

Paper PO8

SAS® Web Editor, is it the right choice for you?

Rebecca Ottesen and Jamelle Simmons, California Polytechnic State University,
San Luis Obispo, CA

ABSTRACT

The SAS® Web Editor was released in SAS OnDemand for Academics as a way for students to learn to use SAS via a Web-based tool. The Web Editor is similar to traditional Windows based SAS with the added advantage that there is no installation and it has platform independence. Being able to access SAS in this way is extremely important for classroom use where the type of machine varies across students. However differences exist in how the data, libraries and files are accessed and the Web Editor also requires an Internet connection. With the release of the Web Editor to the global SAS community in SAS 9.4, we will share our experience and the tradeoff between using SAS in a local installation versus a Web-based environment.

INTRODUCTION

At first glance the SAS Web Editor appears to require learning a new interface with new functionality. However, once the layout of the Web Editor is understood it becomes easy to use for many programming tasks. Our testing environment for the Web Editor was via OnDemand for Academics which is a free SAS service for University instructors and students. As a result, our version of the Web Editor was hosted by SAS which meant that the data needed to be uploaded and code processed on the SAS servers. While this allows platform independence, which is important for students, it also requires an internet connect and a supported web browser. Our experience with the Web Editor was limited to this setting, but it should be noted that at a company certain aspects of our testing could be very different and would depend on the internal server installation of the Web Editor at the institution.

ONDEMAND SET-UP

As noted our setting is classroom based instruction with students having the ability to access SAS from home via SAS OnDemand. This requires the instructor to register the course and to invite students to participate. The description of set-up in this paper may differ in a business setting where the Web Editor for SAS 9.4 would be installed on a company server. However, once the Web Editor is installed we would assume that the functionality would be similar, and only the amount of data storage and speed would differ between the two mechanisms for running the Web Editor.

REGISTERING THE COURSE

Registering a course with SAS OnDemand for Academics is simple. The main SAS OnDemand link can be found at the SAS Customer Support web site (support.sas.com), where there are OnDemand links for Professionals, Academics or K-12. OnDemand has several products available for teaching such as SAS Enterprise Guide, SAS Enterprise Miner, SAS Forecast Server, and the SAS Web Editor. For this paper we explored only the Web Editor.

The first step in the OnDemand environment is for the instructor to create a profile and register their course. There are step by step instructions^{1,2,3} for registration plus additional documentation^{4,5} for review, once registration of the course is complete. After registration, students can be invited to create their own profile and join the course. From the SAS OnDemand for Academics Control Center users can run the Web Editor client. For instructors, the info link at the Control Center is useful as it explains how to upload course data files to the SAS server via file transfer protocol (FTP). The info link also provides some detail about specifying LIBNAME statements properly and a sample email invitation that can be shared with students.

DOCUMENTATION

The best source for documentation is also available at the SAS OnDemand for Academics web site⁶. In addition to the step-by-step registration and installation guide^{2,3}, using the instructors and students links connects to the Instructor User's Guide⁴ and Student User's guide⁵, respectively. Deep within the Instructors User's Guide there is more information and helpful links that provide detail about the uploading of data files for the course via FTP.

In the OnDemand environment it is important to understand that students and instructors have different viewpoints of the system. Instructors have the ability to upload class data sets and example programs, which are accessible for student use. However, students are limited to creating their own programs and uploading data within their user account. In addition, the student's ability to upload data is very limited only allowing for a maximum size of 10MB per upload.

WEB EDITOR LAYOUT

Once the course is registered, the client for the Web Editor can be launched from the SAS OnDemand for Academics Control Center in a supported web browser⁷. The Web Editor application window has a somewhat different look and feel from traditional SAS as seen in figure 1; however, it is easy to navigate with the various sections and icons that are available.

The top part of the window contains application options for extras such as the ability to modify the autoexec.sas, a link to the SAS Web Editor help, and the ODS graphics editor. The navigation pane on the left side includes the ability to manage folders, libraries and file shortcuts. The work area located on the right side is where the programming project including code, log and results are accessed. The Web Editor also has handy keyboard shortcuts that can be utilized such as zoom in (ctrl+plus sign), create a new program (F4), and many more which can be found in detail in the accessibility features documentation⁷.

