Paper 087-30

Across the Great Divide: Creating PC-Ready Formatted Output Using Base SAS® Output Delivery System on the Mainframe

Barbra Lockley, Independent Consultant, Scotch Plains, NJ

ABSTRACT

It may have seemed as if the mainframe programmer was left out of the OUTPUT DELIVERY SYSTEM (ODS) explosion. It may have seemed that by using the SAS System for MVS you could not exploit the more powerful capabilities of ODS. Well in fact, beginning with Version 7, the full power of ODS can be harnessed within the SAS System MVS Operating System environment.

This paper will demonstrate, through the use of examples and sample code, how to create output files which can be rendered via a web browser (i.e. HTML, XML, CSS, etc.) and are compatible with the suite of Microsoft (MS) Office products, i.e. .xls, .doc, etc. You can use the power of BASE SAS ODS to create and format output data once, and then use that data to generate mainframe print, as well as personal computer or server ready output.

INTRODUCTION

A recurring complaint regarding the SAS System output produced on the mainframe is that it has a “dated” look. Web technology and MS Office Suite products have become an integral part of oral and written presentations. This is true regardless of the tool of origin. Output today must be portable, flexible and easy for the layperson to manage.

Base SAS ODS gives the you the ability to create personal computer or server ready data using components available within Base SAS, at no additional cost, using the mainframe platform.

With BASE SAS ODS, you are able to wrap HTML (Hypertext Markup Language), CSS (Cascading Style Sheets), or XML (Extensible Markup Language) around existing mainframe SAS System output presentations or legacy print output. In addition you may create and design new web or Microsoft Office presentation output by also incorporating the TEMPLATE PROCEDURE. How is it possible without the SAS System for personal computer? It is possible through the use of Base SAS ODS

HOW THE OUTPUT DELIVERY SYSTEM (ODS) WORKS

The OUTPUT DELIVERY SYSTEM controls formatting of output objects, e.g. ODS LISTING output (the default), ODS PRINTER output, ODS HTML output. In simplest terms, ODS puts a “wrapper” around the output data. This “wrapper” tells the output object how it should read or interpret the output data so that it is formatted for the intended device. There are various methods for applying more control and using more robust tools, i.e. PROC TEMPLATE , to create the “wrapper”. Below is a sample MACRO for creating a very simple HTML output wrapper.

GENERATING THE ODS OUTPUT, BY FOLLOWING STEPS 1 THROUGH 4

1) CREATING THE PARTITIONED DATA SET EXTENDED (PDSE) LIBRARY OR BINARY OUTPUT FILE.

First create a Partitioned Data Set Extended (PDSE) library with the following Data Control Block (DCB) Information:

LRECL=8196 BLOCKSIZE=27998 ORG=PO RECFMT=VB

This can be done using the ISPF utility, or within the JCL. This will produce a problem free PDSE library that in most cases will handle most output files. Output will be directed to a binary file on the mainframe. **This is an extremely important first step.**

example of creation within JCL

```
//REPORTS1 MY.MAINFRAME.OUTPUT.HTML,UNIT=SYSDA,
           DISP=(NEW,CATLG),DSNTYPE=LIBRARY,
           DCB=(LRECL=8196,BLKSIZE=27998,RECFM=VB,DSORG=PO)
```

There should now be a cataloged PSDE library called MY.MAINFRAME.OUTPUT.HTML with the above DCB information.

Invoke Display Manager System at the site if you would like to see sample output of your data. Provide an input data set to use as an example, the output dataset will be the newly created MY.MAINFRAME.OUTPUT.HTML PSDE.

```
 LIBNAME  MYFILE   'MY.MAINFRAME.INPUT.FILE.DATA' DISP=SHR;
FILENAME REPORTS1 'MY.MAINFRAME.OUTPUT.HTML' DISP=OLD;
```

2) CREATE THE ODS STATEMENT MACRO .

Insert the following sample BASE SAS ODS MACRO code in your MACRO library or at any point prior to the MACRO invocation within the SAS program. The following MACRO demonstrates how BASE SAS ODS may be applied to an existing legacy system in order to redefine the output as specific ODS format. There is no need to change the current DATA OUTPUT statement(s). In addition to the existing output, an additional file is created which contains the same data output in a binary form. In this case the binary data file output will be in HTML format.

