Paper 006-2009

SAS® EBI: What is it, what will it do for me and does it really work?

Frederick Pratter, Ph.D. Associate Professor of Computer Science/Multimedia Studies,
Eastern Oregon University, La Grande OR

Abstract

Increasingly, there are two kinds of SAS programmers in the world: those who know EBI and those who do not. Although the latter are still better off than COBOL programmers, an exclusive dependence on SAS DATA step programming techniques is rapidly becoming inadequate for today's Web-enabled world. This paper will introduce the SAS Business Intelligence Platform, along with some of the most important EBI products, including Web-based applications like Web Report Studio and the Information Delivery Portal, and client-side side utilities such as Information Map Studio and the SAS Management Console. The focus will be on what these products do and why you should know about them. Examples of each product will be presented, along with some empirical performance metrics indicating that yes, indeed, they do work. The intended audience includes both SAS programmers and managers who might be considering an investment in the EBI suite.

Introduction

Traditional DATA and PROC step programming has served us well for over 30 years. Anyone who is paying attention however will have noticed that SAS has introduced a great many products that build on the older technologies but which provide a whole new level of functionality. The documentation for these products consists of literally dozens of reference manuals, user guides, white papers and road maps, not to mention the online help. It would be impossible in one brief presentation even to list all these let alone describe them and provide examples. This paper therefore will focus on just one product, SAS® Enterprise BI Server, often referred to as EBI or Enterprise Business Intelligence (http://support.sas.com/documentation/onlinedoc/entbiserver/). Furthermore, the emphasis will be on one EBI components, SAS Information Delivery Portal 2.0, a Web application that provides an interface to enterprise information. In addition, there are a number of other SAS products that are required to use the portal effectively, and these will be described in turn. Note that these are available in various bundles and of course the best resource for identifying what might be needed for your organization is your SAS sales representative.

EBI is an extremely complex suite of applications. Installing these components is not for the faint of heart. It is strongly recommended that you not try it yourself, but instead get a SAS representative to set up the servers and perform the installation. This paper assumes that you have access to an EBI installation and that it has been properly configured and patched at your site.

The easiest way to think about EBI, in this context is that there are (a) Web application components that run on the mid-tier, along with (b) separate desktop client applications. The following table lists several of the most useful:

Mid-tier	Client
Information Delivery Portal 2.0	Management Console 9.1
Web OLAP Viewer (Visual Data Explorer)	OLAP Cube Studio
Web Report Studio 3.1	Information Map Studio 3.1
Stored Process Web Application	Enterprise Guide 4.1

Table 1. EBI Components

Not all of these are technically part of the Enterprise BI Server suite but they all work together, and in theory should be enough to get you started.

The “mid-tier” or Web applications are JavaServer Page programs that run in a servlet container such as Tomcat 4.1, IBM Websphere or BEA WebLogic. They also require a Web DAV component. You can configure this yourself but SAS recommends the Xythos document manager, and it is a lot easier to go with the default. You will also need a database management system to store the DAV items you create; SAS supplies the PostgreSQL open source database but you can use SQL Server, Oracle or most other database products if you wish. EBI also requires the Open Metadata Architecture (OMA), including the Metadata server, the Object Spawner and the OLAP server.

The client applications are also all Java programs (except for SAS Enterprise Guide) that run on the desktop; these require the Java Runtime Environment (JRE). Note that the SAS supported versions currently are deprecated and are definitely not the most recent ones. This may change when 9.2 is released early next year, but for the time being you must use the SAS recommended versions of Tomcat 4.1 and the JRE 1.4.1.

Logically enough, the gateway to information delivery with EBI is the *SAS Information Delivery Portal* (http://support.sas.com/rnd/web/portal/ doc2/tour/ tour_overview.html). This is a Java Web application that provides a customizable interface to a variety of content including Information Maps, Stored Processes, publication channels and packages, Web reports, text documents, syndication channels, and links to URLs. In other words, pretty much anything that can be displayed on a Web page can be supported in the portal.

The Web OLAP Viewer comes in two flavors, depending on the architecture. The *SAS OLAP Viewer for Java* is a Web application that provides a Web interface for viewing and exploring OLAP data (http://www.sas.com/technologies/ bi/ query_reporting/webolapviewer/). The *SAS Web OLAP Viewer for .NET* is an ASP.NET application that runs on Microsoft Internet Information Server. The two interfaces look the same; they just provide cross-platform versus proprietary access.

SAS Web Report Studio is a Java application for constructing SAS reports http://www.sas.com/technologies/bi/query_reporting/webreportstudio/). Extremely complex reports including tables and charts can be created by users with no Java programming skills, simply by using a standard point-and-click interface. Along with Web report Studio, SAS supplies the *SAS Web Report Viewer* to display the resulting reports in the Portal.

The *SAS Stored Process Web Application* is a Java program that can execute SAS stored processes and return the results to a Web browser. (http://support.sas.com/rnd/itech/doc9/dev_guide/stprocess/stpwebapp.html). The Stored Process Web Application is similar to the SAS/IntrNet Application Broker and has been described in previous papers as “SAS/IntrNet on steroids”. Strictly speaking it is part of the *SAS Web*

Infrastructure Kit, a component of *SAS Integration Technologies*, but it is used in the Information Delivery Portal for displaying stored process output.

SAS Management Console is a desktop client program used to set up data sources and users, handle authentication and authorization, and many other functions (http://support.sas.com/documentation/whatsnew/91x/mcugwhatsnew913.htm). As the documentation indicates, it can be used to manage the following resources:

- server definitions
- library definitions
- user definitions
- resource access controls
- metadata repositories
- SAS licenses
- job schedules
- XML maps

SAS Management Console is required for any BI Server installation.

SAS OLAP Cube Studio, as the name suggests, is used to create Online Analytical Processing (OLAP) cubes (http://support.sas.com/onlinedoc/913/getDoc/en/bidsag.hlp/a003137964.htm). It is just a GUI interface to PROC OLAP, and provides the same functionality. You do not have to have cubes to use the portal, but it definitely helps. A cube is essentially the output of PROC SUMMARY, with some additional features that allow for drill down. The a significant benefit to using pre-summarized data is that performance is greatly enhanced over PROC SQL on row-level data.

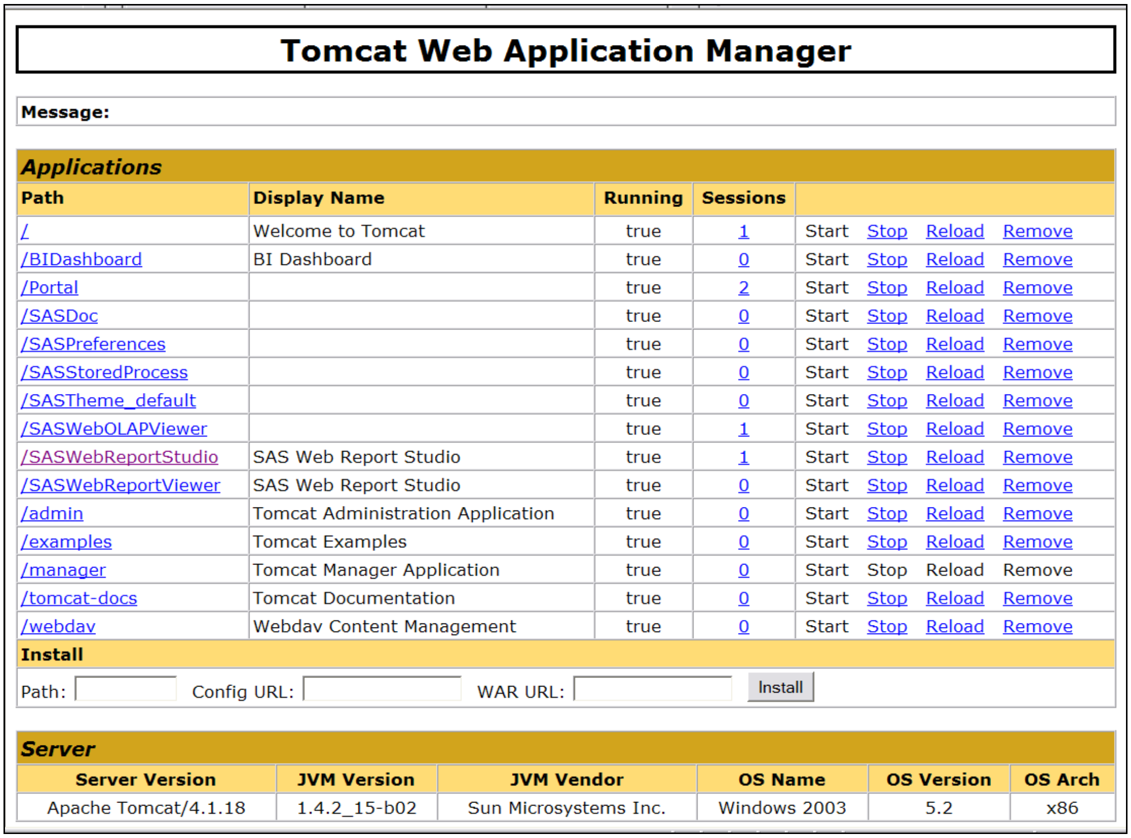
SAS Information Map Studio is GUI interface for creating information maps from cubes or underlying tables (http://support.sas.com/documentation/ onlinedoc/ims/). If the data source is SAS tables (as opposed to cubes) you can specify joins. The idea is that complex views of the data can be constructed by business users with no knowledge of SQL. The Information Map Studio product parallels PROC INFOMAPS in many respects but the two are not identical; the GUI product has much more functionality.

