

Web-Enabling EIS Applications: Take 2

Dave Horne, SAS
Don Henderson, SAS
Vince DelGobbo, SAS

Abstract

This paper provides an overview of the issues involved in moving applications from the SAS® desktop environment to a browser based Web environment.

SAS has a variety of tools that can be used to build new Web-based applications as well to assist customers migrate their existing applications to the Web. The paper describes which tools and approaches are best suited to particular types of SAS/EIS® and SAS/AF® software applications.

Issues to consider in determining the suitability of converting a desktop AF/EIS application to the Web are discussed in the context of criteria which can help the user to determine what tools and methodologies which are available from SAS are appropriate. NOTE: The Web has had a tremendous impact on the time involved in delivering new software technology. At the time this paper was written (early January 1999) it accurately reflected SAS's plans in this area. However, the authors expect that some of the material will be out-of-date by the time the paper is presented at the SUGI 24 Conference. The reader should check with his/her Software Sales representative or the following URL for the most up to date information on SAS/IntrNet™ software: <http://www.sas.com/web>.

Introduction

There are a number of issues to be evaluated when considering moving an existing application to the Web. We will first discuss the impact of the Web on application architecture and how that interacts with the architecture of existing applications to be re-deployed on the Web. This provides the context

necessary to evaluate whether and how an existing application can be moved to the Web. Next, we will briefly discuss the nature of SAS/EIS and SAS/AF software applications as they relate to the above issues.

This will be followed by a discussion of the tools that SAS plans to make available to assist in this conversion.

Impact of the Web on Application Architecture

Applications to be deployed via the Web need to recognize the underlying distributed nature of Web-based applications.

The native Web environment is a two-tier architecture: the Web server/web browser tier and the data/compute services tier. This separation means that many architectural assumptions made in the original development of applications are incompatible with the Web. Such issues can include:

- Many Web applications are stateless (i.e., each request for compute or data services are completely independent of one another). Thus applications that assume exclusive access to transient data are problematic (e.g., in the transaction based world of the Web, the concept of opening a data set for exclusive update access is not the same as in a desktop world).
- The event model in a Web application is very different from the event model of a desktop application. For example, if the client is a browser, the back-end compute engine has no access to key clicks until the browser has forwarded a request for processing to the compute server.
- A corollary to the above point is that even if some of the logic relating to the event model can be surfaced in the client (e.g., using Java, JavaScript, Dynamic HTML, etc.), the separation of the client and the server can be a complicating factor in moving existing applications to a Web environment.

The Architecture of SAS/EIS and SAS/AF Applications

SAS/EIS applications have an underlying architecture that makes it easier for them to be deployed in a Web based environment. Specifically, all the information and parameters necessary to perform the desired functions is encapsulated in SLIST catalog entries and those entries are stored in the SAS/EIS application database. When run from the SAS Desktop, SAS/EIS software accesses the information contained in those lists, and uses SAS FRAME entries to render the desired display. This implies that it is possible to develop alternative applications that will instead:

- generate Internet Content (e.g., HTML, including Java applets)
- *download* the relevant info to a browser based client (e.g., a Java or CGI program) which then can interact with the user.

There are a variety of types of functionality offered by the SAS/EIS objects and the techniques used to Web-deploy them will depend on that. For example, a SAS/EIS variance report must be delivered so that the data are current and live. But a SAS/EIS desktop object can be published, as long as the components that are data based are dynamic.

Moving SAS/AF software applications to the Web is much more difficult. SAS/AF software is a general application development facility that any user can use to build a GUI application. That generality is exactly what makes providing tools to deploy them to the Web problematic.

Unlike SAS/EIS applications where it is reasonably safe to assume the underlying architecture, the same is not true for SAS/AF applications. It is possible, even common, for SAS/AF applications to have the GUI (i.e. the FRAME entry) and the processing logic (i.e., the SCL) intermingled. For such applications, it might well be the case that the application needs to be completely redesigned and rearchitected in order for it to be deployed to the Web.

On the other hand, it is also possible to develop and deploy a SAS/AF application using Object Oriented Methods, specifically the Model-View

architecture. Such applications can be much more easily deployed to the Web because the separation of the GUI (the view) from the processing (the model) is inherent in the underlying architecture. In fact, the reason that SAS/EIS applications are more easily deployed to the Web is because SAS/EIS imposes a simple Model-View architecture on the developer.

