FTP Access Method” explains all of the different options available for this statement.

Since the Linux operating system is case-sensitive, we had to issue a “CAPS OFF” mainframe command in our ISPF edit session, and make sure that we were not using the CAPS option anywhere in our program. This preserves lower-case letters in the SAS programs. Depending on your remote destination, this may be necessary to do in your programs as well.

We use the CD command to change the directory of the destination to point to the remote location where we want to store our file. HOST is the network name of the destination server. It can be specified using an IP address or the actual host name.

USER and PASS specify the UserID and associated Password for access to the remote FTP server. RECFM=S sets the record format type, creating a binary data transfer. The DEBUG option prints log messages from the actual FTP login on the remote server to your SAS Mainframe log.

Notice that even though we are in the Mainframe environment, we did not have to specify any of the typical keywords associated with a fileref on the Mainframe, such as LRECL, DISP, or SPACE. The SAS internal FTP client transfers the file to the remote destination, and no permanent file is written to the Mainframe.

Once you have coded your FILENAME statement, you can use that fileref just as you would in any statement that generates an output file:

```
DATA _NULL_;
 SET ONE;
 FILE OUTFTP;
  PUT …
```

Or, you could combine it with ODS code presented earlier, and have the Mainframe generate the PDF file and automatically transfer it to your remote system. Since the file is never actually stored on the mainframe, it is not necessary to code any DCB parameters in this FILENAME statement.

```
FILENAME OUTPDF FTP "TESTFILE.PDF"
  CD="/home/kjk4/public_html"
  HOST=”111.111.111.111”
  USER="kjk4"
  PASS=”password”
  RECFM=S
  DEBUG;

ODS PDF FILE=OUTFTP STYLE=STYLES.TEST1 NOTOC;
```

MAINFRAME-GENERATED E.MAIL

Our Mainframe offers the capability to generate automated e.mail messages from within SAS programs, and can also include attachments and links to files. We began by trying to send files as attachments, but because of size limitations, we abandoned this method for our project, and instead use e.mail to notify users that files have been created and stored on local servers. These automated e.mails contain hyperlinks to the files, and allows users to quickly access reports in an electronic format. This feature is experimental in Version 8, and production in Version 9, where it is documented.

First, we show code on how to send an e.mail from the Mainframe, and then explain how to send different attachments.

E.mails are also generated on the Mainframe through use of the FILENAME statement. Here is some sample code for generating a plain text message:

```
FILENAME OUTBOX EMAIL
         TO=(’KJK4@XYZ.GOV’ ‘ZQQ8@XYZ.GOV’)
         REPLYTO='TBE2@XYZ.GOV'
         TYPE='TEXT/PLAIN'
         SUBJECT="Your file is ready";
```

In this example, OUTBOX is our fileref for the e.mail that we are going to generate, and EMAIL is the keyword to tell SAS that this file is an e.mail to be sent via SMTP. The TO parameter denotes the e.mail address(es) of the recipients for the e.mail. If you have a single recipient, you may omit the parentheses. You can also code CC= in the same manner if you want to send copies of the message to other users.

By default, the FROM address on messages generated by the Mainframe will not look like a standard e.mail address. In addition, there's probably not a mailbox on that end that you can access. It is good practice to use the REPLYTO parameter with an address you can access so that if there are problems, users can easily contact you. Note that if delivery of a message fails, you may or may not get any type of error in your log, depending on other JCL settings in your job. While we have had very good success with our system, we frequently include developers' e.mail addresses in the list of recipients to monitor performance of the system and make certain that there are no problems.

We code TYPE to make sure that SAS will send the file as plain text. The SUBJECT keyword is optional, but recommended, and values inside quotation marks will have their case preserved.

Once you have issued your FILENAME command, you can then generate an e.mail using PUT statements inside a DATA _NULL_ step, like this:

```
OPTIONS NOCAPS;

DATA _NULL_;
 FILE OUTBOX;
 PUT "The report listed above is now available via the following";
 PUT "link for 21 days:";
 PUT / "http://111.111.111.111/~kjk4/testfile.pdf";
 PUT // "Thanks,";
 PUT   "KJK4";
RUN;
```

The NOCAPS option makes sure to preserve the lower case letters in the body of the message. This is necessary when including hyperlinks that point to a file stored on a case-sensitive system. The FILE statement tells SAS to refer to the OUTBOX fileref, which alerts the system that this is an e.mail. Then, PUT statements are used to specify the text. Our e.mail system, Outlook, automatically recognizes the third line of this e.mail as a hyperlink, and allows the recipient to click on it to access the file specified.

Attachments can also be sent by e.mail, although the maximum attachment size allowed by our Mainframe is 500KB. This maximum is set by an IBM SMTP Configuration File, and is independent of SAS. Text, PDF, HTML, and other types of files can be sent via e.mail.