Finally, *SAS Enterprise Guide* is a Microsoft Windows-only product that is an interactive data exploration and analysis tool (http://www.sas.com/technologies/ bi/query_reporting/guide/). While it is not a part or EBI, it is extremely useful for exploring cubes and building stored processes. The big advantage of EG over the old SAS Display Manager is that the latter only works on the local server, while EG has the capability to run jobs remotely, an essential capability for an EBI site.

Java Web Applications

The EBI Web applications run in a Java servlet container. This is similar to a Web server such as Apache or Microsoft Internet Information Services (IIS), except that it can support dynamic content such as the output from SAS programs. The servlet engine runs in the “mid-tier”, that is, between the client and the SAS application server where the Metadata Server and Object Spawner are running.

The most common servlet engine is Apache Tomcat from the Apache Software Foundation. This is an open source product, freely available via download from http://tomcat.apache.org/. The current version is Tomcat 6.0, but SAS only supports the earlier 4.1 release. Java Web containers are also available from Sun, IBM and BEA, as well as several other open source products.



Tomcat Web Application Manager

Message:

Applications

Path	Display Name	Running	Sessions	
/	Welcome to Tomcat	true	1	Start Stop Reload Remove
/BIDashboard	BI Dashboard	true	0	Start Stop Reload Remove
/Portal		true	2	Start Stop Reload Remove
/SASDoc		true	0	Start Stop Reload Remove
/SASPreferences		true	0	Start Stop Reload Remove
/SASStoredProcess		true	0	Start Stop Reload Remove
/SASTheme_default		true	0	Start Stop Reload Remove
/SASWebOLAPViewer		true	1	Start Stop Reload Remove
/SASWebReportStudio	SAS Web Report Studio	true	1	Start Stop Reload Remove
/SASWebReportViewer	SAS Web Report Studio	true	0	Start Stop Reload Remove
/admin	Tomcat Administration Application	true	0	Start Stop Reload Remove
/examples	Tomcat Examples	true	0	Start Stop Reload Remove
/manager	Tomcat Manager Application	true	0	Start Stop Reload Remove
/tomcat-docs	Tomcat Documentation	true	0	Start Stop Reload Remove
/webdav	Webdav Content Management	true	0	Start Stop Reload Remove

Install

Path: Config URL: WAR URL: Install

Server

Server Version	JVM Version	JVM Vendor	OS Name	OS Version	OS Arch
Apache Tomcat/4.1.18	1.4.2_15-b02	Sun Microsystems Inc.	Windows 2003	5.2	x86

Figure 1. Tomcat Web Application Manager

The Java Servlet architecture is based on the concept of Web Applications, which are packages that contain HTML, Java servlets and/or JavaServer pages and various configuration files. To manage these, Tomcat provides an application called, logically enough, the Tomcat Web Application Manager. The output of this application is shown in **Figure 1**. The URL for this page is http://<server:portnum>/manager/html/ where *server* is the name of the mid-tier server and *portnum* is the TCP port; for Tomcat this is usually 8080. The EBI applications shown include the Portal, the Stored Process Web Application, Web Report Studio and Web Report Viewer and the Web OLAP Viewer, along with BI Dashboard, Preferences, and the default Theme, which are covered in the SAS documentation but not discussed in this paper.

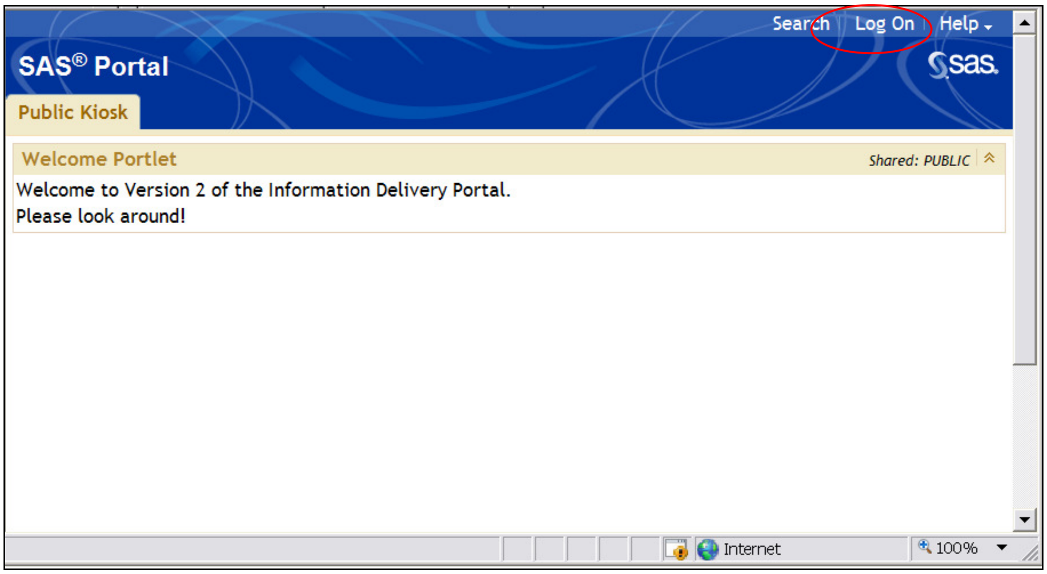
The remainder of this paper provides more detail and examples of each of these products.

SAS Information Delivery Portal

A Web portal is a single point of access to information from a variety of sources. Portals provide a single point of signon and unified access to and control of enterprise information systems. A familiar example of a Web portal is Yahoo. Portals are comprised of portlets, user interface components that provide fragments of markup code that are integrated into a portal page.

The main advantage of a portal is that it provides a customizable interface that can be tailored to a single specific user. The SAS Information Delivery Portal is an example of this concept; the product is designed to allow you to create an individualized Web site that contains just the content you have selected. It should be noted, however, that most SAS Information Delivery Portals sites are not used in this way. It very quickly became clear that this SAS product could be used to develop dynamic web sites to surface information to groups of users. Unfortunately, the software was not designed to make this very easy. The default assumption is that the pages will be viewed by only a single user. You are going to have to jump through some hoops to make it work as an enterprise information delivery system.