SAS/IntrNet™ Software Tools

SAS/IntrNet Software contains a number of components that can be used to assist in deploying SAS/EIS and SAS/AF applications to the Web.

The Application Dispatcher

is a CGI gateway between a Web browser and SAS software. It enables users to build dynamic applications that give users access to the power of SAS software from their Web browsers. It provides the capability to run any SAS program that can be executed in batch mode. The user need only concern themselves with the details of the SAS program. All of the details relating to CGI, the communication of the parameter values from the HTML page, and returning the results to the users browser are handled by the Application Dispatcher framework. The Application Dispatcher allows the deployment of these programs to any number of users (whether they have SAS software installed or not).



Thus, Application Dispatcher programs can be used to access a SAS/EIS SLIST entry and can then generate the appropriate Internet content based on EIS entry type. Subject to the issues discussed above, such programs can also be used as analogs for the SCL code corresponding to SAS/AF applications, instead generating Internet content.

The **Web Publishing Tools** are a collection of macros that enable you to format your SAS data sets and procedure output into IntrNet content (e.g., HTML-formatted pages). Thus they can be used by the Application Dispatcher to create the visual displays the user will see in their browser.



The current Publishing Tools are:

- the HTML Output Formatter;
- the HTML Data Set Formatter;
- the HTML Tabulate Formatter; and
- the GraphApplet HTML Generator;

Additionally, a new publishing tool, the RangeViewApplet (CSF) HTML Generator will be available in Version 8, as well as the next maintenance releases of 6.12 and 6.09E. The **Output Delivery System, ODS**, available starting with Version 7 of the SAS System, will allow for direct creation of HTML files.

The MDDB Report Viewer component of SAS/IntrNet software allows you to Web-enable OLAP reports built with SAS/MDDB® server software. Full-featured multidimensional tables

can be displayed including interactive graphs. The production version of the MDDB Report Viewer will be available for Download on April 19, 1999. This version will include an experimental component to directly replay SAS/EIS applications on the web. By pointing directly to an EIS applicaiton database or metabase, you can run selected reports from your web browser.

The URL to invoke the SAS/EIS application database viewer:

```
http://yourhost/cgi-  
bin/broker.exe?_SERVICE=yourservice&_PR  
OGRAM=sashelp.webeis.app
```

The *&dataset* parameter specifies the name of the EIS application data base.



