Using the Data Step to Output Very Useful HTML and Tips for Making Web Pages More Accessible to the Blind.

Cécilie Birner, Metropolitan Transportation Commission, Oakland CA
Charles Hixson, Metropolitan Transportation Commission, Oakland, CA
Barbara Rhodes, Access Technology Trainer, San Francisco Bay Area

Abstract:
This paper has two main parts: how we used the data step to produce very useful HTML reports and what steps were taken to assure that these HTML pages were accessible to the blind. The reports are written as SAS macros and updated frequently to display important financial data. The HTML pages produced are accessed regularly by transportation project sponsors and the general public in the San Francisco Bay Area. While technology exists for the blind and visually impaired to navigate the web via speaking browsers, there are simple steps that can be taken to insure that your pages will be easily navigated by all web users.

Outline:

Part 1 SAS Macro to output HTML
1 A) Introduction
1 B) SAS Coding required
1 C) Other options web-ready SAS output
SAS internet Software
Adobe PDF format
Other Interactive Software

Part 2 Web page Accessibility Issues
2 A) Introduction
2 B) Basic Concepts of Web Pages Accessibly
2 C) Who else benefits?
2 D) Some references for web page accessibility

Glossary of Acronyms
Acknowledgements
Contacting the Authors

Part 1 SAS Macro to output HTML

1 A) Introduction

This web page program grew out of an attempt to contain a flood of financial data from over 14 billion dollars of projects. One of the functions of a Metropolitan Planning Organization (MPO) is to produce a Transportation Improvement Program (TIP). The TIP is a federally required document which lists all projects which either spend state or federal funds, require a federal permit (touch a state highway, railway, or water), or are considered 'regionally significant.' The TIP is updated frequently and it is important to be able to quickly post clean data on the web.

Until recently, changes to TIP were a black hole. Sponsors were not able to see the changes implemented until the TIP was re-published two years later. With the increased pressure for timely use of funds, TIP users needed real-time access to this important public data. This paper illustrates how we solved this problem by using the data step to output user-friendly data to the World Wide Web and also made the data accessible to all web users.

1 B) SAS Coding required

At my first SAS class the instructor said that "the data step is your best friend." This proved true in many instances, including putting this complicated and constantly changing data on the world wide web. To implement a similar system you need:

- to know basic HTML
- a good understanding of the data step
- a server to store your data

There are four steps necessary to put the data on the web:

1) get the data ready for web publishing
2) create an HTML file to link to SAS output
3) use the data step to output the data in HTML format
4) move data to the web server

Getting the data ready for web publishing:
We entered the data in MS Access and copied it to SAS via DMBScopy (**www.conceptual.com**). DBMSCopy is very reliable and easy to use. The batch feature makes data conversion especially easy and avoids endless mousing.

Next, we ran a SAS program which processed the data and made it ready to be posted to the web page. It is good to separate the data preparation from the HTML conversion. This

practice facilitates error checking and allows the HTML macro to run faster. Lastly, having a separate macro permits less-experienced SAS programmers to run just one macro to update the web page.

Create an HTML file to link to SAS output
The names of the files output from the SAS program need to match the same file locations used in the web page. Translating this technically, the HREF statements must match the SAS files output. In other words, the files output from the SAS code need to match the files that the main HTML page people will be accessing on the web.

For example, to update the San Francisco Municipal Railway (MUNI) projects, the SAS program would output a file called muni.htm

The HTML file must link to the muni.htm file output by the SAS program.

```
View <a href='muni.htm'>MUNI </a
> projects.
```

It is a good idea to use logical names. Use a consistent naming scheme if you will have multiple data files. Since some file servers are case sensitive, make sure both the name and case agree.

