

Paper 042-2011

Surfacing Content from SAS® Enterprise Business Intelligence in Third-Party Portals

Meera Venkataramani, SAS Institute Inc., Cary, NC
Sam Atassi, SAS Institute Inc., Chicago, IL
Gaurav Agrawal, GCE Solutions Inc., Alpharetta, GA

ABSTRACT

Corporate portals have become the preferred choice for many enterprises to surface their company information. These portals provide a single Web site for all users to access and discover services and content provided by the enterprise.

This paper focuses on integrating SAS® content, such as reports, stored processes, and dashboards in an IBM WebSphere Portal. It presents a customer-use case with examples for configuring, deploying, and customizing content using JSR 168 SAS BI portlets. The paper shows how to personalize the portlets to share a pre-populated content view to a group of users in the enterprise.

The paper explores methods to surface SAS content in Microsoft SharePoint.

INTRODUCTION

JSR 168 portlets. What are they? In recent years, organizations have implemented enterprise portals to host internal and external applications. There are numerous portal vendors offering products in this market. In the past, each of these portal offerings defined its own proprietary APIs for building portlets, which are application components that run inside portals. Unfortunately, coding to these various APIs translated into vendor lock-in for portlet developers.

The JSR 168 specification, developed by a committee of J2EE portal vendors, aims to achieve interoperability between portals and portlets. Vendors prove compliancy by passing a series of tests defined in the Sun Microsystems Technology Compatibility Kit (TCK). This standardization helps to simplify portlet development and enables developers to create pluggable components that run on any compliant JSR 168 portal container. The specification defines portlets as Java Web components, managed by a portlet container that processes, requests, and generates dynamic content. Portals use portlets as pluggable user interface components that provide a presentation layer to information systems.

SAS PORTAL VERSUS WEBSHERE PORTAL 6.1.0

SAS® Information Delivery Portal provides a Web-based user interface that enables users to navigate and access a wide variety of SAS information. This information includes reports, charts, Web applications, documents, and links to internal and external Web pages. The layout of pages and portlets on a page can be personalized. Content can be aggregated and shared securely with individuals and groups.

JSR 168 portals are containers for JSR 168 portlets that can be provided by many vendors, including SAS. A container like WebSphere Portal enables the customer to build a portal containing SAS content, together with content and functionality provided from other vendor portlets. For example, a corporation might want to offer a portal (hosted on WebSphere) for its external users to securely view corporate data from varied sources that might include some SAS content, such as SAS reports, collections, and a stored process. The same portal might include a calendar portlet (not from SAS) showing important events for suppliers and contractors.

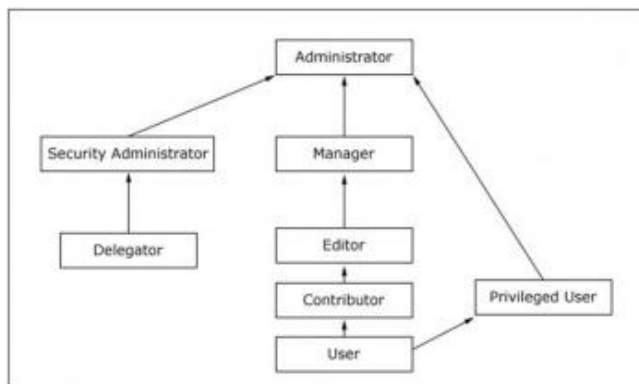
Integrating SAS® Enterprise BI with the WebSphere portal creates a more powerful user experience by combining the capabilities of the two platforms.

ROLES AND RESOURCES IN WEBSHERE PORTAL

Before we venture into this example, it is important that you understand the WebSphere terminology of roles and resources.

Roles access control is based on roles. A *role* combines a set of allowed actions with specific resources. This set of allowed actions is called a *role type*. For example, the role type **Editor** contains allowed actions like view resources,

modify resources, and create new resources. As the following figure illustrates, roles are organized in a hierarchy. Each role type contains all allowed actions that are contained in the role types directly beneath it in the hierarchy. For example, A **Privileged User** and **Editor** can do everything that a **User** can do. A **Manager** can do everything that an **Editors** and a **User** can do.



The following table describes the individual role types.