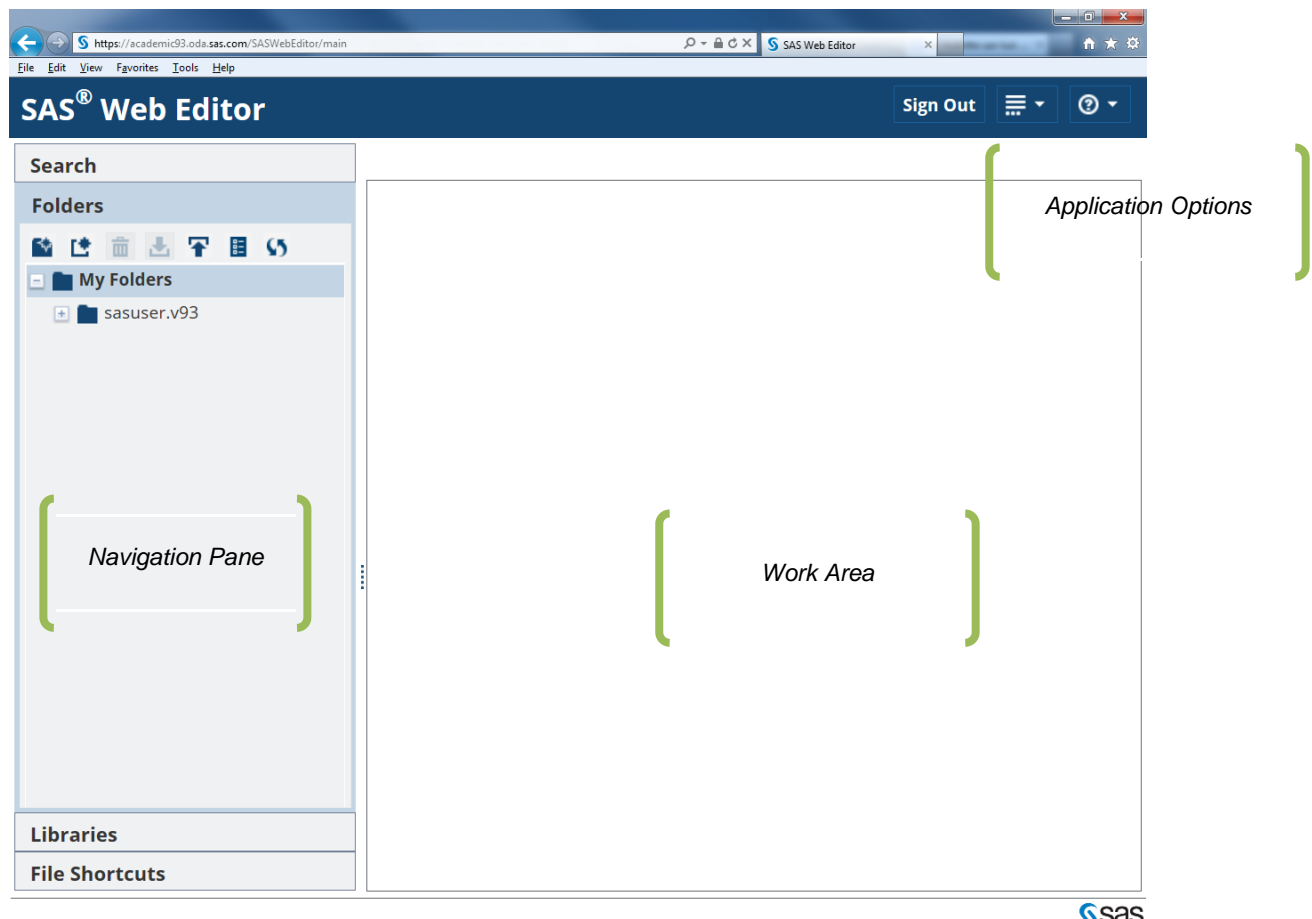



Figure 1. Main Web Editor Interface

NAVIGATION PANE

The specific functionality of the navigation pane is divided into four main sections: 1) searching for files, folders, tables or columns, and libraries; 2) the navigation and creation of new folders on the server as well as the ability to upload and download files to and from the local machine; 3) managing libraries with the ability to access the default SAS libraries (ex. SASHELP, SASUSER, WORK); 4) creating and referencing file shortcuts.

WORK AREA

A programming project session can be started in the folders section by clicking the New SAS Program icon  or F4. A project tab will appear in the work area with working tabs for the code, log and results as seen in figure 2. Within each of these working tabs there are various icons which are mostly self explanatory, however there are a few items

to note. Saving the program will save to the SAS server, not the local machine. There is the ability to download or copy and paste the programming from the web browser to the local machine and if the user does not have SAS installed locally, the code may be viewed in a text editor. A similar feature between the Web Editor coding tab and the traditional SAS enhanced program editor is the coloring of the programming code.

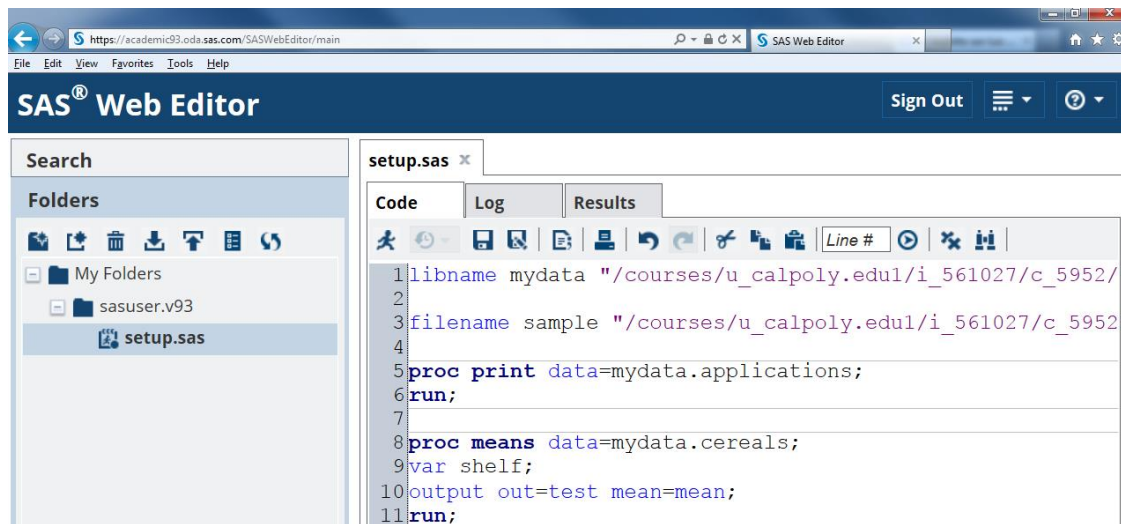



Figure 2. Programming Session

A nice feature in the code tab is an icon  to insert sample code for various tasks such as PROC SQL, importing or generating CSV files, and many code examples for graphics using SG procedures (figure 3).

