Paper 204-27

SAS® USER DOCUMENTATION: WEB PAGE DESIGN MADE EASY

Bruce Gilsen, Federal Reserve Board
Scott Hoenig, Federal Reserve Board

INTRODUCTION

Several hundred pages of SAS-related ® web documents have been developed quickly and easily at the Federal Reserve Board using simple standards and basic HTML features such as unordered lists, tables, and scrollable table of contents pages.

Some users believe they lack the time or web expertise to develop web documents of this type. We disagree. To show how easy it is, we display and review a web document containing a few sections of our Version 8 in-house changes and enhancements guide and the three HTML source files used to create the document. The source files show the simple standards and HTML tools used in several hundred pages of documents. You can copy these source files to a directory and display the actual web document in a browser.

This paper takes into account the recently amended Section 508 of the Rehabilitation Act of 1973, which in part requires Federal agencies to provide comparable access to electronic and information technology to the disabled.

The paper is not an HTML tutorial, but shows how simple HTML features can provide a rich payoff. Basic HTML knowledge is assumed.

WEB DEVELOPMENT STANDARDS

This paper presents some standards for SAS-related web documents. We believe that you can quickly develop web documents using any simple, reasonable set of standards. Our objective is to show our standards as an example of how to proceed, not to tout these specific standards.

SECTION 508

As recently amended, Section 508 of the Rehabilitation Act of 1973 (the federal government's equivalent of the American with Disabilities Act [ADA]), in part requires Federal agencies to provide comparable access to electronic and information technology to the disabled. A detailed discussion of section 508 is beyond the scope of this paper, but compliance is easier if you use simple standards and HTML tools. Standards in this paper motivated by section

508 include the following.

1. <frame> tags have meaningful name and title attributes.

2. Tables have title, summary, and scope attributes.

3. Table headers use the <th> tag for headings.

4. <img> tags have meaningful alt attributes.

Section 508 also requires careful coding of HTML tags. Some web pages with unmatched tags such as missing </tr> or </td> tags in a table can be read as intended by browsers, but not by screen readers employed by the disabled.

For more information, see the Federal IT Initiative home page at http://www.section508.gov or SAS Institute's accessibility page at
http://www.sas.com/solutions/public_sector/accessibility/.

WEB DOCUMENT USED IN THIS PAPER

Figure 1 and **Figure 2** show the web document used in this paper, which is a small subset of a Version 8 in-house changes and enhancements guide available on the Federal Reserve Board's Intranet. It consists of three files included in appendix 1: s8body.htm, s8toc.htm, and s8frame.htm.

To display the web document, copy the three files to a directory (folder) on your PC, name the files as in this paper, and display s8frame.htm in your browser. If you are reading this paper on-line or from the conference proceedings CD-ROM, an easy way to copy the files is to select them (drag your mouse over the file contents), and copy and paste them into files on your PC.

HTML SOURCE FILES

Files and naming conventions

Our web documents consist of the following three HTML source files.

1. a *frame file*, a wrapper file that displays the table of contents file and body file.

2. a *table of contents file* containing links to the body file. The table of contents is scrollable.

3. a *body file* containing text.

The naming convention for the three source files is a prefix followed by either body, toc (table of contents), or frame. The files in this paper are named as follows.

```
frame file:               s8frame.htm
table of contents file:  s8toc.htm
body file:                s8body.htm
```

Frame file

Frame file standards are illustrated in the file s8frame.htm.

```
<frameset cols="25%,*">
<frame src="s8toc.htm" name="tableofcontents" title="table
of contents">
<frame src="s8body.htm" name="maindocument" title="main
document">
</frameset>
```

1. The table of contents file displays in the left-most 25% of the frame, and the body file displays in the rest of the frame. This ratio seems to work well in practice, and users can adjust it with their mouse if they wish.

2. The table of contents file has the title *table of contents* and the frame name *tableofcontents*. The body file has the title *main document* and frame name *maindocument*. Titles are used by screen readers, and frame names are referenced by other HTML elements (though the table of contents name is not referenced elsewhere).