Role Type	Allowed Actions
Administrator	Unrestricted access on resources. This includes creating, configuring, and deleting resources. Administrators can change the access control settings on a resource. In other words, administrators can grant other people access to those resources.
Security Administrator	Creating and deleting role assignments on resources. Being assigned the Security Administrator role on a resource means that the user is allowed to act as a delegated administrator for that resource. In other words, the Security Administrator on a resource is allowed to delegate a subset of his privileges on the resource to other people. For example, a user who is assigned the Security Administrator and Editor role on a resource can assign the Editor role to other people, provided he has the Delegator role on those people. Having the Security Administrator role on a resource alone does not give view or edit access to the resource.
Delegator	Assigning the Delegator role to principals (users and groups) allows roles to be granted to them. Having the Delegator role on other resources, such as specific portlets, is not useful. The set of roles that can be granted to those principals is defined through the Security Administrator and Administrator role types. For example, a user has a Delegator role on the SalesTeam user group, but he does not have the Delegator role on the Managers user group, so this user can grant roles only to the SalesTeam or individual members of the SalesTeam user group, but not to the Managers user group. Having the Delegator role on a resource does not give direct access to the resource. The purpose of the Delegator role type is to allow the granting of roles to users or groups, so assigning the Delegator role on resources or resource types that are not users or user groups does not grant those users additional privileges.

Role Type	Allowed Actions
Manager	Creating new resources and configuring and deleting existing resources that are used by multiple users.
Editor	Creating new resources and configuring existing resources that are used by multiple users.
Contributor	<p>Viewing portal content and creating new resources. The Contributor role does not include the permission to edit resources. It only enables you to create new resources. For example, a user is granted the Contributor role on the Template Category Teamspace. The user cannot modify the category itself, but can create new templates in this category.</p> <p>Note: This role type is available only for the following resources:</p> <ol style="list-style-type: none"> 1. Application Templates 2. Application Template Categories 3. Application Template Root 4. Policies 5. All IBM Workplace Web Content Management related documents
Privileged User	Viewing portal content, customizing portlets and pages, and creating new private pages.
User	Viewing portal content. For example, viewing a specific page.
No Role Assigned	Cannot interact with a resource.

INHERITANCE

Resources are part of a hierarchy. By default, each resource in the hierarchy inherits the role assignments of its parent resource. This inheritance reduces the administration overhead. When you assign a group to a role on a parent resource, the group automatically acquires that same set of allowed actions for all child resources. For example, suppose that a user—Mary—is a member of the Sales group. You can give Mary Editor access to the Market News Page and all pages underneath this page by granting the Editor@Market News Page role to the Sales group. All members of the Sales group implicitly acquire the Editor@Market News Page role. All members of the Sales group inherit the Editor role type on all pages that are beneath the Market News page in the resource hierarchy. So, members of the Sales group automatically inherit the role Editor@USA Market News Page. Inheritance through the resource hierarchy can be blocked at any level to provide more granular access control.