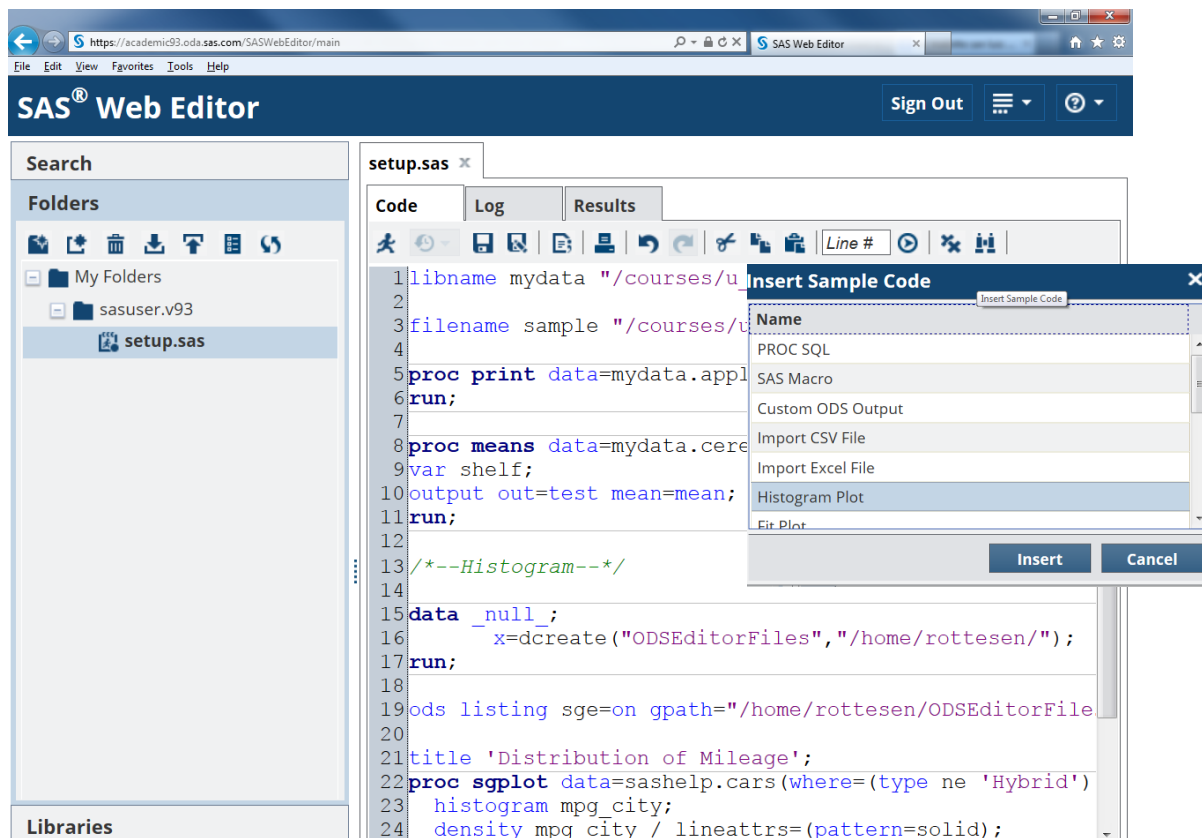



Figure 3. Insert Feature and Sample Code for a Histogram

Another helpful feature appears at the top of the log by default which allows for the tracking of errors, warnings and notes in a collapsible view (figure 4). Each of the messages at the top are linked to the exact location of the message in the log. One important difference about the log in traditional SAS compared to Web Editor is that the log is overwritten with each submission as are the results. However, there is a submission history icon  on the code tab that allows the code from a previous submission to be accessed.