Table of contents file

The table of contents file consists primarily of links to each section of the body file, as in the following statement in the table of contents file s8toc.htm.

```
<a target="maindocument" href="s8body.htm#sect6.1">
6.1. SCAN function can search backward</a><p>
```

maindocument is the name of the frame containing the body file, as assigned in the frame file.

s8body.htm#sect6.1 is the name and location of the body file to display.

Larger documents have a *quick table of contents* at the top, with links to a few major section headings. The table of contents file s8toc.htm includes the following statements.

```
<b>Go directly to a chapter</b><p>
```

```
<a target="maindocument" href=s8body.htm#introbeg">
Intro/Document changes</a><br>

<a target="maindocument"
href="s8body.htm#compatbeg">
Compatibility issues</a><br>

<a target="maindocument" href="s8body.htm#newbeg">
New/improved features</a>
<p><hr>
```

The font size in the table of contents frame is reduced by one unit. This seems to be a good compromise between readability and displaying as much of the table of contents as possible at one time.

```
<font size=-1>
```

Body file

Body file standards are illustrated in the file s8body.htm.

1. We store an entire web document in a single body file, instead of storing each section in a separate body file.

Advantages and disadvantages of a single body file were discussed at length in our SUGI 25 paper (Gilsen and Hoenig 2000). One concern is that a single body file takes longer to load into a browser. Our largest SAS-related web document, 314,000 bytes (307Kbytes), loads without delay. This is another advantage of using simple HTML features. Documents that are very large or contain complex HTML features could take much longer to load.

2. All sections contain anchors. Anchor names are specified as "sect*nnn*, where *nnn* is the section number. For example, the body file s8body.htm contains the following anchor at the beginning of section 6.1.

```
<h3><a name="sect6.1">6.1. SCAN function can
 search backward</a></h3>
```

Section numbering

Advantages and disadvantages of section numbers include the following.

1. Section numbers make it easy to refer users to specific text. For example, users can easily find section 6.2 in the table of contents frame, because section numbers are in ascending order. Users seem to take longer to find specific text by the title or content of a section.

2. A new section should be added in a location that maintains the narrative flow of the document. Usually, this requires renumbering the document. Sometimes, you can avoid renumbering by

adding a new section at the end of the document (for example, adding section 7 after section 6)

adding a new sub-section at the end of a section (for example, adding sub-section 5.3 after sub-section 5.2)

Renumbering can be done manually or with a program written in SAS, Perl, or another language. The SAS program in appendix 2 renumbers the file s8body.htm.

HTML FEATURES

Headings

Our web documents use the following heading tags. If the text displayed by the <h1> tag is too large, you can use <h2>, <h3>, and <h4> tags instead.

<h1>   for major section headings, such as "Part 1"

<h2>   for regular section headings, such as "5"

<h3>   for sub-section headings, such as "5.2"

Displaying code (pre-formatted text)

The following statements in the body file s8body.htm illustrate some standards for displaying SAS code.

```
<pre><blockquote>  /* Illustrate SCAN function */
data two;
 set one;
 /* Scan forward (left to right), just as in Version 6 */
 word3 = scan (var1, 3);

 /* Scan backward (right to left).  This is a new feature
    in Version 8 */
 wordmin2 = scan(var1, -2);
run;</blockquote></pre>
```

1. Code is indented and displayed as pre-formatted text, by surrounding it with the following tags.

start tags: <pre><blockquote>
end tags: </blockquote></pre>

2. Code begins on the same line as the start tags, and ends on the same line as the end tags.

3. Code indentation beyond the <blockquote> tag at the start is done manually, with each indent two additional spaces to the right.

4. Lines containing pre-formatted text (lines between <pre> and </pre>) are limited to 55 columns, as explained in this section.

Two implications of the <pre> tag are as follows.

The reader sees the code exactly as you format it, including indentation and line breaks. This makes the code easier to read in the browser and easier to employ if it is copied to a file or SAS session.

The browser does not format long lines to make them fully visible, so they spill to the right of the display, requiring users to scroll horizontally. This makes a web document hard to read and should be avoided.

We limit lines containing pre-formatted text to 55 columns, to ensure they are completely visible without horizontal scrolling. This limit takes into account the following.

Indentation from the <blockquote> tag makes some columns unavailable.

The body file displays in the right-most 75% of the frame (the table of contents file displays in the rest of the frame).

