

## 7 Steps to a SAS® EBI Proof of Concept Project

Sheryl Weise, Wells Fargo Bank

### ABSTRACT

The Purchasing Department is considering contracting with your team for a new SAS Enterprise Business Intelligence application. He's already met with SAS and seen the sales pitch, and he is very interested. But the manager is a tightwad and not sure about spending the money. Also, he wants his team to be the primary developers for this new application. Before investing his money on training, programming, and support, he would like a proof-of-concept. This paper will walk you through the 7 steps to create a SAS EBI POC project:

1. Develop a kick-off meeting including a full demo of the SAS EBI tools.
2. Set up your Unix filesystems and security.
3. Set up your SAS metadata ACTs, users, groups, folders, and libraries.
4. Make sure the necessary SAS client tools are installed on the developers' machines.
5. Hold a SAS EBI workshop to introduce them to the basics, including SAS Enterprise Guide®, SAS Stored Processes®, SAS Information Maps®, SAS Web Report Studio®, the SAS Information Delivery Portal®, and the SAS Add-In for Microsoft Office®, along with supporting documentation.
6. Work with them to develop a simple project, one that highlights the benefits of SAS EBI and shows several methods for achieving the desired results.
7. Last but not least, follow-up!

Remember, your goal is not to launch a full-blown application. Instead, strive towards helping them see the potential in your organization for applying this methodology.

### INTRODUCTION

This paper will walk you through the steps we use to assist our users in creating a SAS Enterprise Business Intelligence proof-of-concept project. The client and web-based tools used are 9.2 versions of SAS Enterprise Guide, SAS Information Map Studio, SAS Add-in for Microsoft Office, SAS Management Console®, SAS Web Report Studio, and the SAS Information Delivery Portal. In order to keep the POC from getting too complex, this paper does not include cubes. The client server platform was Unix and did not use grid technology.

### STEP 1: THE KICK-OFF MEETING

First, build a **great** demo of your SAS Enterprise Business Intelligence products. The first impression is the most important. If you create a master demo with all the SAS products you offer, then you can whittle it down based on the users' needs. The first part of our demo contains slides with basic information, and the second half is a live demo of the products.

#### DEMO PART 1

Ours contains the following content:

- Basic department information including the services my team provides.
- Description of the SAS vendor, especially as an industry leader in the Gartner Group BI Magic Quadrant.
- Overview of SAS as an analytical tool and a decision making platform. Figure 1 provides an excellent start.

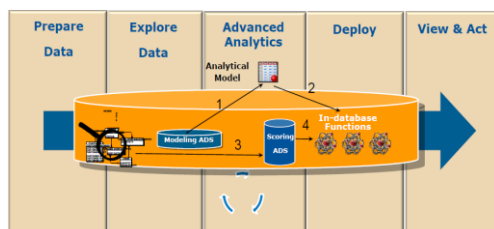


Figure 1. SAS Flow

- Describe the SAS architecture. Make sure to describe the SAS metadata and how it is the center of the SAS EBI platform. Figure 2, from the SAS website, represents all four tiers.

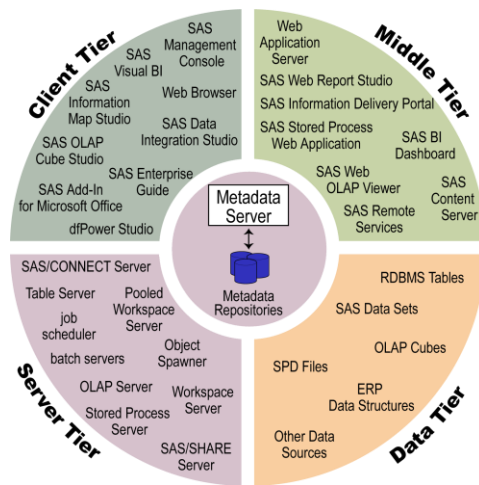


Figure 2. SAS Enterprise Business Intelligence Platform Tiers

- Finally, provide a description of the various tools you'll be showing them in the demo. Include some metadata examples of various groups and security to assure them their data and reports will be secure.