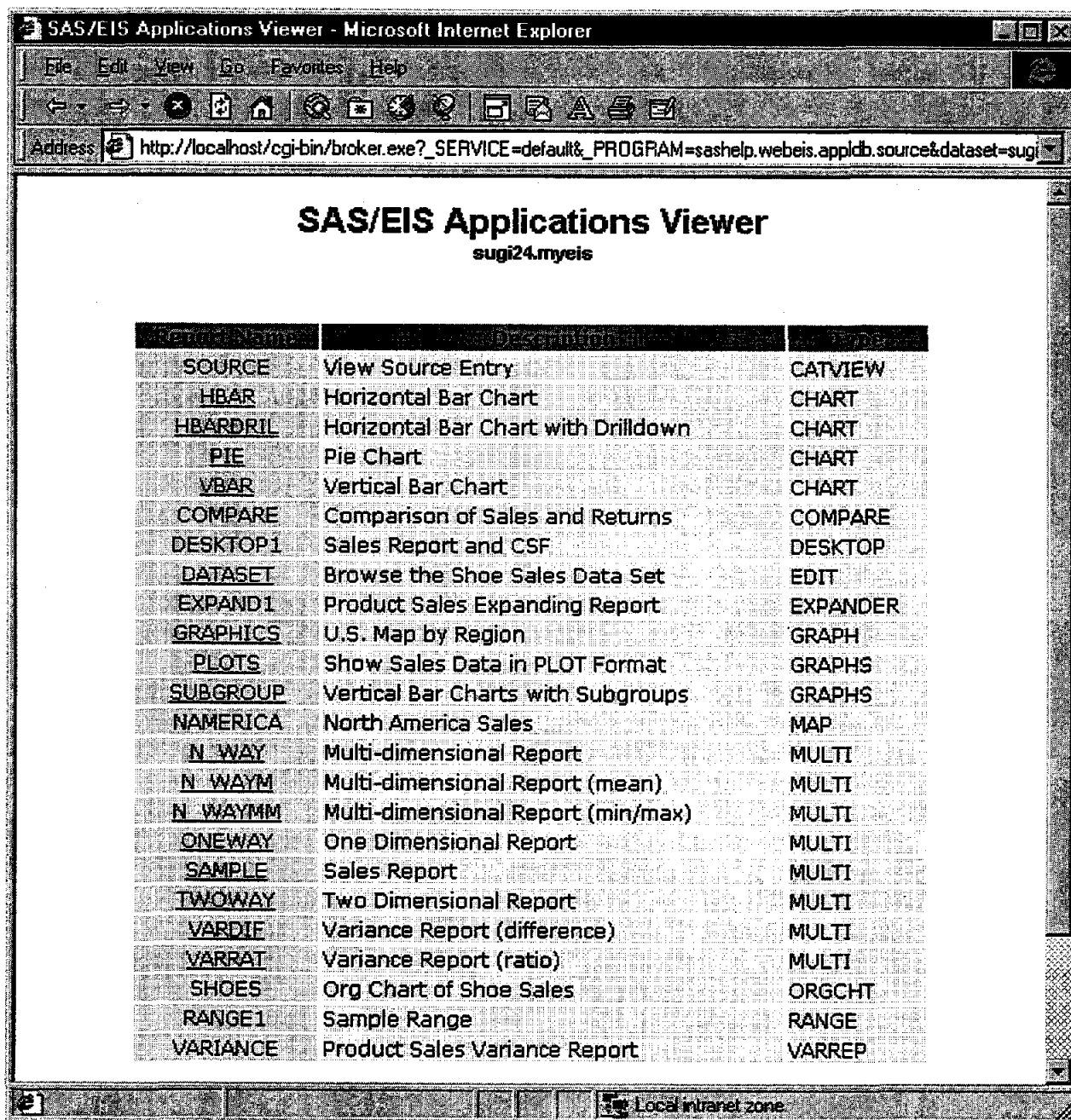


Figure 1: Web-generated SAS/EIS Application Database

Figure 1 contains a sample Web-generated SAS/EIS Application Database. The currently supported objects are hyperlinked. For example, selecting a Multidimensional report will replay that object using the MDDB Report Viewer

application (MRV). The MRV obtains the name of the MDDB, across and down dimensions, and other properties from the EIS object and displays it to the browser.

