

Paper 248-25

Teaching SAS® Web Publishing in a Web Environment

Roderick A. Rose, Jordan Institute for Families, UNC-Chapel Hill, North Carolina

Sally S. Muller, Jordan Institute for Families, UNC-Chapel Hill, North Carolina

ABSTRACT

Experienced SAS users can gain a better understanding of the SAS Web Formatting Tools and the Version 7 Output Delivery System (ODS) when they are taught in a classroom setting with a web page as the primary instructional device. The authors developed a website, on which HTML, FTP, the SAS HTML Formatting Macros and the Output Delivery System are integrated into one interactive session. This website utilizes Javascript and modest client-server interaction. Links that open a new browser when selected are provided throughout the site for additional resources during the lecture or for subsequent review. Examples of programs are provided, by displaying code and output side-by-side using the entire screen rather than the conventional browser window. To avoid mundane tasks such as data entry, data sets can be downloaded with a few simple selections.

The web-based instruction that we developed offers a full range of instructional materials, references, interactive demonstrations, and tutorials -- resulting in much more student interaction with the material than conventional instructional methods. Web-based instruction can go beyond overheads and computer-based slide presentations. This paper is intended for anyone interested in developing a website for instructional purposes.

CONCEPTS TO COVER

The process of developing static SAS-based output on the web involves a background in the client/server application and several iterative steps: HTML, SAS programming, uploading, and debugging. These steps, and the concepts associated with them, serve as the basis of our course.

THE CLIENT/SERVER APPLICATION

By introducing the SAS Web Formatting Tools, SAS has made it practical to store all applications and data on a central server and allow interaction with remote clients using a web browser. Thus, the first concept that the instructors address is the client/server application.

Client/servers applications encompass organization-wide Intranets as well as the Internet, and provide the pipeline over which all web pages are transmitted. Static web pages are created and stored on a web server for distribution. Minimal user interactivity with the page is possible - users can select links to move from one page to the next, for instance - but all elements displayed on a web page must be incorporated into the design of the page by the author before it is published. Still, we tend to think of web pages as "on-the-fly" creations, as authors can instantaneously and continuously update the page.

HTML

With the creation of the SAS Web Formatting Tools, SAS has streamlined the task of creating output that is web-ready. Users no longer have to HTML-format output – as far from "on-the-fly" as one can get – as it is now possible to create SAS output in a completely web-ready format. Using a text editor to draft a web page is still an essential skill. For this, one still needs a basic knowledge of HTML.

For many casual Internet users, the composition of a web page - many files of different types referenced from a hypertext document - is an abstract concept. Hypertext markup language itself can be confusing to a generation of computer users experienced only with WYSIWYG (users who remember the days of markup-enhanced word processing tend to not find it as difficult – an interesting role reversal). It is equally important to dispel the notion of HTML as a "programming" language. The goal is to provide a starting point

without supplying an encyclopedia of information that will overwhelm the learner.

SAS PROGRAMMING

The only prerequisite of the course is basic SAS programming knowledge. To the extent that this prerequisite is satisfied the instruction can focus on the SAS Web Formatting Macros and the Output Delivery System.

The SAS formatting macros can display SAS data sets, PROC TABULATE output, and all SAS procedural output. But sometimes students want to add additional features to their web page – for example, a link to return to the home page. For this purpose the instruction needs to cover the DATA_NULL_ step and PUT statements. These normally simple SAS concepts take on added difficulty when HTML is introduced into the PUT statements.

The Output Delivery System is capable of producing all procedural output in a formatted style. Much like the formatting macros, ODS can also benefit from the inclusion of the DATA_NULL_ step as described above to provide enhancements to the page.

UPLOADING THE STATIC APPLICATION

The applications that are developed must be uploaded from the local (client) machine to the web server to be accessible on the Internet. The students will need to gain familiarity with another widely-used Internet protocol, besides the ubiquitous hypertext transfer protocol (HTTP), the File Transfer Protocol (FTP).

In addition, there is the issue of which particular FTP program to use, and since each program can be considerably different in method, which ones to teach. WS-FTP and the Windows 95/98 FTP Client are useful for general uploads. However, there is an easier method that is applicable to SAS applications called the SAS FTP Access Method, wherein the file transfer protocol is written directly into the SAS programs.

