

Paper 046-2013

## SAS® Stored Processes Are Goin' Mobile! – Creating and Delivering Mobile-Enabled Versions of Stored Process Reports

Michael Drutar, SAS Institute Inc., Cary, NC

SAS® BI programmers have been clamoring for a way to quickly create mobile-enabled versions of existing SAS BI content. This can be difficult because there may be multiple BI reports that need to be converted. Fortunately, PROC STP (new in SAS 9.3) is a solution that can take a system's existing SAS Stored Process Reports and create mobile-enabled versions of them. This paper shows how PROC STP can capture the ODS HTML output from a stored process, create an HTML file from it, and e-mail the file to any mobile device (iPhone, Android, etc). Using this method, any existing Stored Process report's output can easily be mobile-enabled. Also discussed: how to allow users to send Report Output to mobile 'On Demand'.

### CREATING SAS ODS OUTPUT

The first step to solving this problem is to understand how to create and store a ODS HTML file generated from SAS procedures. For the sake of simplicity, the first example in this paper leverages the PROC PRINT procedure to print out the sashelp.class table. PROC PRINT can be run from any SAS programming environment; however, the challenge is how to direct the ODS HTML output to a permanent location. This is achieved by using the ODS HTML option while including the path and file syntax options. An example of this can be found below:

```
ods listing close;
ods html path='C:\temp\SGF' file='proc_print.htm' style=HTMLBlue ;
proc print data=sashelp.class noobs;
run;
ods html close;
ods listing;
```

The resulting HTML file is generated and placed in the path noted in the path= option:

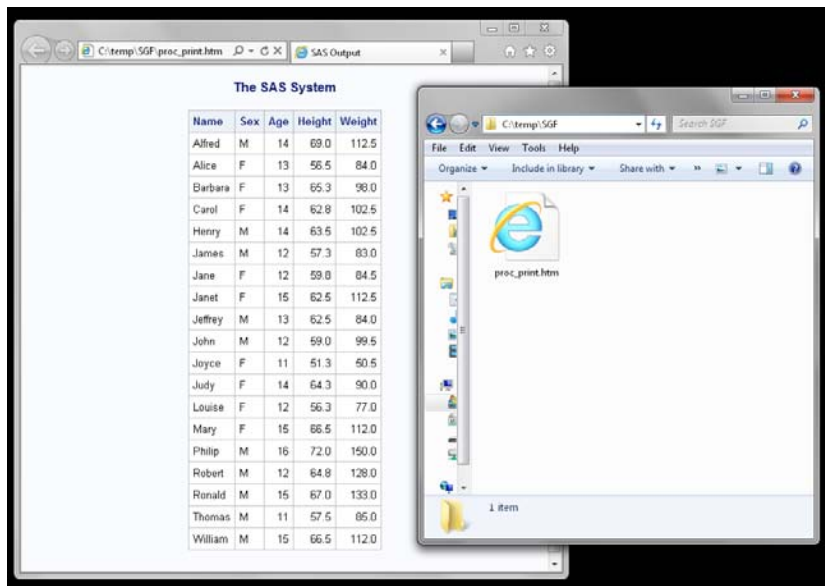


Figure 1. ODS HTML Output

Notice that by using the file= syntax option within the ODS HTML statement, the programmer can also control the name of the produced HTML file.

### DELIVERING SAS ODS OUTPUT TO MOBILE DEVICES

For quite some time now, SAS has had the ability to send e-mail via a SAS program. This is achieved via the FILENAME statement and DATA step. This is well documented on SAS' support site on FILENAME statement, EMAIL (SMTP) access method. For review, the reader can find that documentation here:

<http://support.sas.com/documentation/cdl/en/lrdict/64316/HTML/default/viewer.htm#a002058232.htm>

In particular, the reader should pay attention to the ATTACH statement. By leveraging this statement, the programmer can send an e-mail to one or more recipients, and attach the generated HTML file generated from the example above (which exists on the file system at 'C:\temp\SGF'). This is achieved via the code shown below:

```
filename sendmail email "anybody@company.com"
  subject="Attached Is The Class Dataset Printed"
  attach="C:\temp\SGF\proc_print.htm";

data _null_;
  file sendmail;
  put 'Hi Mike, ';
  put 'Attached is a SAS ODS Report.';
  put 'It is a listing of the class dataset';
run;
```

This code sends the recipient an e-mail with the ODS HTML file attached, as depicted in the following screenshot:

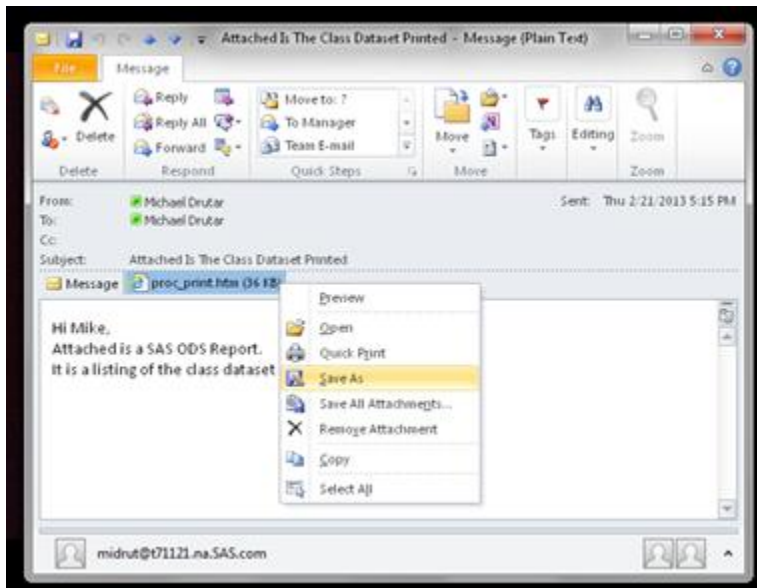


Figure 2. ODS HTML Attached to E-mail

## RECEIVING HTML ATTACHMENTS ON MOBILE DEVICES

Most, if not all, mobile devices include a web browsing client by default. This is good news for the SAS programmer. The web browsing client allows the mobile device's user to view HTML web pages (such as <http://www.sas.com>). Not only can the mobile device view HTML files that are on the web, but the device can also view static HTML files, which exist on the phone's file system. Therefore, when a mobile user receives an HTML file as an e-mail attachment, it can be downloaded to the file system, and then opened by the device's web browser. The downloading and viewing of an HTML file is usually achieved by a single tap on the file attachment within the e-mail. The HTML file is downloaded to the file system and is then automatically opened:

The image shows a hand holding a smartphone. The screen displays a SAS ODS HTML report titled "The SAS System". The report contains a table with the following data:

Name	Sex	Age	Height	Weight
Adrian	M	14	66.0	112.5
Alice	F	13	56.5	84.0
Barbara	F	13	65.5	98.0
Carol	F	14	62.8	102.5
Henry	M	14	63.5	102.5
James	M	12	57.5	83.0
Jane	F	12	59.8	86.5
Jane1	F	15	62.5	112.5
Jeffrey	M	13	62.5	86.0
John	M	12	59.0	95.5
Joyce	F	11	51.5	58.5
Judy	F	14	64.5	90.8
Louise	F	12	56.5	77.0
Mary	F	15	65.5	112.0
Philip	M	16	72.0	136.0
Robert	M	12	64.8	128.0
Richard	M	15	67.8	133.0
Thomas	M	11	53.5	65.0

**Figure 3. ODS HTML Being Viewed on a Mobile Device**

Using the method described above, we now have a solution of how to send SAS ODS reporting directly to mobile devices. The best part of this solution is that the SAS ODS HTML files incorporate all of the customizations one might make to a report directly within a file. For example, a programmer might create a listing table like the one above and customize it to have conditionally highlighted cells, different fonts within cells, and so on. By default, all of these customizations are generated by the SAS ODS HTML statement and embedded into the actual HTML code; therefore, when the HTML file is viewed (be it on a mobile device or on a computer), all customized styles will render correctly. There is one type of SAS ODS HTML output, which cannot be embedded directly into the HTML file. That output is HTML files that contain images.

## REFERENCING SAS-GENERATED IMAGE FILES ON A MOBILE DEVICE

In simple HTML code, an image is referenced by an IMG tag. The image tag code is usually similar to the code below:

```