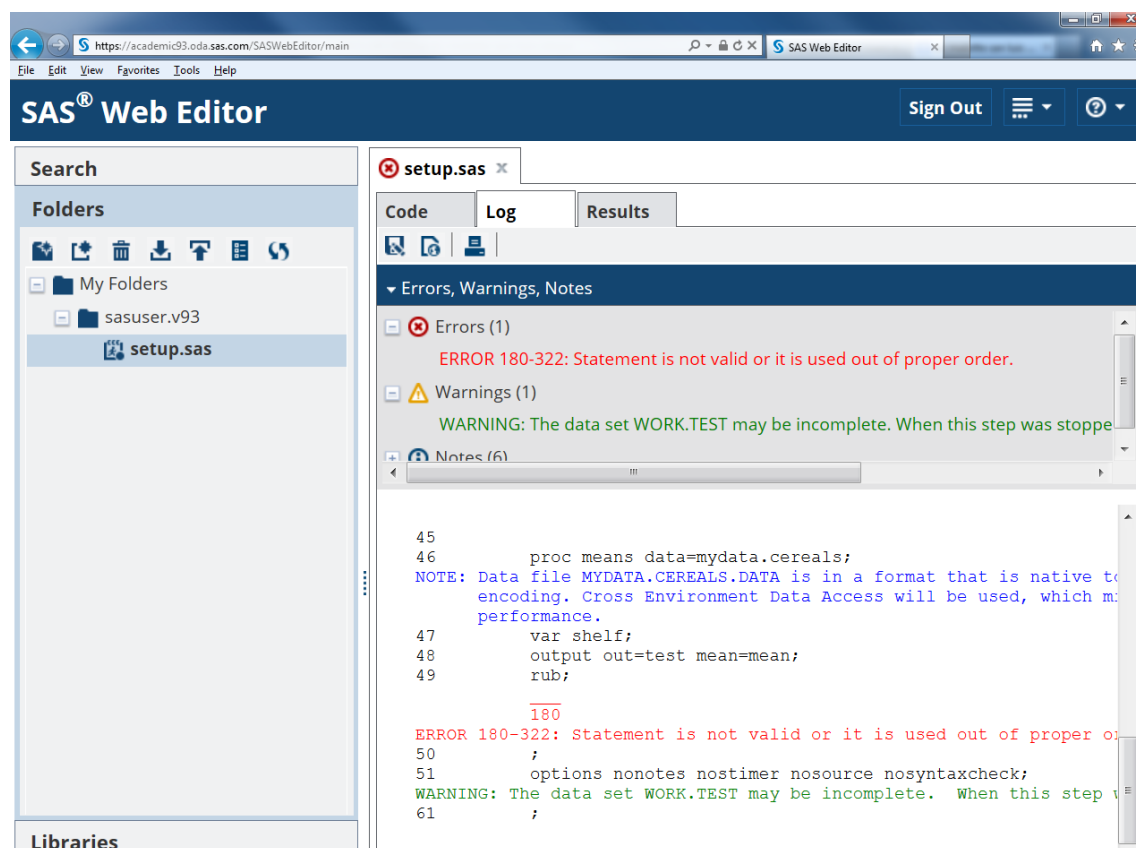





Figure 4. Errors, Warnings and Notes in the Log

The results appear in HTML format in the results tab as shown in figure 5. There are icons that allow the saving of output as an HTML , PDF  or RTF  file without having to code to an ODS destination. An additional difference between the Web Editor and traditional SAS is that there is no expandable results viewer for the output so depending on the number of results, scrolling through lots of pages may be required.

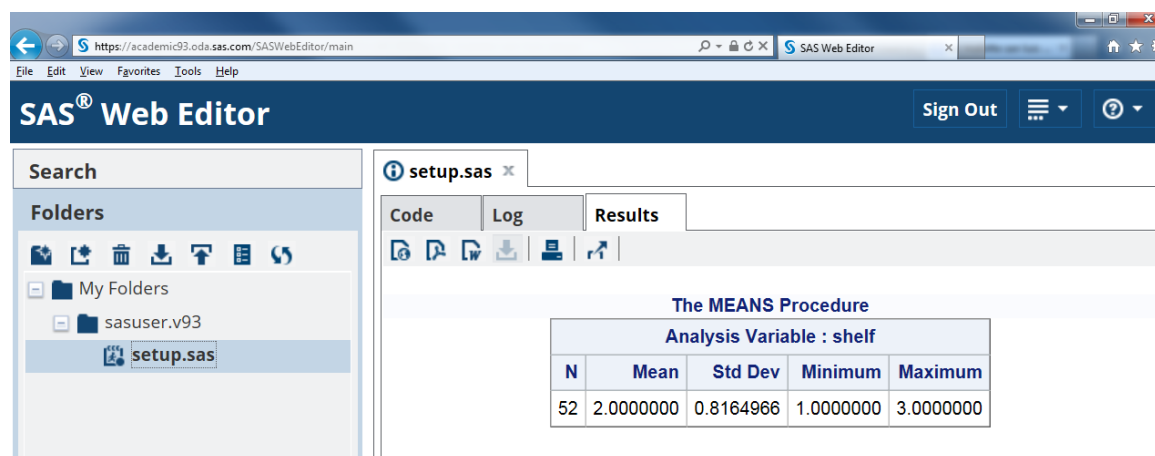



Figure 5. Results Tab

WORKING WITH THE WEB EDITOR

UPLOADING FILES

The SAS server is UNIX based which creates some limitations in how the data sets can be stored. They should be named with lowercase letters, no spaces in file names, and it is important to avoid special characters. In addition to SAS data sets, other data file types such as CSV and ASCII (DAT, TXT) data files can also be uploaded.

Instructors have access to upload files via FTP to the course folders which can be accessed by students in the course. Notably certain data file types such as Excel files, cannot be uploaded via FTP to the SAS server. An important limitation of the OnDemand Web Editor is the size of the data file that can be uploaded via FTP. In addition to overall space restrictions for course data, there are also time out issues in uploading moderately sized (2 GB) data files. Students are restricted even further and can only upload small data files up to approximately 10MB at a time.

Uploading student data takes place via the upload files icon  in the Folders section of the navigation pane as shown in figure 6.

To share programs only the instructor may upload SAS programs for the course, viewed by students via file shortcuts. To use the programming it must be opened in another work area tab and then copied into the code tab of the project.