Sample MACRO (A very simple form, using MVS OS/390 Display Manager)

```
OPTIONS MACROGEN MPRINT MLOGIC SYMBOLGEN DQUOTE ;
************************************************  ;
*                  ODS OUTPUT FILE                ;
* REQUIREMENT: NAME ODS FILEREF ON THE BODY STATEMENT ;
 * USE A URL SUBOPTION TO CREATE PDSE MEMBER  ;
*                                                  ;
* PROBLEM FREE DCB INFO-LRECL=8196 BLOCKSZE=27998 ORG=PO RECFMT=VB ;
* MACRO PARMS:                                    ;
*                                                   ;
*   FILEREF1=OUTPUT FILE NAME FOR PDSE-           ;
*                                                 ;
*   ODSFILE =PDSE MEMBER NAME                     ;
************************************************;

%MACRO HTMLOUT(FILEREF1,ODSFILE) ;
 ODS TRACE ON ;
 ODS &OBJECT1
 STYLE = DEFAULT
 BODY="&ODSFILE"(URL="&ODSFILE..HTML")
 PATH = &FILEREF1(URL=NONE)
 RECORD_SEPERATOR=NONE
 CONTENTS="NAVIGATE"(URL="NAVIGATE.HTML")
 NEWFILE=PAGE
 PAGE="&ODSFILE"(URL="'&ODSFILE..HTML'")
 TRANTAB=ASCII ;
 **************   ODS &OBJECT1 CLOSE ;
 %MEND HTMLOUT ;
```

3) ASSIGN AN ODS MACRO VARIABLE VALUE .

Assign the MACRO variable Object1 to an ODS OUTPUT listing device.

```
%LET OBJECT1=HTML ;
```

4) RUN THE ODS STATEMENT PRIOR TO ANY STANDARD SAS SYSTEM OUTPUT PROCEDURE. CLOSE THE ODS OUTPUT IN ORDER TO DISCONTINUE HTML DATA OUTPUT GENERATION.

Invoke the actual MACRO prior to any OUTPUT PROCEDURE you wish to view as Excel, MS Word, PowerPoint, XML, or CSS within the personal computer or server environment . The PDSE member 'SOMEFILE' will contain the HTML binary data output.

```
%HTMLOUT(REPORTS1,SOMEFILE) ;

 PROC PRINT DATA =MYFILE.SOMEFILE;
 RUN;

ODS &OBJECT1 CLOSE ;
```

```
RUN;
```

HOW DOES THE MACRO ACTUALLY WORK?
Now for a closer look at the macro and exactly what it does.

```
%MACRO HTMLOUT(FILEREF1,ODSFILE) ;
```

This is the MACRO name assignment statement, notice the two variable names which are passed into the MACRO. The variable 'FILEREF1' is used to identify the output HTML PDSE DD name. It will act as a pointer for the actual ODS path. The variable 'ODSFILE' will be used to indicate the actual member name with in the PDSE.

```
ODS TRACE ON ;
```

This turns on a feature which writes any ODS processing directly to the SAS log.

```
ODS &OBJECT1 ;
```

This statement will assign the ODS destination to the variable name 'OBJECT1'. In the example this is initialized via a global variable assignment, it will be defined as HTML. If this is not defined then the default is ALL.

```
STYLE = DEFAULT
```

This identifies the type of style the output will have as well as the coloring, of foreground and background features. Some of the standard style definitions that are currently shipped with the SAS System include:

Default	D3D
Beige	Minimal
Brick	Printer
Brown	Statdoc

PROC TEMPLATE can also be used to define your own individual style, or for internal consistency.

```
BODY="&ODSFILE" (URL="&ODSFILE..HTML")
```

This identifies the actual body portion of the HTML file. In the sample code, this is the actual OUTPUT data.

The MACRO variable "&ODSFILE" represents the member name within the PDSE. Because it is a member name, it should be enclosed in quotes.

This method is particularly useful in legacy systems, where the MACRO could be invoked prior to initial output. The ODS output is generated as a member within the HTML PDSE file. Any original output will continue to be produced. The SAS log will contain the ODS TRACE information indicating what was actually generated.

The value &ODSFILE..HTML represents the personal computer version of the file. In the sample code the personal computer file name is consistent with the member name. One could select any name however this is where the personal computer or server name is defined. This should also be the name of the file in the personal computer directory.

In the example, this code was invoked prior to Legacy systems reports, with the MACRO variable &ODSFILE indicating each individual reporting group. No ODS HEADTEXT= option was used, although it could be incorporated and any text would be specified between the <HEAD> and </HEAD> tags.

```
PATH=&FILEREF1(URL=NONE)
```

This identifies the actual URL path, where the MACRO variable &FILEREF1 is the DD name for the output PDSE. URL=NONE is required because the URL is automatically linked to the URL in the BODY and PAGE options. This will ultimately be on a personal computer or server, where mainframe addressing information will not be useful.

```
RECORD_SEPERATOR=NONE
```

In the mainframe environment, the binary file creates an embedded record separator character. Using NONE immediately signals to SAS that it is environment specific HTML creation.

```
CONTENTS="NAVIGATE"(URL="NAVIGATE.HTML")
```

In the sample this value is hard coded. In a web site or a document which requires a table of contents or directory, the information is automatically routed to the member 'NAVIGATE'. In addition the personal computer file is also called NAVIGATE.HTML.

```
NEWFILE=PAGE
```

This creates a new body file at the specified starting point. In this case the NEWFILE points to PAGE.

```
PAGE="&ODSFILE"(URL="'&ODSFILE..HTML'")
```

PAGE is identified as the MACRO variable &ODSFILE, so each time a new SAS System PROC is encountered, or if a new page is explicitly started (i.e. DATA _NULL step), a member file is created.

