

Paper 257-2008

## **SAS®: The Ultimate Dashboard Machine**

Jeff Wright, ThatWave Technologies, Chapel Hill, NC

### **ABSTRACT**

Leaders want to align their organizations around strategic goals and metrics. Knowledge workers need ready access to data about what is happening right now. IT seeks to make information available and useful to the business, and yet observe appropriate security policies. For many organizations, all of these goals are coming together in a single focal point: dashboards.

SAS is known for its advanced analytics and broad suite of business intelligence offerings, so it should be no surprise that there is more than one way to implement a dashboard initiative using SAS technologies. This paper will begin by reviewing the objectives that organizations normally have in creating dashboards. Next, several best practices for effective visual display of information on dashboards will be presented. Finally, the paper will describe a variety of techniques for creating dashboards using SAS, including the new BI Dashboard tool, SAS/GRAPH, custom web development, and the SAS Portal.

### **INTRODUCTION**

One recurring theme in business intelligence (BI) projects over the last five years has been dashboards. Leaders want to align their organizations around strategic goals and key performance indicators (KPIs). Knowledge workers need ready access to data about what is happening right now. IT seeks to make information available and useful to the business, and yet observe appropriate security policies. Everybody wants dashboards.

In his book *Information Dashboard Design*, Stephen Few defines a dashboard as follows:

A dashboard is a visual display of the most information needed to achieve one or more objectives; consolidated and arranged on a single screen so the information can be monitored at a glance [Few2006]

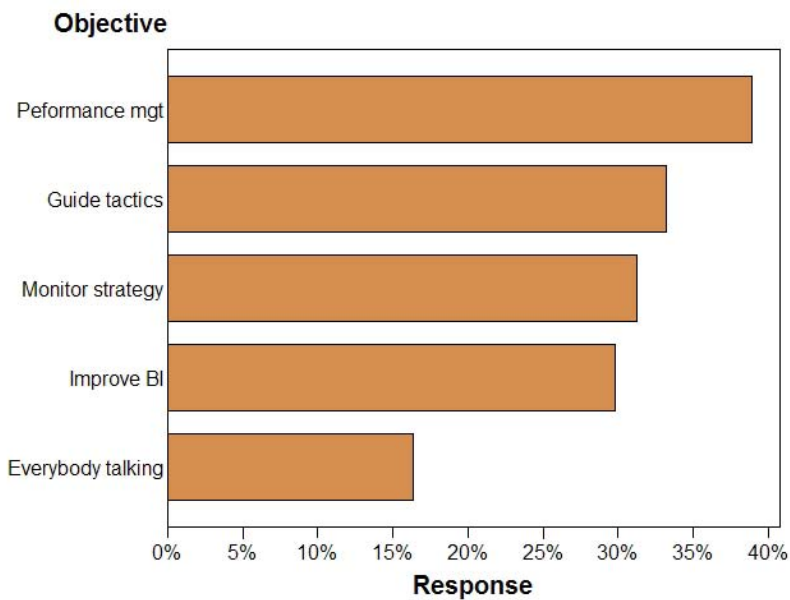
This definition does a good job of not only describing what a dashboard is, but also pointing the way towards dashboards that are valuable to the organizations that create them. This paper will describe and compare four distinct techniques of creating dashboards with SAS. But before getting to that, let's first examine the objectives for creating dashboards, and some principles for effective dashboards.

### **OBJECTIVES BEHIND DASHBOARDS**

In October 2007, ThatWave conducted a survey of over 200 people involved with using SAS to create dashboards. We asked them to choose one or more of the following reasons for interest in dashboards within their organization:

- Implement performance management based on metrics/KPIs
- Monitor strategic health/opportunities in the business
- Guide tactical execution of business operations
- Want to increase usage/relevance of a BI tool
- Everybody is talking about dashboards

The ranking of these reasons is illustrated in Figure 1:



**Figure 1 Dashboard Objectives Survey**

These results indicate that there is no one dominant reason for pursuing dashboards. Dashboards are of interest to executive management all the way down to individual members of staff, and include IT as well.