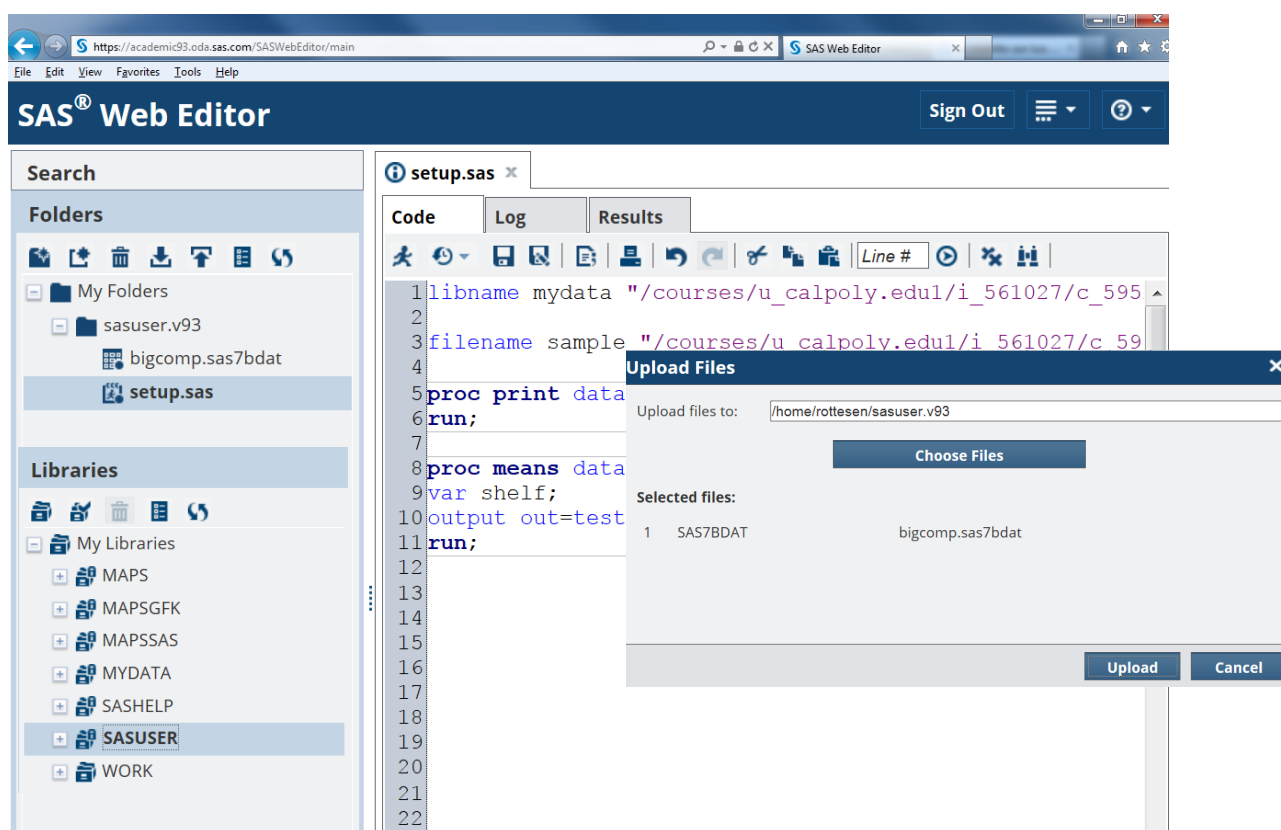




Figure 6. Uploading files to SASUSER

READING AND WRITING FILES

In the OnDemand Web Editor, reading SAS data sets and other raw data files can be accomplished with LIBNAME and FILENAME statements in two different ways: 1) files can be referenced from the data uploaded to the course by the instructor; 2) user specific files can be referenced in the SASUSER directory. In the first case it is recommended in OnDemand that the ACCESS= option on the LIBNAME statement be added to prevent over-writing of uploaded course files.

At first glance the writing of data sets seems to be temporary for example storing SAS data sets to the WORK library. Writing permanent SAS data sets or raw data files can also occur by storing to the SASUSER location; however, the method for saving permanent SAS and raw data sets to SASUSER is not obvious. In order to accomplish this the

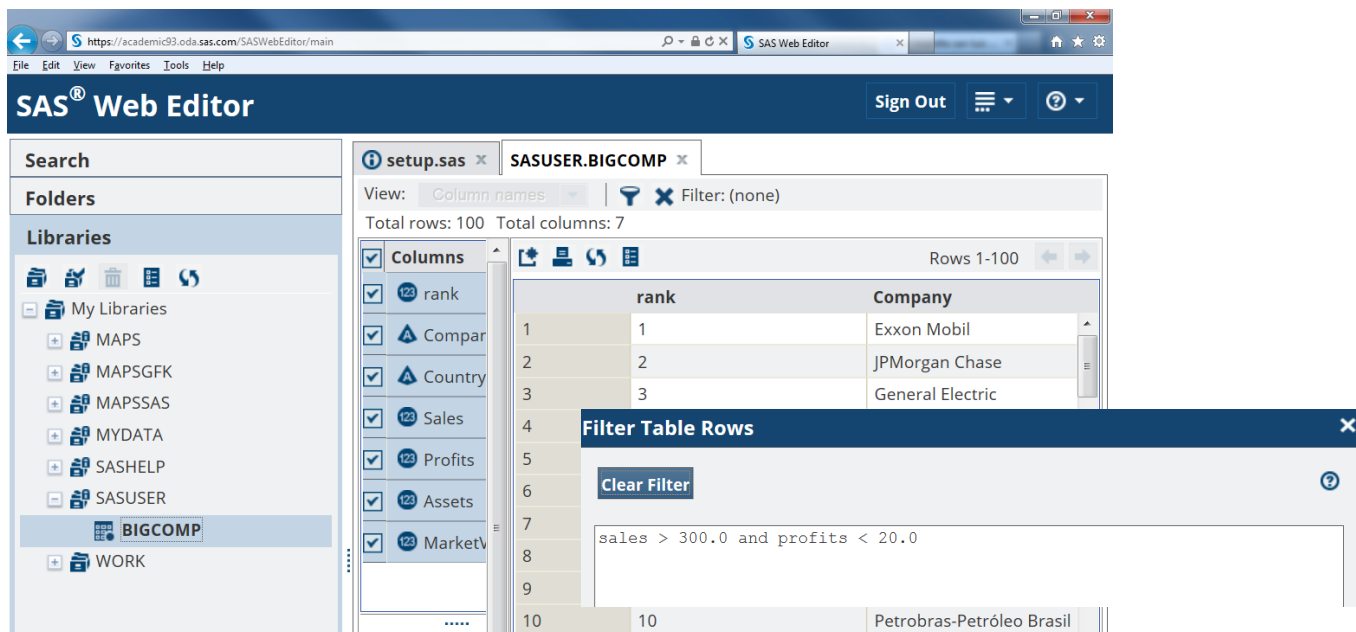
default read-only permission for the SASUSER library needs to be changed by resubmitting a LIBNAME statement without the ACCESS= option. Once submitted any SAS data sets and raw data files may be written to the SASUSER location. From there, utilizing the download icon  in the Folders section enables the user to save the data files to the local machine if desired.

