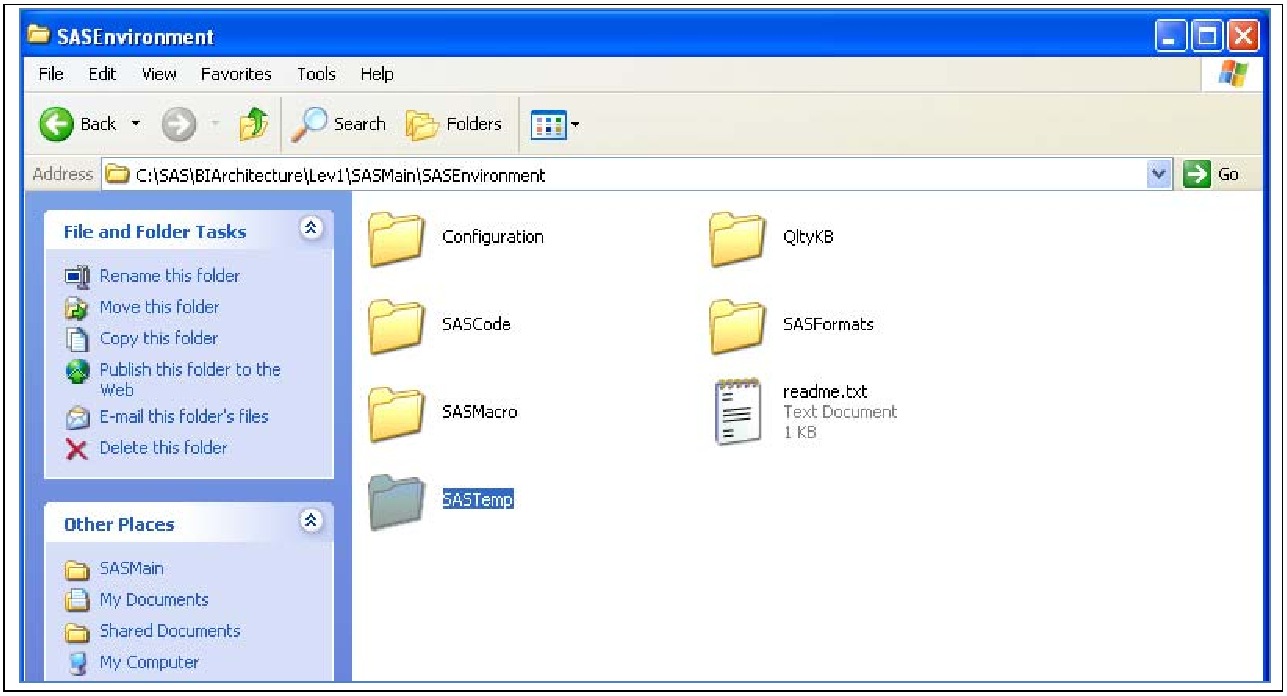
NESUG 2008 Handout: Using Customized Templates with SAS Stored Processes
By Cynthia Zender

These are the general steps for creating a user-defined tagset template in a permanent item store on a single machine install of the Enterprise Intelligence Platform. These same concepts would also apply to style and table templates – for any client application that receives streaming HTML results (**not** WRS or PPT).

1) Create a user-defined directory in a folder that you know the servers can all find. I recommend the SASEnvironment directory (where the formats are recommended to be stored):



