

ODS HTML Evolution: HTML that Scrolls, Panels, Floats, Reads, and Integrates

Eric Gebhart, SAS Institute Inc., Cary, NC

ABSTRACT

The ODS HTML destination has been around a long time and has gone through a number of changes. There are numerous flavors of HTML that can be generated using from tagsets available, but it can be frustrating when you want the features of two or more different tagsets. There are HTML tagsets with scrolling, paneling, and floating tables of contents. There are Web site-enabled tagsets that allow the insertion of JavaScript. There are even a few options that can control Section 508 compliance. This paper will show an HTML tagset that combines these many features into one tagset and then illustrates how to use it.

INTRODUCTION

All of the examples in this paper were created using SAS 9.2 and a new version of the HTML tagsets that was created as an exploration of what the future holds for the next major version of the ODS HTML destination, HTML5. HTML5 is the new HTML standard as defined by the W3C.

The ODS HTML destination was the first destination that I created back in 1995. HTML was a fast moving target back then. The browsers worked very inconsistently, and getting a web page to look the same in Mosaic, Internet Explorer, and Opera was a real challenge. The result was very plain HTML that hugged the middle of the road.

The Internet and HTML has changed a lot since then. HTML has slowed its evolution and become more standardized.

Mosaic turned into Netscape and then Firefox, and Opera faded. Internet Explorer languished and became one of the worst non-compliant browsers. Firefox, Safari, and Google Chrome are now the best browsers, and all three of them are more W3C compliant than any browser that came before.

The good news is that because of the W3C and more W3C-compliant browsers, HTML looks and acts more consistently across browsers than ever before. It's time that the ODS HTML destination got a little love.

The HTML destination has changed over the years. In the beginning, the HTML destination was written entirely in C-code. It was immutable between releases of SAS. It was very important for it to work well in a very mediocre way. Internet Explorer was the target browser with attention to Netscape secondary. Mostly, Netscape worked well, so it wasn't too much of a problem. The HTML3 standard wasn't really settled, and the browser wars were in full swing.

ODS HTML3

The HTML was quite simple but very verbose because it used font tags, and it had no concept of a style sheet. Frames were used to create a table of contents and pages. These days, frames are a real navigational hazard and almost never used in well designed Web sites.

```
ods html3 file="exampleb.html" frame="example.html" contents="examplec.html"
pages="examplep.html";

proc print data=sashelp.class;
run;

proc standard print data=sashelp.class;
run;

proc report data=sashelp.class nowd;
run;

ods html3 close;
```

The output for this program is shown in Figure 1. There is great potential to create HTML that is so much better than this.

Figure 1. ODS HTML3 Output

The screenshot shows a web browser window titled 'SAS Output Frame' displaying the output of a SAS procedure. The browser's address bar shows the file path: `file:///Users/eric/pp/HTML_evolution/example.html`. The page content is organized into a sidebar on the left and a main content area on the right.

Table of Contents (Left Sidebar):

- 1. The Print Procedure
 - [Data Set SASHELP.CLASS](#)
- 2. The Standard Procedure
 - [Standardized data set variables.](#)
- 3. The Report Procedure
 - Detailed and/or summarized report
 - [Table 1](#)

Table of Pages (Left Sidebar):

- 1. The Print Procedure
 - [Page 1](#)
- 2. The Standard Procedure
 - [Page 2](#)
- 3. The Report

Main Content Area:

The SAS System

The STANDARD Procedure

| Name | Mean | Standard Deviation | N |
|--------|------------|--------------------|----|
| Age | 13.315789 | 1.492672 | 19 |
| Height | 62.336842 | 5.127075 | 19 |
| Weight | 100.026316 | 22.773933 | 19 |

The SAS System

| Name | Sex | Age | Height | Weight |
|---------|-----|-----|--------|--------|
| Alfred | M | 14 | 69 | 112.5 |
| Alice | F | 13 | 56.5 | 84 |
| Barbara | F | 13 | 65.3 | 98 |
| Carol | F | 14 | 62.8 | 102.5 |
| Henry | M | 14 | 63.5 | 102.5 |
| James | M | 12 | 57.3 | 83 |
| Jane | F | 12 | 59.8 | 84.5 |
| Janet | F | 15 | 62.5 | 112.5 |
| Jeffrey | M | 13 | 62.5 | 84 |
| John | M | 12 | 59 | 99.5 |

ODS TAGSETS

With SAS 8.0, the ODS HTML destination added style sheets. It was now possible to create a separate cascading style sheet file or have an embedded style sheet at the top of the single HTML file. With the use of style sheets, font tags were no longer needed, but that behavior remains to this day if style sheets are not in use.

With the release of SAS 8.2, the ODS HTML destination became the ODS HTML3 destination, and a new HTML4 destination was created. This new destination was created with a tagset and used the new ODS MARKUP destination. The ODS HTML4 destination is really the incredibly versatile ODS MARKUP destination. The new HTML4 destination changed a few things. First, there are no font tags. All stylistic controls are done with style sheets or an inline style tag.

For the first time, W3C compliance was more important than how the browsers behaved. Netscape and Firefox were more closely targeted because of their higher compliance with the W3C. Safari was new and received more attention than Internet Explorer. Internet Explorer was the least compliant and most ill-behaved of all the browsers, and therefore, fell to a secondary status.

Section 508 compliance became very important at this time, and so it became a basic part of the HTML4 destination as well as the table templates used by the procedures. It became possible to specify summaries, abbreviations, acronyms, and long descriptions for headings, tables, and graphs. Default values were provided in some cases but it is preferable that these attributes be specified in the table template or through the procedure.

Because HTML4 was defined by a tagset, updates to the html destination were no longer tied to releases of SAS. As the demand for better Section 508 compliance grew, the tagset changed and grew with it. The table of contents became more capable with more browsers and not just with Internet Explorer, as it had been in the past.

