

Paper 76-26

Adding A Common Look and Feel to Web Applications Easily

Steve James, Centers for Disease Control and Prevention, Atlanta, GA

ABSTRACT

Having all of your output pages of your SAS/IntrNet™ application with the same look and feel is an important aspect of making your web applications more useful. But adding all of the HTML code to each page can be challenging, as well as difficult to maintain. By using an HTML template file, you can easily add the same banners and links to each output file your application produces. This technique leads to easy maintenance, does not require an extensive knowledge of HTML, but still allows some customization.

This paper assumes a working knowledge of SAS/IntrNet.

INTRODUCTION

Until recently, web pages at the Centers for Disease Control and Prevention (CDC) were as varied as the organization that hosted them. There were few common elements between them, and users had to constantly adjust to a new style when they went to pages from a different Center. In an attempt to minimize the confusion that users would face, the CDC required each Center to use a standard template for all of their web pages whenever possible. Though a small amount of customization was allowed with things such as fonts and colors, this meant that users could generally expect to find things in the same place regardless of where they were within the CDC web site.

The problem was that the HTML code used to create the template was fairly complex. While the webmasters seemed to understand it well, others who did not have such an in-depth knowledge of HTML had their struggles. Combine that problem with having to replicate the screen when producing output from a web application and the complexity mounts. A DATA _NULL_ step with HTML code in PUT statements is difficult to write, and very difficult to debug and maintain. The PREHTML and POSTHTML options of ODS Templates are a reasonable alternative, but can't be customized. Is there a better way? Fortunately there is.

HTML TEMPLATE FILE

By using an HTML Template file, you can create a template for the output that you want, and then simply have SAS® code that reads the file, and displays it to the browser prior to any output your application would send.

In its simplest form, the code would look something like this:

```
data _null_ ;
file _webout_ ;
infile 'template_file.html' ;
input @1 text $80. ;
put text $80.
run ;
```

Using an HTML editor such as FrontPage®, you can create the template file as plain or as complicated as you like. You can also insert trigger statements that could be identified by your program to customize the results. (see figure 1). The template file is then

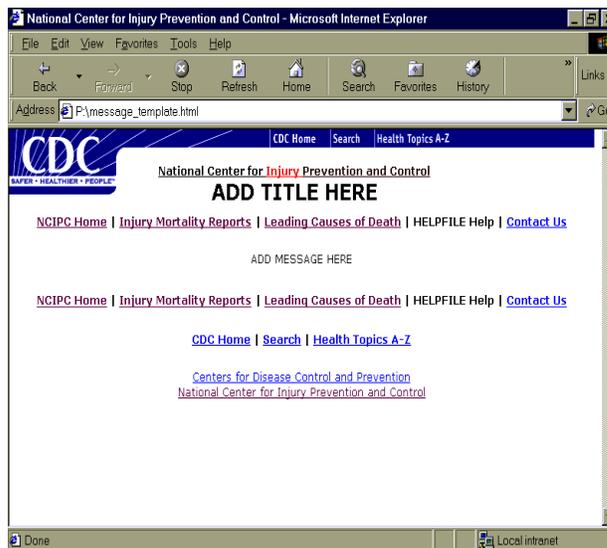


figure 1 - HTML template file

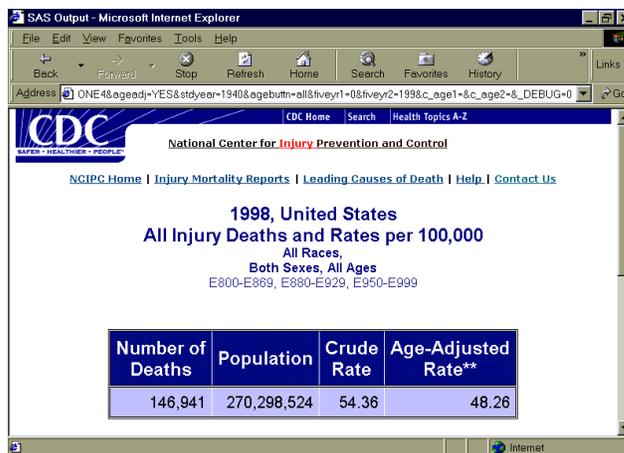


Figure 2 - Results of using HTML Template File (partial output)

stored on the server so that it could be accessed via SAS/IntrNet. A program is written so that it would read the HTML template file and then write it out. If it encountered a trigger statement, then it would insert the appropriate text and continue on. An example of how the finished product might look is in figure 2.

The code for processing the template file is found in Appendix A.

For a working example of this technique, go to <http://www.cdc.gov/ncipc/wisqars>.

Paper 76-26

BENEFITS

The main benefit of using this technique is that you achieve the grand purpose of making your web pages with the same look and feel without a lot of maintenance problems. Modifications can be made using an HTML editor which does not require an extensive knowledge of HTML. You can change the look of all your output pages by modifying the one template file. And you can add the logic to the program to make your messages and links context sensitive.

CONCLUSION

There are several ways of making all your web pages have the same look and feel. An HTML template file is quick and easy and uses the same skills that you have already developed using SAS/IntrNet.

REFERENCES

SAS, SAS/Graph, and SAS/IntrNet are registered trademarks or trademarks of SAS Institute Inc. FrontPage is a registered trademark of Microsoft Corporation.

CONTACT INFORMATION

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

Steve James
Centers for Disease Control and Prevention
4770 Buford Highway, MS-K59
Atlanta, GA 30341
Work Phone: (770) 488-4269
Fax: (770) 488-1665
Email: sjames@cdc.gov

Paper 76-26

APPENDIX A

```

*-----;
* Print out the top part of the CDC Template, up to the      ;
* part where you want to insert the output for the report   ;
* that you are going to create.                             ;
*-----;
* REPT is a parameter from the HTML form requesting the output ;

data _null_ ;
length text $1000 ;
infile "template.html" lrecl=80 pad ;
file _webout ;

input @1 text $80.;

if _n_ = 1
then do ;
    put 'Content-type: text/html';
    put;
    end ;

if index(text,'HELPPFILE') > 0
then do ;
    * Add test here to determine context for
    * identifying Help file ;
    if &rept==mort=
    then text = ><a href=@aboutmort.html@>= ;
    else text = ><a href=@aboutlcd.html@>= ;
    end ;

    if index(text, 'ADD MESSAGE HERE') > 0
then do ;
    text== > ;
    stop ;
    end ;

put text $1000.;
run ;

*-----;
* Print the output for the report                            ;
*-----;

<SAS Code to produce output goes here. PROC PRINT, PROC REPORT, etc.>

*-----;
* Print out the remainder of the CDC Template.              ;
*-----;

data _null_ ;
length text $1000 ;
infile "template.html" lrecl=80 pad ;
file _webout lrecl=1000 ;
retain resume 0 ;

input text $80.;

if index(text,'ADD MESSAGE HERE') > 0
then do ;

```

Paper 76-26

```
resume = 1 ; * Now begin ;
text = ' ' ;
end ;

if index(text,'HELPPFILE') > 0
then do ;
    * Add test here to determine context for
    * identifying Help file ;
    if &rept==mort=
    then text = ><a href=@aboutmort.html@>= ;
    else text = ><a href=@aboutlcd.html@>= ;
    end ;

if resume then do ; put text $1000.; end ;
run ;

* end of code ;
```