Address: C:\SAS\BIArchitecture\Lev1\SASMain\SASEnvironment

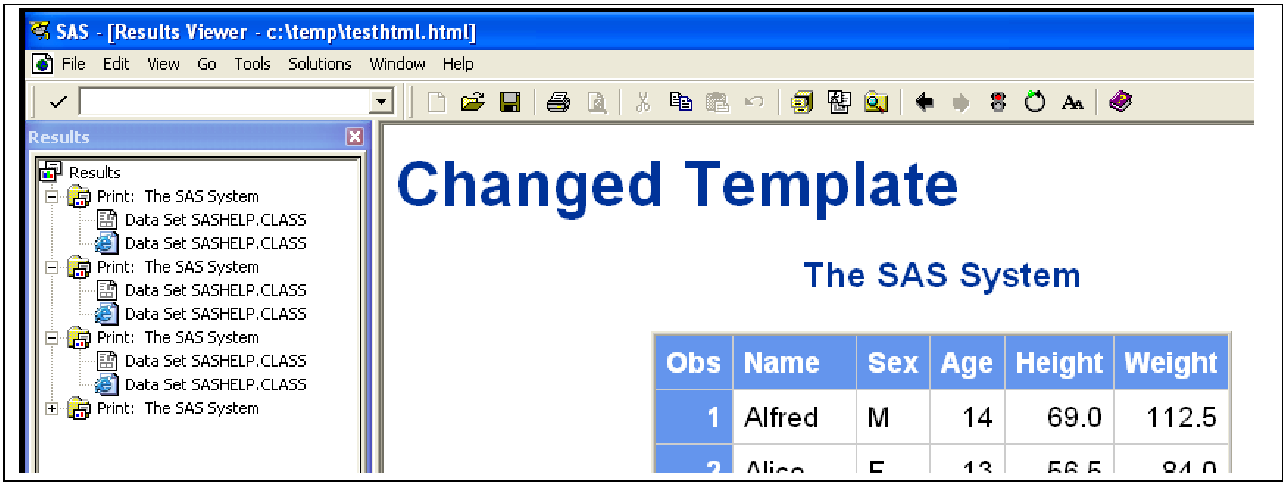
Note how the **SASTemp** folder has been added to this location. In my single machine install, these directories are on my C: drive. In a multi-machine environment, you would need to find this location and possibly ask your administrators to make this directory for you. You MUST know the physical path of this directory. In my local install, this path is:
C:\SAS\BIArchitecture\Lev1\SASMain\SASEnvironment\SASTemp
In a multi machine environment, this location could be:
\\ServerName\SAS\BIArchitecture\Lev1\SASMain\SASEnvironment\SASTemp

2) Now, submit a job (NOT a stored process) that creates an item store in this directory. You may need to ask your SAS Administrator for information on how to run SAS on this server or how to Telnet to this machine. Generally, in a multi-machine environment, you might have to use SAS/Connect or some other interface to submit jobs on a server machine. If you use EG, to create the item store, be SURE that you have write access at the physical disk level to this location. The job to create a new TAGSET template would look something like this (only yours will probably be longer). Do NOT use the META libname engine to point to this location. The META engine does not recognize SAS catalogs or template stores. You MUST use a physical path name for the server location of the template (shown in LINE1 below).

```
libname lb_srv "C:\SAS\BIArchitecture\Lev1\SASMain\SASEnvironment\SASTemp";

ods path lb_srv.templat(update)
         sasuser.templat(read)
         sashelp.tmplmst(read);

proc template;
  define tagset tagsets.myhtml;
  parent=tagsets.htmlcss;
     define event doc_body;
         put "<body onload=""startup()""";
            put " onunload=""shutdown()""";
            put " bgproperties=""fixed""" / WATERMARK;
            putq " class=" HTMLCLASS;
            putq " background=" BACKGROUNDIMAGE;

            trigger style_inline;
            put ">" NL;
            trigger pre_post;
            put NL;
            put "<h1>Changed Template</h1>" NL;
            trigger ie_check;

         finish:
            trigger pre_post;
            put "<h1>Changed Template</h1>" NL;
            put "</body>" NL;
      end;
  end;
run;

ods tagsets.myhtml file='c:\temp\testhtml.html' style=sasweb;
proc print data=sashelp.class;
run;
ods tagsets.myhtml close;
```

This program adds an <H1> tag at the top and bottom of the HTML file – just so you can have visible proof that the changed template is being used.