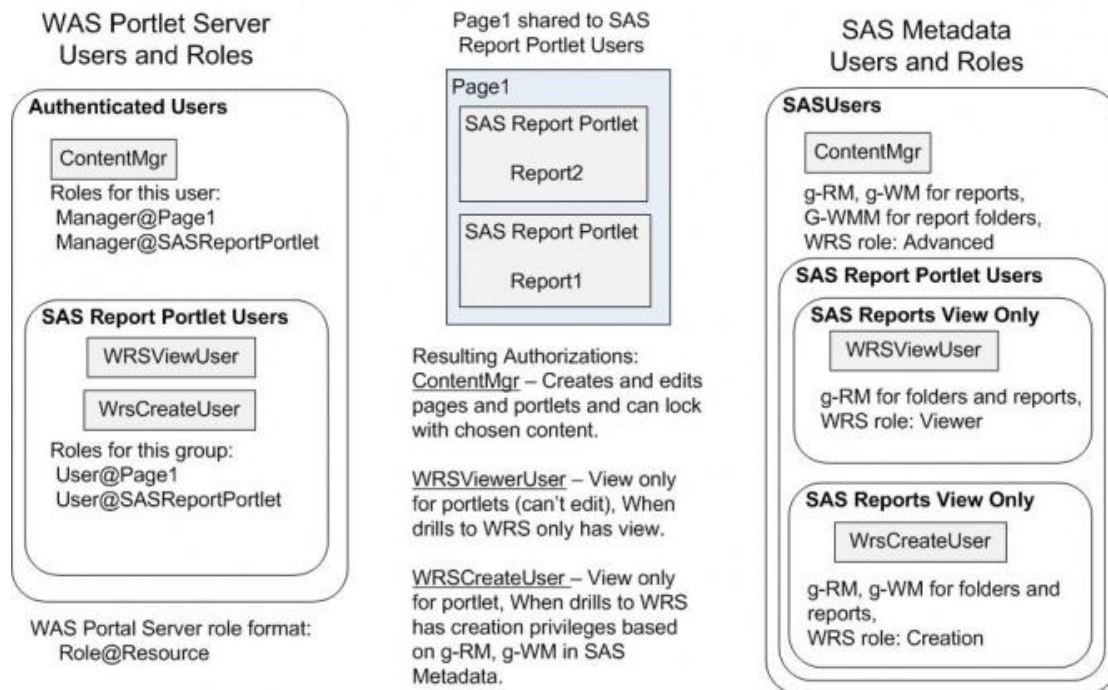
PRIVATE PAGES

A private page can be accessed only by its owner. Privileged Users (users assigned a role of type Privileged User) can explicitly create new private pages that are accessible only by themselves. In addition, a Privileged User on a non-private page can personalize the page, and create new private pages underneath it. Customizing a non-private page usually creates a private copy of the corresponding non-private page. Any changes that a Privileged User makes to a non-private page are not accessible by other users. Private pages cannot be controlled by an external security manager. Access control for private pages is always internally controlled by WebSphere Portal.

SIMPLE USE CASE: USERS AND GROUPS

SASPortalAdmin wants to create a shared page in WebSphere Portal and share it with his users. His shared page will contain SAS reports and SAS stored processes for his users to view. He simply wants his users to view the reports and stored processes. He does not want his users to navigate the SAS® Metadata Repository or personalize the portlets. Personalize is a WebSphere Application Server (WAS) role, which enables the user to navigate the SAS Metadata Repository to pick a report and create a private view for himself. You need to be a privileged user in the WAS portal to be able to do this.

JSR168 Container: IBM WAS Portlet Server
 SAS BI Portlets Authorization Use Case 1



CONFIGURING SAS BI PORTLETS TO DISPLAY PRE-LOADED CONTENT

Portlet preferences store persistent data for a portlet user in the form of name value pairs, which are retrievable during the render phase of the portlet. This property name and the SBIP URL can be passed on to a portlet and be loaded at run time. The portlet can be configured to use this name value pair mapping to pre-load a portlet (SAS report and SAS stored process).

When you add a SAS report portlet to a page, the portlet is always called SAS Report Portlet. To change the header name of the portlet, you need to copy the portlet and rename it to something more meaningful so that your header is indicative of the reports that you are viewing. After you copy the portlet, you need to configure the new portlet with the preference name value pair mapping. This way, you do not end up with all SAS report portlets being called "SAS Report Portlet" and displaying the exact same content. (This mechanism should work for the stored process portlet. It does work for SAS® BI Dashboard.)

COPY PORTLET AND ASSIGN ACCESS

There are two major steps in this section—configure the BI portlets for WAS portal, and create users and groups.