Multiple metaphors have been suggested to explain and guide dashboard creation. The very name suggests an analogy to the view from the driver seat of a car, or the cockpit of an airplane. Like a billboard, a dashboard needs to communicate a message in a short amount of time, usually graphically. Like a scorecard, a dashboard presents numeric measures of performance. A fourth metaphor is a newspaper front page. Consider these parallels:

- There is an audience that has both a shared interest and a diversity of perspective. For a newspaper, the shared interest of the audience revolves around a community. For a dashboard, it is a group of people with a common interest in the organizational function(s) depicted on the dashboard.
- It is designed to communicate at a glance. Newspapers have developed well-understood techniques of using headlines, pictures, and page layout to give emphasis to the information that is judged to be most likely of interest. A dashboard also needs to communicate quickly with graphics, and highlight areas that need attention.
- It invites further exploration. The front page is the starting point for the entire newspaper. Stories continue on other pages. The table of contents and other cues point to other features of the paper. Similarly, a dashboard provides interactivity to allow users to get more detail behind the information that is presented.

## EFFECTIVE DASHBOARDS

Given these objectives, let's examine four principles for effective dashboards.

### PRINCIPLE 1: PRESENT THE RIGHT INFORMATION

Stephen Few's definition points out that the audience for a dashboard has an objective in mind, and is concerned with monitoring a process or situation. Often there are multiple points of view on how best to do that, and gaining consensus on the right content for a dashboard is challenging. Defining dashboard content is particularly important when the dashboard contains KPIs. Performance metrics must be chosen to motivate the right behavior, because they will certainly motivate the behavior that is measured -- one way or another. Scott Adams played on this phenomenon in a Dilbert comic strip that described new bonuses for fixing bugs. The comic ended with Wally exclaiming "Woohoo! I'm gonna code me a minivan!"

It is sometimes the case that a dashboard project is initiated to provide better access to information contained in a particular application or data warehouse. Dashboard implementers should avoid the tendency to be focused on the one data source, and be alert to (possibly unstated) requirements to incorporate other information sources.

Experienced decision makers like to bring a number of perspectives to bear on their management responsibilities. For example, you would naturally expect a district sales manager to closely monitor progress against the sales

quota. But she may also need to keep tabs on high and low performing sales reps, activities that feed the sales pipeline, news about important customers and competitors, and status of marketing promotions. Developers should be prepared to get creative in order to source information from multiple systems or providers.

The principle of right information not only implies appropriate and useful; it also implies correctness. Dashboards share the characteristic of all BI tools, that users will avoid them in droves if they don't trust the data. Modern BI suites enable quick turnaround on producing data displays, but this is no excuse for skipping QA cycles focused on the data itself.

#### **PRINCIPLE 2: PROVIDE CONTEXT FOR MEASURES**

Again, imagine a dashboard that presents sales information. If last month's sales are baldly presented as a dollar amount – what does that actually mean? People require some context to determine whether a measure requires action to be taken. A measure in the context of a target is much more meaningful. A sales measure of "\$50M" is subject to interpretation, but "50% of target" communicates clearly that something is wrong. Note that obtaining and updating target data may be as significant a data sourcing challenge as some of the measures themselves, because targets and forecasts are frequently determined in an ad hoc way.

The importance of context may seem obvious, but it is easy to overlook. One scenario is where dashboards are intended to communicate across functional areas within an organization. The user lead from Sales may think it's obvious what "50" means, but if people from Manufacturing and Purchasing are using the dashboard also, they need the context to be explicit. This has implications for the visual display, and also suggests that there be convenient access to the end-user definition of dashboard metrics.

Another aspect of context is to represent how a measure has changed over time. Decision makers often start by asking: are we getting better or worse? This context can be shown as a simple trend over time. It is also common to compare discrete time periods, such as this month versus last month, or this month versus the same month last year.

#### **PRINCIPLE 3: COMMUNICATE EFFECTIVELY**

To be able to communicate the maximum amount of information at a glance, dashboard designers should understand and use principles of human perception, such as how people react to color. Colors naturally act as an attention-getting mechanism. It is common, at least in the US, to use a traffic light metaphor. The colors red-yellow-green are intended to communicate a status of needs attention, warning, and meets expectations.

If applied simplistically, this technique results in a dashboard that shows a lot of green under normal conditions – and this could be distracting. It is easier to assimilate the dashboard at a glance if no special coloring is used for measures that meet expectations, and their attention is drawn to those data points which actually need attention.