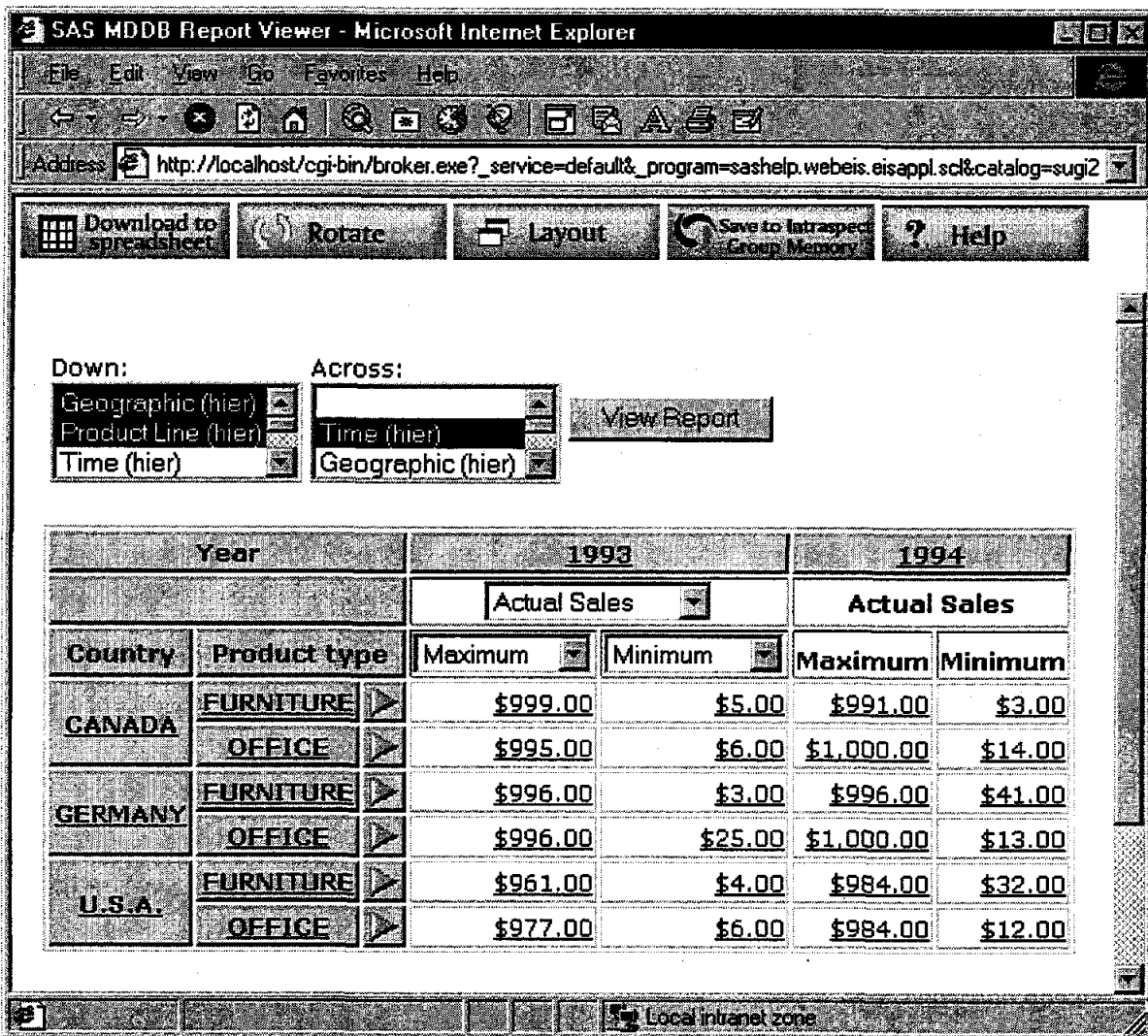


Figure 2: SAS/EIS Multidimensional report object Web generated .

Figure 2 contains a sample SAS/EIS multidimensional report object that uses an Application Dispatcher program to generate IntrNet content. It was merely necessary to identify the SAS/EIS application database and object. It then generated the appropriate content. Our plans are to support most of the reporting and graph objects.

Another experimental component of the MRV is the ability to view the registered data in an EIS Metabase. By pointing directly to an EIS metabase, you can display MDDB's or SAS data sets and views.

The URL to invoke the SAS/EIS Metabase viewer:

`http://yourhost/cgi-bin/broker.exe?_SERVICE=yourservice&_PROGRAM=sashelp.webeis.meta`

The `&dataset` parameter specifies the name of the EIS Metabase.