```

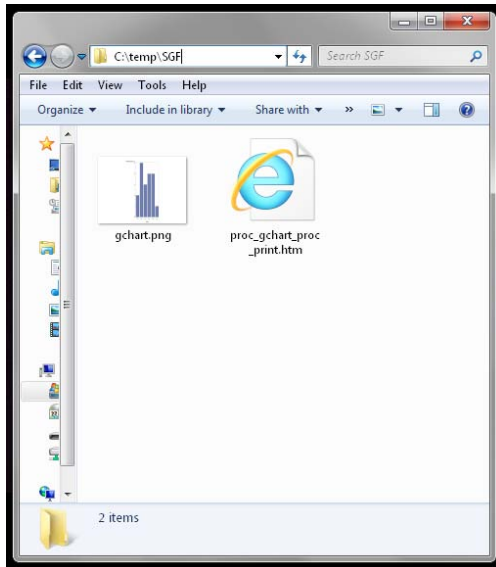
This means that when the HTML is rendered by a web browser (be it mobile or computer based), as the browser reaches the line that contains the IMG tag, it sees the src= statement. The browser is smart enough to travel through the web to the location of that image (in this case gchart.png). The browser then even places that image directly in the HTML code at that line. Hence, the web page is displayed with the image in the correct place.

When the SAS ODS HTML file is generated and it contains graphics, SAS creates image file(s) (for example gchart.png) and places it on the file system. Then, within the HTML code of the final ODS HTML file, it references this file and location via an IMG tag similar to the one from the example HTML code above. For example, if we take the code from the previous ODS HTML file example and add a graphics procedure (PROC GCHART in this case).

```
ods listing close;
ods html path='C:\temp\SGF' file='proc_gchart_proc_print.htm' style=HTMLBlue;
proc gchart data=sashelp.class;
  vbar age / discrete
  des="" name="gchart";
run; quit;

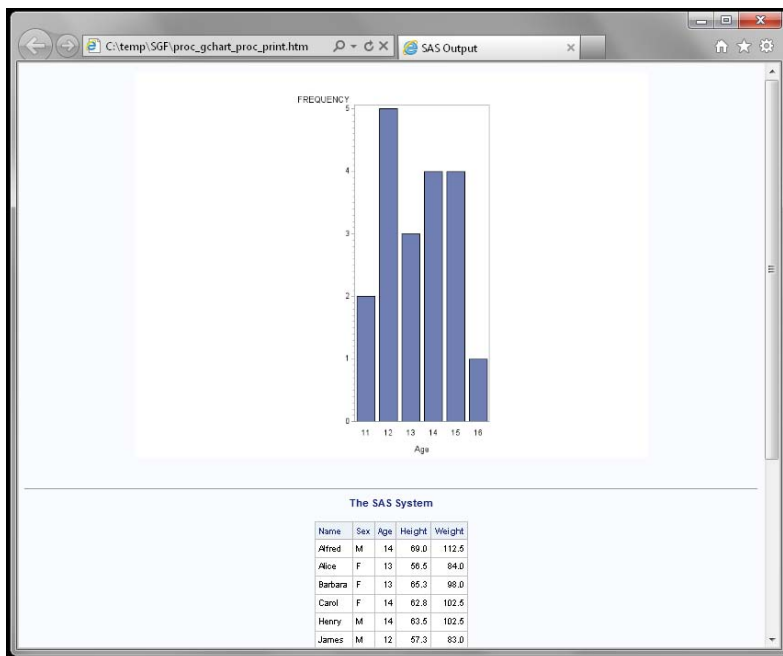
proc print data=sashelp.class noobs;
run;
ods html close;
ods listing;
```

SAS creates the expected HTML file at the correct path; but SAS also creates the gchart.png file at the same location.



**Figure 4. ODS HTML Output in the Specified Location**

Here is the browser rendered ODS HTML file as viewed on a computer:



**Figure 5. ODS HTML File as Viewed on a Computer**

As you can see, the HTML file has the PROC GCHART generated image above the PROC PRINT table. And within the actual HTML code, the ODS HTML generated there is an image tag:

```

```

So the question is how to deliver an ODS HTML file that contains SAS ODS graphic images similar to the one above and have it render correctly on a mobile device. If we use the HTML attachment method from above, the mobile device's web browser will become confused. At the `img src=` section of the HTML file, the browser is expecting to see a value such as `http://some-web-page/gchart.png`, not `C:\temp\SGF\gchart.png`. The mobile browser will not know where to retrieve the image for placement on the HTML page when it is rendered. There is the option of possibly attaching the ODS HTML generated graphic images along with the HTML file. But this solution will not work because the `img src=` tag will be referencing a location, which is on the mobile device's file system itself. Therefore, the programmer must find a way to create the HTML file in one location, but place the actual image files on a web server, which can be referenced via the HTTP protocol. The programmer must then customize the ODS HTML generated file to have the `img src=` tags reference the web server's location, while placing the HTML file itself in a different location. Fortunately, the ODS HTML statement has built-in options for achieving all of these needs.