## DEMO PART 2 (LIVE)

Present an outline, screenshots, and description of the various SAS products and some of the capabilities. Here's the one I follow:

- Development and Analytics
  - SAS Enterprise Guide
  - SAS Access Engines
  - SAS Statistics
  - SAS Graph
  - SAS ODS
  - Teradata In-Database Analytics
- Staging Tools
  - SAS Information Map Studio
  - SAS Web Report Studio
  - SAS Enterprise Guide
  - SAS Stored Process Server
- Delivery Tools
  - SAS Information Delivery Portal (various portal types)
  - SAS Web Report Studio
  - SAS Add-in for Microsoft Office (AMO)

Before beginning the demo, go over the two paths you'll be following to create BI reports:

### Path 1:

- Build a SAS data set via SAS Enterprise Guide and register it
- Build a SAS Information Map on the data from #1
- Using SAS Web Report Studio, build a report using the Info Map from #2
- Surface the report on the SAS Portal

### Path 2:

- Build a SAS data set via SAS Enterprise Guide and register it
- Create a report using SAS Enterprise Guide
- Turn the report from #2 into a SAS Stored Process and test on the SAS Stored Process Server
- Surface the stored process on the SAS Portal

## CONCLUDE

If they are interested in proceeding, finish by filling out a SAS Active Project form. This can be done using MS Word, MS Excel, or, as we prefer, MS Sharepoint. The following information should be gathered:

- A 3-5 digit acronym representing the project/team. For this paper we'll use ZIP.
- A description of the project.
- Count of approximate number of developers and consumers.
- SAS Client software – determine versions and what is needed. Currently, we suggest PC SAS 9.2, SAS Enterprise Guide 4.3, SAS Information Map Studio 4.3, and, optionally, SAS Add-in For Microsoft Office 4.3. We do not offer SAS Management Console to our users but maintain that on the SAS Support team.
- SAS web tiers – determine which products they'll use: SAS Information Delivery Portal, SAS Web Report Studio, SAS Stored Process Server, and/or SAS Enterprise Miner.
- A lead contact with email address.
- Names, email addresses, roles, and IDs for each person to be set up in the metadata. For developers they must have a Unix ID whereas the consumers can use their ad-ent IDs.
- Include notes from each meeting, takeaway tasks, milestones met, etc.

## STEP 2: UNIX FILESYSTEMS AND SECURITY

Start with a test environment – you don't want newbies working with SAS metadata in a production environment. Set up numerous 50G POC filesystems ahead of time, so space is available as needed. This can then be broken up into multiple projects, using their chosen acronym. Within each filesystem, set up a standard set of directories. See Figure 3 for an example.

```
[srvr] sas /workspace/poc2/zip> ls -l
drwxrwxrwx  2 sas  sas           256 Aug 16 21:28 data
drwxrwxrwx  2 sas  sas           256 Aug 15 20:19 output
drwxrwxrwx  2 sas  sas           256 Aug 16 21:12 programs
drwxrwxrwx  2 sas  sas           256 Aug 15 20:19 reports
drwxrwxrwx  2 sas  sas           256 Aug 15 20:19 scripts
drwxrwxrwx  2 sas  sas           256 Aug 16 21:24 stp
```

Figure 3. Standard Unix Filesystems

Set up security at the acronym level, using Unix ACLs, to prevent others from accessing the filesystem. See a sample in Figure 4.

```
[srvr] sas /workspace/poc2/zip> aclget .
*
* ACL_type  AIXC
*
attributes:
base permissions
  owner(sas):  rwx
  group(sas):  rwx
  others:  ---
extended permissions
  enabled
  permit  rwx    u:sassvr
  permit  r-x    u:sastrust
  permit  rwx    u:a777777
  permit  rwx    u:a987654
```

Figure 4. Unix ACL




We use a SAS program to read the SAS metadata and get the users IDs to build Unix ACLs.

## STEP 3: SAS METADATA

The SAS metadata for this project consists of four parts: users, ACTs (Access Control Templates), folders, and libraries. They should be entered in the order indicated below. Log on as an unrestricted user (eg: sasadmin@saspw) for everything except defining the data library.

### 1. Groups:

Using the User Manager, set up a general group with the acronym name. This group does not have any roles – it is only for information to be shared on the SAS Portal with everyone on the project.