Not all graph types that are commonly used actually communicate well. Pie charts are over-used, ignoring that humans are much better at judging length than area. So while there are certainly applications for dividing a fixed whole into a pie chart, in cases where it is important to judge the size of each slice, or to make careful comparisons of slices, it would be better to use a bar chart instead. Some new chart types such as bullet graphs and spark lines are especially appropriate for dashboards [Few2006].

#### **PRINCIPLE 4: FACILITATE TAKING ACTION**

Dashboards should do more than communicate information; they should facilitate acting on that information. We've already discussed the importance of presenting information in context, and using appropriate visual design to highlight what needs attention. Both these techniques facilitate action by identifying exceptions.

Dashboards should also serve as a launch pad to explore the data. Normally the first response to an exceptional data point is to explore to find out why the measure is out of tolerance. This makes it vital that dashboards be seamlessly integrated with other typical BI tools such as standard reports, ad hoc queries, and OLAP analysis.

Ideally, the BI platform should also enable collaboration around problem solving. As different people investigate business conditions reflected on the dashboard, they should be able to annotate their findings into the BI system so that others can benefit from their insights.

### **SAS DASHBOARD TECHNIQUES**

So far we have looked at objectives for dashboards and principles for effective dashboards. Now we turn to a survey of four techniques for creating dashboards with SAS.

## TECHNIQUE 1: SAS BI DASHBOARD

The SAS BI Dashboard tool became available in 2007 as an add-on to SAS Enterprise BI Server. The breadth of capability, integration with the SAS9 intelligence architecture, and ease of use make BI Dashboard a logical starting place for a discussion of dashboard implementation techniques with SAS.

In order to talk about the BI Dashboard, it's helpful to define the terminology that is used in working with the tool:

- A *dashboard* is made up of indicators. Each *indicator* is a visual representation of a performance measure, or in some cases a collection of performance measures. For example, an indicator might be a bullet graph that shows year-to-date sales as a percent of target. It is both a measure and a visualization of the measure.
- Indicators obtain their data from a *data model*. In addition to the numeric value for a measure, data models can provide additional details such as a formatted text version (*label*) of the value and a contextual *name*.
- Indicators can also be associated with a *range*, which defines threshold values of interest for measures.

BI Dashboards are created and viewed in the Information Delivery Portal, so there is a relationship between this SAS dashboard technique and the fourth one that will be described below.

### DATA SOURCES

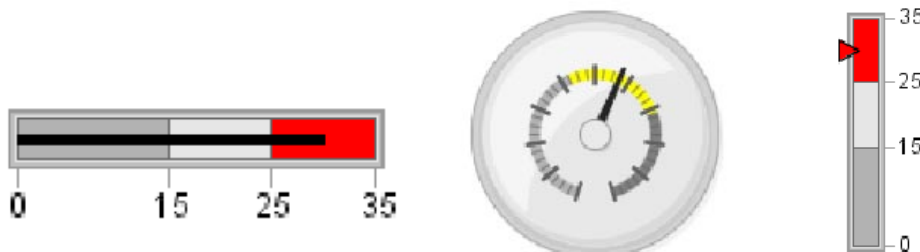
Data for dashboards can be obtained from three sources:

- A SAS Information Map.
- A SQL query.
- A SAS Strategic Performance Management scorecard.

Note that SQL queries use SAS pass-through SQL against a SAS library. So the SQL queries can go against any type of database or data store for which you have the corresponding SAS/ACCESS product. The same holds for Information Maps, which can pull data from any SAS/ACCESS source in addition to native SAS storage.

It is possible to enable caching with BI Dashboard data sources. This not only improves response time in displaying dashboards, but also improves multi-user scalability by allowing multiple users to share the same data source query results.

### INDICATOR TYPES



**Figure 2 BI Dashboard  
Dynamic Gauges**

Many users picture dashboards as an array of gauges, and you can certainly achieve that with BI Dashboard. The tool comes with dials, sliders, and bullet graphs that dynamically display a KPI in the context of a range. Each has a variety of display options for setting colors, size, and 2D/3D appearance. Some examples are shown about in Figure 2.

Another common paradigm for dashboards and other management reports is the traffic light. To increase information density, it is useful to boil down some business process or measure to a status of green, yellow, or red. If this is represented with an actual traffic light icon, even a user who is red-green color blind can spatially determine the status color. BI Dashboard includes a library of traffic lights and other static images that can be used to indicate status by range.