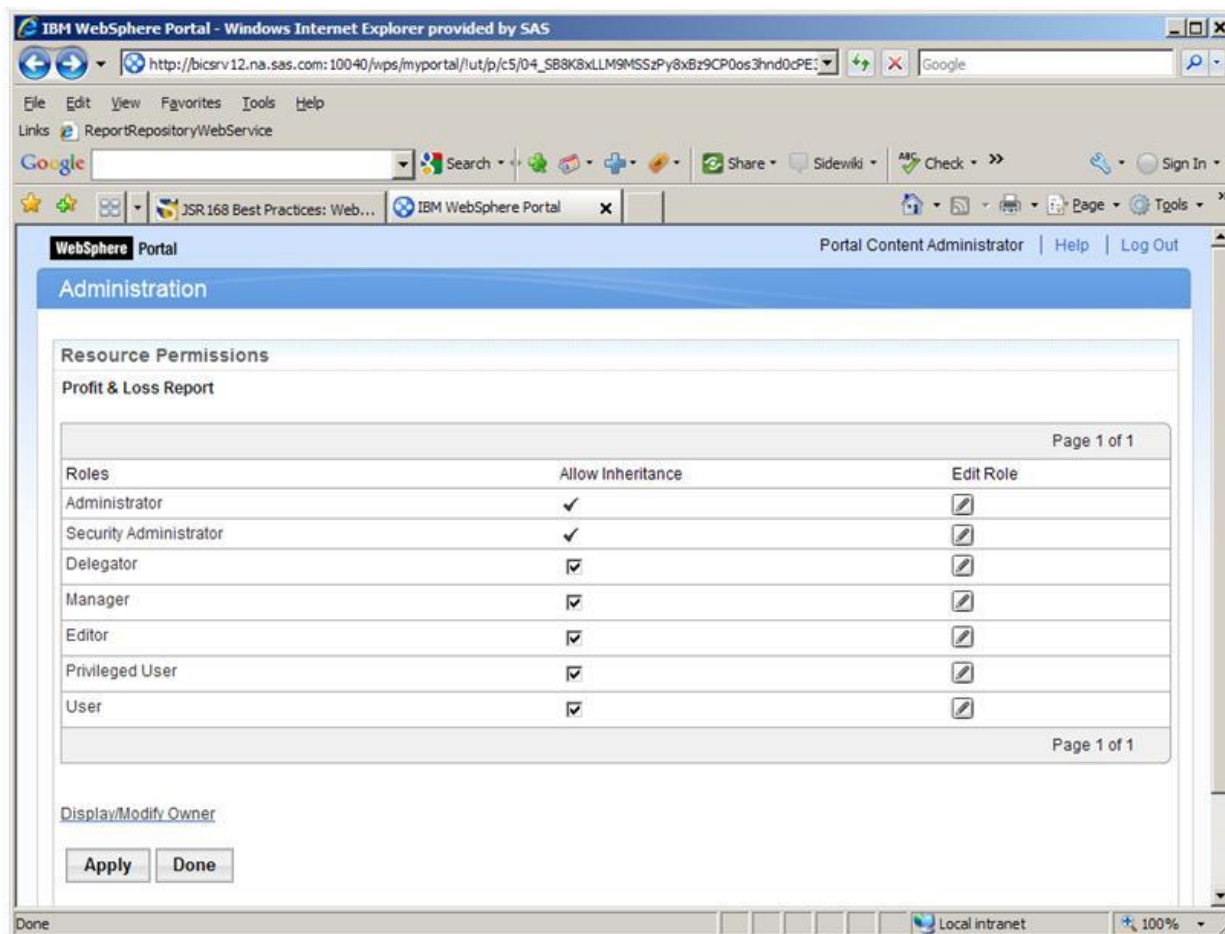
For simplicity, users and groups are limited to the following:

UID	GRP	WAS Role	SAS Metadata Auth
SASPortalAdmin	Portal Content Admin Group	admin	Creates and edits pages and portlets and can lock with chosen content.