It was also now possible to create entirely new HTML destinations with new capabilities just by creating a new tagset. HTML was the first ODS markup destination to experience this sort of growth. The result was a number of different tagsets that filled specific needs. There is the slider tagset which uses graph bars for percentage columns, and the sidebar tagset with stacked graph bars for survey results. The HTMLscroll tagset added scrolling tables, and the HTMLpanel tagset added the ability to panel output together side by side and beyond. The Website tagset made it easier to integrate HTML into Web sites automatically by allowing the inclusion of other HTML files in the construction of its output files. Most recently, the TableEditor tagset allows scrolling, sorting, and rearranging of columns in an HTML table, although currently, it only works with Internet Explorer.

THE OPTIONS OPTION

With the release of SAS 9.1.3, came a new option on the ODS MARKUP statement, the OPTIONS option. The ExcelXP tagset was created about this same time, and because new behaviors were needed, this new option was added. Using options was a much better solution than creating a new tagset for each new functionality. After SAS 9.1.3, options were added to various tagsets, and HTML tagset was one of the first tagset to benefit from these options although most of these options had to do with Section 508 compliance, not major functionality like scrolling, editing, or panelling.

With options being added to the tagsets, it became each tagset's responsibility to create documentation explaining all of its options. The Doc option is supported by all tagsets that have options.

Running the following program with SAS 9.2 produces the help text for the HTML4 tagset.

```
ods html file="t1.html" options(doc='help');
ods html close;
```

The help text generated by these statements is printed to the SAS log. Most of these options are for Section 508 compliance or, in the case of the CSS_TABLE option, are used to turn on behavior that is more W3C compliant but which cannot be the default because some browsers behave badly.

```
=====
The tagsets.HTML4 Tagset Help Text.

This Tagset/Destination creates output in HTML format.

=====

These are the options supported by this tagset.

Sample usage:

ods html options(doc='Quick');

ods html options(header_dots='yes' summary_byvars='yes');

Doc:  No default value.
     Help: Displays introductory text and options.
     Quick: Displays available options.

header_data_associations:  Default Value 'no'
                          Associates data cells and header cells by adding an ID attribute
                          to each header cell and listing the IDs of associated headers in
                          a HEADERS attribute added to each data cell. (PROC REPORT only)

header_dots:  Default Value 'no'
             Puts hidden dots before the text in all table headers

summary_as_caption:  Default Value 'no'
                   Causes a table caption to be created from the table summary.

summary_byvars:  Default Value 'no'
                Adds a list of by variable names to the table summary

summary_byvals:  Default Value 'no'
```

Add the values of the by variables along with the names in the table summary
This works with summary byvars but not without.

```
summary:    Default Value ''
            Text for the table summary

summary_prefix:    Default Value ''
            Text to place at the beginning of table summary

summary_suffix:    Default Value ''
            Text to place at the end of table summary

page_break:    Default Value 'yes'
            If yes, the usual pagebreak style attribute will be used to create
            what becomes the page separator.  Usually that is an HR line.
            If No, then no pagebreak will be output.
            If anything else, the value given will be output as the pagebreak.

css_table:    Default Value 'no'
            If yes, the table tags will not have any style attributes but will
            rely entirely on the table style defined in the stylesheet.  This may
            create undesirable table rendering in some browsers.

percentage_font_size:    Default Value 'no'
            If yes, titles and footnotes will allow font sizes to be specified as
            a percentage of the font size.
```

=====

These options allow considerable control of the HTML, but the HTML destination never really recovered from the splintering caused by the creation of all of the new tagsets that occurred early on. All of those other tagsets still remain and have not been merged back into the HTML tagset with the appropriate options. It's time the HTML4 tagset received a little love.

ODS HTML4

More recently, there has been a growing desire to have a table of contents without the use of frames. With HTML5 becoming a new standard, we decided it was time to show our affection for HTML and bring it out of the shadow cast by Excel. As an experiment, we incorporated a new scheme to have a better table of contents without frames. The aim is to incorporate these changes, remain W3C compliant and, hopefully, have everything still work reasonably well in all the browsers. All of this has to happen while also preserving the current behavior of the ODS HTML destination.

The table of contents was so nice that I decided to try to incorporate as many of the features as possible from the various HTML tagsets that we had created along the way. The functionality of the htmlscroll, website, and htmlpanel tagsets were incorporated into this new tagset. The slider and survey tagsets proved to be too invasive to do at this time, but may be incorporated in the future.

This new HTML tagset has much more extensive documentation. The documentation can be obtained with the following statement, after the new tagset has been run.

```
ods html file="example.html" options(doc='help');
```

The resulting documentation is written to the SAS log and looks like this:

=====

The tagsets.HTML4 Tagset Help Text.

This Tagset/Destination creates output in HTML format.

=====

These are the options supported by this tagset.

Sample usage:

```
ods html options(doc='Quick');
```

```
ods html options(header_dots='yes' summary_byvars='yes');  
ods tagsets.htmlpanel file='test.html' options(doc='Quick');
```

```
ods tagsets.htmlpanel options(panelColumns='3'  
                             embedded_titles='No'  
                             bylines='No'  
                             bylabels='No');
```

See Also:

<http://support.sas.com/rnd/base/topics/odsmarkup/>

Doc: No default value.

Help: Displays introductory text and options.

Quick: Displays available options.

body_toc: Default Value 'no'
Creates a floating table of contents in the main output file.

toc_type: Default Value 'tree'
menu: display as pulldown menus.
tree: display as a hierarchical list.
Determines whether the table of contents will show as pulldown
menus or as a tree.

header_data_associations: Default Value 'no'
Associates data cells and header cells by adding an ID attribute
to each header cell and listing the IDs of associated headers in
a HEADERS attribute added to each data cell. (PROC REPORT only)

header_dots: Default Value 'no'
Puts hidden dots before the text in all table headers

head_file: Default Value 'none'
File to read in and put in place of the head section of a document
A foot file must also be specified as well as the notop option on the
ods statement

foot_file: Default Value 'none'
File to read in and put in place of the foot section of a document
A head file must also be specified as well as the notop option on the
ods statement

Panelling: Default Value 'no'
Turn on automatic panelling for graph procedures.

scroll_tables: Default Value 'no'
Causes tables to scroll vertically.

scroll_batch_size: Default Value '8'
Number of observations to display in a scrolling table.
This can also be set through a macro variable of the same name.
This value can also be provided on the url when loading the output.
http://www.foo.com/mytables.html?batch_size=3;

scroll_control_images: Default Value 'no'
Causes the scroll controls to use images rather than ascii characters
By default these images are named, top.png, pageup.png, up.png, down.png,
pagedown.png, and bottom.png. These images and more information about
table scrolling can be found here. <http://support.sas.com/rnd/base/ods/odsmarkup/htmlscroll.html>

scroll_control_text: Default Value 'no'
Causes the scroll controls to use textual words, top, pageup, up, down, page
down, and bottom rather than ^, ^, +, -, v, vv.

scroll_long_table_length: Default Value (2 * scroll_batch_size)+1
Number of rows a table must have before scrolling is turned on for that table.