BI Dashboard enables not only the typical dials and traffic lights, but also more sophisticated graphs, including:

- Trend graph

- Scatter plot
- Bar with reference lines
- Heat map

Finally, if these options are not enough, it is possible to associate any URL with an indicator, so that a completely custom display may be used. Figure 3 shows an example of SAS stored process output created to serve as a dashboard indicator display:

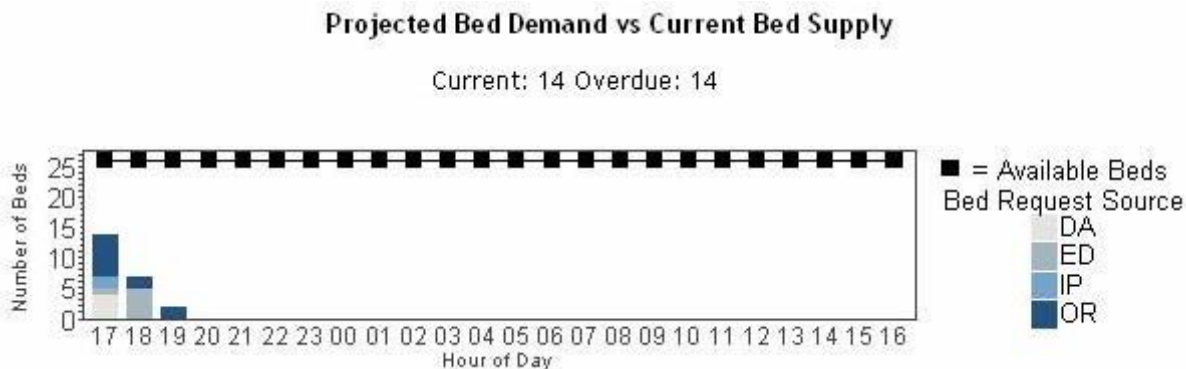


Figure 3 BI Dashboard Custom Graph

**RANGE MANAGEMENT**

Code Interval (For static gauges only)	Interval	Label	Color (For dynamic gauges only)	Delete
Below Target	<= 0	Below Target	Red	X
On Target	> 0 and <= 15	On Target	Gray	X
Above Target	> 15 and <= 25	Above Target	Yellow	X
Below Target	> 25 and <= 35	Below Target	Red	X
Below Target	> 35	Below Target	Red	X

Figure 4 BI Dashboard Range Definition

One of the principles for effective dashboards cited above is to present measures in context. BI Dashboard allows creation of ranges as one way of providing that context. Each interval of measure values is given a text label, such as "Below Target," and a color, as shown in Figure 4. These ranges are named and can be shared by more than one indicator. Dashboards are often used to present information to executives and across functional boundaries. Ranges enable users that are less than expert with a particular KPI to quickly judge whether the target for a value is low, middle, or high.

In the above example a shade of gray is used as the color for "On Target". While green is often associated with measures that are on target, from a human factors standpoint this can sometimes detract from the ability of a dashboard to highlight areas that need attention. It is often a better choice to use a neutral hue for on target values, and reserve colors for measures that are exceptional.