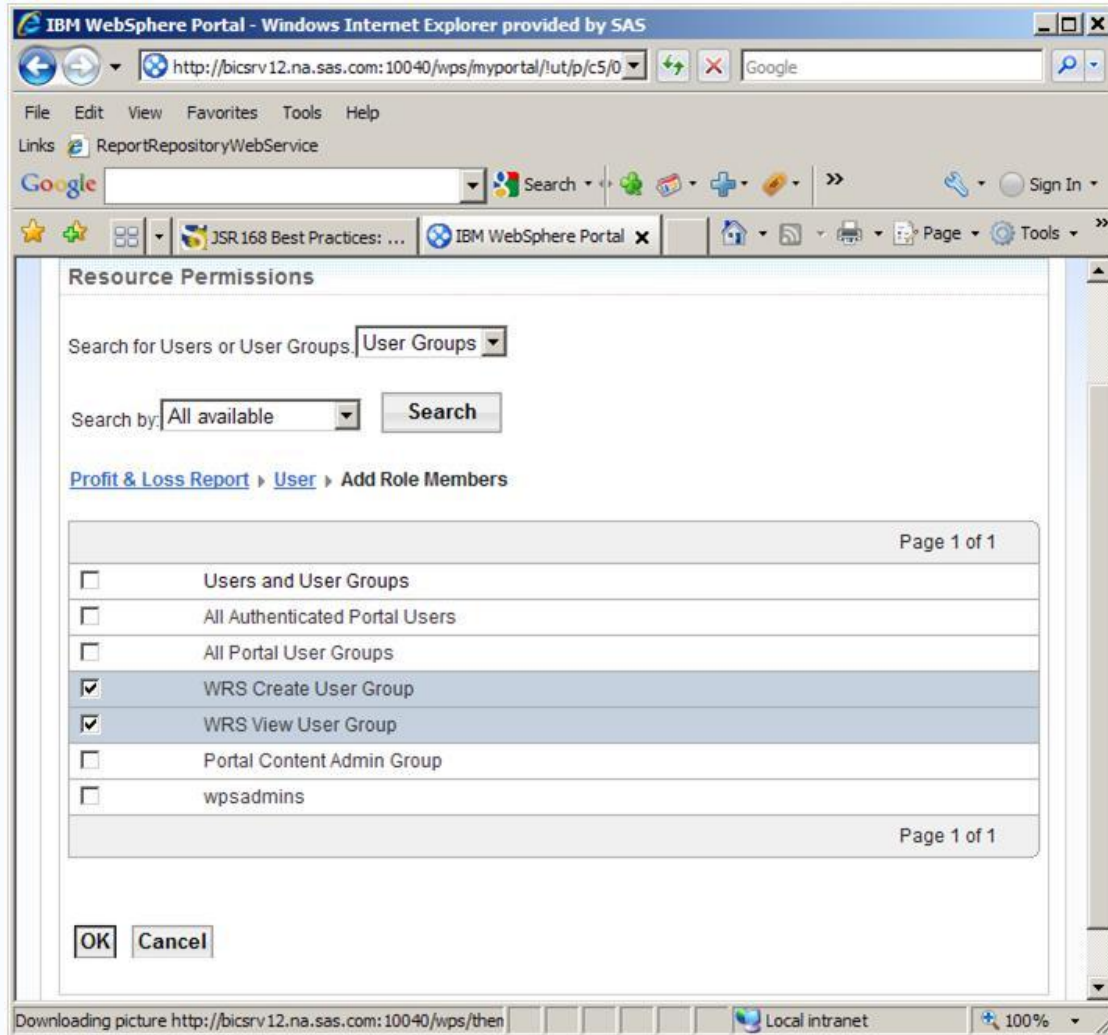
WRSCreateUser	WRS Create User Group	User	View only for portlet: Report creation capability for WRS.
WRSViewUser	WRS View User Group	User	View only for portlets: Report viewing capability for WRS.

Note: These users have to be created in the SAS Metadata Repository using the SAS® Management Console with appropriate privileges.

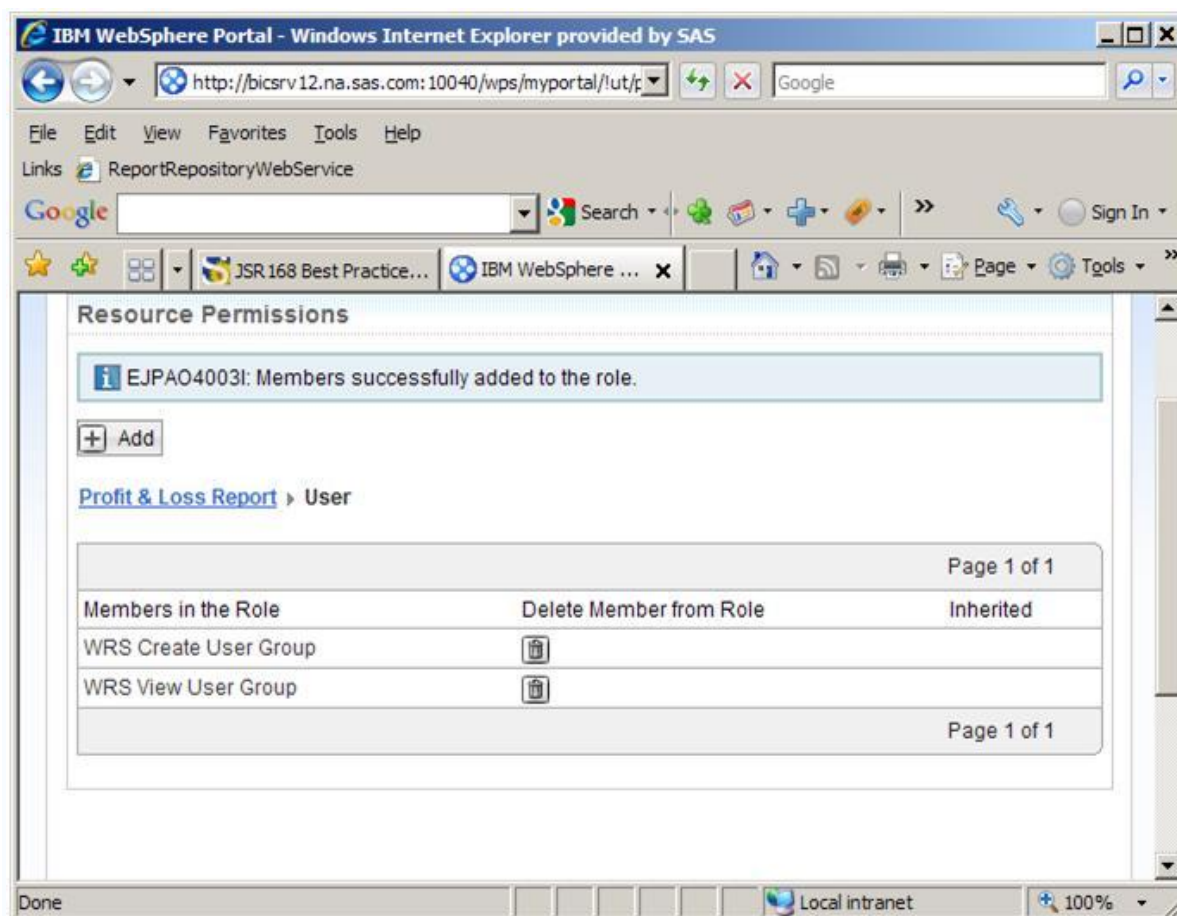
1. Log in as SASPortalAdmin in WAS portal. Go to the **Administration** tab. Click **Portlets**.
1. Search for SAS Report Portlet. Click on the Copy Portlet icon on the right.
2. In the **Name** field, enter Profit & Loss Report.
3. Repeat this process to create another portlet named **Balance Sheet Report**.
4. In the Profit & Loss Report portlet that you created, click on the Assign Access icon on the far right.



