**Paper 407**

# Add A Common Look and Feel to Your Web Application Easily

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

## ABSTRACT

Having all of your output pages of your SAS/IntrNet™ application have 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.

## INTRODUCTION

Until the last year or so, web pages at the Centers for Disease Control tended to be 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 Centers to use a standard template for all of their web pages whenever possible. Though a small amount of customization was possible with things such as fonts and colors, this meant that users could expect to find things in the same place regardless of where they were in the web site.

The problem was the HTML code that was 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. The challenge was producing that code for every page of a dynamic web application in a way that makes debugging and maintenance relatively easy while maintaining the flexibility and customization that a web application requires. A DATA _NULL_ step with HTML code in PUT statements is difficult to write, and very difficult to debug and maintain Is there a better way? Fortunately there is.

## HTML TEMPLATE FILE

Using an HTML editor such as Microsoft FrontPage, creating a template file was easy and allowed the insertion of trigger statements that could be identified by a program (see Appendix 1). The template file was then stored on the SAS/IntrNet™ server. A program was written so that it would read the HTML template file and then write it out “as is”. If it encountered a trigger statement, then it would insert the appropriate text and continue on. All output to the user used this template file and thereby maintained a common ‘look and feel’ throughout the application.

The advantage is that there’s a single point to make changes than will be reflected in all of your output pages.

Note: the template file need not be used to create every page the user sees. Static forms such as where query parameters are entered need not be generated this way.

## EXAMPLE OUTPUT

One of the common uses for this technique is creating error messages (see Appendix 2). The SAS Code in Appendix 3 was “macro-ized” so that it could accept parameters such as the type of report and the error message to display to the user. Then while writing the program if you wanted to report an error condition, you simply called the macro with the appropriate parameters, such as the condition to look for and the error message to print.

## CONCLUSION

The use of a template file has greatly enhanced developing the WISQARS™ web application at CDC. It’s used to produce all output to the user, including error messages. Changes can be made at one central location and propagate to the entire application without other modifications.