Use the data step to output the data into HTML
Once you've got the HTML file names set up, the SAS code is quite simple. The SAS program outputs a text file with put statements. The text file ends in ".htm", thus allowing web browsers to open it. The key is that the put statement outputs BOTH the data and the HTML coding.

```
data _null_;
     set alldat2 end=endfile;
     by sponsor proj_id;
file &outfile notitles
linesleft=space;

if _n_=1 then do;
put '<HTML>' /
    '<BODY TEXT="#000000"
BGCOLOR="#FFFFFF">' /
    '<P valign=center>'

<!MORE HTML CODING>
```

It is important, as you will read in the next section on accessibility, to output all necessary links for each web page so the user will not be lost in cyberspace. Links to your main web page, links to top and so on facilitate navigation. This is very easy to do with lines like:

```
if last.project then do;
put '</TABLE> <BR>' /
        "<STRONG>End of &desc
</STRONG>" / '<BR><BR>'
    '<A HREF="#top">Top of
Page</a><br>' /
```

Next, output the data along with the necessary HTML coding to display it:

```
put '<TD align=right>' FY97_dol
comma9. '<BR></TD>'
    '<TD align=right>' FY98_dol
comma9. '<BR></TD>'
```

If you will be using a put statement and a character ID, note that when outputs a SAS character variable it automatically puts in an extra space after it. This may cause problems if you are using a character in an href statement. There are several workarounds to this problem. Please contact the primary author for code samples.

Move the data to the web server:
Lastly, port (move) the data to the server. It is a good idea to have an intermediate step between the HTML SAS output and the web page so you can check the data.

1 C) Other options web-ready SAS output
SAS Internet Software:
SAS provides an internet software module. MTC's network configuration has SAS for Windows and a Unix Server so this solution would not work for me. SAS Internet software is covered in other papers in the SAS internet

section. The internet SAS module must be purchased separately.

Adobe Portable Data Format (PDF):
You can download programs to read data in Adobe PDF (**www.adobe.com/**) for free, but you must pay for the software to create PDF files. Many users are already familiar with Adobe, but it is time consuming to talk users through download the free Adobe reader. The Adobe web page is well set up and easy to navigate.

Adobe format is not suitable for blind readers. The Adobe Access tool extracts all readable text and converts the document to HTML. However, it has been known to mangle the contents the data files.

While Adobe PDF is a convenient format to use there are four downsides:

- The Adobe Access tool more often then not hashes data on pages
- If you do not have Adobe on your computer opening and Adobe file can crash your computer. Not all users know about Adobe PDF format.
- You must convert each file to Adobe PDF format
- Adobe PDF files can be slow to download

I do use Adobe PDF format and find it very convenient. However, I always provide an alternative format such as HTML or Excel or Word.

Interactive Software:
There are currently interactive data-driver packages on the market such as Cold Fusion. There is not enough room in the paper to completely evaluate all these options. Make sure to test them out using a speaking browser to insure that the data will be accessible to all web users. This applies not only to the visually impaired, but also to people using old browsers and other computer systems.

Perl:
Perl is a programming language which allows interactive web pages. Perl is extremely powerful and can be fast and flexible. It can also be a very contorted language. Perl shares many features in common with the C programming language. There are many Perl scripts (programs) you can download for free from the web. Please see **http://www.perl.com/pace/pub** for more information.

Part 2 Web page Accessibility Issues

2 A) Introduction

Every day the World Wide Web becomes a bigger part of our economy and culture. Therefore, it needs to be accessible to all users. At press time for the 1999 WUSS conference, there are some pending Americans with Disabilities Act (ADA) complaints and lawsuits against several governmental agencies regarding website accessibility.

2 B) Basic Concepts of Web Pages Accessibly

JAWS (Job Access With Speech) for Windows is a screen review program, that allows persons who are blind/visually impaired to interface with the Windows environment. It allows you to use many programs, including Word, WordPerfect, Excel, Access, Internet Explorer, Netscape, and other applications. People who can not see have been able to use computers for many years using adaptive equipment such as JAWS.