Programming for the project is stored to the SAS server and there is a way to store the code locally by utilizing the download icon  in the Folders section. While downloading the code creates a SAS program that can be opened in a local SAS session, it should be noted that the user may not have a local installation of SAS which is why they are using SAS OnDemand Web Editor in the first place. If this is the case, opening the downloaded SAS program in a simple text editor such as Notepad makes it appear without any line returns and renders it virtually unreadable. A solution to this readability problem is to open the program in a more sophisticated text editor such as WordPad or Microsoft Word®. This text editor issue is also the case for raw data files that are saved to the local machine as mentioned above.

In summary, users of the Web Editor in OnDemand have the ability to read data from a course folder or their own uploaded data, and then edit and save to their WORK or SASUSER folder. Additionally they can download their data sets to a local machine if the data files are stored in the SASUSER folder. User programming is also saved in the server account but can be downloaded to the local machine as necessary.

VIEWING DATA SETS

SAS data sets can be viewed by double clicking on them in the Navigation pane which renders them to a new tab in the programming project. The Web Editor viewer shows 100 observations at a time and allows the selection of which observations will appear in the table. There is also the ability to add filtering capabilities which are programmed similar to a SQL WHERE clause (figure 7). By clicking on one of the variables in the columns box on the left side of the table, users are able to view the properties and values of each variable such as its label, name, length, type, format, and informat.



The screenshot shows the SAS Web Editor interface. On the left is a navigation pane with 'Libraries' including 'SASUSER' and 'BIGCOMP'. The main area displays a data table for 'SASUSER.BIGCOMP'. The table has columns 'rank' and 'Company'. The first three rows are visible: rank 1 (Exxon Mobil), rank 2 (JPMorgan Chase), and rank 3 (General Electric). A 'Filter Table Rows' dialog box is open, showing a filter expression: 'sales > 300.0 and profits < 20.0'. The dialog also has a 'Clear Filter' button.

rank	Company
1	Exxon Mobil
2	JPMorgan Chase
3	General Electric
10	Petrobras-Petróleo Brasil

Figure 7. Data View and Filter

CODING FUNCTIONALITY

The Web Editor is capable of traditional SAS programming syntax. We tested code including DATA step functionality, arrays, do loops, setting, merging, PROC SQL, SORT, PRINT, many of the statistical procedures, ODS graphics, and macros to name a few. While we tested as much SAS syntax as we could, it proved difficult to explore every possible coding situation. Most of the differences that were noted revolved around processing speed, running out of memory for large amounts of output, building larger data sets from smaller data sets, and limitations in graphics editing. For example, currently the Web Editor only supports ODS graphics Editor and not ODS Graphics Designer for editing graphics. To run the ODS Graphics Editor a download package must be installed on the local machine.

The largest functionality issue experienced was processing, where working with complex code on even moderately sized data sets eventually timed out. For example, running a program that utilizes do loop replications over thousands of variables, on a main data set with ~200,000 observations that was also merged with several other data sets, was unable to complete when it took 25 minutes to run locally. However, after narrowing the main data set down to ~50,000 observations, the job was able to complete in approximately 2 minutes where it took approximately 7 minutes locally on that same subset. This implies that while there is a limit to the number of observations or size of the working data set, if the data are under this limit the processing time may be faster on the SAS server.

BROWSER AND CONNECTIVITY

The Web Editor is compatible with certain web browser versions, and one of these is Apple Safari, which makes running SAS in the classroom platform independent. This is a huge benefit of using the Web Editor in that the student with a Mac does not have to jump through endless hoops to get a local installation of SAS running. While we were unable to test the OnDemand Web Editor on a Mac computer, we are hopeful that the functionality on computer based Safari would be similar to a PC.