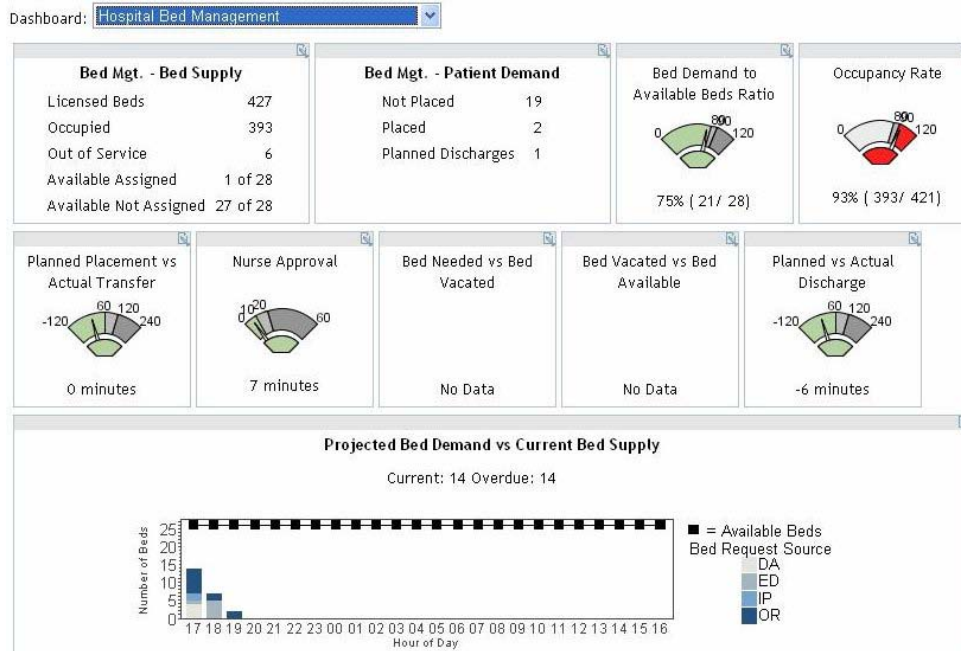
**DRILL DOWN**

Another principle for effective dashboards is to enable action. Often the next logical action for a measure that is out of range is to get more information. To facilitate that, BI Dashboard enables indicators to have a drill down link to:

- Another dashboard
- A web report
- A stored process
- A portal page
- An external URL

In addition to defining a drill down link in the indicator definition, some indicator displays allow a hyperlink to be taken from the data model, so that hyperlinking is dynamically associated with data values.

### CASE STUDY



**Figure 5 Dashboard Case Study**

The dashboard in Figure 5 was created to complement a hospital operational management application. This application deals with non-clinical functions such as bed management, environmental services, transport, and surgery scheduling. Timely and clear display of information is critical in a health care setting, which motivated the need for dashboards.

In contrast to many BI applications which are based on a data warehouse with daily updates, for this application the goal was to display data that is within 5 minutes of current. With this constraint, it was necessary to pull data from transactional data models that were not fully summarized and optimized for reporting. In order to address these issues, the implementation included:

- Stored procedures running with the database (SQL Server) to query and transform transactional data
- A SAS query stack including information maps, stored processes, and SAS/ACCESS to OLEDB
- Caching to improve dashboard performance

## TECHNIQUE 2: SAS/GRAPH FOR DASHBOARDS

It is also possible to create dashboards using only Base SAS and SAS/GRAPH. There are some nice examples of SAS/GRAPH dashboards on the SAS web site [graph]. In Figure 6, notice the use of quickly digested bar graphs and spark lines, and how bright colors (red/orange) are reserved for items that need attention.