The PC resolution most users currently employ at the Board is 800 by 600 pixels, with a fixed font in Netscape of Courier New, size 10. This is common, but could vary at other sites.

The 55-column limit adds a few restrictions to web development. These restrictions seem insignificant after repeated use. We must

break up code (and especially comments) more than in typical web documents or SAS applications programs

use hard returns to split the lines

5. No HTML tags are used within the code.

6. Where feasible, the code is self-contained, so that users can copy and paste the code and execute it.

Tables

The following statements in the body file s8body.htm show the first, second, and last row of a two column table.

```
<blockquote>
<table cellpadding=8 title="Maximum length of some
common SAS language elements"
summary="Two column table.  The left column lists SAS
language elements.  The right column has their length.">
<tr valign=top align=left>
<th>SAS Language elements</th>
<th scope="col">Length</th>
```

3

```
</tr>

<tr valign=top><td>Informats</td>
<td>7</td>
</tr>

<tr valign=top>
<td>Label associated with a variable</td>
<td>256</td>
</tr>
</table></blockquote>
```

Some standards for tables are as follows.

1. Tables are indented with the <blockquote> tag.

2. The first row contains column headings, which are left aligned and specified with the <th> tag. Column headings are omitted on occasion if not needed.

3. No borders are used.

4. The cellpadding attribute is used to increase the space between columns. This makes the tables easier to read.

5. The title, summary, and scope attributes are included.

Unordered lists

The following statements in the body file s8body.htm illustrate how unordered lists are employed.

```
<ul>
<li>In the first observation: word3=fg, wordmin2=fg </li>
<p>

<li>In the second observation: word3=p, wordmin2=mno
</li>
</ul>
```

The <p> tag follows all but the last list item. The <p> tags are sometimes omitted if there are only a few items (5 or less), all just a few words long.

What's missing

As previously noted, our web documents are developed using only simple HTML features. Some HTML features not used in our web documents include the following.

1. Colors.

2. Forms.

3. Images, other than *New* and *Updated* images (not shown in the paper) that are used on many Board web pages.

4. JavaScript (drop-down lists, image rollovers, etc.).

5. Java applets.

6. Fonts, with two exceptions.

<font size=-1> in the table of contents frame

Pre-formatted text (text between <pre> and </pre>) is automatically changed to a fixed font

"CHANGES TO THIS DOCUMENT" SECTION

Web documents can be updated quickly and easily. However, frequent updates can actually make web documents *less* effective. If users check a web document periodically for changes, but cannot easily determine what changed since they last checked, they might miss important information or stop checking the document. To avoid this problem, our web documents include a "Changes to this document" section. As shown below, each entry has the date, a brief summary, and a link to the section.

```
<li>May 8, 2000.  Updated the section
<a href="#sect6.1">SCAN function can search backward
</a> to include the results.</li>
```

Entries are ordered from newest to oldest. With a few entries, the order does not seem to matter. When this section becomes larger than one page, it is much easier to read when ordered from newest to oldest.

DOCUMENTING WEB OUTPUT FROM THE SAS SYSTEM

In Version 8 of the SAS System, you can "web enable" your SAS output. This section suggests how to document web output from the SAS system in a web document.

1. The Output Delivery System (ODS) allows you to generate output in a variety of formats, including web pages. To document ODS, you can include the SAS code that generates the output and a link to the output, which is one of the following HTML files.

If your ODS HTML statement includes the FILE= keyword but not the CONTENTS= and FRAME= keywords, link to the file specified by FILE=.

If your ODS HTML statement includes the FILE=, CONTENTS=, and FRAME= keywords, link to the frame file (the file specified by FRAME=).

2. Web output from SAS/GRAPH® software can be

documented as follows.

> The GIF and JPG drivers generate a GIF or JPG file, but not the HTML needed to display the file. In a web document, include the SAS code and a link to the GIF or JPG file, using the <img>tag with the alt attribute. For example, the following statement is a link to iowaann.gif, a GIF file in the sasweb subdirectory of the current directory that displays annual sales for Iowa.

> <img src="sasweb/iowaann.gif" alt="Annual sales for Iowa">

> The HTML, ACTIVEX, and JAVA drivers create one or more GIF files and an HTML file that displays the GIF file(s). In a web document, include the SAS code and a link to the HTML file.