Of course, in accessing web sites, access also depends on entities making their sites accessible!!! *This is where SAS programmers come in.*

For web pages to be accessible to those using speaking browsers, the web page user needs to know what is in the page, how to get to the information, and be able to easily exit the page. There are simple ways to set up web pages that will make them accessible to all users.

Avoid navigation bars:
Web site indexes or links on the bottom of each page are preferable to navigation bars. Navigation bars are nightmares for those using talking browsers. They must read down the entire navigation bar to get to reach the main part of each page.

Use a table of contents:
The top of each page should have a table of contents with links to get to each. The table of contents should be both inclusive and clear and

concise. It is helpful to include both the link and actually spell out the web page URL. This is especially important if you are sending someone to a separate page. For example:

```
Go to the <a
href='www.barneyonline.com'>
Barney's</a> page:
www.barneyonline.com
```

Colors/Text:
Make sure that your choice of colors provides sufficient contrast. Some people are color blind and do know it. A good contrast will make the page readable to them, as well as more visually impaired readers. If you use bold or italics to stress a point, text should relay its importance as well.

If you use a style sheet, code so that the pages have a consistent background and text. The pages need to be readable both with and without the style sheet.

Avoid rotating and moving graphics:
Blinking and rotating graphics are problematic for people using adaptive browsers as well as other people who are sensitive to certain light patterns. A speaking browser gets caught in an infinite loop while reading a flashing graphic. Flashing graphics have been known to cause bad reactions including a few epileptic fits.

Using tables:
Tables are difficult for adaptive browsers to read. Speaking browsers read tables down, whereas they are meant to be read across.

You will either need to simplify the table or provide the data in an alternative format. For example, use a text delimiter such as '/'. Unfortunately, this does not work well as tables for users who are not using speaking browsers. Fortunately, it is easy to produce data in multiple formats using a macro.

If you must use tables for clarity's sake, then be prepared to provide an alternative format such as Excel or table-free HTML. Any table used must have all rows and columns identified and include a table summary.

In the TIP web pages, I used a summary table with links to more detailed financial tables. The detailed summary tables are not accessible, but most users just want to know the project totals available in the simple summary table. An excel version of the data is also available. If you do not have a unique link, such as project ID, which you can use to link the summary table to the detail table, it is easy to create one in the datastep. For example, you can use the SAS _n_ function.

The data step's flexibility made it easy to create clear, interactive HTML data. You can run the step twice, once to create a summary table and a second time to create detailed financial data linking to this summary table. Since each TIP project has an unique project ID, the project ID is a handy ready-made link.

The SAS program created an individual summary for each table by putting the following code in the beginning of the HTML table definition:

```
Put ' <TABLE> '
' Summary= " This table is a
summary of TIP project listings,
project ID, sponsor, description,
and total funding for
&description."  >' /
'<TR Colspan=10> Please click on
the Project ID for full financial
information.</TR><BR>' /
'<TR><TH valign= top>TIP 99
ID</TH>' /
'<TH valign= top>Sponsor</TH>'  /
<!more HTML coding>
```

Each detailed financial table also had a summary statement:

```
put 'summary=" Financial detail
for project:' proj_id '">'
<! print out data>
'<TD align=right>'fy97_tot
comma9.'<BR></TD>'
```

Note the use of column header definitions and '<br>' after each data cell. The break makes tables read more normally in browsers which can not read tables (such as old versions of Netscape).

Acronyms:
Acronym abuse is an epidemic problem in many organizations, especially among government agencies. Specify the expansion of each abbreviation or acronym in a document where it first occurs. It is also helpful to have a glossary of terms on your website. For example

```
The <ABBR title="Barney Fan
Club">BFC</ABBR> is great!
```

Use alt tags for graphics:
If you have a graphic, also put an alt tag describing what the graphics shows. For example:

```
See what's new <img
src="http://nice
place.greatpicture.gif"
alt='new'> 02-31-99<br>
```

Let people out!
Reading down a very long page with no way out can be very tedious. Having a few 'top of page' links in each page will make reading web pages much easier for those using accessible browsers.