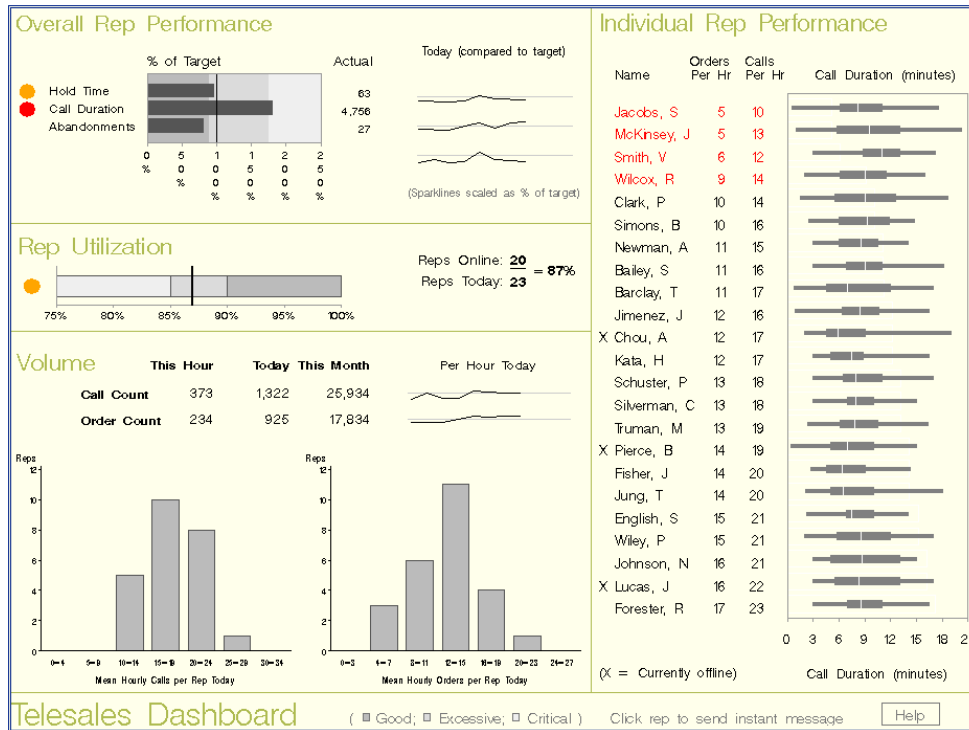


Figure 6 SAS/GRAPH Dashboard

This technique involves programming, which can be broken down into four steps:

- Obtain data using the DATA step or PROC SQL (and SAS/ACCESS if your data is in a non-SAS data store).
- Create tabular and graph displays using Base SAS procedures and SAS/GRAPH.
- Layout the visualization components on the page.
- Generate the desired output file (such as HTML or PDF) using the Output Delivery System (ODS).

Recall that the definition of a dashboard includes the concept of fitting information onto a single screen. This implies a need to lay out components carefully in order to achieve a dashboard that is both information dense and understandable. Some techniques to consider include:

- If your dashboard consists entirely of SAS/GRAPH output, as in the above sample, you can save graph output to a catalog, then use PROC GREPLAY to position the graph displays on the page.
- Similarly, you can use the ODS DOCUMENT destination to send any SAS output (not just graphs) to an item store, and then replay using PROC DOCUMENT. The positioning on the page can be controlled using ODS LAYOUT, but note that this facility is marked experimental in SAS 9.1.
- If you are generating HTML output, you can embed HTML or CSS formatting into your output to control layout. Ideally you would create a tagset to make this logic reusable. The htmlpanel tagset is a SAS 9.1 tagset available from the SAS web site [htmlpanel] for just this purpose.

Once a SAS/GRAPH dashboard has been created, it can be made available to users in any number of ways, including:



- Copy to a web server
- Copy to the WebDAV content repository and view on the SAS portal (more on this later)
- Directly email to a distribution list
- Publish to a channel
- Or even (shudder) print it out on paper

### TECHNIQUE 3: CUSTOM FRONT-ENDS TO SAS

Most dashboard projects aim to deliver information to users via a web browser. This raises an interesting question: is it possible to use conventional web development techniques for dashboards, and to simply use SAS as a back-end service for data integration and analytics? The answer to this question is not only yes, but there are quite a large number of integration options from which to choose. These techniques apply for both Java and .NET web development, and some of them are even usable from LAMP (open source scripting) web applications.

The SAS integration options can be categorized as three levels, ranging from data services to analytic services to reporting services. *Data services* are the lowest level of integration, and imply that we are only accessing data via SAS. The data being accessed could reside in some form of SAS Intelligence Storage, such as traditional SAS data sets or an OLAP cube. Alternatively, the data may be in a database or application for which a SAS/ACCESS product exists. SAS provides a federated data abstraction across all the configured data sources, enabling a consistent access mechanism and even the ability to join data from disparate sources. Clients can invoke SAS data services using standards such as JDBC, ODBC, OLE DB, OLE DB for OLAP, and ADO.

*Analytic services* add the ability to tap into the computational procedures for which SAS is so well known. The primary mechanism for accessing analytic services from SAS is the Integrated Object Model (IOM) programming interface. IOM is available to Java and .NET programmers, and allows client programs to submit SAS code and call SAS stored processes, as well as accessing data. In addition to IOM, it is possible to access the analytic capabilities of SAS via SAS BI Web Services. These services follow the XML for Analysis version 1.1 specification. SAS BI Web Services provide a SOAP-based interface for SAS stored processes.

*Reporting services* add presentation-level formatting to the analytic and data access features. When SAS is used as a reporting service, it delivers pre-formatted tables, graphs, and complete documents to the client application. Once again, the IOM programming interface is probably the first choice as the mechanism for this integration. A typical scenario would be to define a SAS stored process that produces streaming output, and then to retrieve this result via IOM. However, other approaches are possible. It might be desirable to run SAS reports asynchronously, and send the output to a directory or content repository that is accessible to the web tier application.