> The WEBFRAME driver creates GIF files and HTML files that display thumbnail and full-size versions of the graphs. In a web document, include the SAS code and a link to the HTML file index.html, which is created by the WEBFRAME driver.

Currently, SAS/GRAPH software drivers such as HTML and WEBFRAME generate <img> tags without alt attributes, and the WEBFRAME driver generates some <frame> tags without title and name attributes. For section 508 compliance, add this information manually or with a program written in SAS, Perl, or another language. A SAS program could use a DATA step analogous to the DATA step in appendix 2.

EDITORS AND HTML SHELLS

Many fine HTML editors are available at little or no cost. However, if an HTML editor is not available, you can develop web documents with the Windows Notepad editor (the web document in this paper was developed using Notepad). This section suggests how Notepad users can quickly insert HTML tags in their documents, a capability normally provided by an HTML editor.

If you repeatedly use the same few HTML features, you can maintain a *shell file*, with shells of frequently used HTML features. The shells include tags and whitespace, but little or no text, and can be quickly copied and pasted into a document.

Our shell file currently includes the following three HTML shells. It also includes tables with three and four columns.

Code:

> <p>

<pre><blockquote> /* Create sample data */

</blockquote></pre>

Unordered list:

>
> <p>
>
> <p>
>
>
>

Two column table:

> <blockquote><table cellpadding=8
> title=""
> summary="" >
> <tr valign=top align=left>
> <th scope="col"> </th>
> <th></th></tr>
>
> <tr valign=top><td></td>
> <td></td>
> </tr>
>
> <tr valign=top><td></td>
> <td></td>
> </tr>
> </table></blockquote>

CONCLUSION

This paper showed that users lacking time or web expertise can quickly and easily develop SAS-related web documents. We reviewed three HTML source files used to create a web document containing a few sections of our Version 8 in-house changes and enhancements guide, and illustrated the simple standards used in hundreds of pages of SAS-related documentation at the Board.

The standards in this paper are overly simple, but provide an example of how to proceed. We believe that by employing any simple, reasonable set of standards, you can quickly develop web documents.

For more information, contact

Bruce Gilsen
Federal Reserve Board, Mail Stop 157
Washington, DC 20551
phone: 202-452-2494
e-mail: gilsenb@frb.gov

Scott Hoenig
Federal Reserve Board, Mail Stop 172
Washington, DC 20551
phone: 202-452-3516

e-mail: scott.hoenig@frb.gov

REFERENCES

Gilsen, Bruce (2000), "In-house SAS User's Guides: Site-Specific Information and Conversion Guides," in the Proceedings of the Twenty-Fifth Annual SAS Users Group International Conference, 25, 1305-1310.

Gilsen, Bruce, and Hoenig, Scott (2000), "SAS User Support and the Intranet: Web Pages, In-house Documentation, and Sample Library," in the Proceedings of the Twenty-Fifth Annual SAS Users Group International Conference, 25, 1015-1024.

ACKNOWLEDGMENTS

TRADEMARK INFORMATION

SAS and SAS/GRAPH software are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries. ® indicates USA registration.

Other brand and product names are registered trademarks or trademarks of their respective companies.

APPENDIX 1. FILES USED IN THIS PAPER

Frame file, s8frame.htm

```
<html>
<head>
<title>SAS Version 8 changes and enhancements: a
summary</title>
</head>
<frameset cols="25%,*">
<frame src="s8toc.htm"
name="tableofcontents" title="table of contents">
<frame src="s8body.htm"
name="maindocument" title="main document">
</frameset>
</html>
```