5. Click **Edit Role** for **User** role, and add **WRS View User Group** and **WRS Create User Group** as role members.



You end up with something like the following:



6. Click on the **Profit & Loss Report** breadcrumb to go back. Click **Apply**, and then **Done**.
7. Repeat this process for the Balance Sheet Report portlet.

NAME VALUE PAIR CONFIGURATION

Next, you need to configure the portlets with the preference name value pair parameters as inputs to be loaded at run time. This gives you pre-loaded content for the portlet.

Before you do this, the administrator has to decide which report or stored process in the SAS folders that he wants displayed in the portlet. The configuration for the portlet is on a per-portlet basis, and it persists for all instances of that portlet.

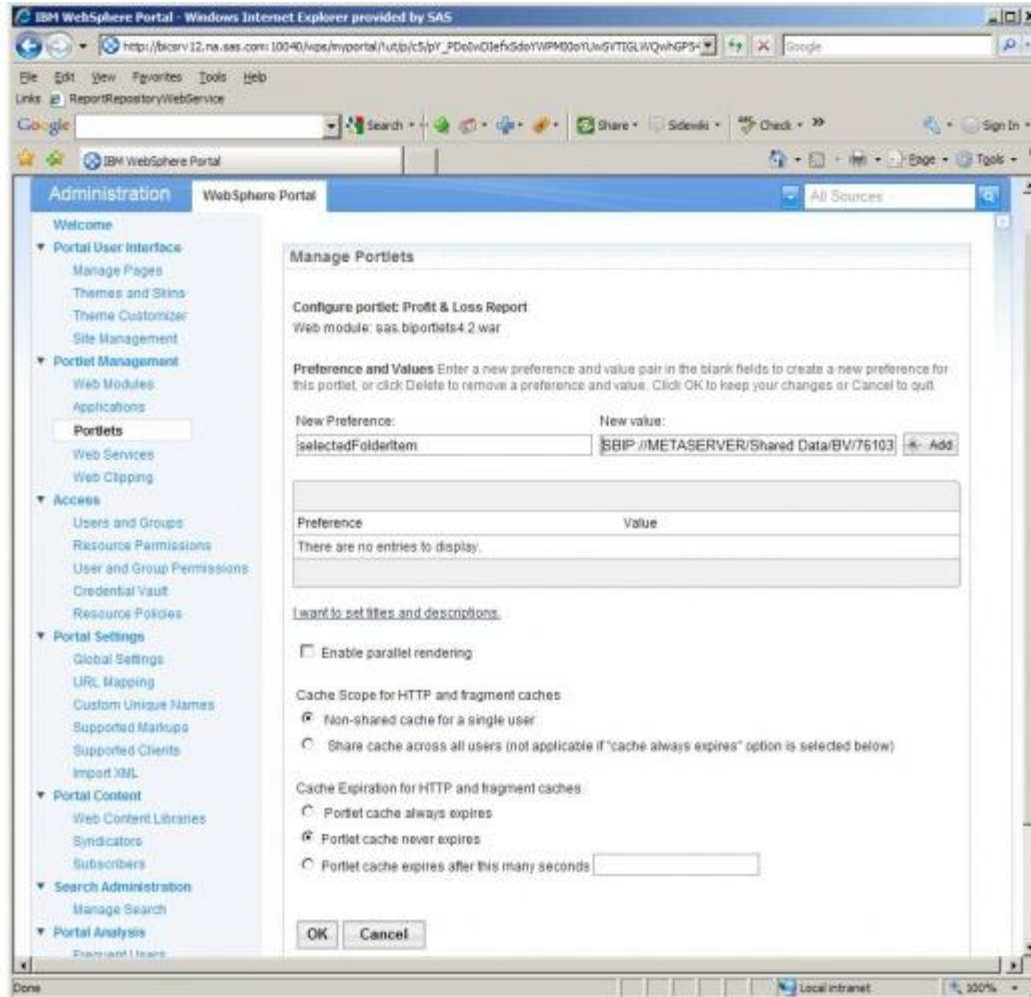
The SBIP URL is in the form:

```
SBIP://METASERVER/Shared Data/BV/BV Solutions/BV Revenue_Cube.srx
SBIP://METASERVER/Shared Data/BV/7610322350/7610322350_report.srx
```