When the results of this program are done, you should see the screenshot below to show the <H1> tag was used (only showing top of HTML file in the SAS Results – you can see that I tested the PROC PRINT several times before I was finally done!):



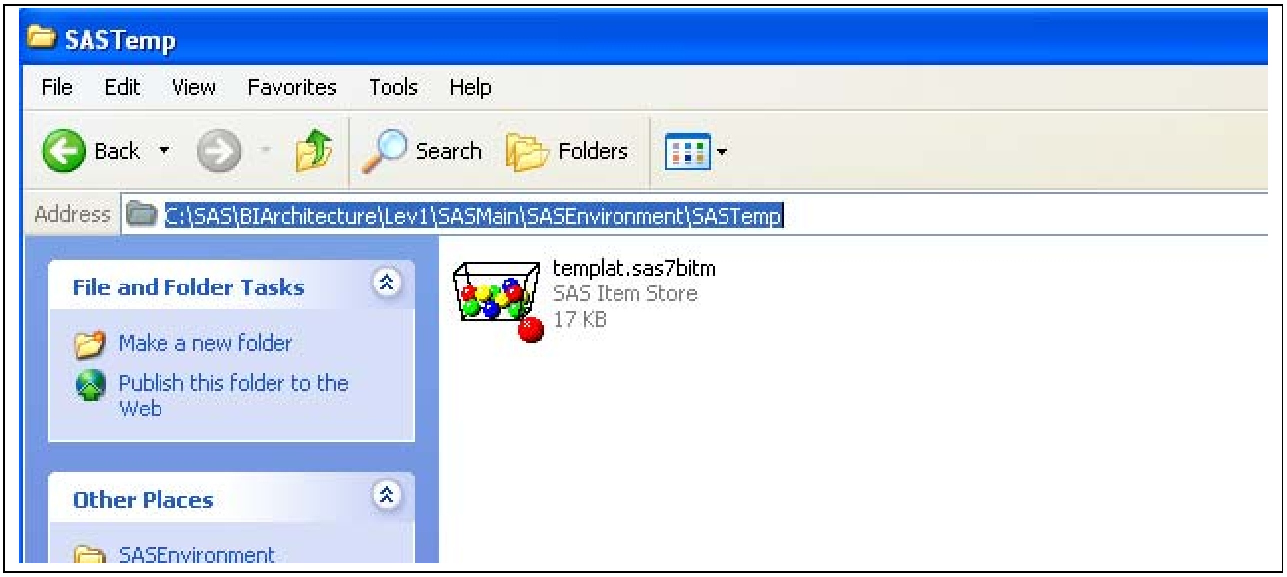
Changed Template

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

SAS Output Using New Template in Session that Created Server Template Store

3) Of course, instead of the C:\ drive location, you have to use the server location that both the SP Server and the Workspace server know about. After this job is done, if you navigate or issue a DIR command against the server location, you should see your new item store:



C:\SAS\BIArchitecture\Lev1\SASMain\SASEnvironment\SASTemp

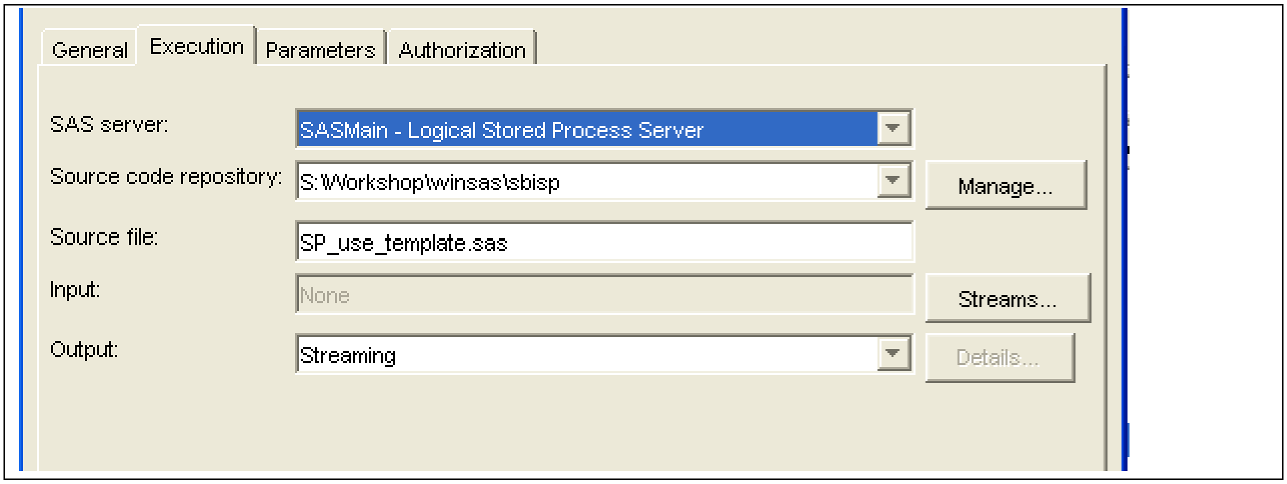
templat.sas7bitm
SAS Item Store
17 KB

4) Now, you need to write your stored process to USE this new tagset template. Remember that this template will ONLY work in client applications (like EG, the Portal, or the Add-in for Microsoft Office that ACCEPT HTML results) – this stored process will NOT work in Web Report Studio or in PowerPoint with the SAS Add-in.

My stored process uses an ODS PATH to point to the permanent item store. My stored process code is in a repository location named: **SP_use_template.sas**

```
*ProcessBody;

libname lb_srv "C:\SAS\BIArchitecture\Lev1\SASMain\SASEnvironment\SASTemp";
** note, this is the libname for a single machine install;
** in a production environment, your path name would be something;
** else, like:
libname lb_srv "\\ServerName\SAS\BIArchitecture\Lev1\SASMain\SASEnvironment\SASTemp" ;
**;

ods path(prepend) lb_srv.templat(read);

%let _odsdest=tagsets.myhtml;
%let _odsstyle = sasweb;
%let _odsstylesheet=;

%stpbegin;
proc print data=sashelp.class;
run;
%stpend;
ods path(remove) lb_srv.templat(read);
```