Table of contents file, s8toc.htm

```
<html>
<head>
<title>SAS Version 8 changes and enhancements: a
summary</title>
</head>
<body>
<font size=-1>

<b>Go directly to a chapter</b><p>

<a target=
"maindocument" href=s8body.htm#introbeg">
Intro/Document changes</a><br>

<a target=
"maindocument" href="s8body.htm#compatbeg">
Compatibility issues</a><br>

<a target=
"maindocument" href="s8body.htm#newbeg">
New/improved features</a>
<p><hr>

<b>Intro/Document changes:</b><p>

<a target="maindocument" href="s8body.htm#sect1">
1. Introduction</a><p>

<a target="maindocument" href="s8body.htm#sect2">
2. Changes to this document</a><p>

<hr>
<b>Compatibility Issues:</b><p>

<hr>
<b>New and improved features:</b><p>

<a target="maindocument" href="s8body.htm#sect5">
5. Changes to variable names and other
SAS names</a><p>

<a target=
"maindocument" href="s8body.htm#sect5.1">
5.1. Maximum length of variable names</a><p>

<a target=
"maindocument" href="s8body.htm#sect5.2">
5.2. Maximum length of some common SAS language
elements</a><p>

<a target="maindocument" href="s8body.htm#sect6">
6. Miscellaneous enhancements</a><p>

<a target=
"maindocument" href="s8body.htm#sect6.1">
```

6.1. SCAN function can search backward<p>

</body>
</html>

Body file, s8body.htm

```
<html>
<head>
<title>SAS Version 8 changes and enhancements: a
summary</title>
</head>
<body>
<b><font size=+2><p align=center>
SAS Version 8 changes and enhancements: a summary

<p align=center>
August 2, 2000
</font></b>
<hr>

<h1><a name="introbeg">Part I.  Introduction and changes
to this document</a></h1>

<h2><a name="sect1">1. Introduction</a></h2>

<h2><a name="sect2">2.  Changes  to  this
document</a></h2>

<p>
<ul>
<li>May 8, 2000.  Updated the section
<a  href="#sect6.1">SCAN  function  can  search
backward</a> to include the results.</li><p>

<li>March 6, 2000.  Added the section
<a  href="#sect5.1">Maximum  length  of  variable
names</a>.</li><p>

<li>October 27, 1999.  Initial version of this document
released.</li>
</ul>

<h1><a  name="compatbeg">Part  II.     Compatibility
issues</a></h1>

<p>
<h1><a name="newbeg">Part III.  New and
improved features</a></h1>

<h2><a name="sect5">5. Changes to variable names and
other SAS names</a></h2>

<h3><a name="sect5.1">5.1. Maximum length of variable
names</a></h3>
```

```
<p>
The maximum length of variable names has increased from
8 characters to 32.  The first character must be a letter or
underscore.  Subsequent characters can be letters,
numbers, or underscores.

<p>
<h3><a name="sect5.2">5.2. Maximum  length  of  some
common SAS language elements</a></h3>
<p>
<blockquote>
<table cellpadding=8 title="Maximum  length  of  some
common SAS language elements"
summary="Two column table.  The left column lists SAS
language elements.  The right column has their length.">
<tr valign=top align=left>
<th>SAS Language elements</th>
<th scope="col">Length</th>
</tr>

<tr valign=top><td>Informats</td>
<td>7</td>
</tr>

<tr valign=top>
<td>Formats, filerefs, and librefs</td>
<td>8</td>
</tr>

<tr valign=top>
<td>Names of: arrays, catalog entries, catalogs, data sets,
integrity  constraints,  macros,  macro  variables,  SAS
variables</td>
<td>32</td>
</tr>

<tr valign=top>
<td>Label associated with a variable</td>
<td>256</td>
</tr>
</table></blockquote>

<p>
<h2><a name="sect6">
6. Miscellaneous enhancements</a></h2>

<p>
<h3><a name="sect6.1">6.1. SCAN function can search
backward</a></h3>
<p>
In Version 8, the SCAN function can scan from right to left
by specifying a negative number as an argument.

<p>
<pre><blockquote>  /* Create sample data */
data one;
 length var1 $12;
```

7

```
  input var1 $12.;
  cards;
ab de fg hi
jkl mno p
;run;

data two;
  set one;
  /* Scan forward (left to right), just as in Version 6 */
  word3 = scan (var1, 3);

  /* Scan backward (right to left).  This is a new feature
     in Version 8 */
  wordmin2 = scan(var1, -2);
run;</blockquote></pre>
```

```
<p>
Results are as follows.

<ul>
<li>In the first observation: word3=fg, wordmin2=fg </li>
<p>
<li>In the second observation: word3=p,
wordmin2=mno</li>
</ul>

<hr>
<b><i>This FedWeb page is available to staff throughout
the Federal Reserve System.</I></b>
<br>Maintained by Bruce Gilsen, <i>Board of
Governors</I>
<br><b>Last update: 8/2/00</b>
</body>
</html>
```

APPENDIX 2.　　SAS PROGRAM TO RENUMBER THE BODY FILE

The following SAS program renumbers the body file s8body.htm when a new section is added. It could be enhanced to renumber the body file when a sub-section is added or to renumber the table of contents file.