This can also be set through a macro variable of the same name.
This value can also be provided on the url when loading the output.
http://www.foo.com/mytables.html?batch_size=3&long_table_length=20;

`summary_as_caption:` Default Value 'no'
Causes a table caption to be created from the table summary.

`summary_byvars:` Default Value 'no'
Adds a list of by variable names to the table summary

`summary_byvals:` Default Value 'no'
Add the values of the by variables along with the names in the table summary
This works with `summary_byvars` but not without.

`summary:` Default Value ''
Text for the table summary

`summary_prefix:` Default Value ''
Text to place at the beginning of table summary

`summary_suffix:` Default Value ''
Text to place at the end of table summary

`page_break:` Default Value 'yes'
If yes, the usual pagebreak style attribute will be used to create what becomes the page separator. Usually that is an HR line.
If No, then no pagebreak will be output.
If anything else, the value given will be output as the pagebreak.

`css_table:` Default Value 'no'
If yes, the table tags will not have any style attributes but will rely entirely on the table style defined in the stylesheet. This may create undesirable table rendering in some browsers.

`percentage_font_size:` Default Value 'no'
If yes, titles and footnotes will allow font sizes to be specified as a percentage of the font size.

=====

Additional inline format functions that may be useful.
These functions may also be called with the ods html event= statement

`inline_image:`
~{inline_image test.jpg}
This function will place an image in the current output.

`include_file:`
~{include_file test.txt}
This function will read an external file and place it's contents in the output.

`include_newlines:`
~{include_newlines yes}
This function will cause `include_file` to place a break at the end of each line that it reads from external file so that newlines are preserved.

=====

Paneling Options

`PanelColumns:` Default Value '2'
Current Value: 2
How many columns of panels to create when doing automatic or semi-automatic paneling. The default is to put everything 2 up.
Also available as a macro variable

`PanelBorder:` Default Value '0'

Current Value: 0
 This is border width, 0 means no borders. Bigger numbers make it wider.
 Also available as a macro variable.

Embedded_Titles: Default Value 'No'
 Current Value: No
 If 'Yes' titles and footnotes will appear inside each panel as if each panel where a miniature page. If 'No' the titles and footnotes appear once above and below the entire panel grouping.
 Also available as a macro variable.

ByLabels: Default Value 'Yes'
 Current Value: yes
 If 'No' remove the variable name and equals sign from the bylines.
 Also available as a macro variable.

ByLines: Default Value 'Yes'
 Current Value: yes
 If 'No' do not generate any bylines.
 Also available as a macro variable.

Event Controls

Semi-automatic and manual panelling is controlled by calling events from the ODS statement. They are started and stopped by specifying start and finish as arguments to the event. Row_panel and column_panels can be nested to create complex layout configurations

Automatic panelling must be turned on with the panelling option.

Panel:
 This event starts and stops semi-automatic panelling. All output between the start and finish will be panelled according to the settings of the option variables above.

```
ODS tagsets.htmlpanel event=panel(start);
/* Run Procs here */
ODS tagsets.htmlpanel event=panel(finish);
```

row_panel:
 This event starts a row-wise panel. Each output object will be placed horizontally while this panel is active.

```
ODS tagsets.htmlpanel event=row_panel(start);
/* Run Procs here */
ODS tagsets.htmlpanel event=row_panel(finish);
```

column_panel:
 This event starts a column-wise panel. Each output object will be placed vertically while this panel is active.

```
ODS tagsets.htmlpanel event=column_panel(start);
/* Run Procs here */
ODS tagsets.htmlpanel event=column_panel(finish);
```

A more complex example.

```
|-----|
||-----||-----|
||         ||         ||
||         ||         ||
||         ||         ||
||         ||         ||
||         ||         ||
||         ||         ||
||-----||-----|
```

|-----|

```
ODS tagsets.htmlpanel event=row_panel(start);

ODS tagsets.htmlpanel event=column_panel(start);
/* Run Procs here */
ODS tagsets.htmlpanel event=column_panel(finish);

/* Run a graph to be placed in the middle

ODS tagsets.htmlpanel event=column_panel(start);
/* Run Procs here */
ODS tagsets.htmlpanel event=column_panel(finish);

ODS tagsets.htmlpanel event=row_panel(finish);
```

=====

The old frames and table of contents are still available with this new tagset, but the new floating table of contents is much nicer. However, it must be turned on. The option that does this is called BODY_TOC for “body file table of contents”. The default type for the table of contents is a tree much like what has always been available. The difference is that it floats on the page, and there are no frames.

Here's a simple example:

```
ods html file="example2.html" style=seaside
  options(body_toc='yes');

proc print data=sashelp.class;
run;

proc print data=sashelp.class;
run;

ods html close;
```


Figure 2. Floating Table of Contents

The screenshot shows a SAS Output window with a floating table of contents on the left and two data tables on the right. The table of contents is a vertical list with three main sections: 1. Print, 2. Standard, and 3. Report. Each section has a sub-item: 1. Print has 'Data Set SASHELP.CLASS'; 2. Standard has 'Standardized data set variables'; 3. Report has 'Detailed and/or summarized report' with a sub-item 'Table 1'. The right side of the window displays two tables. The first table is titled 'The SAS System' and 'The STANDARD Procedure', and contains three columns: Name, Mean, Standard Deviation, and N. The second table is titled 'The SAS System' and contains five columns: Name, Sex, Age, Height, and Weight. Both tables list data for 14 individuals.