	ZIP	Overview folder for IDP only (no users)
	ZIP Consumers	Group for consumers of the POC
	ZIP Developers	Group for SAS developers with full read/write access

Then set up a group for each level of user. For this paper we're setting up the ZIP Developers and ZIP Consumers, both a part of the ZIP group. Add the appropriate roles depending on the capabilities you want them to have, as shown in Figure 5.

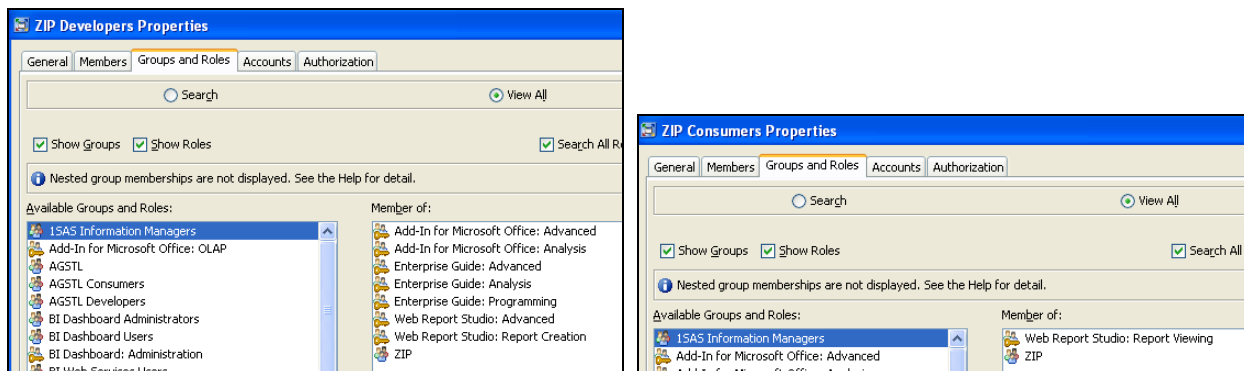


Figure 5 ZIP Groups in SAS Metadata

## 2. Members:

Each developer should have a Unix ID and is set up with default authorization (DefaultAuth), and add them to the ZIP Developer group. We set each user up with just the most basic information and their email address. This enables us to send emails from SAS. The users are part of the ZIP Consumer group using ad-ent authorization. Figure 6 shows one of each type of user.

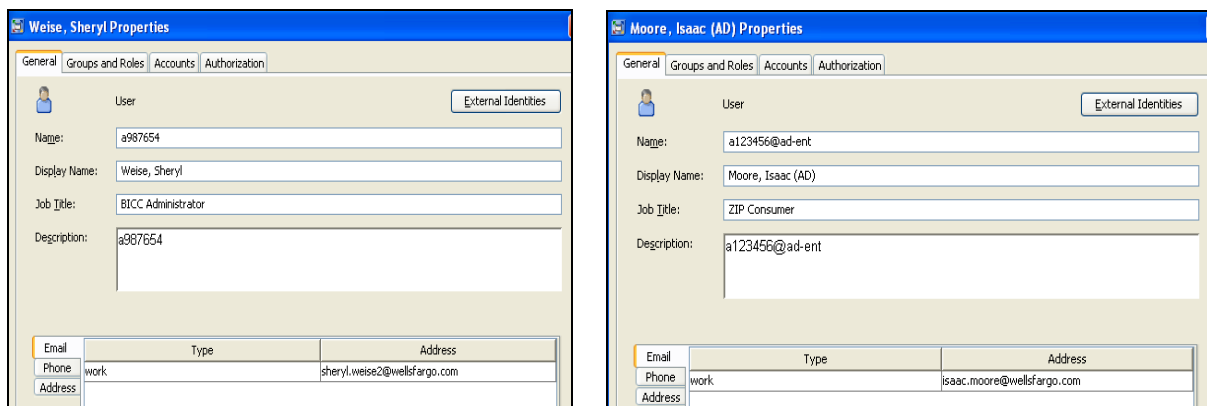


Figure 6 ZIP Developer and ZIP Consumer Users in SAS Metadata

We also add an internal account for the ZIP Developers to use to test their results the same as if they were a consumer (see Figure 7). This account exists only in the SAS metadata and uses the SAS System Services ID for Unix access.