Within the ODS HTML statement there are three supplemental options which solve this problem. First: the need to place the ODS HTML file in one location on the file system, but place the generated graphic images in another. This is achieved by leveraging the `gpath=` option. Similar to the `path=` statement which allows the programmer to control where on the file system the ODS-generated HTML is placed (`path='C:\temp\SGF'` in the previous example); the `gpath=` option will place the generated graphics where the programmer desires. Therefore, the programmer can now place the ODS generated graphics on a web server that can be referenced later by a web browsing client. Additionally, the programmer can create subfolders within file system on the web server and place the graphic images to certain reports' images organized when they are referenced later. An example of this is `gpath="<-- path to web server -->report1"`.

Now that the programmer can control where the ODS images are being placed there is an issue of referencing those images via the HTTP protocol within the IMG tags of the generated HTML. The appropriate ODS HTML option to achieving this is the `url=` option. By leveraging this statement and placing the appropriate web address to the programmers web server (such as `http://programmers-web-server/report1`), the ODS-generated HTML file will automatically have that value within the `img src=` tag of the HTML code.

Here is the previous example using the new ODS HTML options discussed above:

```
ods listing close;
ods html path='C:\temp\SGF' file='proc_gchart_proc_print.htm' gpath="<-- path to web
server -->\report1" (url="http://programmers-web-server/report1/") style=HTMLBlue;

proc gchart data=sashelp.class;
  vbar age / discrete
    des="" name="gchart";
run; quit;

proc print data=sashelp.class noobs;
run;

ods html close;
ods listing;
```

When this program runs, it creates the graphic images and places them on the web server's file system. It also creates the HTML file. Within that file, it generates the `img src=` tag with the appropriate reference. The IMG tag from the program above is the following:

```

```

The ODS HTML statement is an extremely powerful piece of the SAS Output Delivery System. With all these features being utilized, the programmer can create an HTML file that includes graphic images that can be sent to a mobile device as an attachment. When the mobile device opens the generated HTML file, the ODS generated HTML renders correctly. Also, where graphic images are located within the HTML code, the mobile device's browser sees the `img src=` tag reference to `http://programmers-web-server/report1/gchart.png`. The browser retrieves the

image and then displays it in the correct location on the rendered view of the HTML page. \*NOTE: some mobile devices standard web browser might now display the references graphic images correctly. If this occurs, try using a different browser on the mobile device.

## PROC STP

Using the method described in this paper, a SAS programmer can easily create an HTML file and send it to a mobile device. With this method in our arsenal, the challenge is how the programmer takes existing BI content (specifically SAS Stored Processes) and easily convert it to ODS HTML. The first option is to extract the source code from the SAS Stored Process and manually convert it so that it will generate the ODS HTML. But this method is very manual and not sustainable, especially because SAS Stored Procedures can be very extensive with lots of graphics, tables, and other content. Fortunately, a new procedure in SAS 9.3 can do the heavy lifting for the programmer.

PROC STP is a new procedure that can execute an existing stored process from a SAS Metadata Server. Not only will PROC STP execute the stored procedure, but it can capture the output from the stored procedure for archiving. In order to solve the 'mobile conversion' problem described in this paper, the programmer can leverage PROC STP and combine it with the ODS HTML method from above to create a mobile-ready version of the stored procedure.

Before the PROC STP procedure can be executed, a few options must run. First, PROC STP needs to be directed toward a SAS Metadata Server on which the target stored process is stored. In order to connect to a SAS Metadata Server from a BASE SAS session, the options `metaserver=`, `metaport=`, `metauser=`, `metapass=` values with the appropriate SAS Metadata Server, metadata port, and the user ID and password. An example of these options will be shown later.

When the `metaserver` option has been executed with the appropriate values, the SAS session is now connected to the desired SAS Metadata server and PROC STP can be executed. First, the programmer must specify the metadata location of the stored process that is to be executed. This value is placed in the `program=` argument. When the metadata location is in place, PROC STP can execute the stored process on the remote server.