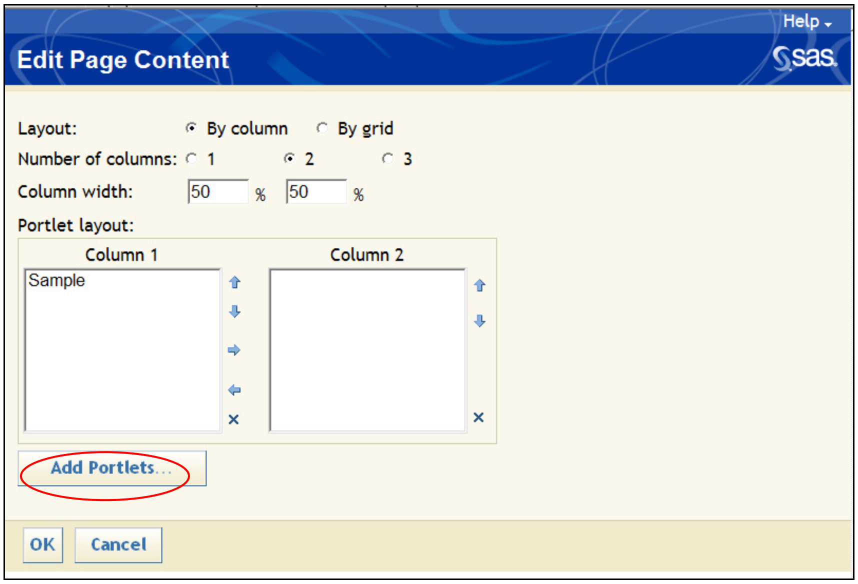
The default home page for SAS Information Delivery Portal is the Public Kiosk, as shown in **Figure 2**:



Welcome to Version 2 of the Information Delivery Portal.
Please look around!

Figure 2. SAS Information Delivery Portal

The URL for this page is http://<server:portnum>/Portal/main.do; alternately you can go straight to the login screen to access your content. Once logged in, you can add pages. Initially the new pages are empty. After adding a page, you need to edit the content to add portlets, as shown below:



Layout: ● By column ○ By grid
Number of columns: ○ 1 ● 2 ○ 3
Column width: 50 % 50 %
Portlet layout:

Column 1	Column 2
Sample	

Figure 3. Information Delivery Portal: Edit page content

The page referenced above contains a single portlet, called “Sample”. Clicking on “Add Portlets” allows you to add one or more portlets; some of these (shown in **Figure 3**) are supplied by SAS or if you are an experienced Java programmer you can create your own portlet types. At a minimum you need to give the portlet a name, which will be displayed on the portal page.

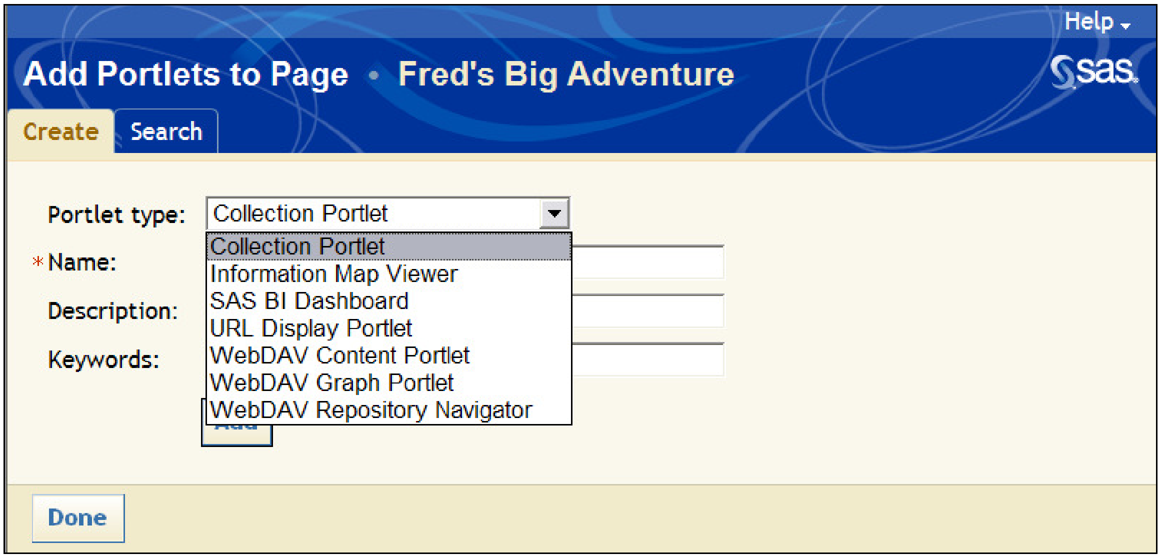


Figure 4. Add portlets to page

The resulting portlet is available only to the user who created it. As noted above, the assumption is that you are the only one who will be looking at the data. You can set various levels of security, down to the row level, so that different users can see different versions of the portal.

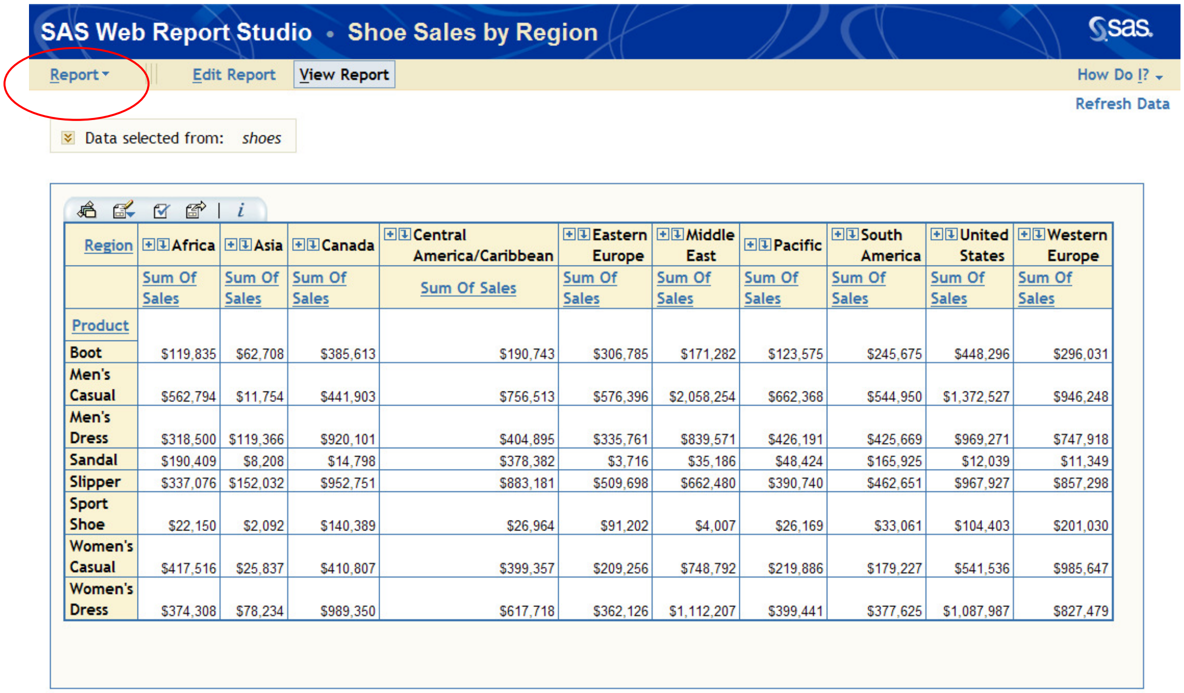
In order to share the page, you must create a user group in SAS Management Console and then make yourself the content administrator for the group. The details of this are beyond the scope of this presentation, but some help is available in the section “Configure a Group Content Administrator” in the *SAS® 9.1.3 Intelligence Platform Web Application Administration Guide, 2nd Ed.* (http://support.sas.com/ documentation/configuration/biwaag.pdf); careful study of this document is essential for anyone wanting to administer the BI Server suite. This document also includes sections on configuring Web Report Studio and the OLAP Web Viewer. In addition, you should be conversant with the *SAS® 9.1.3 Intelligence Platform Security Administration Guide, 2nd. Ed.* (http://support.sas.com/ documentation/configuration/bisecag.pdf), which as the title suggests covers security administration for your site.