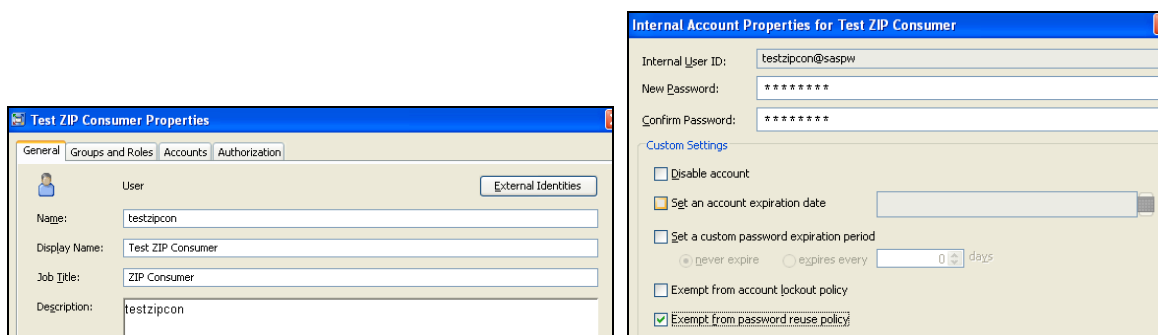


Figure 7 Internal Account to Test ZIP Consumer Capabilities

### 3. ACTs

Using the Authorization Manager, set up an ACT for each of the groups, adding the appropriate group and setting up permissions, as show in Figure 8.

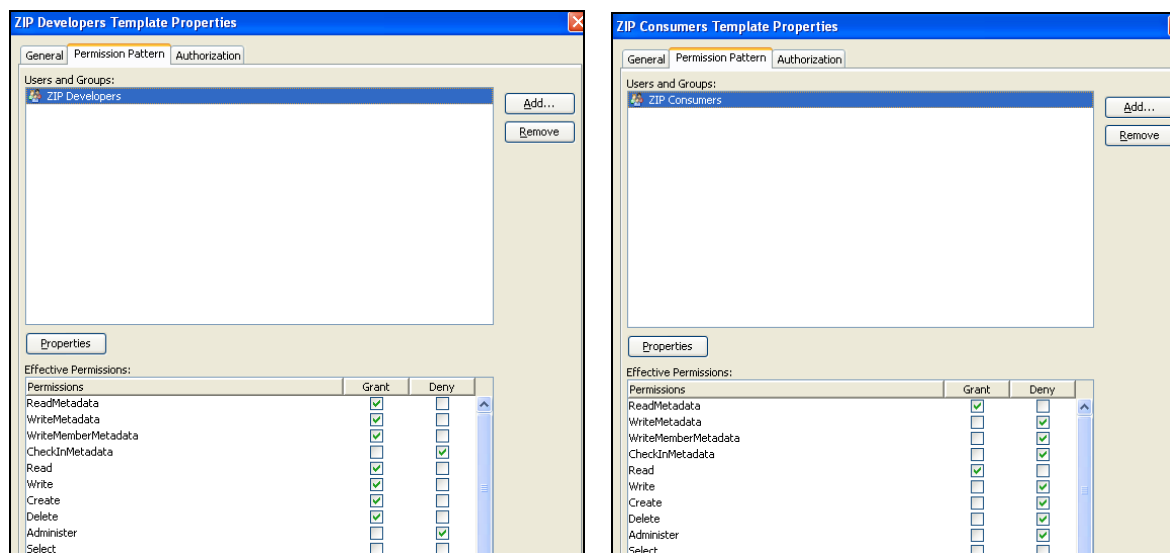


Figure 8 SAS Metadata ACTs

### 4. Folders

From the Folders tab, create a folder structure similar to Figure 9. SAS recommends you keep the structure as flat as possible and name each folder clearly,

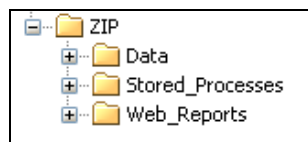


Figure 9 SAS Metadata Folders

Next, right-click on the ZIP folder, and go to the Authorization tab to add the Access Control Templates. Make sure to add the Private User Folder ACT to prevent other users from having access, and then add the 2 ZIP ACTs, as show in Figure 10.