DEBUGGING

SAS-generated Internet applications are comprised of SAS and HTML-based components. Unfortunately, mistakes and bugs are comprised of both SAS and HTML-based components as well. The best method for debugging a SAS application is to study the log file; the best method for debugging HTML is to study the output. The trick is to approach the programming in an iterative fashion – design the HTML first, then build the SAS code around it. This does not prevent all problems from arising, but it does reduce them. The authors thus chose to emphasize the iterative approach to building static SAS applications.

CONSTRUCTING THE WEBSITE

Once we determined the concepts that we wanted to communicate, the construction of the website became straightforward.

The website consists of several layers of user interaction. The first, and most important layer, consists of a series of sequential pages that provide the foundation for the class. Additional layers allow the students to learn more about the concepts of greatest importance to them, by providing demonstrations, links to other websites and further instruction.

The first section begins with an introduction to SAS web publishing, and tries to answer a basic question: What advantages accrue to the SAS user who can publish static online content "on-the-fly"?

The second section is devoted to generating basic online output and

consists of three lessons:

- The first lesson is an instruction in basic HTML, in which elements, attributes and tags are defined and reviewed. After learning several simple but useful HTML elements, the students author a simple web page.
- The second lesson combines HTML and SAS code by showing the students how to construct a DATA _NULL_ step with PUT statements with which the SAS System will author the HTML for them. In particular, it will author the same page that they wrote in the first lesson.
- The final lesson of the section explains that the web pages must be placed on a server to be made Internet-accessible. In addition, this section introduces methods of uploading files to a server using FTP. Three different FTP methods are covered: WS-FTP, the Windows 95/98 FTP Client, and the SAS FTP Access Method.

When the students complete this section, they have a home page accessible online which was authored by using the SAS system.

Macros are the focus of the third section. First, since no assumptions have been made about the students' understanding of macros, an introduction to macros is given. The remainder of the section consists of three lessons, each one designed around one of three HTML Formatting Macros:

- The Data Set Formatter is the first lesson, since this is the easiest of the macros to understand.
- The Output Formatter, is the second lesson and much of the instruction is focused on this lesson. An interactive demonstration ensues and when the demo is complete, the page created by using the Output Formatter is uploaded to the server.
- The Tabulate Formatter, is the third lesson and focuses on the many options that the formatting macros provide.

The fourth section is an introduction to ODS. This includes a lesson in the ODS HTML destination. An interactive session is provided in which a multi-frame web page is developed and uploaded to the server automatically by using the FTP Access Method.

When the student has completed these five sections, the student should have three pages online, beginning with the home page created by the DATA _NULL_ step in the first interactive demo. This home page links to the pages created by the Output Formatter and the ODS output as generated in the subsequent two interactive demos.

A review of the material is presented in a final section.

FLOW, INTERACTIVITY AND FURTHER REVIEW

Our website adheres to three principles which can serve as useful guidelines for any instructional website: flow, interactivity, and further review.

FLOW

By its very nature a website allows students to take any "path" they desire. They can take a short excursion into supplementary material, they can back-up and review material, they can forge ahead and skip material they already know. This is the beauty of the web and our pages are designed to facilitate this behavior. However, as with any instruction, the flow of topics sometimes needs to be linear. In our case, we wanted the students first to learn basic HTML and then how to upload files using FTP, before moving to producing web output with a SAS DATA _NULL_ step. We accomplished this simply by the sequence of the pages, as we carefully evaluated how to make a logical progression from one topic to the next.

Just as the sequence of the pages facilitates flow, so does the use of hyperlinks whenever there is a need for an example. The student who has no need for an example can ignore the link and stay focused on the lesson. But for the students who need an example, or another reference, the link can make a significant difference in their

understanding of the material. A tip when creating such links is to eliminate the need to navigate backwards through a maze of pages to return to the lesson. To do this, add the attribute `target="_noframes"` to the hyperlink tag as follows:

```
<a href="http://anotherwebsite.com/"
target="_noframes">
```

A new browser window will open when selected. If one has already been opened and has not been closed, the linked page will load in that window. Besides providing links to examples, we also found it was useful to provide links to data files. We were pleased to discover how much valuable class time this saved, as compared to having the students key in the data themselves.

As said, we wanted to capitalize on the freedom that the web gives you for jumping to supplementary material, for reviewing, and for going where you need to go. To facilitate this we employ three links. The first link takes the student to the next lesson in the linear fashion as described above. It is placed directly after the lesson and is clearly marked as "Next".