Obviously, there is a lot more to the Portal than can be shown in this example. There is a good demo of the Portal interface available on the SAS technical support site, with different examples of the kinds of content that can be displayed. This presentation will look at two types of content that are extremely useful: Web Reports and the Visual Data Explorer.

SAS Web Report Studio

For many users, the most common content will consist of SAS reports. These are dynamic views of a SAS Information Map (see the section on *SAS Information Map Studio* below for more information about maps). Once an Information Map has been constructed, it can be displayed as a table or chart (or both) as a report. Web Reports are created using SAS Web Report Studio, which as noted is a “mid-tier” application that runs in a servlet container, that is, as a Web program. You do not need to have anything installed on your desktop except for Internet Explorer versions 6 or 7 (don't try it in Firefox or Opera!)

The link to the login page is http://<server:portnum>/SASWebReportStudio/ webreportstudio.jsp. Note that this is a JavaServer Page. Once logged in, you are given two choices, either to open an existing report or create a new one.



Data selected from: *shoes*

Region	Africa	Asia	Canada	Central America/Caribbean	Eastern Europe	Middle East	Pacific	South America	United States	Western Europe
	Sum Of Sales	Sum Of Sales	Sum Of Sales	Sum Of Sales	Sum Of Sales	Sum Of Sales	Sum Of Sales	Sum Of Sales	Sum Of Sales	Sum Of Sales
Product										
Boot	$119,835	$62,708	$385,613	$190,743	$306,785	$171,282	$123,575	$245,675	$448,296	$296,031
Men's Casual	$562,794	$11,754	$441,903	$756,513	$576,396	$2,058,254	$662,368	$544,950	$1,372,527	$946,248
Men's Dress	$318,500	$119,366	$920,101	$404,895	$335,761	$839,571	$426,191	$425,669	$969,271	$747,918
Sandal	$190,409	$8,208	$14,798	$378,382	$3,716	$35,186	$48,424	$165,925	$12,039	$11,349
Slipper	$337,076	$152,032	$952,751	$883,181	$509,698	$662,480	$390,740	$462,651	$967,927	$857,298
Sport Shoe	$22,150	$2,092	$140,389	$26,964	$91,202	$4,007	$26,169	$33,061	$104,403	$201,030
Women's Casual	$417,516	$25,837	$410,807	$399,357	$209,256	$748,792	$219,886	$179,227	$541,536	$985,647
Women's Dress	$374,308	$78,234	$989,350	$617,718	$362,126	$1,112,207	$399,441	$377,625	$1,087,987	$827,479

Figure 5. SAS Web Report Studio

Selecting `Report` from the page menu bar allows you either to create the report manually, use a wizard, or utilize an existing organization template. Note that you can also export the report to Excel, schedule it to be run at specific times, or distribute via any one of a number of subscription channels. The menu choice `Edit Report` allows you to manage the content of the report, as expected, but a limited amount of edit capability is also included in the `View Report` selection. For example, you can change the variables that are displayed with a simple point-and-click interface.

The dataset used for the examples is SASHELP.SHOES, supplied by default with the SAS installation disks. For this presentation, it was necessary first to create a cube, using OLAP Cube Studio, and then an Information Map of the cube with Information Map Studio. Although the Web OLAP Viewer can access either cubes or maps, Web Report Studio can only use information maps.

To create a new report, you need at a minimum to select an information map as the data source and then drag and drop report widgets onto the page. There are a variety of components available, including crosstabulation tables, color mapped tables, and line, bar and pie charts. More than one component can be added to the page, so for example you can have a table and the corresponding graph displayed side-by-side.

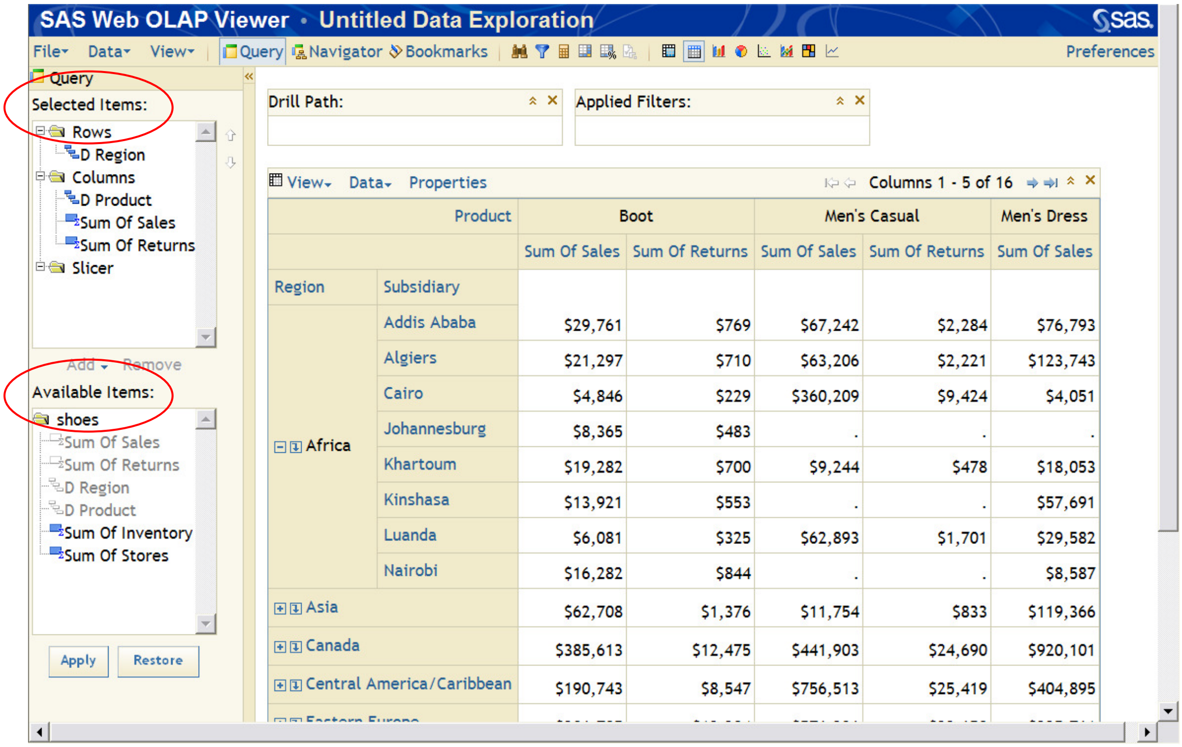
Web reports also allow data to be displayed in tabs, called *sections*. Each section can have its own data source, so that it is possible in a single report to include information from several different maps. The report illustrated only has a single section, but Web Report Studio provides a simple way of copying pages to new sections so that they can be customized. For example, it would be possible to have a tab for "Overall" values, and then by using filters ("where" clauses for you old folks), each section could display the same report for a subset of the data.

For more information about Web Report Studio, see the *Web Report Studio 3.1: User's Guide* (http://support.sas.com/documentation/onlinedoc/wrs/ug31.pdf).

SAS Web OLAP Viewer

As noted above, the OLAP Viewer can use several different types of data sources, either cubes, maps or Data Explorations (see below). The advantage of the OLAP Viewer, as shown in **Figure 6** is the ability to "slice and dice" data using the Visual Data Explorer.

Available items are all those in the source data; *selected* items are those chosen for display using the Add and Remove buttons. (Note that Web Report Viewer also allows you to select fields from the data source, using a completely different point and click interface). Two kinds of elements can be selected: dimensions, or class variables, and measures, the continuous variables. Each of these can be displayed as row elements, columns, or slicers, which must be measures. Using this simple interface, it is possible to display n-way tabulations and graphs of the underlying cube.



