

Sorting ODS HTML Tables by Column Headers Using Javascript

Laura W. Covington and Neal Oden, The EMMES Corporation, Rockville, MD

ABSTRACT

ODS output can be used together with javascript to generate HTML tables that sort themselves on the client side when the user clicks on a column header. Sorttable.js is an already existing javascript that allows HTML tables to have sortable columns (<http://www.kryogenix.org/code/browser/sorttable/>). You need to do three things to adapt this javascript to SAS ODS output:

- (1) Create an ODS HTML table by PROC REPORT or PROC PRINT with a simple title and column names defined using the LABEL statement.
- (2) Use a DATA statement to incorporate into the ODS output HTML code that defines the class and creates a unique ID.
- (3) Store sorttable.js in the same folder as the ODS HTML table.

INTRODUCTION

A client asked for columns that could be sorted in a table posted on a website. Then the requests expanded: "What about dates? Can I sort on multiple columns? What about different browsers? Does the user need to download anything to make it work?"

The following sections show how to make the columns sort to meet the client demands. First, I show what type of PROC outputs can generate the SAS .html. Then I show the code to modify the SAS .html file in order to get the sorting functionality into the table. Next, I discuss the .js file and its use. Finally, I outline how to implement additional features.

GENERATING THE .HTML FILE IN SAS

In SAS 9.1, use the following ODS statement:

```
ods html3 file="C:\TEMP2\sample1_9.html" style=sasweb;
```

In SAS version 8.2, use the standard ODS HTML.

There are two kinds of reports that can be used to generate the table to be sorted: PROC PRINT and PROC REPORT.

PROC PRINT

It is easy to generate the ODS HTML output using PROC PRINT. The variable name for the column designation can be kept as the variable name or defined using labels. The javascript will ignore SPLIT=characters in the labels; don't use them. Keep your labels simple.

The following code will generate Table 1 in SAS 9.1.

```
ods html3 file="C:\TEMP2\sample1_9.html" style=sasweb;
```

```
PROC PRINT DATA=indata.DEMO noobs label;
  var projid site age dx dxdate event eventdat;
  label site='Site'
        projid='Project ID'
        eventdat='Event Date'
```

```

age='Age'
dx='Diagnosis'
dxdate='Diagnosis Date'
event='Event';
title1 'Sample Sortable HTML Using PROC PRINT';
format dxdate mmddyy10. eventdat mmddyy10.;
RUN;

ods html3 close;

```

Table 1: Sample Sortable HTML Generated Using PROC PRINT

Sample Sortable HTML Generated Using PROC PRINT

Project ID	Site	Age	Diagnosis	Diagnosis Date	Event	Event Date
14762	019	16	BIPOLAR	11/13/2003	INFECTION	01/17/2004
14982	003	16	DEPRESSION	07/22/2003	HEADACHE	11/13/2004
14982	003	16	DEPRESSION	07/22/2003	HIGH BP	09/17/2005
15023	009	15	ANXIETY	02/28/2005	WEIGHT GAIN	08/06/2005
15032	025	18	ANXIETY	09/01/2005	HIGH BP	12/12/2004
15032	025	18	ANXIETY	09/01/2005	INSOMNIA	10/04/2005
15165	005	14	BIPOLAR	05/18/2005	INSOMNIA	06/05/2005
15432	017	17	ANXIETY	10/31/2004	BACK PAIN	04/29/2005
15576	021	15	DEPRESSION	08/29/2004	MIGRAINE	10/02/2004
15723	023	14	DEPRESSION	03/01/2005	HEADACHE	05/12/2005

PROC REPORT

Label use and restrictions are the same for PROC REPORT as they are for PROC PRINT.

When using PROC REPORT if you have multiple rows with duplicate information (in this case Project ID, Center, Age, Diagnosis, and Diagnosis Date), the variables that are duplicated need to be described as DISPLAY in the define statement or you will get output like Table 2, which is unsortable using this technique.

Table 2: Sample of Unsortable HTML Generated Using PROC REPORT

Sample of Unsortable HTML Generated Using PROC REPORT

Project ID	Site	Age	Diagnosis	Diagnosis Date	Event	Event Date
14762	019	16	BIPOLAR	11/13/2003	INFECTION	01/17/2004
14982	003	16	DEPRESSION	07/22/2003	HEADACHE	11/13/2004
					HIGH BP	09/17/2005
15023	009	15	ANXIETY	02/28/2005	WEIGHT GAIN	08/06/2005
15032	025	18	ANXIETY	09/01/2005	INSOMNIA	10/04/2005
					HIGH BP	12/12/2004
15165	005	14	BIPOLAR	05/18/2005	INSOMNIA	06/05/2005
15432	017	17	ANXIETY	10/31/2004	BACK PAIN	04/29/2005
15576	021	15	DEPRESSION	08/29/2004	MIGRAINE	10/02/2004
15723	023	14	DEPRESSION	03/01/2005	HEADACHE	05/12/2005