Before running the program, add text like the following at the location where the new section will be added. Use _newsec_ instead of the section number.

```
    <h2><a name="sect_newsec_">_newsec_.
    SAS/GRAPH software changes</a></h2>


data _null_;
    length intext $200;

        /* HTML files to read and write */
    infile 's8body.htm' length=leng ;
    file 's8bodyn.htm' ;

        /* make secnum at least as large as number of
        sections of the document after updating */
array secnum (10) $ _temporary_
    ("1" "2" "3" "4" "5" "6" "7" "8" "9" "10");

        /* nadds is number of sections added so far */
        /* i is number of sections encountered so far */
retain nadds 0 i 0;

        /* read record, check for section tag */
input  intext $varying200. leng;
temp1 = substr(intext,1,4) ;
if temp1 = "<h2>" then i+1;

        /* add a new section number? */
if temp1 = "<h2>" and index(intext,"_newsec_")
ne 0 then do;
    intext = tranwrd(intext,
            "_newsec_", trim(secnum(i)));
    nadds + 1;
    leng = length(intext);  /* line is shorter now */
end;

        /* update section or sub-section number? */
else if temp1 in("<h3>","<h2>") and nadds gt 0 then
do;
    intext = tranwrd(intext,
            trim("sect" ||secnum(i-nadds)),
            "sect" || trim(secnum(i)));
    intext = tranwrd(intext,
            ">" || trim(secnum(i-nadds)) || ".",
            ">" || trim(secnum(i)) || ".");
            /* line could be longer now */
    leng = length(intext);
end;

    /* write out the record */
    put intext $varying200. leng;
run;
```
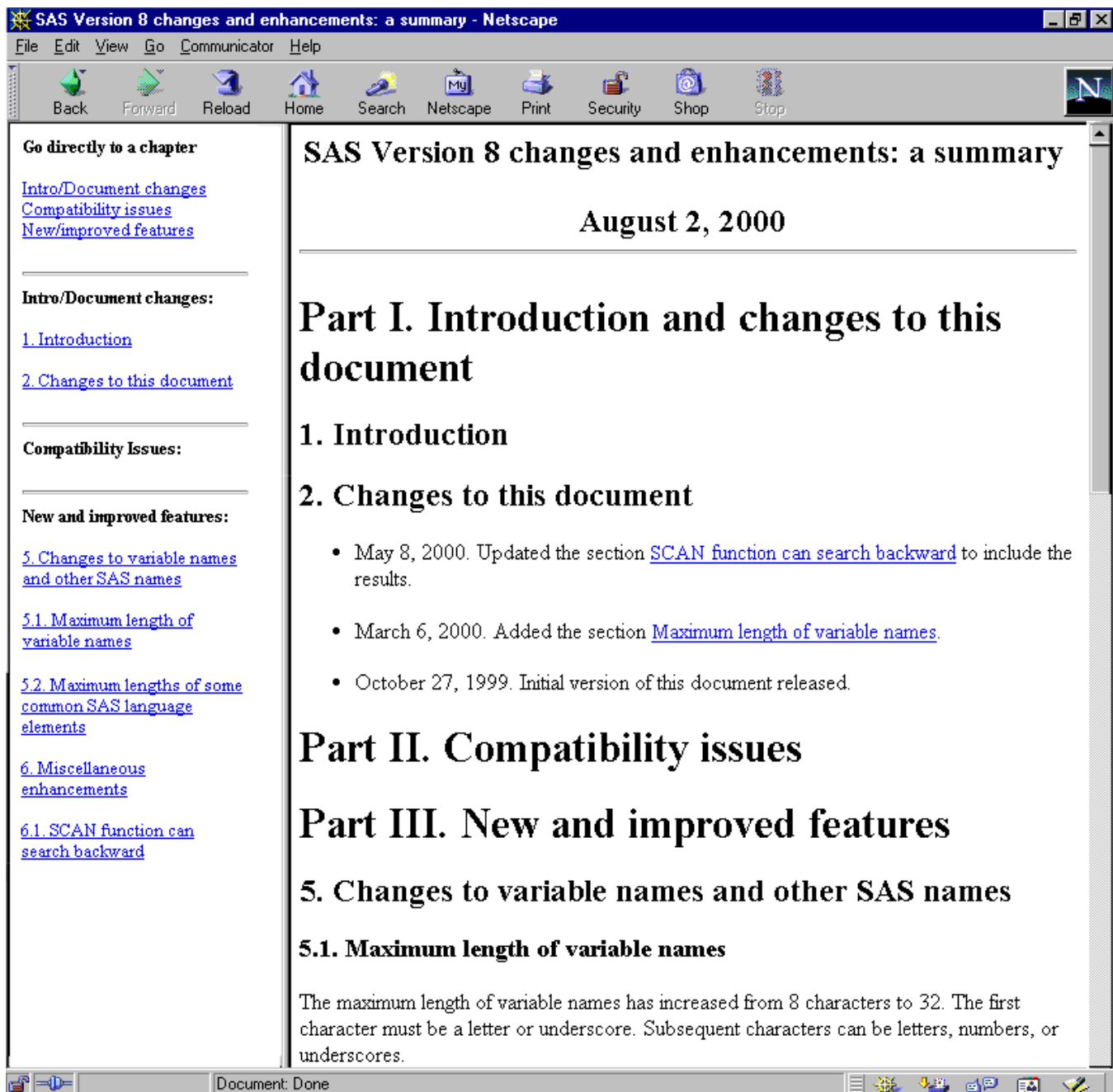
SAS Version 8 changes and enhancements: a summary - Netscape