Table 1: Standardized data set variables

| Name | Mean | Standard Deviation | N |
|--------|------------|--------------------|----|
| Age | 13.315789 | 1.492672 | 19 |
| Height | 62.336842 | 5.127075 | 19 |
| Weight | 100.026316 | 22.773933 | 19 |

Table 2: Detailed and/or summarized report

| Name | Sex | Age | Height | Weight |
|---------|-----|-----|--------|--------|
| Alfred | M | 14 | 69 | 112.5 |
| Alice | F | 13 | 56.5 | 84 |
| Barbara | F | 13 | 65.3 | 98 |
| Carol | F | 14 | 62.8 | 102.5 |
| Henry | M | 14 | 63.5 | 102.5 |
| James | M | 12 | 57.3 | 83 |
| Jane | F | 12 | 59.8 | 84.5 |
| Janet | F | 15 | 62.5 | 112.5 |
| Jeffrey | M | 13 | 62.5 | 84 |
| John | M | 12 | 59 | 99.5 |
| Joyce | F | 11 | 51.3 | 50.5 |
| Judy | F | 14 | 64.3 | 90 |
| Louise | F | 12 | 56.3 | 77 |

Adding the TOC_TYPE= option and setting it to menu will change the table of contents into an expanding series of menus.

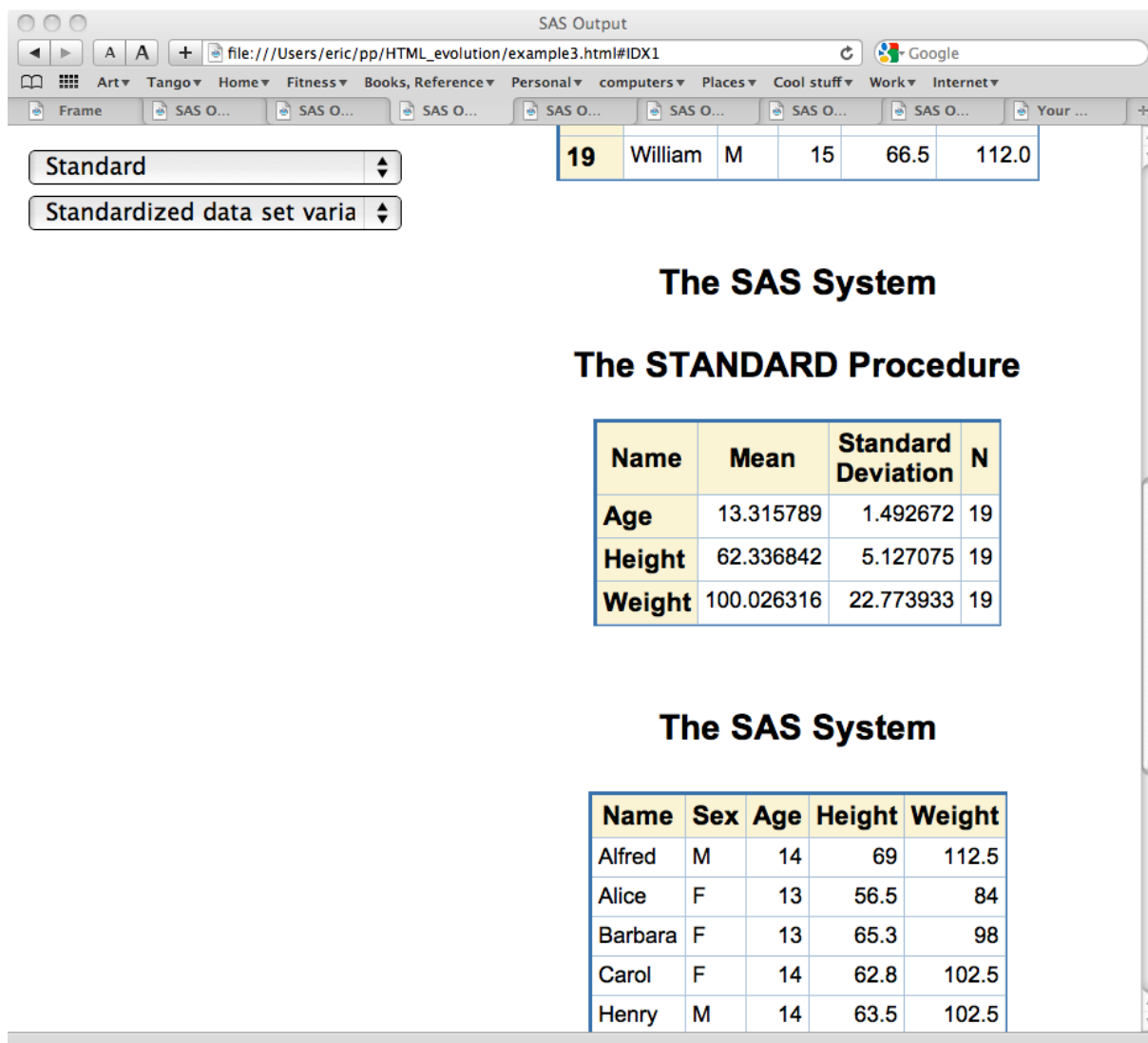
```
ods html file="example3.html" options(body_toc='yes' toc_type='menu');

proc print data=sashelp.class;
run;

proc print data=sashelp.class;
run;

ods html close;
```

Figure 3. Table of Contents Using Menus



Adding scrolling tables to the mix is just a matter of adding one more option, the SCROLL_TABLES= option.

```
ods html file="example4.html" style=Journal
      options(body_toc='yes' toc_type='menu'
              scroll_tables='yes');

proc print data=sashelp.class;
run;

proc standard print data=sashelp.class;
run;

proc report data=sashelp.class nowd;
run;

ods html close;
```

By default, the controls for scrolling are simple characters, but they can be changed to words or images.