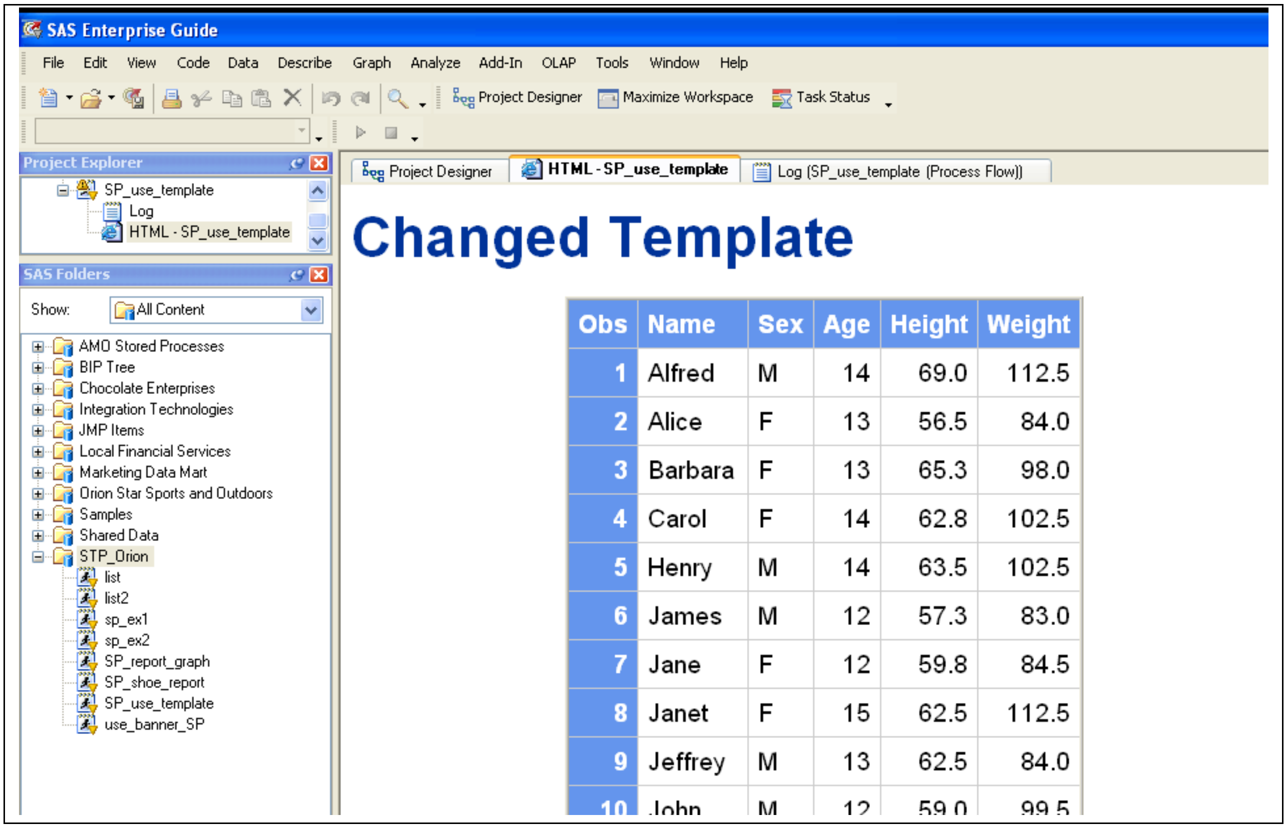
5) Now, you need to register the SP that uses the new tagset template as a destination. In the screen shot below I have saved the SP into a location on a shared drive, the S: drive. My stored process does not use any parameters, so I'm only showing the execution parameters. On my single machine install, this repository location is known as **STP_Orion** in SAS Management Console and I named the SP the same as the SAS program, **SP_use_template**:



SAS server: SASMain - Logical Stored Process Server
Source code repository: S:\Workshop\winsas\sbisp
Source file: SP_use_template.sas
Input: None
Output: Streaming

6) I know that EG will ACCEPT HTML streaming results. So, if the SP works when tested in EG, then it should also work when invoked from the Portal. So, now, I switch over to EG, add the

SP_use_template stored process to my project (in Project Explorer), then right mouse click and run the SP. The results I see in EG are the same as when I tested the template creation.



Changed Template

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
9	Jeffrey	M	13	62.5	84.0
10	John	M	12	59.0	99.5

HTML Results Using New Template TAGSETS.MYHTML using stored process SP_use_template which is in the STP_ORION repository

7) Now, you also need to test the template in other client applications. As I explained earlier in #4, you should NOT expect this tagset template to work in Web Report Studio or the Add-in for Office with PowerPoint, because those 2 client applications do not ACCEPT HTML results.
8) This same method also applies to any other templates that you want to update and use in the BI platform. The need to use a physical path name also applies to any SAS catalogs or item stores (like template item stores or document item stores or SCL BUILD catalogs, or format catalogs, etc) that you might want to use. Any permanent "non-dataset" library should be referenced in your code with the physical server location for the resource. For SAS item stores, you always need the ODS PATH statement to establish the search path for the item store.

For more help on using custom templates with SAS Stored Processes in the Enterprise Intelligence Platform, you should contact SAS Technical Support. The steps above were performed on a single-machine Platform install, which may be different code than that needed on your particular Platform configuration. If you have questions about **this** example, please contact the author: **Cynthia.Zender@sas.com** by email.