	Product	Boot		Men's Casual		Men's Dress
		Sum Of Sales	Sum Of Returns	Sum Of Sales	Sum Of Returns	Sum Of Sales
Region	Subsidiary					
Africa	Addis Ababa	$29,761	$769	$67,242	$2,284	$76,793
	Algiers	$21,297	$710	$63,206	$2,221	$123,743
	Cairo	$4,846	$229	$360,209	$9,424	$4,051
	Johannesburg	$8,365	$483	.	.	.
	Khartoum	$19,282	$700	$9,244	$478	$18,053
	Kinshasa	$13,921	$553	.	.	$57,691
	Luanda	$6,081	$325	$62,893	$1,701	$29,582
	Nairobi	$16,282	$844	.	.	$8,587
Asia		$62,708	$1,376	$11,754	$833	$119,366
Canada		$385,613	$12,475	$441,903	$24,690	$920,101
Central America/Caribbean		$190,743	$8,547	$756,513	$25,419	$404,895

Figure 6. SAS Web OLAP Viewer for Java

Once a particular collection of elements have been selected from a data source, it is possible to save the view as a Data Exploration. What is more, one Data Exploration can contain several Bookmarks, which are just saved views of a particular data source. A Data Exploration is saved by default in the user's home folder in the metadata. It is possible, however, to save a data exploration in a shared space, where some or all of the users of a site can access it.

SAS Web Stored Process Web Application

SAS Stored Processes are just “canned” SAS programs that are registered in the metadata. They can be viewed in various ways, including Enterprise Guide, the Stored Process Web Application, and the SAS Add-in for Microsoft Office. Any kind of ODS output can be displayed by a Stored Process. The example shown in **Figure 7** is one supplied with the installation disks.

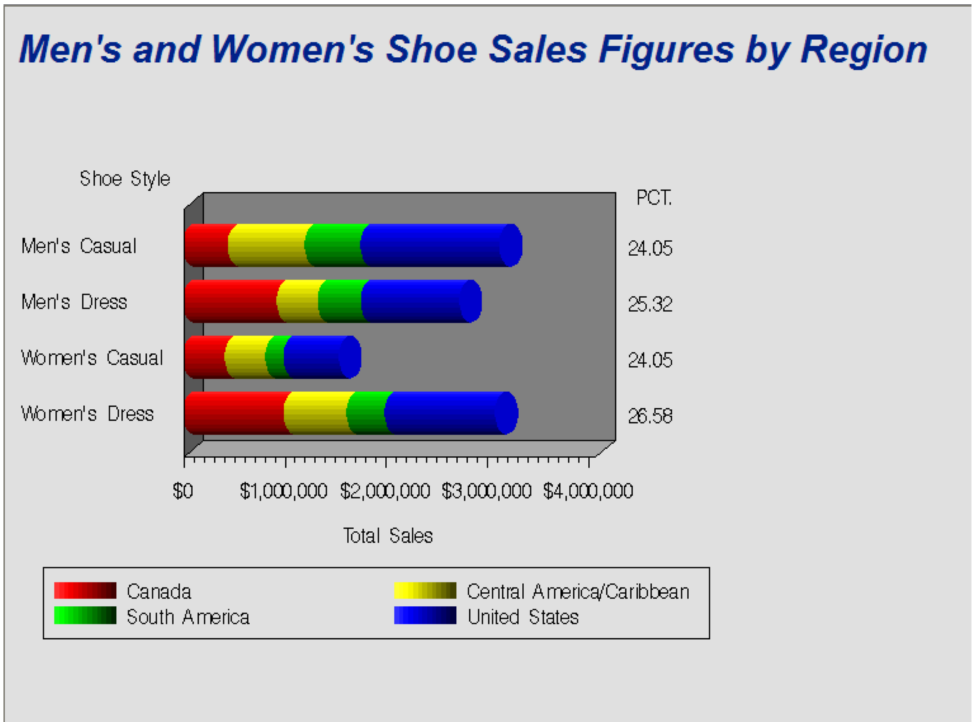


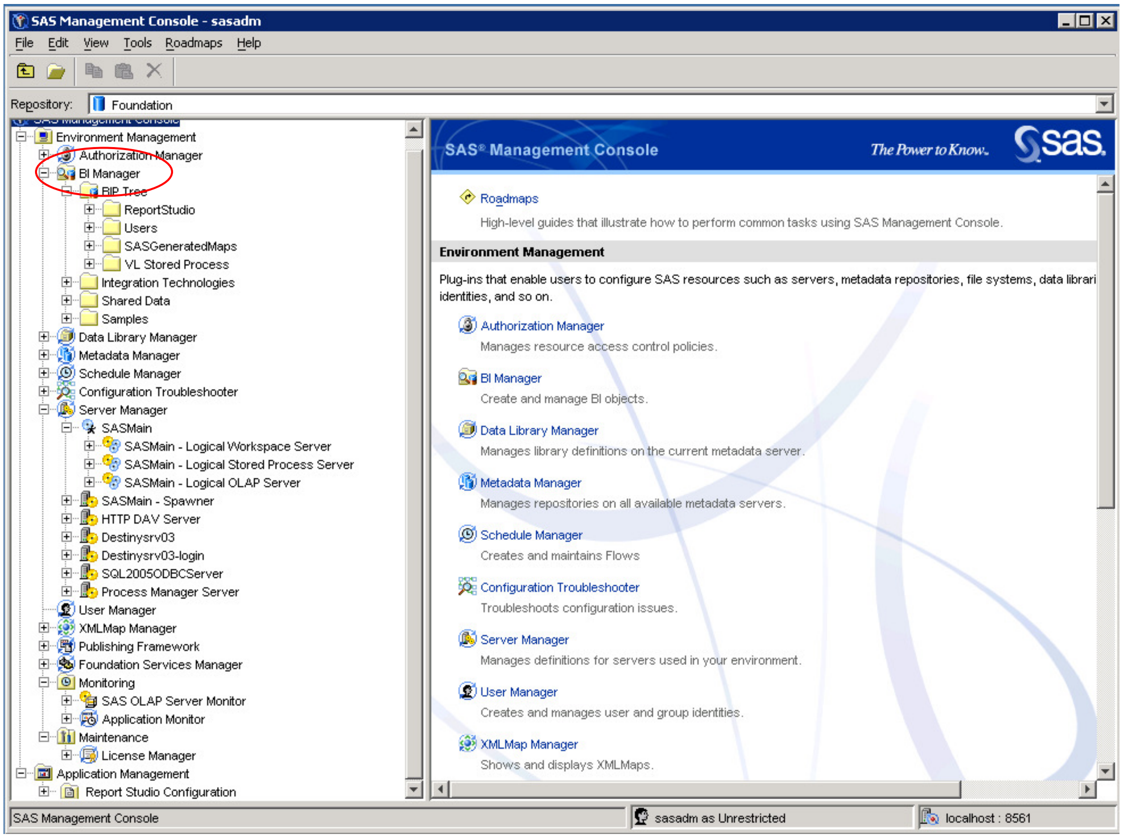
Figure 7. SAS Stored Process Web Application

You can add a stored process to a Collection Portlet for display in the Portal, which then uses the Stored Process Web Application to show the result. The advantage of stored processes is that you can supply run-time parameters. These are nothing more than macro variables; use the macro variable in the SAS code, save the stored process, and then supply values for the variables when you run it. Typically these are utilized for “where” statements to subset the data.

SAS Management Console

All four of the applications discussed so far have run in the servlet container. The remainder of the products discussed in this presentation are client applications. They can be installed on the server, but it is more usual for them to be run from the users’ desktop. Through the magic of TCP/IP, it does not matter where the server is located as long as the client can access it.

Foremost among these desktop applications is the SAS Management Console, illustrated in **Figure 8**. SAS Management Console has many uses, but for the EBI developer the most important of these are the BI Manager, the Data Library Manager, and the User Manager. These plug-ins are used to configure authorization and authentication for the site, and are something of a black art. Your best friend is the security and web administration documentation, and of course your SAS installation guru.