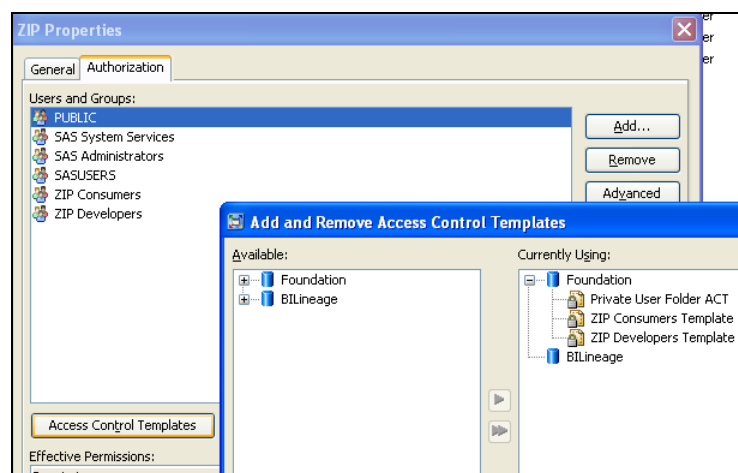


Figure 10 Adding SAS Access Control Templates

As you click on each group you'll see the permissions in green, showing they are inherited via the ACTs.

## 5. Data libraries

Using the Data Library Manager, set up a SAS metadata definition for the library on Unix. When you set the Location (see Figure 11) to the folder created in #4 it inherits the permissions and thus prevents anyone outside of the ZIP group from accessing.

The figure consists of two screenshots from the SAS Data Library Manager. The left screenshot shows the 'ZIP\_DATA Properties' dialog box with the 'General' tab selected. It displays the Name as 'ZIP\_DATA', the ID as 'A56N7ETV.AY00003G', and the Location as '/Wells\_Fargo/ZIP/Data'. The right screenshot shows the 'Options' tab for the same library. It shows the Libref as 'zipdata' and the Engine as 'BASE'. Under 'Path Specification', there is a list of 'Available items' including '/workspace/wbrcia/wbrciab/data', '/workspace/wbrcrm', '/workspace/wbrcrm/cubes', '/workspace/wbrcrm/data', and '/workspace/wrsdist'. The item '/workspace/poc2/zip/data' is selected and moved to the 'Selected items' list.

Figure 11 ZIP Data Library Definition

## STEP 4: SAS CLIENT TOOLS

This paper assumes the SAS web tier applications with assisting components (eg: Java, servlet container, Web Dav component, etc.) are already installed. Some companies use packages to push out the software, while others use a SAS depot for the users to pull their own software. We recommend packages so that every install is consistent and contains all the necessary software. The client software for this paper includes:

- SAS Enterprise Guide 4.3
- SAS Information Map Studio 4.31
- SAS Add-in for Microsoft Office 4.3
- SAS Management Console 9.2 (for SAS Support team not user teams)

Once the developers have the software installed, you provide instructions to configure the Unix connection. The users must be set up in the SAS metadata first (see Step 3). I created a basic SAS Enterprise Tutorial for experienced PC SAS Developers. For people new to development I recommend they take the beginning SAS programming class which is now taught using SAS Enterprise Guide. I also provide a SAS Developer Reference Guide with recommended training, links to helpful internet sites, etc.

## STEP 5: SAS EBI WORKSHOP INCLUDING SAS ENTERPRISE GUIDE

Set up a 3 hour session with your developers to teach them how to create 2 basic BI reports, following the same 2 paths as you showed in the Live Demo. I have a 'Workshop' area set up but you can just use their POC area. This is the agenda I follow (this assumes you have SAS EBI content development knowledge):

1. Review folder/server views, log arrows, hover and automatic statement, etc.
2. Write a new program to create the data set. A sample is in Figure 12.

```
libname sh '/opt/sas92/SASFoundation/9.2/sasHELP' ;
libname zip '/workspace/poc2/zip/data' ;

PROC SQL;
CREATE TABLE zip.zipcd AS
SELECT t1.ZIP,
       t1.CITY,
       t1.COUNTYNM,
       1 as Total
FROM SH.ZIPCODE t1
WHERE t1.STATECODE = 'VT'
ORDER BY t1.COUNTYNM, t1.ZIP;
QUIT;
```

Figure 12 Sample SAS Code

I have them put their initials on the end of the data set name so we're not all creating the same one.