The following is a sample of code using PROC REPORT that will generate a table that can be sorted using this technique. The table it generates can be found in Table 3.

```
ods html3 file="C:\TEMP2\sample1_9_report.html" style=sasweb;

proc report data=indata.demo nowindows;
  title1 'Sample of Sortable HTML Using PROC REPORT';

  column projid site age dx dxdate event eventdat;

  define projid / display 'Project ID' style={font_weight=bold};
  define site / display 'Site';
  define age / display 'Age';
  define dx / display 'Diagnosis';
  define dxdate / display 'Diagnosis*Date' format=mmddyy10.;
  define event / display 'Event';
  define eventdat / display 'Event*Date' format=mmddyy10.;

run;

ods html3 close;
```

Table 3: Sample Sortable HTML Generated Using PROC REPORT

Sample Sortable HTML Generated Using PROC REPORT

Project ID	Site	Age	Diagnosis	Diagnosis Date	Event	Event Date
15723	023	14	DEPRESSION	03/01/2005	HEADACHE	05/12/2005
15432	017	17	ANXIETY	10/31/2004	BACK PAIN	04/29/2005
14762	019	16	BIPOLAR	11/13/2003	INFECTION	01/17/2004
14982	003	16	DEPRESSION	07/22/2003	HEADACHE	11/13/2004
15023	009	15	ANXIETY	02/28/2005	WEIGHT GAIN	08/06/2005
14982	003	16	DEPRESSION	07/22/2003	HIGH BP	09/17/2005
15032	025	18	ANXIETY	09/01/2005	INSOMNIA	10/04/2005
15165	005	14	BIPOLAR	05/18/2005	INSOMNIA	06/05/2005
15576	021	15	DEPRESSION	08/29/2004	MIGRAINE	10/02/2004
15032	025	18	ANXIETY	09/01/2005	HIGH BP	12/12/2004

THE CODE

The next step requires modifying the SAS .HTML in order to make it work. The code used to insert the required information into the SAS generated .HTML code follows.

```

1 data _null_;
2 length buffer $2000;
3 retain seed tablenum 0;
4 infile "c:\temp2\sample1_9_report.html" truncover;
5 file "c:\temp2\sample1_out_9_report.html";
6 input buffer $char3000.;

7 if 0 ^= index(buffer, "<TABLE ") then do;
8     tablenum = tablenum + 1;

9     if mod(tablenum, 2) = 0 then buffer = '<TABLE class="sortable"
id="' || put(floor(uniform(seed) * 1e8), 8.) || '"' || substr(buffer, 7);
end;

10 put buffer;

11 if 0 ^= index(buffer, "<HEAD>") then do;
12     buffer = '<script src="sorttable.js"></script>';

13 put buffer;
14 end;
15 run;

```

There are two required elements that need to be incorporated into the HTML code in order to make the sortable tables. The first is addressed on line 9 of the code. It is used to find the number of the table we want to sort. It is important to review the HTML code to determine the HTML table number to be sorted. In most cases a title is used, and is defined as the first HTML table in the HTML code. The second HTML table usually contains the data you want to sort. The code above assumes this scenario. It is possible to have sorting if multiple tables are generated using the ODS HTML statement.

Each table in HTML starts with a line of code defining it:

```
<TABLE cellspacing=1 cellpadding=1 rules=NONE frame=VOID border=0 width=100% bgcolor="#FFFFFF">
```

When the above program executes, this line is modified from the above to the following.

```
<TABLE class="sortable" id="82949177" cellspacing=1 cellpadding=7 rules=NONE frame=BOX  
bgcolor="#CCCCCC">
```

Note that the table now has a unique ID that is a random number.

The next required step occurs at line 11 of code. This statement adds to the HTML output a reference to the required .js file.

Before the code is implemented, the HTML code reads:

```
<HEAD>  
<TITLE>SAS Output</TITLE>
```

After the code is run, it is modified to:

```
<HEAD>  
<script src="newsorttable.js"></script>  
<TITLE>SAS Output</TITLE>
```

The result of these two lines is a table like Table 4, where the Project ID column has been selected for sorting. The table headers now are links and the Project ID column has an arrow showing which sort order (up or down) is occurring.