Environment Management
- Authorization Manager
- BI Manager
  - BIP Tree
    - ReportStudio
    - Users
    - SASGeneratedMaps
    - VL Stored Process
  - Integration Technologies
  - Shared Data
  - Samples
- Data Library Manager
- Metadata Manager
- Schedule Manager
- Configuration Troubleshooter
- Server Manager
  - SASMain
    - SASMain - Logical Workspace Server
    - SASMain - Logical Stored Process Server
    - SASMain - Logical OLAP Server
  - SASMain - Spawner
  - HTTP DAV Server
  - Destinysrv03
  - Destinysrv03-login
  - SQL2005ODBCServer
  - Process Manager Server
- User Manager
- XMLMap Manager
- Publishing Framework
- Foundation Services Manager
- Monitoring
  - SAS OLAP Server Monitor
  - Application Monitor
- Maintenance
  - License Manager

Application Management
- Report Studio Configuration

Roadmaps
High-level guides that illustrate how to perform common tasks using SAS Management Console.

Environment Management

Plug-ins that enable users to configure SAS resources such as servers, metadata repositories, file systems, data librari identities, and so on.

Authorization Manager
Manages resource access control policies.

BI Manager
Create and manage BI objects.

Data Library Manager
Manages library definitions on the current metadata server.

Metadata Manager
Manages repositories on all available metadata servers.

Schedule Manager
Creates and maintains Flows

Configuration Troubleshooter
Troubleshoots configuration issues.

Server Manager
Manages definitions for servers used in your environment.

User Manager
Creates and manages user and group identities.

XMLMap Manager
Shows and displays XMLMaps.

Figure 8. SAS Management Console

SAS OLAP Cube Studio

A second general purpose utility is OLAP Cube Studio, used to create OLAP cubes. As noted, cubes are not exclusive to EBI but they are a common way to surface information. The OLAP Cube Studio is a point-and-click GUI interface to construct cube definitions and add them to the metadata.

Cubes are n-way aggregation of data. Each variable in a cube is either a *dimension* or a *measure*. By analogy with PROC SUMMARY, dimensions are the class variables, used for the aggregation. Measures are the continuous variables. Most of the statistics available from PROC SUMMARY can be added to a cube, such as means, sums, standards errors, and so forth.

What makes cubes different is the concept of a *hierarchy*. You can specify a nested series of dimensions, such as region/state/zip or department/project, and SAS will allow drill-down capability to display the nested categories. The example cube shown in **Figure 9** provides this capability to the level of the individual observation, but it is possible to restrict drill down just to categories of data such as region and subsidiary.

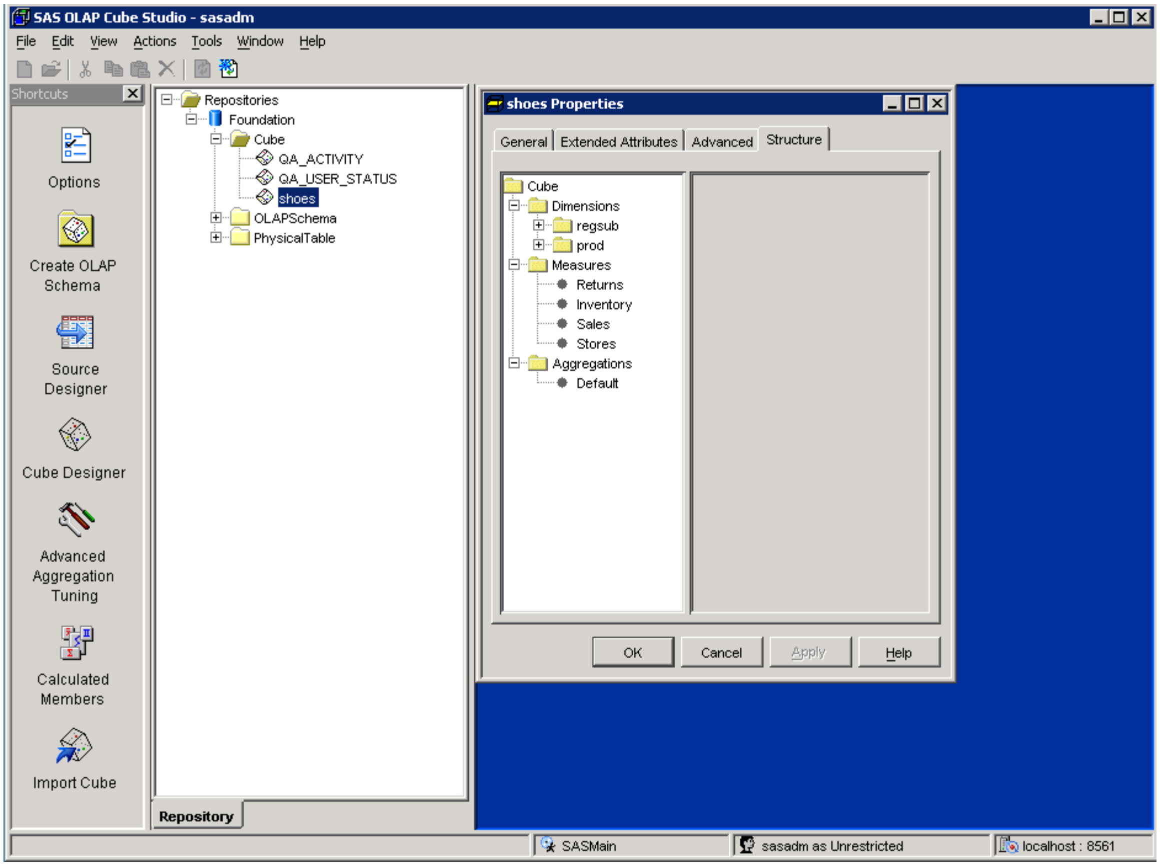


Figure 9. SAS OLAP Cube Studio

The OLAP Cube Studio can be used to write SAS batch code, which can optionally be saved to a folder on the user's system. There is no difference between running this code in batch or using the OLAP Cube Studio to run it. For the example, the program to construct the cube is as follows (reformatted slightly for clarity):

```
PROC OLAP DATA=sashelp.shoes
      DRILLTHROUGH_TABLE=sashelp.shoes
      CUBE=shoes PATH="c:/data/samples"
      DESCRIPTION="Shoe sales by region";

      METASVR host="<host-name>" port=8561 protocol=bridge
            userid="<sas-user>" pw="<password>"
            repository="Foundation"
            olap_schema="SASMain - OLAP Schema";

      DIMENSION reg_sub
            HIERARCHIES=(reg_sub )
            CAPTION='Region/subsidiary'
            DESC='Subsidiary by region'
            SORT_ORDER=ASCENDING;


      HIERARCHY reg_sub ALL_MEMBER='All reg_sub'
            LEVELS=( Region Subsidiary )
            CAPTION='reg_sub' DEFAULT;
```

```
        LEVEL Subsidiary
              CAPTION='Subsidiary'
              SORT_ORDER=ASCENDING;

        LEVEL Region
              CAPTION='Region'
              SORT_ORDER=ASCENDING;

         DIMENSION prod
              HIERARCHIES=(prod )
              CAPTION='Product'
              SORT_ORDER=ASCENDING ;

        HIERARCHY prod ALL_MEMBER='All prod'
                  LEVELS=( Product )
                  CAPTION='product' DEFAULT;

        LEVEL Product
              CAPTION='Product'
              SORT_ORDER=ASCENDING;

        MEASURE Sales
             STAT=SUM
             COLUMN=Sales
             CAPTION='Sum of Sales'
             FORMAT=DOLLAR12.
             DEFAULT
             ;
         MEASURE Returns
             STAT=SUM
             COLUMN=Returns
             CAPTION='Sum of Returns'
             FORMAT=DOLLAR12.
             ;
         MEASURE Inventory
             STAT=SUM
             COLUMN=Inventory
             CAPTION='Sum of Inventory'
             FORMAT=DOLLAR12.
             ;

        MEASURE Stores
             STAT=SUM
             COLUMN=Stores
             CAPTION='Sum of Stores'
             FORMAT=12.
             ;
        AGGREGATION Region Subsidiary Product/ NAME='Default';

   RUN;
```

The code illustrates the basic structure of the cube, and how it is constructed via PROC OLAP. The initial statement specifies the name of the cube, the location of the source (which MUST be registered in the metadata), where the cube is to be stored and a description. The user is allowed to drill down to the level of the rows of the table.