3. Register the data sets to the SAS Metadata using the Tools/Update Library Metadata. Have your users watch as only one person needs to do the tasks shown in Figure 13.

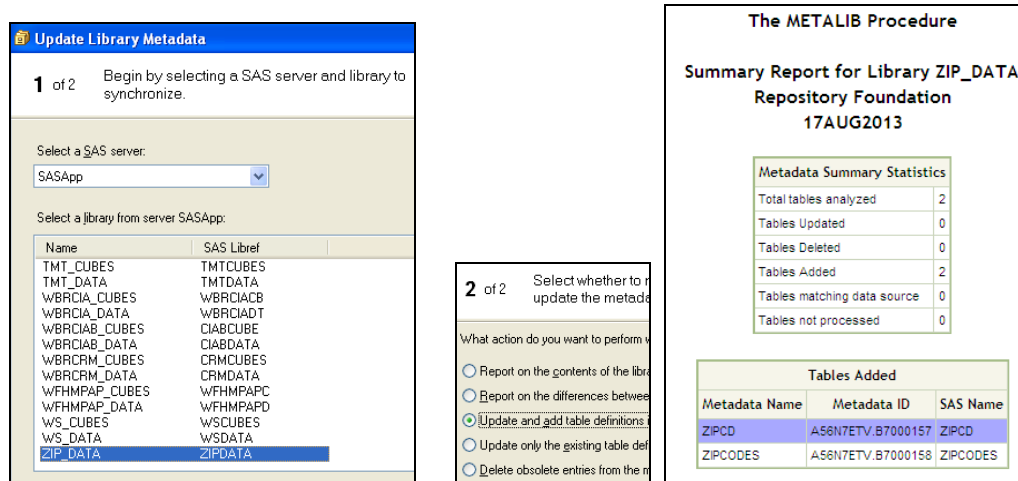


Figure 13 Registering SAS Data

4. Create an information map from that data set. Point out how the icons change as the variables become categories and measures. Change the zip variable to a category, change some names to better labels, add a format to the Total variable, etc. Figure 14 shows some of these changes.