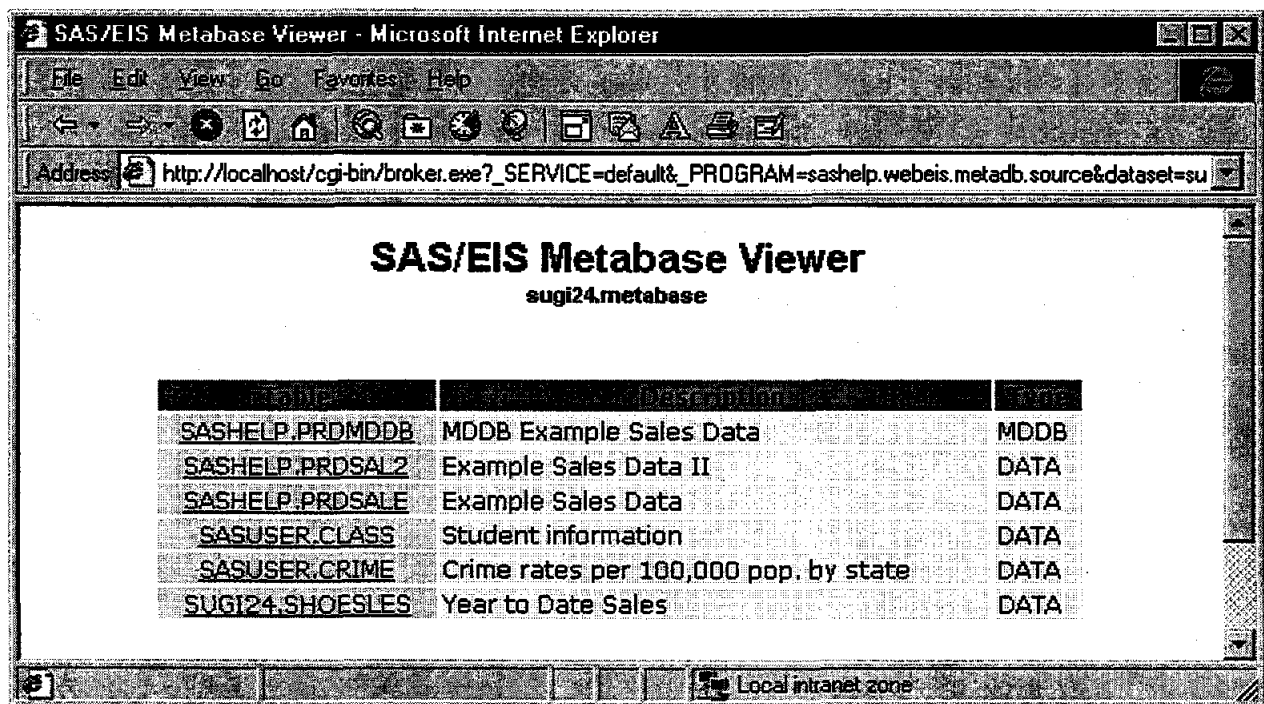
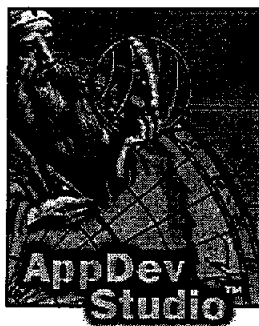


Figure 3: Web-generated SAS/EIS Metabase

Selecting an MDDB will invoke the MDDB Report Viewer application. Selecting a data set will invoke the Xplore sample application to view these objects on the Web.

AppDev Studio™ Software Tools

The webEIS™ software component of AppDev Studio™ is a Java application that makes it easy to create interactive, EIS-style documents containing charts and multidimensional tables that can be published on



the Web. A webEIS document is published on the Web as a Java applet that communicates with a remote SAS server to retrieve live data. Additionally, a webEIS document is a JavaBean, which makes it easy to create dynamic reports that can be incorporated into larger, more complex webAF applications. SAS customers looking for a flexible, powerful, easy-to-use Java-based OLAP tool will find the webEIS software component of AppDev Studio to be a key technology.

Using the webEIS development facility, it is possible to build an applet that merely points to the SAS/EIS object and will render the desired display in the applet. This is similar to what was described above for the MDDB Report Viewer.