Figure 4. Scrolling Tables

The screenshot shows a web browser window titled "SAS Output" with a Google search bar. Below the browser window, there is a report titled "Table 1" with the following data:

| Name | Mean | Standard Deviation | N |
|--------|------------|--------------------|----|
| Age | 13.315789 | 1.492672 | 19 |
| Height | 62.336842 | 5.127075 | 19 |
| Weight | 100.026316 | 22.773933 | 19 |

Below the report, there is a section titled "The SAS System" with a table of data and scrolling controls:

| Name | Sex | Age | Height | Weight | Scrolling Controls |
|---------|-----|-----|--------|--------|--------------------|
| Alfred | M | 14 | 69 | 112.5 | ^^ ^ + |
| Alice | F | 13 | 56.5 | 84 | 1-8 of 19 |
| Barbara | F | 13 | 65.3 | 98 | - |
| Carol | F | 14 | 62.8 | 102.5 | v vv |
| Henry | M | 14 | 63.5 | 102.5 | |
| James | M | 12 | 57.3 | 83 | |
| Jane | F | 12 | 59.8 | 84.5 | |
| Janet | F | 15 | 62.5 | 112.5 | |

This default scrolling behavior has a scroll batch size of eight and controls that are created from text characters. The batch size can be changed with the `SCROLL_BATCH_SIZE=` option, and the controls can be changed to images or words with the `SCROLL_CONTROL_IMAGES=` or `SCROLL_CONTROL_TEXT=` options. Additionally, there is a setting that controls the size of the tables to which scrolling will be added. This is essentially the minimum table size that scrolling will be applied to. The default table size is twice the `SCROLL_BATCH_SIZE` + 1, or 17 with the default batch size. The batch size and long table length can also be set on the URL, so that these settings can be manipulated after the HTML page has already been created.

Here is the help text for these options:

```
scroll_batch_size:   Default Value '8'
                    Number of observations to display in a scrolling table.
                    This can also be set through a macro variable of the same name.
                    This value can also be provided on the url when loading the output.
http://www.foo.com/mytables.html?batch\_size=3";

scroll_control_images:   Default Value 'no'
                    Causes the scroll controls to use images rather than ascii characters
                    By default these images are named, top.png, pageup.png, up.png, down.png,
                    pagedown.png, and bottom.png. These images and more information about table
                    scrolling can be found here.
http://support.sas.com/rnd/base/ods/odsmarkup/htmlscroll.html
```

`scroll_control_text:` Default Value 'no'
Causes the scroll controls to use textual words, top, pageup, up, down, page down, and bottom rather than ^, ^, +, -, v, vv.

`scroll_long_table_length:` Default Value (2 * scroll_batch_size)+1
Number of rows a table must have before scrolling is turned on for that table.
This can also be set through a macro variable of the same name.
This value can also be provided on the url when loading the output.
http://www.foo.com/mytables.html?batch_size=3&long_table_length=20;

Here is an example that uses images for the scrolling controls:

```
ods html file="example5.html" style=Journal
  options(body_toc='yes' toc_type='menu'
    scroll_tables='yes' scroll_control_images='yes');

proc print data=sashelp.class;
run;

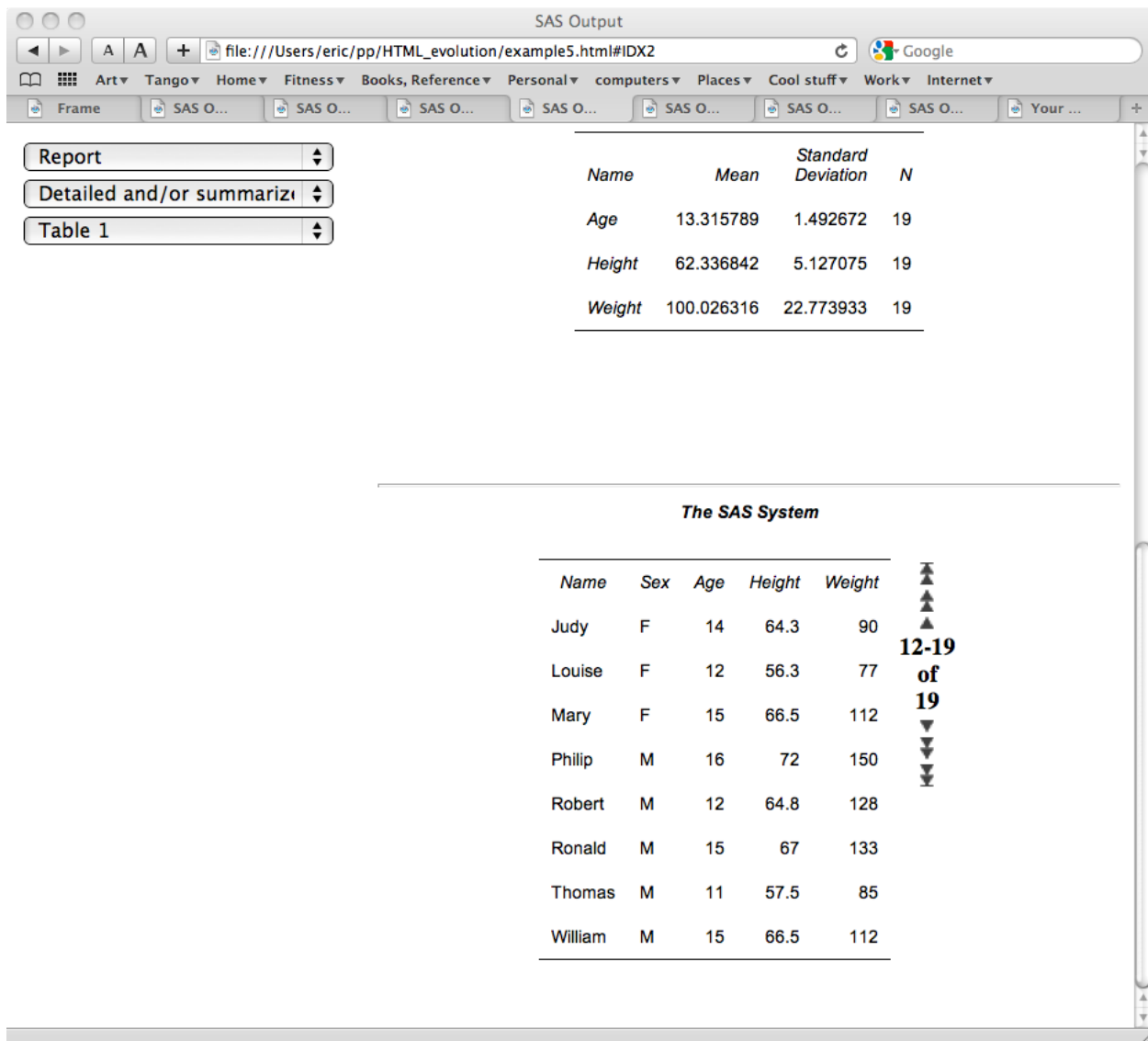
proc standard print data=sashelp.class;
run;

proc report data=sashelp.class nowd;
run;

ods html close;
```