The second and third links are in the address bar at the bottom of the page. The second link takes the student to a site index that allows the student to jump to any page on the website if they choose to review a topic later. The third link takes the student to a "Frequently Asked Questions" page. The FAQ extends the interactivity between the instructor and the student by providing a forum outside of the classroom for students to ask questions and receive answers.

INTERACTIVITY

A substantial body of research indicates that students learn more when they interact in the classroom as opposed to listening to a "talking head." The authors heeded this research and designed the website to maximize student interaction. For example, rather than just showing the student the appropriate SAS program for the job, the student has to write the SAS program, run it, and thereby generate the HTML output themselves. The trade-off is that this strategy takes more time, especially if some of the students are less experienced than others.

To help alleviate this problem, we use web scripting. For instance, to link to an example of code and the output it generates, we enhance the hyperlink, by referencing a Javascript function:

```
<A onClick="this.href='javascript:
openExample('\splitscreen.html\')'" href..>
```

"openExample" is a Javascript function which generates a full-screen window (without toolbars or menu bars, etc.) when the link is clicked. This full-screen window consists of three frames: a menu from which several examples can be selected; and windows for the SAS code and output, which appear side-by-side. Selecting one item in the menu opens the SAS code and output windows. This results in a significant time saving as the student need not download any code nor run the code to see the output.

We found that by displaying the code and output simultaneously the student gets a better understanding of how the codes produces the output. In keeping with our efforts to maximize interactivity, we provide another link on the menu that gives the student the option of downloading the code and running it on their own computer.

Another opportunity we discovered for saving time using web scripting techniques, involves the students acquiring server permission for web publishing. On the University's server, as with many other academic servers, students are allowed an online directory called "public_html" for web publishing. The students must register to receive the necessary permission before the pages become accessible online. At UNC, this process can take up to an hour.

Our instruction requires that the students use this directory for storing their static web applications. To ensure that students have the necessary permissions set -- before they need them -- we use a very simple Javascript-enhanced HTML form at the beginning of the website:

```
<FORM ACTION="" NAME="myform" METHOD="GET"
NCTYPE="text/plain"
onSubmit="document.myform.action =
'http://www.unc.edu/~' +
document.myform.userid.value" + '/'>
<P>Your UserID:
<INPUT TYPE="TEXT" NAME="userid"> </P> </FORM>
```

This form allows students to check whether they already have permission, and if not, allows them the opportunity to obtain the permission. The form reappears on the site wherever such access is required.

FURTHER REVIEW

The website concept empowers the student by encouraging them to revisit the material for review and further development. The two-hour class allows only an overview of several concepts and tools. In a situation where copious note-taking is not worthwhile, this provides the student with an opportunity to pick up what they can in class with the reassurance that they can return to the material later.

COMPUTER BASED TRAINING AND THE WEBSITE

Although our website was not meant to compete with CMC (Computer Mediated Communication) nor any other asynchronous or dispersed means of computer-based instruction, we do share many of the same goals. One such goal is the delivery of a portable and platform-independent means of instruction. The website is portable and non-proprietary, as is true with most Internet applications. There is modest platform dependence in the sense that some applications perform well on certain browsers and poorly on others.

Another goal that website instruction shares with other methods of computer-based instruction, is the opportunity for self-instruction. Self-instruction when learning how to create static web pages seems especially appropriate since the pages themselves serve as examples of the desired results.

Also as with most other computer-based training, website instruction requires that the webserver is up and running. If the server goes down, which can happen at our site, a back-up plan is needed. In our case this meant making copies of the website on both the instructors and students local hard drives.

CONCLUSION

This presentation covers many of the advantages of using website instruction. Although our subjects, the use of the SAS Web Formatting Tools and ODS, are particularly amenable to website instruction, we believe other topics are equally appropriate for website instruction.

The important thing is that the website incorporate the principles of flow, interactivity, and review. If these guidelines are followed and the web pages present a logical progression of ideas and materials, the website can provide a powerful tool for instruction.

CONTACT INFORMATION

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

Roderick A. Rose
Jordan Institute for Families
CB 3550
301 Pittsboro St.

Chapel Hill, NC 27599
Email: rarose@email.unc.edu

Sally S. Muller
Jordan Institute for Families
CB 3550
301 Pittsboro St.
Chapel Hill, NC 27599
Email: sally@email.unc.edu
Work Phone: 919-843-7798