For a stored process, it is in a form similar to:

```
SBIP://METASERVER/Products/SAS Intelligence Platform/Samples/Sample: Hello
World(StoredProcess)
```

1. Go back to the **Profit & Loss Report** portlet, and click on the Configure icon on the far right.
2. For the **New Preference** value, select **selectedFolderItem**. For the **New value** field, enter the SBIP URL.



3. Click **OK**.
4. Repeat this process for the Balance Sheet Report portlet.
5. Create a page named **Financial Statements Summary**. Share the page.
6. Add the configured **Profit & Loss Report** portlet and the **Balance Sheet Report** portlet to this page.
7. Log off as SASPortalAdmin. Log in as WRSViewUser or WRSCreateUser. You should get a **Financial Statements Summary** page with pre-loaded content for both of the portlets, and the page should look like the following display. Notice that you only get the page, and you cannot change, modify, or personalize the content. You can launch a live Web report in WRS or refresh the report.

The screenshot shows a web browser window displaying an IBM WebSphere Portal. The main content area is divided into two sections: 'Profit & Loss Report' and 'Balance Sheet Report'. The 'Profit & Loss Report' table lists sales data for various countries and divisions, with columns for Actual Sales, Country, Division, Month, Predicted Sales, and Product Type. The 'Balance Sheet Report' table shows financial data for the year 2009, with columns for Year, Invoicing Backlog, and Organization Level2. The portal interface includes navigation tabs like 'Home', 'Getting Started', and 'Web 2.0 Introduction'.

PORTLET PREFERENCE NAME VALUE PAIRS EXAMPLES

SAS REPORT PORTLET

```
selectedFolder:
SBIP://METASERVER/Products/SAS Intelligence Platform/Samples(Folder)
```

```
selectedFolderItem:
SBIP://METASERVER/SAS Projects/Portal/IDP 5 Sections.srx(Report)
```

SAS STORED PROCESS PORTLET

```
prompt_<prompt_name>:
String (prompt value)
promptNames:
Array of Strings (prompt names)
```

```
selectedFolder:
SBIP://METASERVER/Products/SAS Intelligence Platform/Samples(Folder)
```

```
selectedFolderItem:
SBIP://METASERVER/Products/SAS Intelligence/Samples/Sample: European Demographic
Data(StoredProcess)
```