The images are a nice improvement over the ASCII art that is used by default.

Figure 5. Images as Scrolling Controls



Using a URL similar to the following will cause the batch size to change from 8 to 5.

```
file:///Users/eric/pp/HTML_evolution/example5.html?batch_size=5';
```

Figure 6 shows what the same output file looks like with the new batch size.

Figure 6. Changing the Batch Size for Scrolling

The SAS System

| Obs | Name | Sex | Age | Height | Weight |
|-----|---------|-----|-----|--------|--------|
| 1 | Alfred | M | 14 | 69.0 | 112.5 |
| 2 | Alice | F | 13 | 56.5 | 84.0 |
| 3 | Barbara | F | 13 | 65.3 | 98.0 |
| 4 | Carol | F | 14 | 62.8 | 102.5 |
| 5 | Henry | M | 14 | 63.5 | 102.5 |

The SAS System

The STANDARD Procedure

| Name | Mean | Standard Deviation | N |
|--------|------------|--------------------|----|
| Age | 13.315789 | 1.492672 | 19 |
| Height | 62.336842 | 5.127075 | 19 |
| Weight | 100.026316 | 22.773933 | 19 |

Finally, we can see about doing some panelling. Panelling and scrolling tables work really well together. If panelling is turned on, the behavior is what we call automatic. It is automatic in that graph output will be panelled automatically. But if anything else needs to be panelled, then we have to turn on semi-automatic panelling or control every aspect of the panelling with explicit panelling commands. Panelling still needs to be turned on even though, in this case, nothing will be panelled unless it is told to do so.

There is also something strange here. ODS HTML cannot do this! But ODS TAGSETS.HTML4 can! They are really the same thing, but the interface (grammar) used by the ODS statement sees them as two different things. The syntax for ODS HTML is more limited than the syntax for ODS TAGSETS.tagset-name. That means that for this to work, the ODS HTML statements must be changed to ODS TAGSETS.HTML4. If we try to use ODS HTML statements, there will be a syntax error when we try to use the ODS HTML EVENT= statement.

Here is the resulting program, which is virtually the same as the last example, but it has semi-automatic panelling added:

```

ods tagsets.html4 file="example6.html" style=seaside
  options(body_toc='yes' toc_type='menu'
    scroll_tables='yes' scroll_control_images='yes'
    panelling='yes');

ods tagsets.html4 event=panel(start);

proc print data=sashelp.class;
run;

proc standard print data=sashelp.class;
run;

proc report data=sashelp.class nowd;
run;

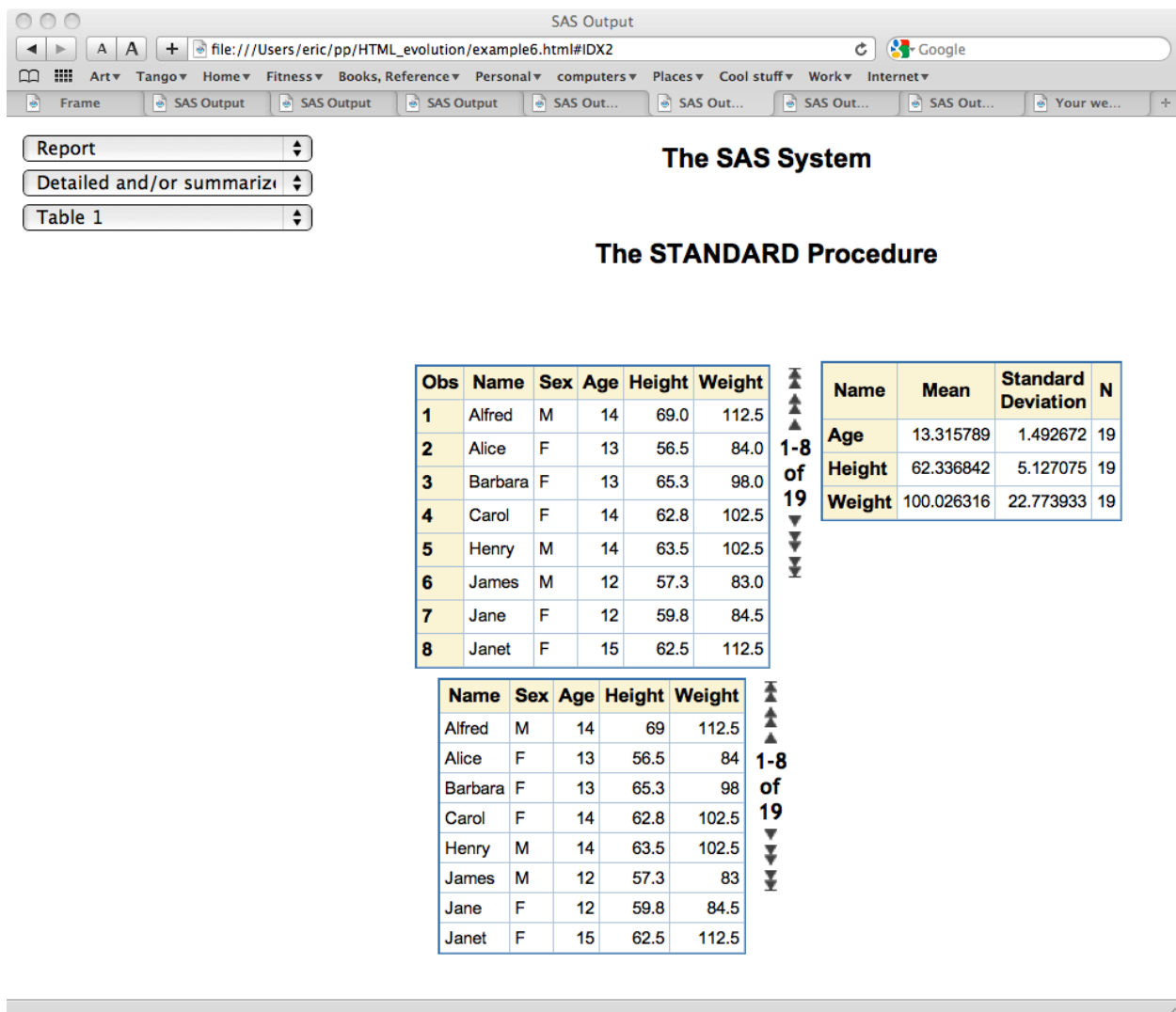
ods tagsets.html4 event=panel(finish);

ods html close;