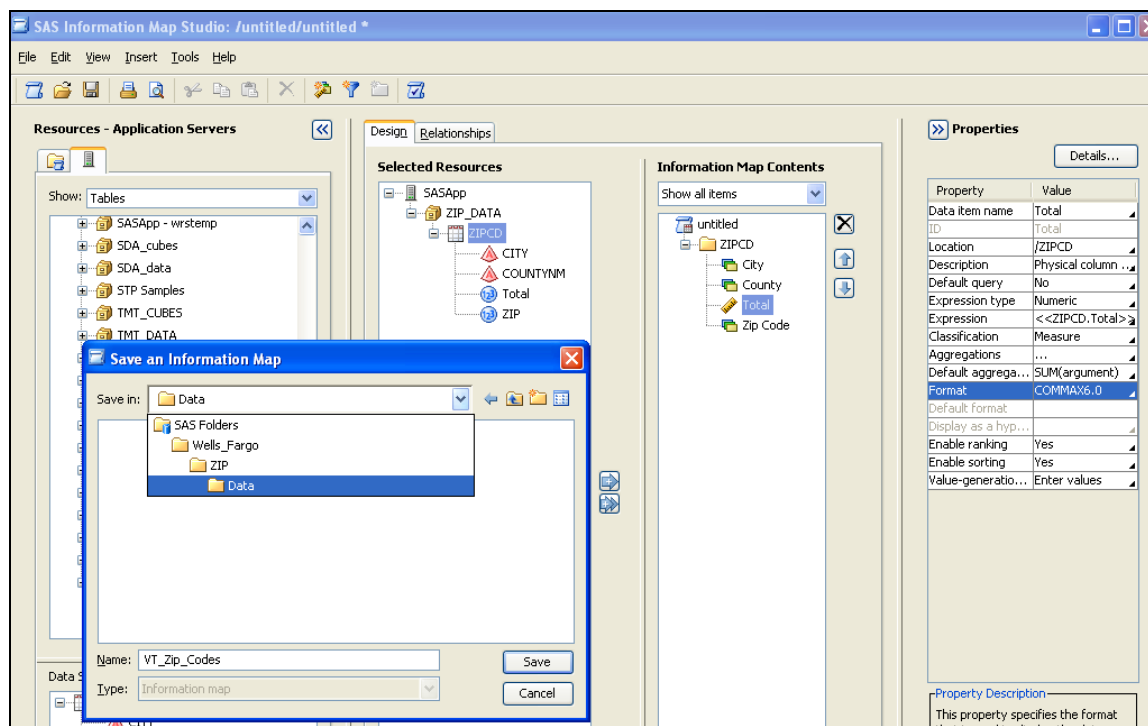


Figure 14 Creating a SAS Information Map

Note: I like to put the SAS Information Maps in the same SAS metadata folder as the data, and I name them to reflect their contents. Remember – do as much formatting when building the data set and less in the map.

5. Create a web report, using the map from #4, consisting of a graph and list. I like to make the graph a horizontal bar chart by county, and the list of counties/cities/zip codes.
6. Create a SAS Information Delivery Portal page with 2 columns. Add a SAS Report Portlet and add the report from #5 to it.

7. Using SAS Enterprise Guide, recreate the same reports as #5 but try adding a few more bells and whistles. Turn this report into a stored process, adding the same style sheet used in #5.
8. Add a SAS Stored Process Portlet to the SAS Information Delivery Portal, and add the report from #7. One version looks like Figure 15.