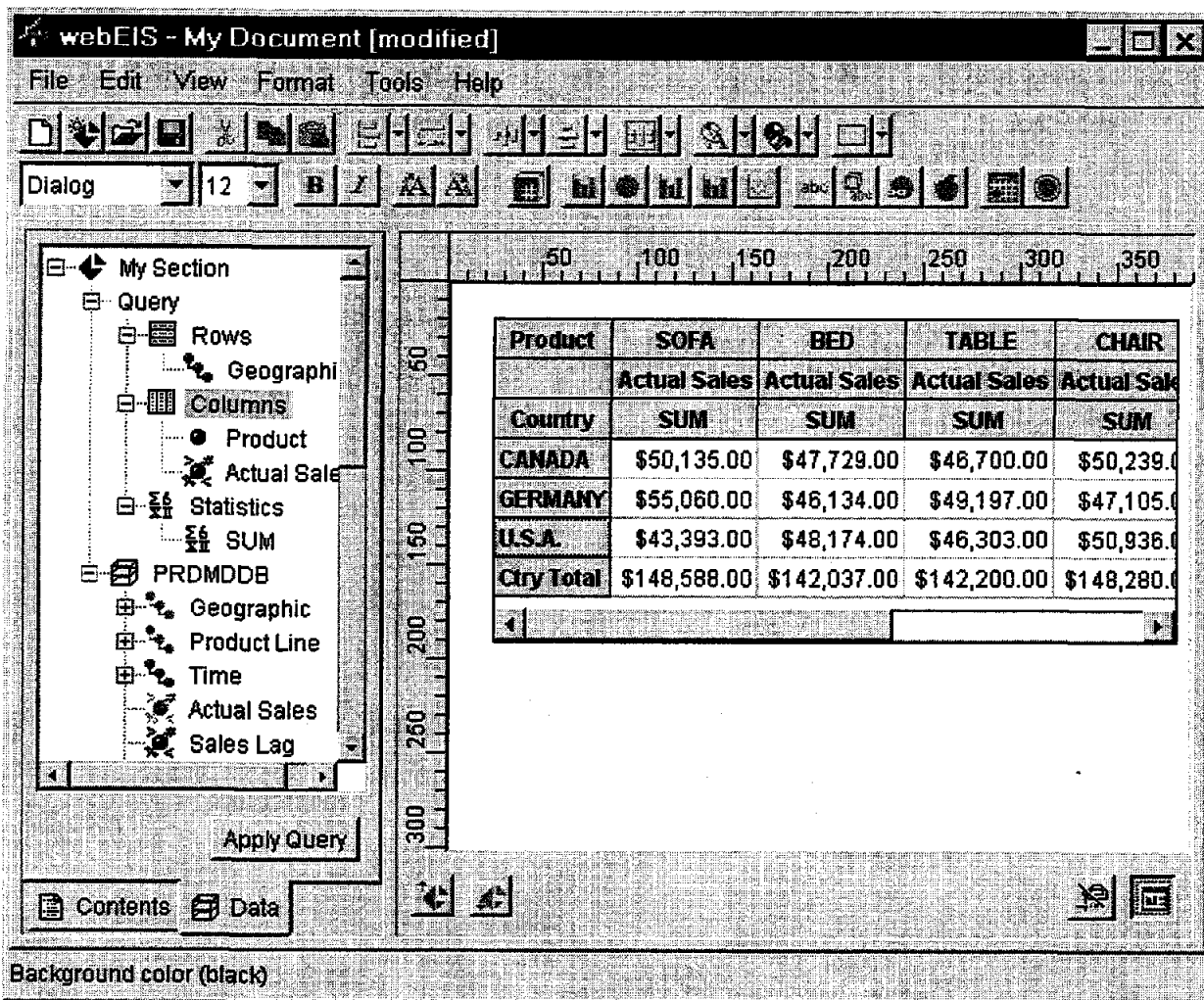


Figure 4: Sample webEIS application

The Available Options

The availability of these tools provides a variety of options for the conversion of existing SAS/EIS and SAS/AF applications.

As described above, some SAS/EIS objects can be dynamically executed in a Web environment. This includes graphical objects such as 3D Business Graphs, Charts, CSFs, Grouped Bar charts, Plots, and Graphics output objects. Multidimensional and other tabular reports such as the Multidim, Comparison Report, Expanding Report, Multi-Column, and Variance Reports will be available.

The Institute also intends (note that at the time this paper is being written, these plans are very rough and preliminary and may change by the time for the SUGI 24 conference) to provide an interim tool that will read a SAS/EIS desktop object or a SAS/AF FRAME entry and attempt to publish an HTML page corresponding to the visual display. The initial HTML produced by this tool can then be modified as needed by a developer to realize the desired visual display and performance. In the case of SAS/EIS desktop objects, it will generate HTML with embedded framesets to dynamically access current data for those objects which are based on data.