```

This program creates a panel with four regions in it. Since there are only three tables, the last region is left empty.

Figure 7. Using Panels (Two Columns)



The SAS System

The STANDARD Procedure

| Obs | Name | Sex | Age | Height | Weight |
|-----|---------|-----|-----|--------|--------|
| 1 | Alfred | M | 14 | 69.0 | 112.5 |
| 2 | Alice | F | 13 | 56.5 | 84.0 |
| 3 | Barbara | F | 13 | 65.3 | 98.0 |
| 4 | Carol | F | 14 | 62.8 | 102.5 |
| 5 | Henry | M | 14 | 63.5 | 102.5 |
| 6 | James | M | 12 | 57.3 | 83.0 |
| 7 | Jane | F | 12 | 59.8 | 84.5 |
| 8 | Janet | F | 15 | 62.5 | 112.5 |

| Name | Mean | Standard Deviation | N |
|--------|------------|--------------------|----|
| Age | 13.315789 | 1.492672 | 19 |
| Height | 62.336842 | 5.127075 | 19 |
| Weight | 100.026316 | 22.773933 | 19 |

| Name | Sex | Age | Height | Weight |
|---------|-----|-----|--------|--------|
| Alfred | M | 14 | 69 | 112.5 |
| Alice | F | 13 | 56.5 | 84 |
| Barbara | F | 13 | 65.3 | 98 |
| Carol | F | 14 | 62.8 | 102.5 |
| Henry | M | 14 | 63.5 | 102.5 |
| James | M | 12 | 57.3 | 83 |
| Jane | F | 12 | 59.8 | 84.5 |
| Janet | F | 15 | 62.5 | 112.5 |

We can use the PANELCOLUMNS= option to change the number of columns per panel that is used by semi-automatic panelling.

```
ods tagsets.html4 file="example7.html" style=seaside
  options(body_toc='yes' toc_type='menu'
    scroll_tables='yes' scroll_control_images='yes'
    panelling='yes' panelColumns='3');

ods tagsets.html4 event=panel(start);

proc print data=sashelp.class;
run;

proc standard print data=sashelp.class;
run;

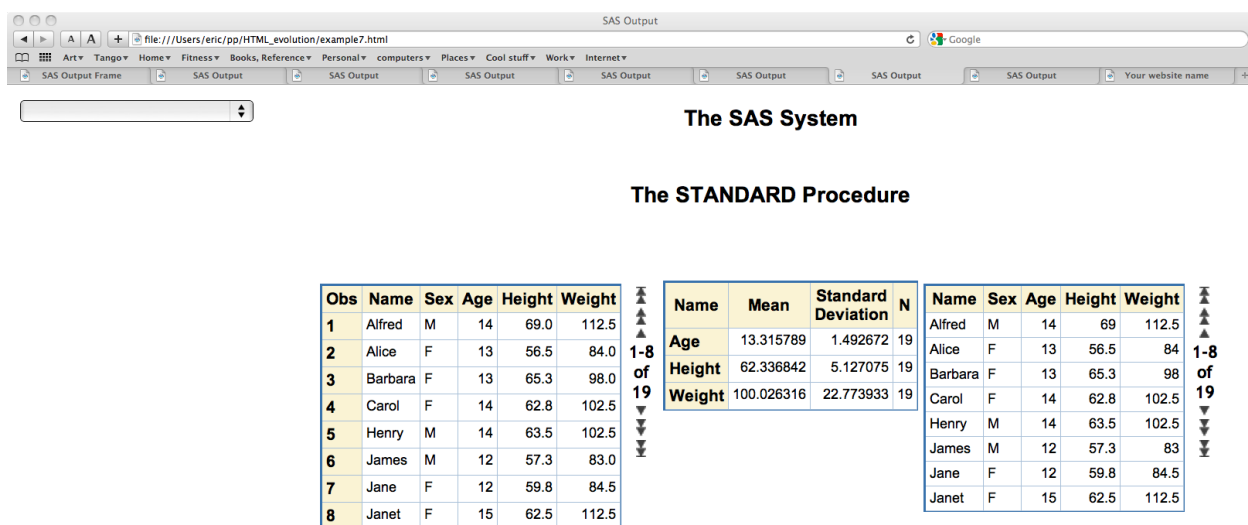
proc report data=sashelp.class nowd;
run;

ods tagsets.html4 event=panel(finish);

ods html close;
```

This positions all three tables crosswise on the page.

Figure 8. Using Panels (Three Columns)



Panelling can also be controlled explicitly if you want to create more complex layouts.

Using the website functionality poses problems if the desire is to have scrolling tables or the floating table of contents. Both of those features require a fair amount of JavaScript, which is included in the head section of the HTML file. The website functionality omits that section. The head file that is provided by your website administrator is read in and scanned for the ending </head> tag. When this tag is found, the scrolling JavaScript is written out in addition to any style sheet links that are needed. This ensures that the ODS styles will not conflict with any styles that are included in the head file.