To send an attachment, you will need to have already created a file on the Mainframe, using a standard fileref earlier in your program. Then, you simply add one additional option to your FILENAME EMAIL statement that denotes the attachment:

```
ATTACH=("KJK4.T1030.TXT” NAME=”KJK4”);
```

This statement attaches a text file that is named KJK4.T1030.TXT on the Mainframe. Since PC users may be confused by a 3-level file name, I can use the NAME= option to specify a new name for the file, to which SAS automatically adds the correct file type extension. So, the above statement sends the file KJK4.T1030.TXT as an attachment, but in the e.mail, it appears as a file named KJK4.TXT.

This statement attaches a PDF file directly to the e.mail message:

```
ATTACH=("KJK4.T1030” CT='APPLICATION/PDF' EXT='PDF');
```

Since we have coded the EXT keyword, the Mainframe attaches the file denoted by the filename in quotes, and adds the extension, which becomes KJK4.T1030.PDF in this case. The CT parameter specifies to the receiver's e.mail system what to do when the attachment is opened. With the value of ‘APPLICATION/PDF’, the recipient's e.mail program should automatically open the attachment in Adobe Acrobat when clicked.

The syntax for sending attachments is not always obvious, but there are a number of examples located on the SAS website, and in SUGI papers. The documentation for Version 9 also gives additional examples.

CONCLUSION

We were able to take advantage of a number of technologies that we had not previously explored in the Mainframe environment, mostly because we were unsure of their stability and usability. Although it took some time to become comfortable with the new options, they have greatly enhanced our efficiency, and allowed us to provide users with better and faster access to data.

REFERENCES

SAS Institute Inc. 2003. SAS OnlineDoc® 9.1. Cary, NC: SAS Institute Inc.

SAS Institute Inc. "Sending Electronic Mail within the SAS System under OS/390."
<http://support.sas.com/techsup/technote/ts605.html> (01 February 2005).

SAS Institute Inc. "Tips for using ODS in the OS/390 Environment."
<http://support.sas.com/rnd/base/topics/templateFAQ/MVSODS3.pdf> (01 February 2005).

CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the author at:

Kryn Krautheim
Centers for Disease Control and Prevention
National Center for Health Statistics
PO Box 12214
Research Triangle Park NC 27709
(919) 541.0197
kjk4@cdc.gov

APPENDIX

This is a sample of PROC TEMPLATE Code that we use to create a very simple and clean user-specified Style for ODS:

```
LIBNAME NEWTEMP 'DWJ2.TEMPLATE.LIB';

PROC TEMPLATE;
  /* HEADINGFONT CONTROLS COLUMN HEADERS.  DOCFONT CONTROLS THE CELLS.
   * EMPHASISFONT CONTROLS THE TOTAL LINES.
   */
  DEFINE STYLE STYLES.TEST1 / STORE=NEWTEMP.TEMPLAT;
    PARENT=STYLES.SASDOCPRINTER;
      REPLACE FONTS /
        'TITLEFONT2' = (ARIAL,9PT,BOLD)
        'TITLEFONT' = (ARIAL,7PT,BOLD)
        'STRONGFONT' = (ARIAL,9PT,BOLD)
        'EMPHASISFONT' = (ARIAL,9PT,ITALIC)
        'FIXEDEMPHASISFONT' = (ARIAL,9PT,ITALIC)
        'FIXEDSTRONGFONT' = (ARIAL,9PT,BOLD)
        'FIXEDHEADINGFONT' = (ARIAL,9PT,BOLD)
        'BATCHFIXEDFONT' = (ARIAL,6PT)
        'FIXEDFONT' = (ARIAL,9PT)
        'HEADINGEMPHASISFONT' = (ARIAL,10PT,BOLD ITALIC)
        'HEADINGFONT' = (ARIAL,9PT,BOLD)
        'DOCFONT' = (HELVETICA,9PT)
        ;
      STYLE TABLE FROM OUTPUT /
        RULES=NONE
        ;
      REPLACE OUTPUT FROM CONTAINER /
        FRAME=HSIDES
        CELLPADDING=1PT
        CELLSPACING=0
        ;
      REPLACE SYSTEMTITLE FROM TITLESANDFOOTERS /
        FONT=(ARIAL, 10PT, BOLD)
        ;
      REPLACE SYSTEMFOOTER FROM TITLESANDFOOTERS /
        FONT=(ARIAL, 7PT, BOLD)
        ;
  END;
RUN;

PROC TEMPLATE;
 LIST / STORE = NEWTEMP.TEMPLAT;
 RUN;
```

For additional information about PROC TEMPLATE, there are many resources available from proceedings of previous years’ SUGI conferences.