One note, when this member file is created, providing an ODS Close is not encountered, each PROC operation will generate a new member by sequentially appending the number one onto the member name. E.g., Dept_CD member will be followed by Dept_CD1, Dept_CD2, etc.

```
TRANTAB=ASCII;
```

This translates the mainframe EBCDIC into ASCII code for view on the personal computer or server.

```
ODS &OBJECT1 CLOSE;
```

This is the ODS CLOSE statement. It has been commented out in the body of the MACRO. You can include it in the MACRO, however once encountered, ODS CLOSE stops the ODS output. In the example &OBJECT1 will be initialized to HTML. The ODS HTML CLOSE statement is included in the SAS code in the example. If you had several OUTPUT Procedures, and wanted to capture them with the member name LEGACY1, LEGACY2, LEGACY3, etc., your OUTPUT will continue to sequentially number these files, until it encounters the ODS HTML CLOSE statement.

HOW IS THE MACRO USED?

This example shows usage of the actual MACRO. The MACRO variable OBJECT is initialized to HTML, the FILEREF is called REPORTS1, and the member name of the file with in the PDSE is called LEGACY.

```
%LET OBJECT1=HTML
       %HTMLOUT(REPORTS1,LEGACY) ;

           Additional Legacy system SAS code.

       ODS &OBJECT1 CLOSE ;
 ODS TRACE OFF ;
```

Please note the sample SAS Log.

```
683          +%HTMLOUT(REPORTS1,LEGACY);
684
MLOGIC(HTMLOUT):  Beginning execution.
MLOGIC(HTMLOUT):  Parameter FILEREF1 has value REPORTS1
MLOGIC(HTMLOUT):  Parameter ODSFILE has value LEGACY
SYMBOLGEN:  Macro variable OBJECT1 resolves to HTML
SYMBOLGEN:  Macro variable ODSFILE resolves to LEGACY
SYMBOLGEN:  Macro variable ODSFILE resolves to LEGACY
SYMBOLGEN:  Macro variable ODSFILE resolves to LEGACY
SYMBOLGEN:  Macro variable ODSFILE resolves to LEGACY
NOTE: Writing HTML Body file: LEGACY
NOTE: Writing HTML Contents file: NAVIGATE
NOTE: Writing HTML Pages file: LEGACY
MLOGIC(HTMLOUT):  Ending execution.
```

Sample SAS Log

NOW WHAT?

At this point the data created on the mainframe is ready to move to the personal computer or server environment. There are many tools to port the file, just remember that this is a binary file. Therefore the appropriate tool will need to be used in order to put the binary files in the personal computer directory.

The files at this point must simply be called by the member name, however once downloaded, they can easily be renamed.

In this example, the data are used in an Excel spreadsheet. The resulting files, LEGACY.HTML through LEGACY10.HTML , were opened using Microsoft Excel and the data was used for analysis and presentation. They were also included in a Microsoft Word document, where the specific reporting pages were included for presentation. Because of it's portability, the same file was provided to another user, who wished to use the information within an Access database, and yet another user included it in SQL processing. In addition, Legacy System reports were retained for easy reference . The same reporting previously only available through printed reports, and more time consuming CSV files was now easily dispensed to business analysts.

The HTMLOUT macro was included in an existing Production process in order to allow the same flexibility through out an auditing process.

CONCLUSION

In this example HTML was used, however CSS or XML can easily be substituted. CSS files need only be identified as using the STYLESHEET option. In addition to defining XML through PROC TEMPLATE XML output types are now available as well.

The robust power comes from being able to fully harness the web client foundational tools of HTML, CSS, and XML, and "wrap" your SAS System OUTPUT in these tools, coupled with MS Office Suite products' ability to read these file formats with minimal manual intervention.

You, the mainframe user are no longer confined to formatting data output as .CSV or .TXT flat files for downloading, then exporting, importing to MS Office product and manually re-formatting the data.

You have freedom to span across the great divide, and create pc ready output data via Base SAS ODS, using only the mainframe platform.

For a more in depth discussion of these individual components there is "The Complete Guide to SAS Output Deliver System, Version 8'. In addition this there are excellent SAS User Group International papers on the specific topics of HTML, XML, ODS, Java script by Eric Gebhart; PROC TEMPLATE, ODS, CSS, MS Suite Products, etc. by Chevell Parker ODS; and XML by Miriam and Ricardo Cisternas, just to name a few.

REFERENCES

SAS Institute Inc.(1999), *The Complete Guide to the SAS Output Delivery System*, Version 8, Cary, NC SAS Institute, Inc.

Chevell Parker, "Generating Custom Excel Spread Sheets Using ODS"
SAS Institute, Inc. 2003

Vive la difference? Miriam G. Cisternas, Ricardo A. Cisternas
Proceedings of the Twenty-Ninth Annual SAS Users Group International Conference, 29, 119-29.

CONTACT INFORMATION

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

Barbara Lockley
Email: blockley.inet1@worldnet.att.net