Before running PROC STP, the programmer should specify what to do with the resulting output from the stored procedure. For this task, there is a second optional argument in PROC STP called `odsout`. For the `odsout` option, there are two possible values. The first value is `odsout=replay`. If PROC STP is run with the `replay` value specified, the procedure executes the stored process on the remote server and then simply displays the results within the output window. While it is nice to see the output, this is not the programmer's goal. The programmer needs to capture and store the generated output from the stored process for mobile delivery. Therefore, when running PROC STP the programmer specifies `odsout=store`. An example of a program running PROC STP is displayed below:

```
options metaserver = 'server-name'
        metaport   = 9999
        metauser   = 'user-id'
        metapass   = 'password';

proc stp program='/SGF/Report1'
odsout=store;
run;
```

In this example, first the SAS session connects to the metadata server and locates the stored process; the stored process is named 'Report1' and is in the metadata folder '/SGF'. Then, PROC STP executes the stored process and stores the output for use later in the SAS Session. It should be noted that if the programmer runs this code alone, no output is generated in the results window.

## PROC DOCUMENT

Now that the programmer has successfully executed the stored process and stored the output, the next challenge is creating an ODS HTML file from this stored output. This is achieved by leveraging the SAS procedure PROC DOCUMENT. The PROC DOCUMENT procedure is used by the SAS system to create or modify documents.

Because a copy of the output of the PROC STP procedure is needed, PROC DOCUMENT will be used to create a document. The best part of the PROC DOCUMENT procedure is that it can be used in conjunction with the ODS HTML statement(s) described earlier in this paper. By submitting an ODS HTML statement with the path=, gpath= and url= options correctly specified, the programmer can place the stored results from PROC STP in all the same locations as the previous ODS HTML examples. Example code for PROC DOCUMENT with the ODS HTML statement is displayed below:

```
goptions device=png;
ods listing close;
ods html path='C:\temp\SGF' file='Report1.htm' gpath=""<-- path to web server --
>\report1" (url="http://programmers-web-server/report1/") style=HTMLBlue;

proc document name=&_ODSDOC (read);
  replay / levels=all;
run;

ods html close;
ods listing;
```

By using the ODS HTML statement with the appropriate options and values, the code above creates an HTML file that is 'mobile-enabled'. The ODS HTML exists in a single HTML file, the graphical images (if any) are generated and placed on the web server, and the IMG tags (if any) within the HTML file that references the correct web location via the HTTP protocol. All that remains is to e-mail the file to the user.

## END TO END AUTOMATION

To achieve the goal of creating mobile-ready versions of existing SAS Stored Processes and deliver them to a mobile device, all three methods described in this paper are necessary. First, the programmer executes PROC STP that is directed at a particular stored process, and the output will be stored. Then, the programmer uses the ODS HTML statement in conjunction with PROC DOCUMENT to create an HTML output file containing the results generated from PROC STP. Finally, that output file will be e-mailed to a user's mobile device. This will all be achieved by piecing together the examples from above, with one exception.

Thus far, the ODS HTML statement has been used to place the resulting HTML file in one location (C:\temp\SGF) and the graphic images generated on a web server's file system. In the final example below, the actual HTML will also be placed on the web server's file system. The reason for this is to help account for all mobile device hardware and operating systems. If a user's mobile device is unable to display the attached HTML file, the user will have a second option to view the produced HTML file by navigating to it directly on the web. Therefore, the example below has identical values for path= and gpath=. After submitting the code, the graphic image will exist at <http://programmers-web-server/report1/gchart.png> and the actual HTML file will exist at <http://programmers-web-server/report1/Report1.htm>. The programmer can now place a link to a 'live web' version of the stored process output directly in the body of the e-mail message.

A full example of this method is below:

```
options metaserver = 'server-name'
      metaport      = 9999
      metauser      = 'user-id'
      metapass      = 'password';

proc stp program='/SGF/Report1'
odsout=store;
run;
ods html path=""<-- path to web server -->\report1" file='Report1.htm' gpath=""<-- path
to web server -->\report1" (url="http://programmers-web-server/report1/")
style=HTMLBlue;
```



```
proc document name=&_ODSDOC;
  replay / levels=all;
run;

ods html close;
ods listing;

filename sendmail email "anybody@company.com"
  subject="Attached Is The Class Dataset Printed"
  attach="<-- path to web server -->\report1\Report1.htm";

data _null_;
  file sendmail;
  put 'Hi Mike,';
  put 'Attached is a SAS ODS Report.';
  put 'It is a listing of the class dataset';
  put ' ';
  put 'If you are having trouble viewing the attached file, you can also view it by
clicking the link below';
  put 'http://programmers-web-server/report1/Report1.htm';
run;
```

The following display shows the success of this code and how it resulted in an existing SAS Stored Process output being sent and displayed on a mobile device:

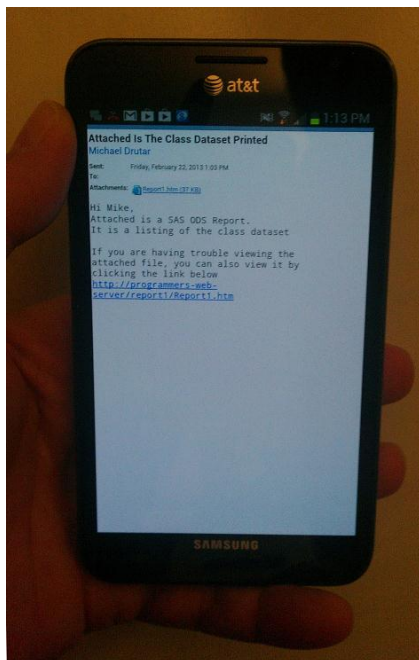


Figure 6. E-mail on Mobile Phone with Both HTML File Attached and Link Reference in Message Body



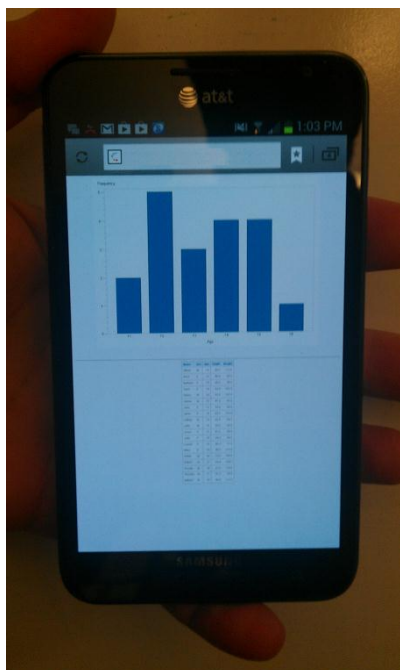


Figure 7. ODS HTML File Displayed on a Mobile Device

## SENDING STORED PROCESS OUTPUT TO A MOBILE DEVICE ON DEMAND

The method of delivering stored process output to mobile devices can be automated end to end by running the code that has been provided in this paper. Using this method, if the programmer needs to convert twenty five stored processes and deliver their outputs, the solution could be to write twenty five SAS programs with a different stored process in each PROC STP statement. However, this method is not very efficient. Instead, the programmer could write one such program and have it saved on their machine. And then when a request comes in for a stored process to be delivered remotely, the programmer could open the program, edit the PROC STP procedure by inserting the name and location of the stored process being requested and then submit the code. However, this process is still quite manual.

A different approach would be to create a new stored process users can access. This new stored process will have code similar to the example above as its source code; however, a necessary change would be for the programmer to the call to the stored process to be a macro variable. The user can then add a prompt to the new stored process, which in turn, prompts the user to choose an existing reporting stored process on the system. Then, when the stored process is executed, it replaces the macro variable in PROC STP accordingly and the SAS system will continue running the overarching stored process. The stored process runs, and almost instantly the user receives an e-mail with an attached HTML file that contains the output from the stored process they selected. Thereby, a user can send a stored process output to their mobile device on demand.

## CONCLUSION:

PROC STP is an extremely powerful procedure within the SAS system. By leveraging it, and using some other tricks from SAS Output Delivery System, a user can now create a file with any existing stored process output. The ODS HTML procedure also places the graphic images (if any) on a web server. ODS HTML references the correct HTTP protocol location of the images within the produced HTML. This file can now be sent to a mobile device using the established filename e-mail method. In short, as programmers struggle with how to display their existing SAS BI content on mobile phones, leveraging PROC STP and the method described in this paper solves a big piece of the puzzle.

## REFERENCES

The SAS Institute. "FILENAME Statement, EMAIL (SMTP) Access Method." Support.sas.com. Nov, 15<sup>th</sup> 2012. Available at <http://support.sas.com/documentation/cdl/en/lrdict/64316/HTML/default/viewer.htm#a002058232.htm>

The SAS Institute. "STP Procedure" Support.sas.com. Jan, 12<sup>th</sup> 2013. Available at <http://support.sas.com/documentation/cdl/en/stpug/62758/HTML/default/viewer.htm#p1oh9e7gnrxg4yn1lqfeefi2355l.htm>

## CONTACT INFORMATION

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

Michael Drutar  
SAS Campus Drive  
SAS Institute Inc.  
E-mail: [Michael.Drutar@sas.com](mailto:Michael.Drutar@sas.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 countries.