## REFERENCES

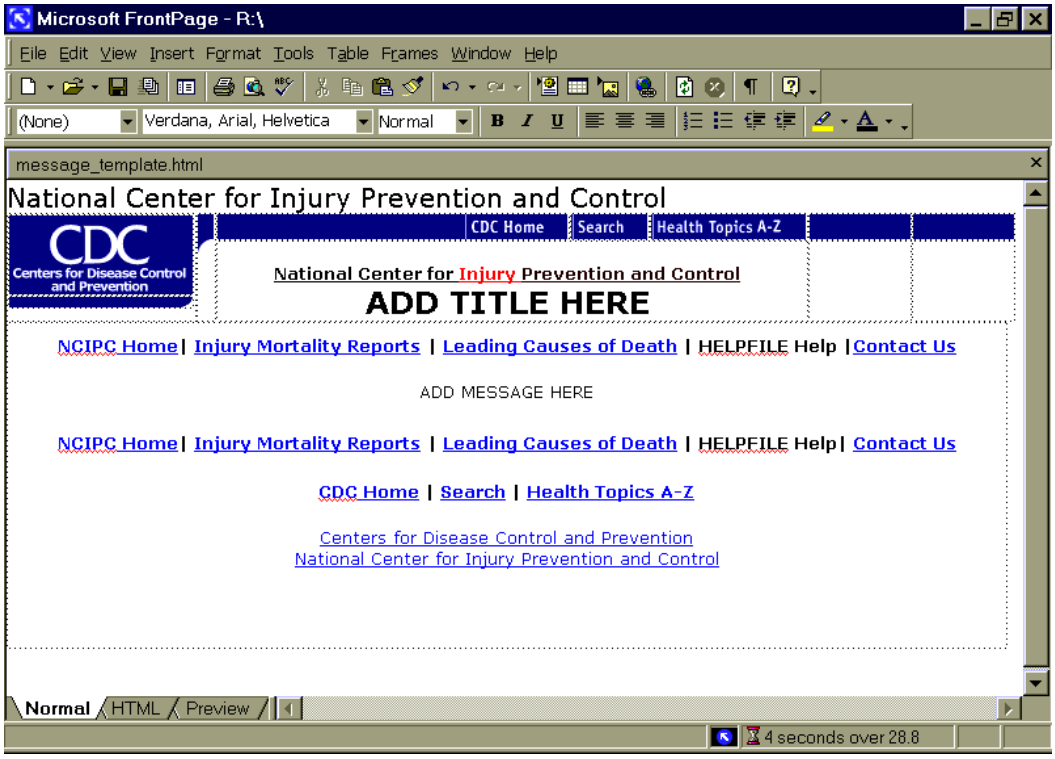


## CONTACT INFORMATION

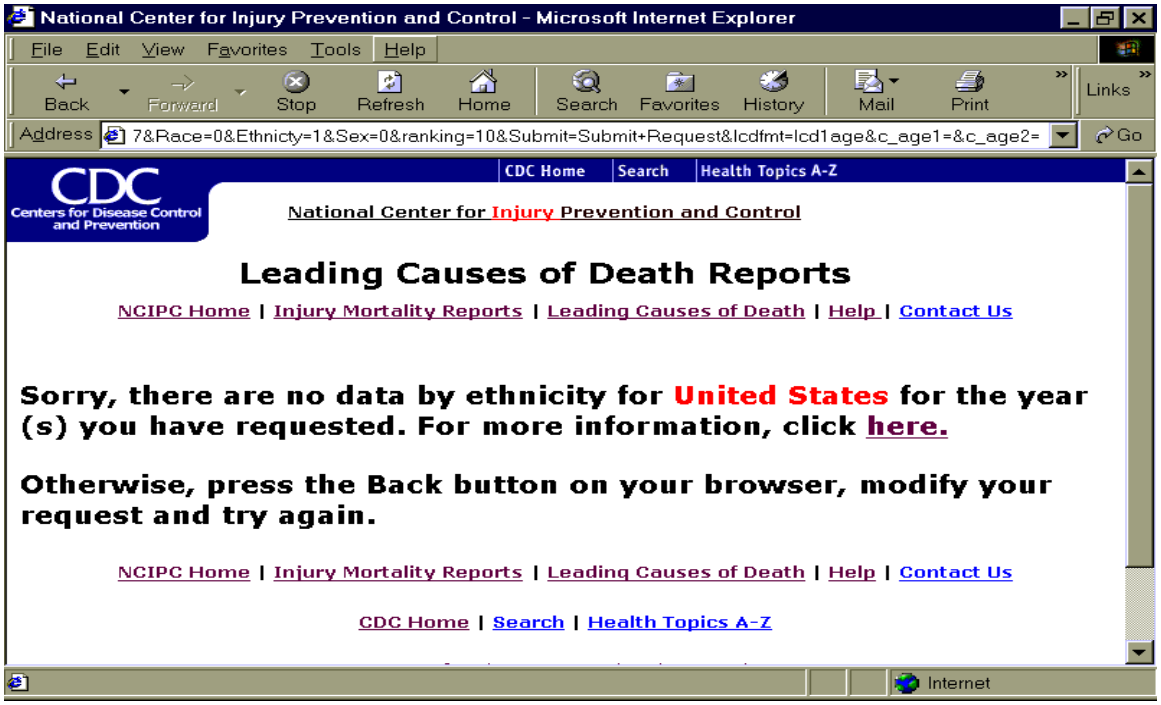
Your comments and questions are valued and encouraged. Contact the author at:

Steve James
Centers for Disease Control and Prevention
Atlanta, GA 30341
Work Phone: 770-488-4269
Fax: 770-488-1665
Email: sjames@cdc.gov

## APPENDIX 1 – HTML TEMPLATE FILE IN FRONTPAGE




## APPENDIX 2 – SAMPLE OUTPUT



## APPENDIX 3 – SAS CODE

The SAS code used to actually produce the output pages follows.

```
 *-------------------------------------------------------------------;
 * 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 $80 ;
      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,'HELPFILE') > 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=
           'add whatever text you want here, including HTML code' ;
           stop ;
           end ;

      put text $80.;
      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 $80 ;
infile "template.html" lrecl=80 pad ;
      file _webout lrecl=80 ;
retain resume 0 ;

      input text $80.;

      if index(text,'ADD MESSAGE HERE') > 0
      then do ;
           resume = 1 ;  * Now begin ;
           text = ' ' ;
           end ;

      if index(text,'HELPFILE') > 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 $80.; end ;
      run ;
```