Contact information:
Let readers know when the web page was last updated. It is important to give people an email address or phone number in case they have questions. When listing an email address, provide both a link and the email address. For example,

```
Please contact Cécilie if you
have any questions:
cbirner@mtc.ca.gov
```

Not all browsers are set up to send an email and users will be frustrated if there is not email address they can copy.

Get web page testers:
After checking errors and rechecking a page, one soon reaches a point where you no longer see errors or potential problems. Ask librarians (an often underutilized resource), friends or co-workers to visit the web page. It is helpful to give them an 'assignment' which closely matches the information most people will need to get from your site

Try reading your web page with an reading browser and the screen turned off. It is guaranteed to be an eye opening experience.

Above are the basics needed for web page accessibility. Most of these are common sense. If you have more questions, please see the references in the last section.

2 C) Who else benefits?
With the sprawling growth of the web, programmers may be impatient about adhering to new standards. The question "do I have to?" frequently is the initial reaction.

Web pages which are accessible to the blind are by definition more organized. This helps all users. When I did the necessary conversions to make my web site accessible I received showers of praise from all the web page users. The novice users were especially grateful for the changes.

Some automobiles are being equipped with built in web browsers (which hopefully will not be used by the driver). They may use speaking browsers to give information to the driver. This will spur the demand for an accessible internet (and hopefully not increase the accident rate).

The increase in carpal tunnel and other Repetitive Strain Injuries (RSI) is adding to the list of people who can not use keyboard. These users will benefit from accessible web pages, as well.

2 D) Some references for web page accessibility

- The Center for Applied Special Technology (CAST), **http://www.cast.org/bobby/**, contains a program that checks web pages for accessibility and has useful web programming links.
- Designing a More Usable Website, **http://www.trace.wisc.edu/world/web/index.html**, lists a wide range of links to other web accessibility pages.
- JAWS Adaptive Web Server (speaking web browser), **http://www.cs.wustl.edu/~jxh/research/**, contains links to papers about adaptive browsers.
- National Arts and Disability Center, **http://nadc.ucla.edu/dawpi.htm**, is a multi-faceted site which deals with making multi-media art accessible.
- City of San Jose Accessible web page standards, **http://www.ci.san-jose.ca.us/oaacc/disacces.html**, explains the ADA web site standards used in the City of San José's very well received and accessible web site.
- Web Accessibility Initiative, **http://www.w3.org/WAI/**, WAI recently published new accessibility guidelines.

Acronyms Used:

ADA= Americans with Disabilities Act
DBMSCopy: Database Management Software
HTML= HyperText Markup Language
JAWS = Job Access With Speech
MTC= Metropolitan Transportation Commission
MPO= Metropolitan Planning Organization
RSI= Repetitive Strain Injury
TIP= Transportation Improvement Program
URL= Universal Resource Locator (Web Address)

Acknowledgements

♥ This paper is dedicated to Georgia, for her 11 year service and partnership as a guide dog to Sheila, and especially for being so good natured about training her successor.
♥ Thanks to Jean Nandi for her technical and personal support to so many new web users who have disabilities.
♥ Thanks to Sue Hodges for her support and wit and to her service dog Felix.

Contacting the Authors

Cécilie Birner can be reached at: cbirner@mtc.ca.gov (work) or cecilie@igc.org (personal).
Her web page: www.mtc.ca.gov/projects/tip/tipindex.htm

Barbara Rhodes can be reached at brhodes@pacbell.net. She is an Access Technology Trainer and advocate for accessible transportation and web sites for all persons who are disabled. She has been a member of the MTC (Metropolitan Transportation Commission) EDAC (elderly and Disabled Advisory Committee) since its inception in 1991 and is very active in her local community.