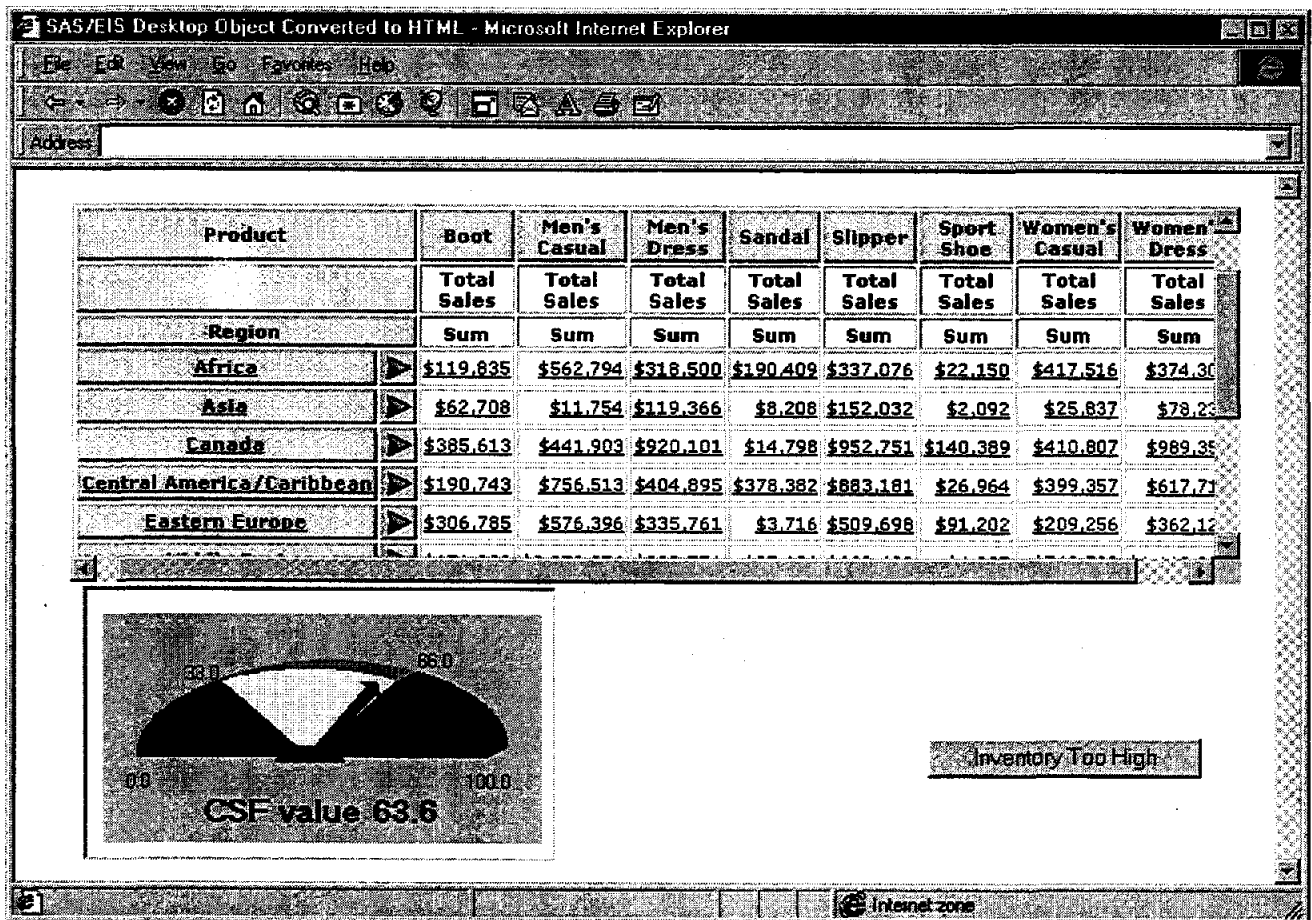


Figure 5: Sample Published EIS Desktop Object

Figure 5 shows such a published HTML page. Note that it contains a toolbar, multidimensional report, and 3D graph. Both the graph and the multidim object are generated at the time the user displays the object. Thus the data is current. Furthermore, the user can interact with the multidim object to dynamically drill down. This facility can also be used with SAS/AF FRAME entries leaving the responsibility to the developer to re-architect the SCL as appropriate.

Future Plans

This section will overview our future plans and will be updated after the SUGI Proceedings are published.

Conclusion

This paper summarizes several of the existing technologies in place today to deploy SAS/EIS and SAS/AF application to the Web. SAS will provide the latest tools and solutions available for deploying existing applications and for creating new and dynamic Web applications.

Acknowledgments

Appreciation is extended to Rob Stephens, Aaron Hill, Deva Kumar, Carl LaChapelle, Glen Walker, David Malkovsky, and Jean Moorefield for their input and review of this paper.

A special thanks to Vince DelGobbo for all his development efforts!

SAS is a registered trademark or trademark of SAS Institute Inc. in the USA and other countries. ® indicates USA registration. Other brand and product names are registered trademarks or trademarks of their respective companies.

Author Contacts:

Dave Horne, SAS
dave.horne@sas.com
Don Henderson, SAS
don.henderson@sas.com
Vince DelGobbo, SAS
vince.delgobbo@sas.com