Some general browser issues to note are that the Web Editor is sensitive to display resolution, losing work upon exiting or refreshing the browser (be careful and save often!), and browser memory limits for large amounts of output. Additionally, due to the fact that the Web Editor is running on a browser there are obvious connectivity issues that must be tolerated depending on network performance. To help users pinpoint connectivity issues SAS has a web site where the status of their OnDemand servers can be verified. Additionally, care must be taken that a session is not left open for too long (approximately one hour) without saving which may result in work being lost due to timing out. Last, there are also documented exceptions to accessibility standards that can be referenced in the Web Editor accessibility documentation⁷.

iPRODUCTS

Although we were unable to test on a Mac, the Web Editor was tested using Safari on the iPad. The ability to run SAS code on the iPad is impressive and liberating. The functionality is similar to what is previously mentioned; however, there were a few differences to note. While the general layout areas are the same (work area, navigation pane, and application options), both the navigation pane and work area cannot not be open in the same window.

This causes users to have to switch back and forth either by swiping left/right or by using the browse button Browse.

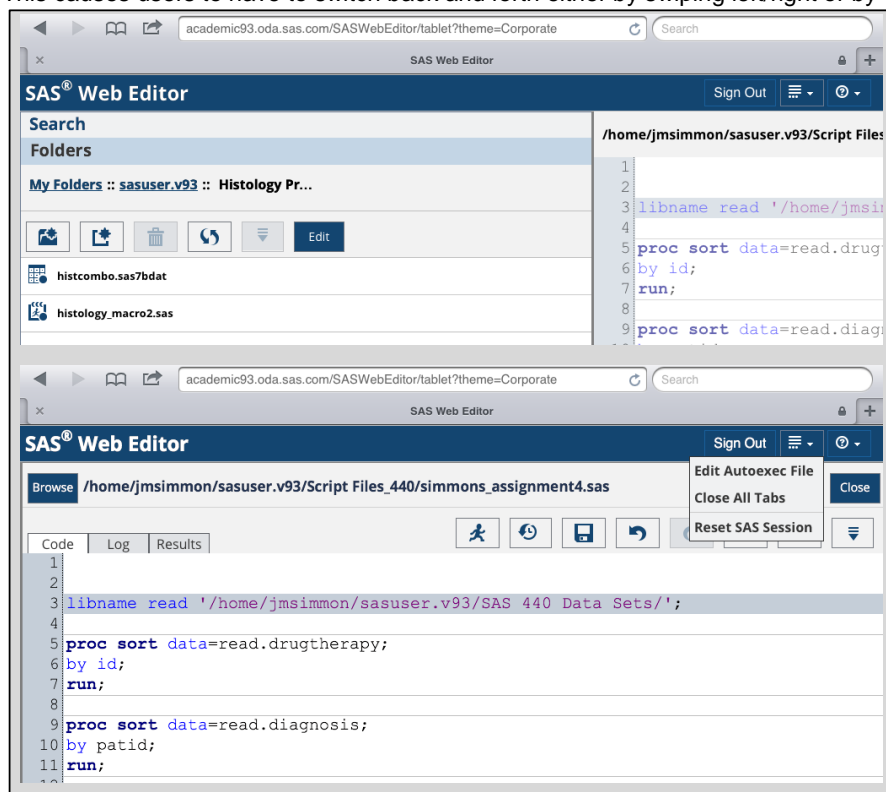


Figure 8. iPad Working Area

Users also lose the expandable log display (shown in figure 4) of errors, warnings, and notes, making scrolling through the log after code submission critical. The Graphics Editor download option which is available on the PC Internet Browser version is not available in Safari for download on an iPad; this would normally be found on the dropdown next to the Sign Out option. On the iPad the option of saving results in the PDF, RTF, or HTML format is not available, but the results may still be opened in new tabs of the browser.

As the mobile Web Editor is on the iPad everything is single touch and users no longer have the option of right-click or left-click to view the properties of files and copy file locations for less typing. Instead, users can tap on “Edit” where they can select a file and then select the dropdown where they have the option to open the file properties or create a shortcut to the file for use in code as seen in figure 9. An additional feature lost is the ability to hover over icons to know what they represent, so it is beneficial that users are familiar with the Web Editor on a computer prior to working with the mobile Web Editor on an iPad. Although there are differences between the web browser and iPad, which mainly surround screen layout and options, these differences are not significant and users can easily adjust between them with continued usage.