The METASVR statement specifies the connection to the metadata server that will be stored to store the cube.

There are two DIMENSION statements, each followed by a HIERARCHY statement and one or more LEVEL statements.

There are four measures, and for each the required statistic is the sum over the class variables.

Finally, the AGGREGATION statement, the equivalent of a PROC SUMMARY class statement, indicates the desired levels of aggregation.

Running this program from the command line on the server or from within Enterprise Guide will construct the cube, or you can use the OLAP Cube Stdio itself to submit the code.

SAS Information Map Studio

Once the cube has been constructed, if you want to use it in Web Report Studio you need to create an Information Map. Maps can be based on cubes, or optionally one or more joined tables. In the latter case, a relationships tab is used to specify the required joins. In **Figure 10** the map shown is simply the cube structure with no additional fields.

It is possible to use Information Map Studio to create new *expressions*, that is, new variables not present in the input data. These must be aggregations however. There is no way in a map to copy a source field. If the map is not an aggregation, you have to fool it by specifying a sum statistic, where of course there is only one observation in each category.

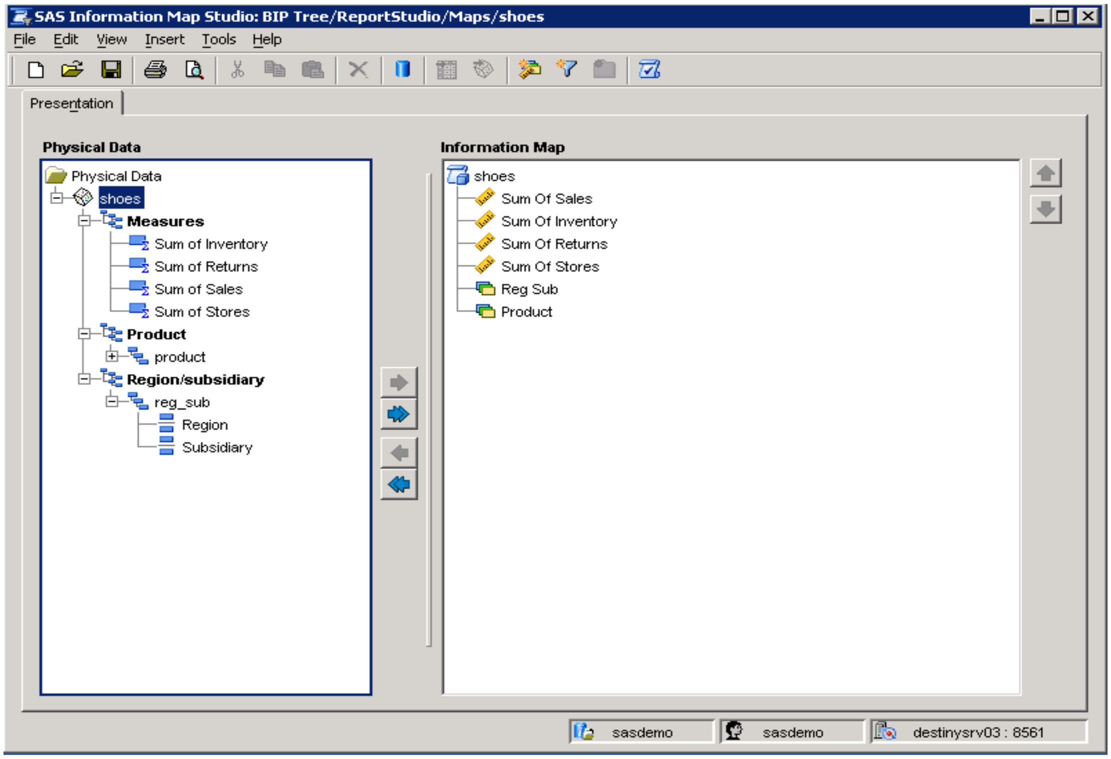


Figure 10. SAS Information Map Studio

As noted above, the idea of EBI is that a business analyst can construct reports using a GUI, without being familiar with the details of SAS batch commands or SQL statements. In fact, this is not how it is usually done. Most SAS programmers will find it easier to specify the structure of the maps and cubes using code, rather than the GUI. Unfortunately, unlike the OLAP Cube Studio, Information Map Studio provides functionality that is not available in a batch program.

For the example shown, it is possible to construct the map using something like the following code.

```
%MACRO mapper(cube=,mapname=);

      PROC INFOMAPS
           METAUSER="&userid"
           METAPASS="&pw"
           METAPORT=8561
           METASERVER="&host"
           METAREPOSITORY=Foundation
           MAPPATH="&infomap_path" ;

           %* if map exists, delete it;
            DELETE INFOMAP "&mapname";

           %* create new map from cube;
            OPEN INFOMAP "&mapname";

           %* add all of the variables in the cube to the map;
            INSERT DATASOURCE
                SASSERVER="SASMain"
                CUBE="&olap_schema"."&cube"
                _ALL_;

           %* don't forget to save the resulting map;
            SAVE;

      RUN;

%MEND mapper;
```

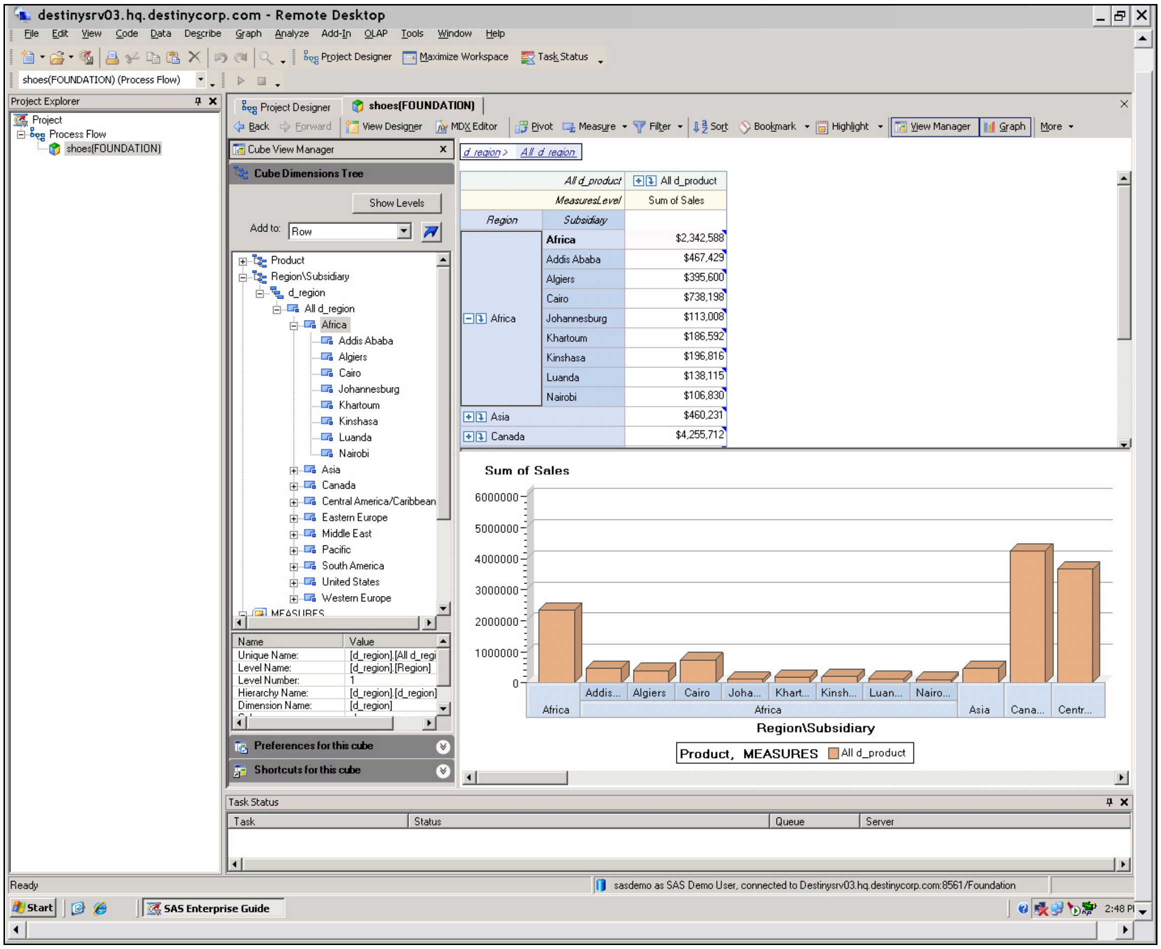
This program uses PROC INFOMAPS to construct the map. If the map exists, the program will fail, so the first step is to delete on if it is there. If it is not, this statement has no effect. The OPEN INFOMAP statement create a new map, while the INSERT DATASOURCE _ALL_ indicates that the map should have a one to one correspondence to the cube. Note that the source must be registered in the metadata. Finally it is important to save the map to the metadata.