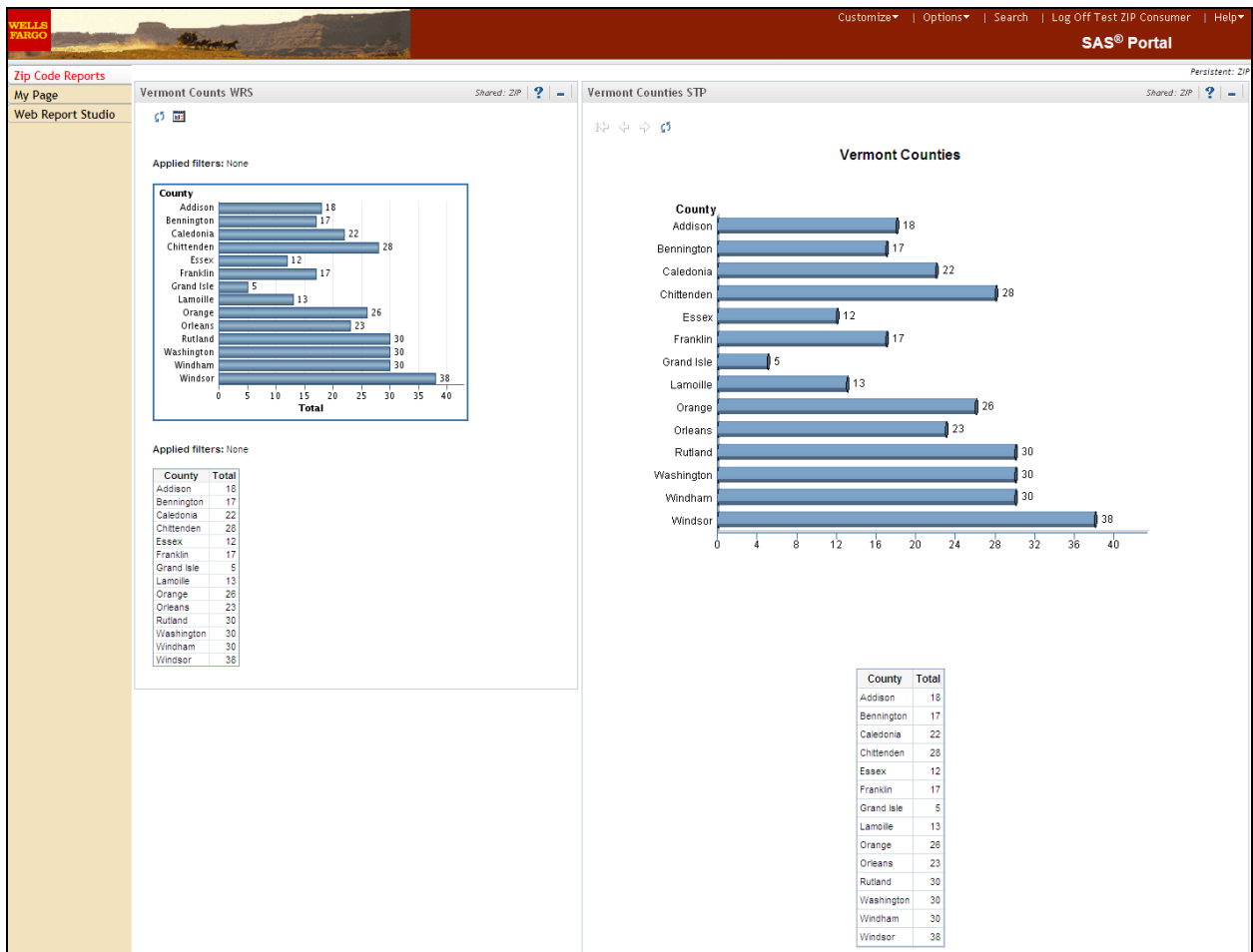


Figure 15 SAS Information Delivery Portal

9. Make sure your page and portlets are all created as Persistent and shared with only the ZIP group. They will have to create their versions in their personal folders, since they are not Content Administrators.
10. If they are interested in SAS Add-in for Microsoft Office, have them open up MS Excel and pull in the results of this workshop (data, report, and stored process).
11. Go over additional features of SAS Enterprise Guide, especially if their prior experience is with only PC SAS.
12. After getting permission, show them other SAS EBI projects that have been created at your company.

## STEP 6: DEVELOP A PROJECT

I've found the best way is for them to identify a repetitive SAS process they already perform. For example, a weekly process they run to create some specific output which they then email to a group of users. Help them map out their design: input, output, processing, prompts, delivery tool, etc. Go over their SAS folders and Unix filesystems. Some examples I have helped with:

- A chained stored process for querying servers. The first screen provided drop-down menus, date selection (calendars), entry boxes, and radio buttons for output type, to query against 60K plus servers. The result provided the output in the format requested, or a report stating no results were found matching the selection criteria.
- A Web Report built pulling data from 3 different database nightly, and summarizing to create 4 different graphs.



- A marketing map of products by state and county, along with the detailed data, accessible from Microsoft products only.

I recommend setting up weekly meetings to continue providing assistance, until they complete their proof of concept. Be available when they are ready to present to management, in case additional questions arise.

## **STEP 7: FOLLOW UP**

I recommend setting up weekly meetings to continue providing assistance until they complete their proof of concept. Be available when they are ready to present to their management, in case additional questions arise.

After the presentation conduct a follow-up meeting with the team, both consumers and developers. Gather feedback. Would they like an additional POC project? Are they ready to do a full-blown project? Even if they decide not to proceed at this time, follow-up with them in 3 months and see if there is any additional assistance you can provide.

## **CONCLUSION**

Remember to be flexible. No two projects and no two user groups are alike.

## **RECOMMENDED READING**

- SAS® EBI: What is it, what will it do for me and does it really work?  
- presented at SESUG 2009 by Frederick Pratter, PH.D.

## **CONTACT INFORMATION**

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

Sheryl Weise  
Wells Fargo Bank  
Charlotte, NC  
Sheryl.Weise2@wellsfargo.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.

*The views and opinions expressed here are those of the author and do not necessarily reflect the views and opinions of Wells Fargo Bank. Wells Fargo Bank is not, by means of this article, providing technical, business, or other professional advice or services and is not endorsing any of the software, techniques, approaches, or solutions presented herein. This article is not a substitute for professional advice or services and should not be used as a basis for decisions that could impact your business.*