The bottom line is that there are a lot of options for how to integrate SAS into a web application. Choosing the best design for your application needs to start with your functional requirements. What are your data sources, and what processing needs to be done with SAS? The capabilities of your web development environment need to be lined up against the capabilities of SAS. Is your project team more comfortable with using the web development tools to prepare tabular output, or are they comfortable with SAS ODS formatting? Which tools can create the graphs that users need? Finally, license considerations play a role in deciding the architecture. Some of the integration options require optional SAS products such as Integration Technologies or SAS/SHARE. Also, pay attention to whether your SAS license has per-user restrictions which are inconsistent with a web interface.



## TECHNIQUE 4: SAS PORTAL

The SAS Information Delivery Portal is an integral part of SAS Enterprise BI Server, providing a unified web interface for the various SAS BI tools, and even allowing access to non-SAS content. The Portal interface is made up of *pages* and *portlets*. Portlets are the lowest unit of content, and make up a fragment of a page. Pages are accessed using a tab control.

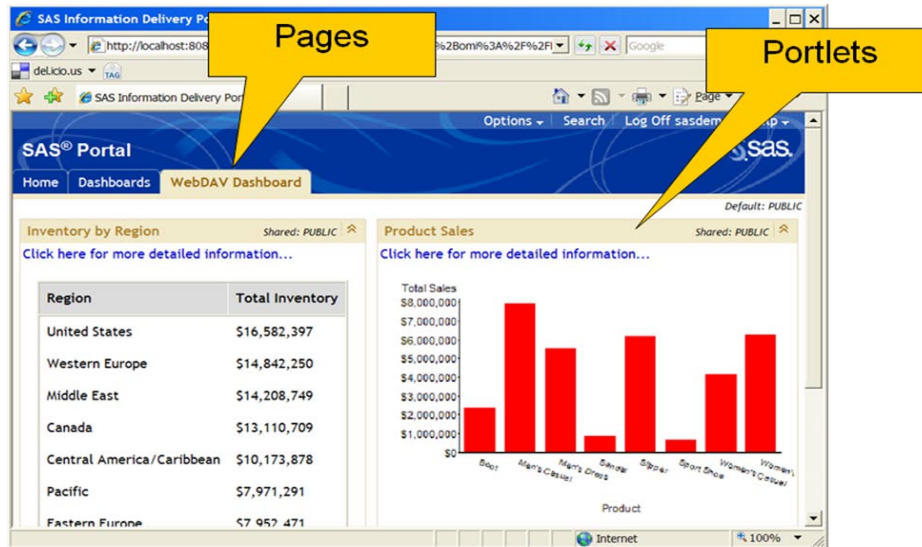


Figure 7 SAS Information Delivery Portal

The SAS Portal provides our fourth and final method of defining a dashboard: create a dashboard as a portal page made up of one or more portlets. So what can be displayed in a portlet?

- Information maps
- Stored processes
- HTML page fragments stored in the WebDAV content repository
- Portal-generated graphs of data in the WebDAV content repository (in XML form)
- "Syndicated" content from any web page with a URL
- And don't forget: SAS BI Dashboards

This set of portlet choices means that the SAS Portal is not only an additional way to create a dashboard, it's also a way to mix and match dashboards created via the previous three techniques. Just because you're using the BI Dashboard doesn't mean that its portlet has to be the only thing on the portal page. It could be side by side with a SAS/GRAPH dashboard and a custom portlet that is making IOM access to SAS analytic services.

In order to get the most out of the portal, it's important to understand the role of the WebDAV content repository in the SAS intelligence architecture. The WebDAV server provides a network accessible storage location for documents and pre-generated reports. Its security is integrated with SAS, so that authorization can be configured in the SAS Management Console. Consider the example of a SAS/GRAPH dashboard. Such a dashboard could be generated on demand using a stored process. While this may be the right approach for some applications, in other cases the data behind the dashboard may only change once a day, when a data warehouse is updated. If that's the case, it is probably less demanding for server resources and more responsive to users if the dashboard is created offline immediately after the data warehouse is refreshed. The output can be written to WebDAV and accessed via the WebDAV content portlet.