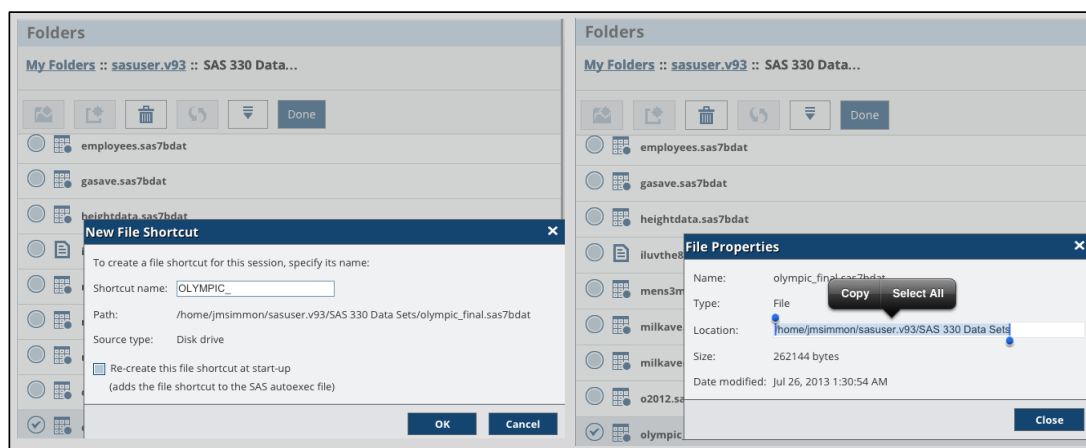


Figure 9. File Properties and Shortcuts

CONCLUSION

For students learning SAS for the first time, using the Web Editor with OnDemand is a good choice for basic instruction if there is no other access to SAS available at the University. If access to Macs is not required then coding can be deployed in OnDemand for Enterprise Guide (EG) or the Web Editor. If Mac access is required then the Web Editor is the only OnDemand option. For use in industry the Web Editor would work well for a company with the ability to install it on a server that has plenty of hard drive space for data as well as processing power and large amounts of RAM to enhance run times. For professionals who want to keep up with their SAS programming skills but do not have access to SAS, the Web Editor through OnDemand would be one way to stay current with the software and certification.

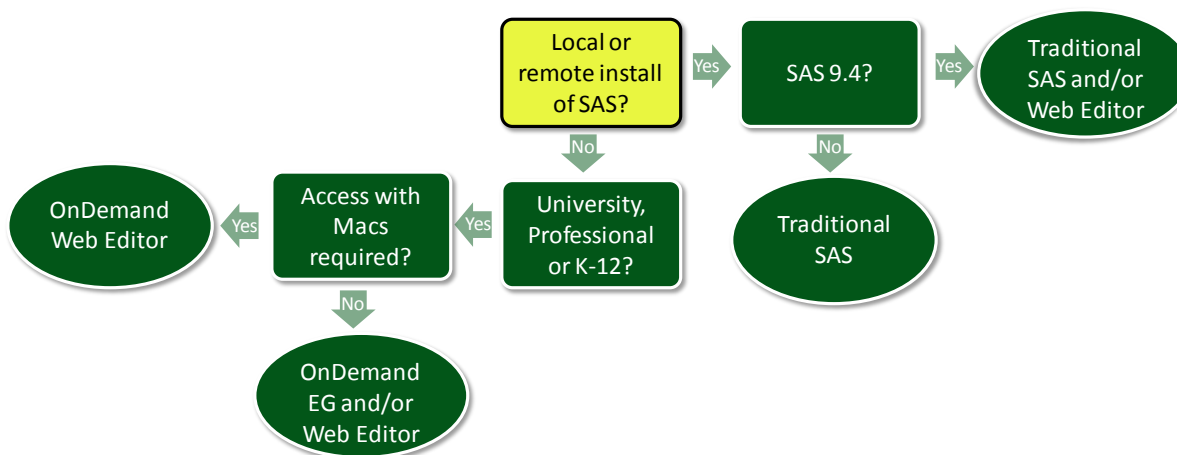


Figure 10. Various Pathways for Basic SAS Access

As with any software there is always room for improvement. Items that would be extremely helpful are a break key for escaping a program submission, although eventually there is a time out due to the web browser and server environment. An expandable results section would be a nice feature to allow for better navigation of the output. Finally, an easier method for saving data sets locally, including directly from the course folder, would be helpful. While it would be fabulous if the OnDemand server could allow larger data sets to be stored and processed it is understandable that SAS cannot get into the business of hosting large amounts of data. This is an unfortunate limitation of the OnDemand Web Editor because the best way for students to learn SAS is to challenge them with gigantic data sets and complex processing.

REFERENCES

- ¹ "SAS OnDemand for Academics Step-by-Step Registration and Installation Guides." SAS. 2013. Available at <http://support.sas.com/ondemand/steps.html>
- ² "SAS OnDemand for Academics: SAS Web Editor Instructor Registration Instructions." SAS. 2013. Available at <http://support.sas.com/ondemand/manuals/SASWebEditorInstructor.pdf>
- ³ "SAS OnDemand for Academics: SAS Web Editor Student Registration Instructions." SAS. 2013. Available at <http://support.sas.com/ondemand/manuals/SASWebEditorStudent.pdf>
- ⁴ "SAS OnDemand for Academics Instructor User's Guide Third Edition." SAS. 2013. Available at <http://support.sas.com/ondemand/manuals/AcadInstrucManual.pdf>
- ⁵ "SAS OnDemand for Academics Student User's Guide Third Edition." SAS. 2013. Available at <http://support.sas.com/ondemand/manuals/StudentManual.pdf>
- ⁶ "SAS OnDemand for Academics." SAS. 2013. Available at <http://support.sas.com/ondemand/index.html>
- ⁷ "Accessibility features for SAS Web Editor 2,2 (for Browsers on Microsoft Windows and Mac OS X)." SAS. 2013. Available at <http://support.sas.com/documentation/onlinedoc/webeditor/2.2/webeditora11y.html>

CONTACT INFORMATION

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

Name:	Rebecca Ottesen
Address:	1 Grand Avenue, Statistics Dept
City, State ZIP:	San Luis Obispo, CA, 93407-0405
Work Phone:	805-756-2709
Fax:	805-756-2700
E-mail:	rottesen@calpoly.edu
Web:	http://statweb.calpoly.edu/rottesen/
Name:	Jamelle Simmons
Address:	Virginia Polytechnic & State University
City, State ZIP:	Blacksburg, VA 24060
E-mail:	jmsimm13@vt.edu

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.