The problem is that PROC INFOMAPS uses a different MDX engine than the Information Map Studio, so there are some things that can be done in the latter that are not possible in batch. It usually only takes a moment to create a map, so you may as well use the GUI, not matter how dedicated you may be to SAS batch programming.

SAS Enterprise Guide

All of the applications described so far are written in Java and run on any SAS supported platform. Enterprise Guide, however, as is well known, is a MS Windows only product. Most people just assume EG is a replacement for the Display Manager, but in fact it is considerably more powerful, since it is not restricted to running jobs locally. If the remote server is registered with the metadata you can prepare all your batch code locally on the client and then run it on the server.

The other cool thing you can do with EG is to view cubes. **Figure 11** illustrates the sample shoes cube in EG, along with the capabilities of drill-down and slice and dice. Before surfacing the cube to the Web in the OLAP viewer or using a Web report, it is extremely useful to be able to look at the cube and verify that in fact it has been set up successfully.



		All d_product
	MeasuresLevel	Sum of Sales
Region	Subsidiary	
Africa	**Africa**	$2,342,588
	Addis Ababa	$467,429
	Algiers	$395,600
	Cairo	$738,198
	Johannesburg	$113,008
	Khartoum	$186,592
	Kinshasa	$196,816
	Luanda	$138,115
	Nairobi	$106,830
Asia		$460,231
Canada		$4,255,712

Name	Value
Unique Name:	[d_region].[All d_regi
Level Name:	[d_region].[Region]
Level Number:	1
Hierarchy Name:	[d_region].[d_region]
Dimension Name:	[d_region]

sasdemo as SAS Demo User, connected to Destinysrv03.hq.destinycorp.com:8561/Foundation

Figure 11. SAS Enterprise Guide

Metrics

The obvious question then is how well does all this stuff work? The following examples were run on a Dell PowerEdge server with dual Intel 2.33 GHz Xeon processors and 4 GB RAM. While this is a modestly powerful system, it is by no means a supercomputer. You should expect to get similar results in your environment.

A 6-way aggregation using PROC OLAP on 598,000 records took about 4 minutes. Once the cube was created, loading the data into the OLAP Web Viewer took only a few seconds. The Visual Data Explorer was able to display multiple views virtually instantaneously.

The largest cube tested was an 11-way aggregation where one of the dimensions had 188 levels and the others cardinality anywhere between 5 and 25 levels. The cube has about 14 million rows. The longest it took to load into the Web Viewer was about 15 seconds. Once loaded, the cube could be sliced into various n-way views in less than 5 seconds.

Although response time is important, the more crucial resource is development time. Over the course of a six-week project, it was possible for an relatively experienced developer (me) to create a dozen cubes and nearly 500 Web reports, using the tools available in the BI Server suite. The client was happy with the result, and I was very happy that the client was happy. I do not believe this experience was unique; with a little practice and a lot of study you too can achieve similar results.

Conclusion

So yes, it does work, most of the time. It would be disingenuous to say that it was easy however. The most important thing is to get the installation right, and in particular, to set up the security correctly. If you have problems, they are going to be about permissions. SAS Technical Support is pretty good about helping, but your responsibility is to be familiar with the available documentation. SAS BI Server is a large and complex application, and the relatively few developers with experience in these applications can stay quite busy, thank you. While traditional DATA and PROC step programming will accomplish much of what you need, the extended capabilities of EBI will allow your organization to move into the new millennium with state-of-the art Web reporting capabilities.

Acknowledgements

The examples for this paper were run at Destiny Corporation, Rocky Hill CT, and I owe a debt of gratitude to Dana Rafiee for supporting and encouraging my adventures in EBI development. As always, I need to thank the Division of Science, Mathematics and Technology at Eastern Oregon University for sending me to user group conferences and accepting my frequent distractions from teaching, in particular my chair Dr. Anna Cavinato and Dean Dr. Marilyn Levine.

References

The following is a partial list of some of the SAS sources consulted for this paper; all of the documents referenced can be downloaded in PDF form from the SAS support website.

- SAS® 9.1 Open Metadata Interface: Reference. Cary, NC: SAS Institute Inc. 2004.
- SAS® 9.1 Open Metadata User's Guide: Reference. Cary, NC: SAS Institute Inc. 2004.
- SAS® 9.1.3 Intelligence Platform: Application Server Administration Guide. Cary, NC: SAS Institute Inc. 2006.
- SAS® 9.1.3 Intelligence Platform: Data Administration Guide. Cary, NC: SAS Institute Inc. 2006.
- SAS® 9.1.3 Intelligence Platform: Desktop Application Administration Guide. Cary, NC: SAS Institute Inc. 2006.
- SAS® 9.1.3 Intelligence Platform: Security Administration Guide, Second Edition. Cary, NC: SAS Institute Inc. 2006.
- SAS® 9.1.3 Intelligence Platform: Web Application Administration Guide, Second Edition. Cary, NC: SAS Institute Inc. 2007.

- SAS® 9.1.3 OLAP Server: MDX Guide, Third Edition. Cary, NC: SAS Institute Inc. 2006.
- SAS® 9.1.3 OLAP Server: User's Guide, Second Edition. Cary, NC: SAS Institute Inc. 2006.
- SAS® 9.1.3 Open Metadata Interface: Reference, Second Edition. Cary, NC: SAS Institute Inc. 2005.
- SAS® BI Dashboard 3.1: User's Guide, Second Edition. Cary, NC: SAS Institute Inc. 2007.
- SAS® Guide to Applications Development, Second Edition. Cary, NC: SAS Institute Inc. 2004.
- SAS® Information Map Studio 3.1: Tips and Techniques. Cary, NC: SAS Institute Inc. 2006.
- SAS® Intelligence Platform: Overview, Second Edition. Cary, NC: SAS Institute Inc. 2006.
- SAS® Web Infrastructure Kit 1.0: Developer's Guide, Fifth Edition. Cary, NC: SAS Institute Inc. 2007.
- SAS® Web Report Studio 3.1: User's Guide. Cary, NC: SAS Institute Inc. 2006.
- Base SAS® Guide to Information Maps. Cary, NC: SAS Institute Inc. 2006.
- Getting Started with SAS® 9.1.3 Open Metadata Interface, Second Edition. Cary, NC: SAS Institute Inc. 2006.

In addition, there are numerous papers from SUGI and SGF in the last few years on the topic of the SAS Business Intelligence suite and BI Server in particular. See for example the "BI Forum/User Applications" section from SGF 2008 at http://www2.sas.com/proceedings/forum2008/TOC.html#biua.

Contact

Frederick Pratter
Computer Science/Multimedia Studies Program
Eastern Oregon University
One University Blvd
La Grande OR 97850
fpratter@eou.edu