Table 4: Sortable HTML File Using PROC REPORT**Sortable HTML File Using PROC REPORT**

<u>Project ID</u> ↓	<u>Site</u>	<u>Age</u>	<u>Diagnosis</u>	<u>Diagnosis Date</u>	<u>Event</u>	<u>Event Date</u>
14762	019	16	BIPOLAR	11/13/2003	INFECTION	01/17/2004
14982	003	16	DEPRESSION	07/22/2003	HEADACHE	11/13/2004
14982	003	16	DEPRESSION	07/22/2003	HIGH BP	09/17/2005
15023	009	15	ANXIETY	02/28/2005	WEIGHT GAIN	08/06/2005
15032	025	18	ANXIETY	09/01/2005	INSOMNIA	10/04/2005
15032	025	18	ANXIETY	09/01/2005	HIGH BP	12/12/2004
15165	005	14	BIPOLAR	05/18/2005	INSOMNIA	06/05/2005
15432	017	17	ANXIETY	10/31/2004	BACK PAIN	04/29/2005
15576	021	15	DEPRESSION	08/29/2004	MIGRAINE	10/02/2004
15723	023	14	DEPRESSION	03/01/2005	HEADACHE	05/12/2005

JAVASCRIPT FILE

The final step in this three-part process is to include the javascript file in the folder on the server where the HTML file will be stored to be viewed. The javascript file used for generating the tables that could be sorted is available at <http://www.kryogenix.org/code/browser/sortable> and was created by Stuart Langridge. The author notes that the date function sorts using the UK dd/mm/[ccyy] format, but will need additional tweaking for dates in other formats. I address this in the Additional Information section below. The content of javascript file follows:

SORTTABLE.JS

```
function addEvent(elm, evType, fn, useCapture)
// addEvent and removeEvent
// cross-browser event handling for IE5+, NS6 and Mozilla
// By Scott Andrew
{
  if (elm.addEventListener){
    elm.addEventListener(evType, fn, useCapture);
    return true;
  } else if (elm.attachEvent){
    var r = elm.attachEvent("on"+evType, fn);
    return r;
  } else {
    alert("Handler could not be removed");
  }
}
```

ADDITIONAL INFORMATION

DATE FORMATS

The sortable.js available on the <http://www.kryogenix.org/code/browser/sorttable> does not allow for standard U.S. date formats. The programming staff at EMMES has modified the original javascript to allow for dates in the standard SAS format of mmddyy10.

The code is not available on the Internet, but is listed below.

NEWSORTTABLE.JS

```
function ts_sort_default(a,b) {
    aa = ts_getInnerText(a.cells[SORT_COLUMN_INDEX]);
    bb = ts_getInnerText(b.cells[SORT_COLUMN_INDEX]);
    if (aa==bb) return 0;
    if (aa<bb) return ret(-1);
    return ret(1);
}

function addEvent(elm, evType, fn, useCapture)
// addEvent and removeEvent
// cross-browser event handling for IE5+, NS6 and Mozilla
// By Scott Andrew
{
    if (elm.addEventListener){
        elm.addEventListener(evType, fn, useCapture);
        return true;
    } else if (elm.attachEvent){
        var r = elm.attachEvent("on"+evType, fn);
        return r;
    } else {
        alert("Handler could not be removed");
    }
}

function ts_sort_default(a,b) {
    aa = ts_getInnerText(a.cells[SORT_COLUMN_INDEX]);
    bb = ts_getInnerText(b.cells[SORT_COLUMN_INDEX]);
    if (aa==bb) return 0;
    if (aa<bb) return -1;
    return 1;
}
```

SORTING ON MULTIPLE COLUMNS

Recent versions of Internet Explorer evidently use a stable sorting algorithm, so it is possible to sort on more than one column by working backwards. For example, if we wanted to sort on Age, and Event Date within Age, we would sort first on Event Date and then on Age.

BROWSER COMPATIBILITY

The examples in this paper are generated using Internet Explorer 6.0. When we used Mozilla 5.0 and Netscape 8.0, date sort function worked appropriately for the mmddyy10. format and the multiple sort function worked as well.

CONCLUSION

It is easy to use javascript to make sortable tables from the ODS HTML. This functionality improves the presentation of your data. In the future, we hope to explore whether the sorting capability can be added to PROC TABULATE, PROC FREQ, or PROC MEANS output.

REFERENCES

The javascript file used for generating the tables that can be sorted is available at <http://www.kryogenix.org/code/browser/sortable> and was created by Stuart Langridge.

ACKNOWLEDGMENTS

I would like to acknowledge the team who adapted the sortable.js file to create the newsortable.js: Jack Anderson, Kuo Guo, and Fenhua He.

CONTACT INFORMATION

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

Laura W. Covington
The EMMES Corporation
401 N. Washington Street, Suite 700
Rockville, MD 20850
Work Phone: 301-251-1161 ext. 187
Email: lcovington@emmes.com
Web: <http://www.emmes.com>

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.