File Edit View Go Communicator Help

Back Forward Reload Home Search Netscape Print Security Shop Stop

Go directly to a chapter

Intro/Document changes
Compatibility issues
New/improved features

Intro/Document changes:

1. Introduction

2. Changes to this document

Compatibility Issues:

New and improved features:

5. Changes to variable names and other SAS names

5.1. Maximum length of variable names

5.2. Maximum lengths of some common SAS language elements

6. Miscellaneous enhancements

6.1. SCAN function can search backward

SAS Version 8 changes and enhancements: a summary

August 2, 2000

Part I. Introduction and changes to this document

1. Introduction

2. Changes to this document

- May 8, 2000. Updated the section SCAN function can search backward to include the results.

- March 6, 2000. Added the section Maximum length of variable names.

- October 27, 1999. Initial version of this document released.

Part II. Compatibility issues

Part III. New and improved features

5. Changes to variable names and other SAS names

5.1. Maximum length of variable names

The maximum length of variable names has increased from 8 characters to 32. The first character must be a letter or underscore. Subsequent characters can be letters, numbers, or underscores.

Document: Done

Figure 1. The web document (continued in Figure 2)

```
SAS Version 8 changes and enhancements: a summary - Netscape
File   Edit   View   Go   Communicator   Help
Back   Forward   Reload   Home   Search   Netscape   Print   Security   Shop   Stop
```

5.2. Maximum length of some common SAS language elements

SAS Language elements	Length
Informats	7
Formats, filerefs, and librefs	8
Names of: arrays, catalog entries, catalogs, data sets, integrity constraints, macros, macro variables, SAS variables	32
Label associated with a variable	256

6. Miscellaneous enhancements

6.1. SCAN function can search backward

In Version 8, the SCAN function can scan from right to left by specifying a negative number as an argument.

```
    /* Create sample data */
data one;
  length var1 $12;
  input var1 $12.;
  cards;
ab de fg hi
jkl mno p
;run;

data two;
  set one;
  /* Scan forward (left to right), just as in Version 6 */
  word3 = scan (var1, 3);

  /* Scan backward (right to left).  This is a new feature
     in Version 8 */
  wordmin2 = scan(var1, -2);
run;
```

Results are as follows.

- In the first observation: word3=fg, wordmin2=fg

- In the second observation: word3=p, wordmin2=mno

This FedWeb page is available to staff throughout the Federal Reserve System.
Maintained by Bruce Gilsen, *Board of Governors*
Last update: 8/2/00