In our previous discussion of principles of effective dashboards, we observed that experienced managers like to bring a number of perspectives to bear on their decisions. The versatility of a portal paradigm facilitates delivering that kind of 360 degree perspective, because it provides access from one place to many types of information, including static reports prepared off-line, dynamic reports prepared on demand, and syndicated information from other applications.

## COMPARISON OF DASHBOARD TECHNIQUES

Four techniques for creating dashboards with SAS have been presented. Let's recap and examine how they compare to each other. The *BI Dashboard* tool is the newest technology, and it exists specifically to create dashboards. It is heavily oriented towards KPIs, and contains an extensive palette of gauges, traffic lights, and graphs with which to visualize them. Each indicator can have a drilldown link. With BI Dashboard, creation of dashboards occurs without programming (if the data is prepared!). Not only that, but BI Dashboard provides a user interface for managing ranges, and has a fine-grained permission model that allows end-user customization of dashboards to be enabled or prohibited. This flexibility gives organizations the opportunity to distribute some aspects of dashboard definition to end-users, where appropriate. However, be aware that data must be prepared in a specific format for use by BI Dashboard, and the options for controlling layout of indicators on dashboards are somewhat simplistic. BI Dashboard requires SAS Enterprise BI Server.

*SAS/GRAPH dashboards* are most useful for producing a limited number of carefully designed dashboards. Because they are created with the Base SAS programming language, there is excellent control over layout and you can create highly dense, customized displays. It does not require an Enterprise BI Server license to create this style of dashboard – only Base SAS and SAS/GRAPH. In fact, organizations with a strong SAS programming history may actually be more comfortable with this approach because it doesn't require the full SAS9 intelligence architecture. However, Enterprise BI Server customers can use its facilities to publish dashboards created with SAS/GRAPH. Organizations without a strong SAS programming skill set may be daunted by this technique, but the skill set concern can potentially be mitigated by deploying Enterprise Guide with its wizard-based approach to queries, analytics, and graphs.

Dashboards can also be created using a *custom front-end to SAS*. As with SAS/GRAPH dashboards, a custom front-end offers the opportunity to use programming techniques to create exactly the desired dashboard design. It could also be appealing where dashboards are intended to be integrated into an application that has many non-SAS functions. This technique is in some cases a better fit to the skill set of an organization, or the distribution of skills within the organization. A custom front-end does not require Enterprise BI Server, but it will typically require either Integration Technologies, SAS/SHARE, or both.

Finally, the *SAS Portal* can also be used to create dashboards. The portal paradigm of composing pages out of portlets offers the opportunity to mix and match BI content created with a variety of techniques, including the other dashboard techniques and even non-SAS sources of information. Like the BI Dashboard, there is the opportunity to configure local permissions to enable some user self-service of their pages and portlets. For full capability, the portal requires Enterprise BI Server, and also requires that Xythos WebDAV be installed as the content repository.

## CONCLUSION

This paper began by describing common objectives for dashboards, and presented four principles for effective dashboards. These principles can be expressed as the following questions for evaluating a dashboard design:

- Does this dashboard contain the right information for its audience?
- Has enough context been provided for the dashboard measures, both to understand the measures and to evaluate their status?
- Does the visual design communicate as clearly as possible?
- Does the dashboard provide interactivity to react to its content?

Finally, four specific techniques for creating dashboards with SAS were presented. Which one is right for your problem?

## REFERENCES

[Few2006] Few, Stephen. 2006. *Information Dashboard Design*. Sebastapol, CA: O'Reilly Media, Inc.

[graph] SAS Institute Inc. "SAS/GRAPH Dashboard Samples."  
<http://support.sas.com/rnd/datavisualization/dashboards>.

[htmlpanel] SAS Institute Inc. "The htmlpanel Tagset Controls Paneling."  
<http://support.sas.com/rnd/base/ods/odsmarkup/htmlpanel.html>.

**ACKNOWLEDGMENTS**

Thanks very much to my colleagues at ThotWave for their collaboration on numerous dashboard projects, and especially to Danny Grasse and Richard Phillips for their suggestions for this paper.

**CONTACT INFORMATION (HEADER 1)**

Contact the author at:

Jeff Wright  
ThotWave Technologies, LLC  
510 Meadowmont Village Circle, #192  
Chapel Hill, NC 27517  
E-mail: [jwright@thotwave.com](mailto:jwright@thotwave.com)  
Web: <http://www.thotwave.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.