Here is an example of using everything together:

```
ods tagsets.html4 file="example9.html" (notop) style=seaside
  stylesheet="example9.css"
  options(body_toc='yes' toc_type='menu'
    scroll_tables='yes' scroll_control_images='yes');
```



```

panelling='yes'
head_file='default_head.html'
foot_file='default_foot.html');

ods tagsets.html4 event=panel(start);

proc print data=sashelp.class;
run;

proc standard print data=sashelp.class;
run;

proc report data=sashelp.class nowd;
run;

ods tagsets.html4 event=panel(finish);

ods html close;

```

This example has a few problems. The first problem was that I had to add `onload="setupLongTables()"` to the body tag of the DEFAULT_HEAD.HTML file. That was all that was needed to get scrolling tables working. However, the floating table of contents is a real problem. It ended up at the bottom of the page. With some creative HTML, it can be moved around, but it is in direct conflict with all of the navigation menus that are included in the DEFAULT_HEAD.HTML file. This could be fixed, but it would require the help of your local web administrator or HTML wizard. The easiest solution is to not use the floating table of contents when you are also using a complex head section from your Web site. It would probably work with a simpler example that didn't include all of the navigation controls, so I would encourage trying it to see what happens.

Here is the same example, but this time, it uses panelling and scrolling tables, and it includes complicated website header and footer files:

```

ods tagsets.html4 file="example10.html"(notop) style=seaside
  stylesheet="example10.css"
  options(
    scroll_tables='yes' scroll_control_images='yes'
    panelling='yes'
    head_file='default_head.html'
    foot_file='default_foot.html');

ods tagsets.html4 event=panel(start);

proc print data=sashelp.class;
run;

proc standard print data=sashelp.class;
run;

ods tagsets.html4 event=panel(finish);

proc report data=sashelp.class nowd;
run;

ods html close;

```

This output works a little better out of the box.

Figure 9. Panels, Scrolling Tables, Headers and Footers

Logo Image alt text: Your Company: Slogan

Select your Region:
Region 1 Go

News
Announcements
In the Press
More...

Company
Contact
Information
About

Search for:
Go

The SAS System

The STANDARD Procedure

| Obs | Name | Sex | Age | Height | Weight |
|-----|---------|-----|-----|--------|--------|
| 1 | Alfred | M | 14 | 69.0 | 112.5 |
| 2 | Alice | F | 13 | 56.5 | 84.0 |
| 3 | Barbara | F | 13 | 65.3 | 98.0 |
| 4 | Carol | F | 14 | 62.8 | 102.5 |
| 5 | Henry | M | 14 | 63.5 | 102.5 |
| 6 | James | M | 12 | 57.3 | 83.0 |
| 7 | Jane | F | 12 | 59.8 | 84.5 |
| 8 | Janet | F | 15 | 62.5 | 112.5 |

| Name | Mean | Standard Deviation | N |
|--------|------------|--------------------|----|
| Age | 13.315789 | 1.492672 | 19 |
| Height | 62.336842 | 5.127075 | 19 |
| Weight | 100.026316 | 22.773933 | 19 |

The SAS System

| Name | Sex | Age | Height | Weight |
|---------|-----|-----|--------|--------|
| Alfred | M | 14 | 69 | 112.5 |
| Alice | F | 13 | 56.5 | 84 |
| Barbara | F | 13 | 65.3 | 98 |
| Carol | F | 14 | 62.8 | 102.5 |
| Henry | M | 14 | 63.5 | 102.5 |
| James | M | 12 | 57.3 | 83 |
| Jane | F | 12 | 59.8 | 84.5 |
| Janet | F | 15 | 62.5 | 112.5 |

Copyright 2004 Your Company name.

2 errors occurred in opening the page. For more information, choose Window > Activity.

There are other ways to read an external file into the output file that ODS is creating. There is an inline format function called `INCLUDE_FILE` that can be used anywhere. It can be called directly as an event, or it can be embedded in a title, in an ODS text statement, or even in the data. See my paper from SAS Global Forum 2009 titled "Inline Formatting with ODS Markup".

Other functionality, such as the slider and survey HTML tagsets, proved to be too invasive for inclusion in this tagset at this time. However, that functionality is still in consideration. The slider functionality has been especially popular. This new HTML4 tagset is really a stepping stone to the next major HTML tagset release, which is going to be HTML5.

HTML5 AND THE FUTURE

We have a number of goals in mind with the HTML5 tagset. First, it should be W3C compliant as well as Section 508 compliant. It should provide a solid platform that will grow in features much like the Excel tagset has in the last few years. We want to incorporate features such as the floating table of contents and scrolling tables with real scroll bars. Also, to ease publishing woes, it should also be able to easily integrate HTML snippets as required by web administrators and your company's website.

Panelling will most likely be left out, because ODS layout will be available. It will be easier to use and much more powerful than the panelling features that are demonstrated in this paper.

Other features that are under consideration are the features of the table editor tagset, which allows for scrolling, sorting, and rearranging of table columns. A style explorer that works better than the htmlstyle tagset will also most likely be included. Also, with the advent of ODS lists, it is possible that we may be able to easily implement word clouds and menus for the creation of forms. All of this is yet to be seen, but is entirely possible as we begin this new era in which browsers are more W3C compliant than ever before.

Enjoy this latest HTML4 tagset and think about the needs you have that the new HTML5 destination could fulfill. HTML is an amazing markup language that is full of possibilities. It is about time that the ODS HTML destinations received a little love.

RECOMMENDED READING

Gebhart, Eric. 2009. "Inline Formatting with ODS Markup". Proceedings of the SAS Global Forum 2009 Conference. Cary, NC: SAS Institute Inc. Available at support.sas.com/resources/papers/proceedings09/222-2009.pdf.

CONTACT INFORMATION

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

Eric Gebhart
SAS Campus Drive
SAS Institute Inc.
E-mail: Eric.Gebhart@sas.com
Web: <http://support.sas.com/rnd/base/ods/odsmarkup/>

SAS and all other SAS Institute Inc. product or service names are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries. ® indicates USA registration.

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