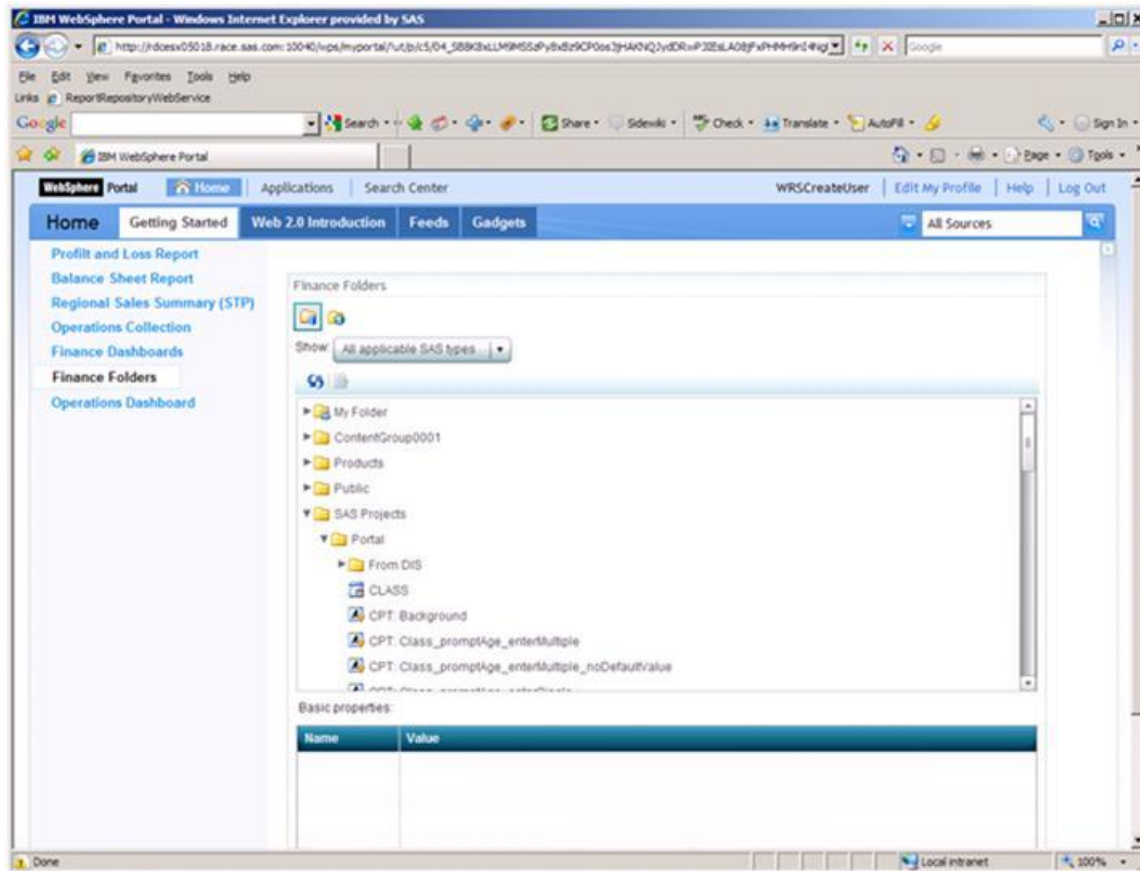
SAS COLLECTIONS PORTLET

```
packageSortOrder:
String (like "ascending")
```

```
showLocation:  
String (like "true")  
  
showPackage:  
String (like "true")  
  
showDescription:  
String (like "true")  
  
collectionItems:  
Array (containing MetadataInterface objects)  
  
collectionSelectedContentTypes:  
Array of Strings (like "Link")
```

Here is an example on how to pass multiple collection items as preferences. The property name to use is **collectionItems**, and the property should be defined as follows:

```
SBIP://METASERVER/SAS Projects/Portal/IDP Prdsale Cube Map(InformationMap)  
SBIP://METASERVER/SAS Projects/Portal/IDP Student Information Cube  
Map(InformationMap)  
SBIP://METASERVER/SAS Projects/Portal/IDP Class Relational Report.srx(Report)  
SBIP://METASERVER/SAS Projects/Portal/CPT: Hello World (no prompt)(StoredProcess)  
SBIP://METASERVER/System/Publishing/Channels/Parent(Channel)  
SBIP://WebDAV/public2/pictures/yawningSquirrel.JPG(Content)  
SBIP://Portal/IdentityGroup/Portal Application Tree/PUBLIC Permissions Tree/PUBLIC  
Link0001(Link)  
SBIP://Portal/IdentityGroup/Portal Application Tree/PUBLIC Permissions Tree/PUBLIC  
SyndicationChannel0001(ContentChannel)  
SBIP://Portal/IdentityGroup/Portal Application Tree/PUBLIC Permissions Tree/PUBLIC  
WebReportStudio0000(WebApplication)
```



SAS NAVIGATOR PORTLET

```
selectedFolder:
String (like 'SBIP://METASERVER/Products/SAS Intelligence
Platform/Samples(Folder)')
```

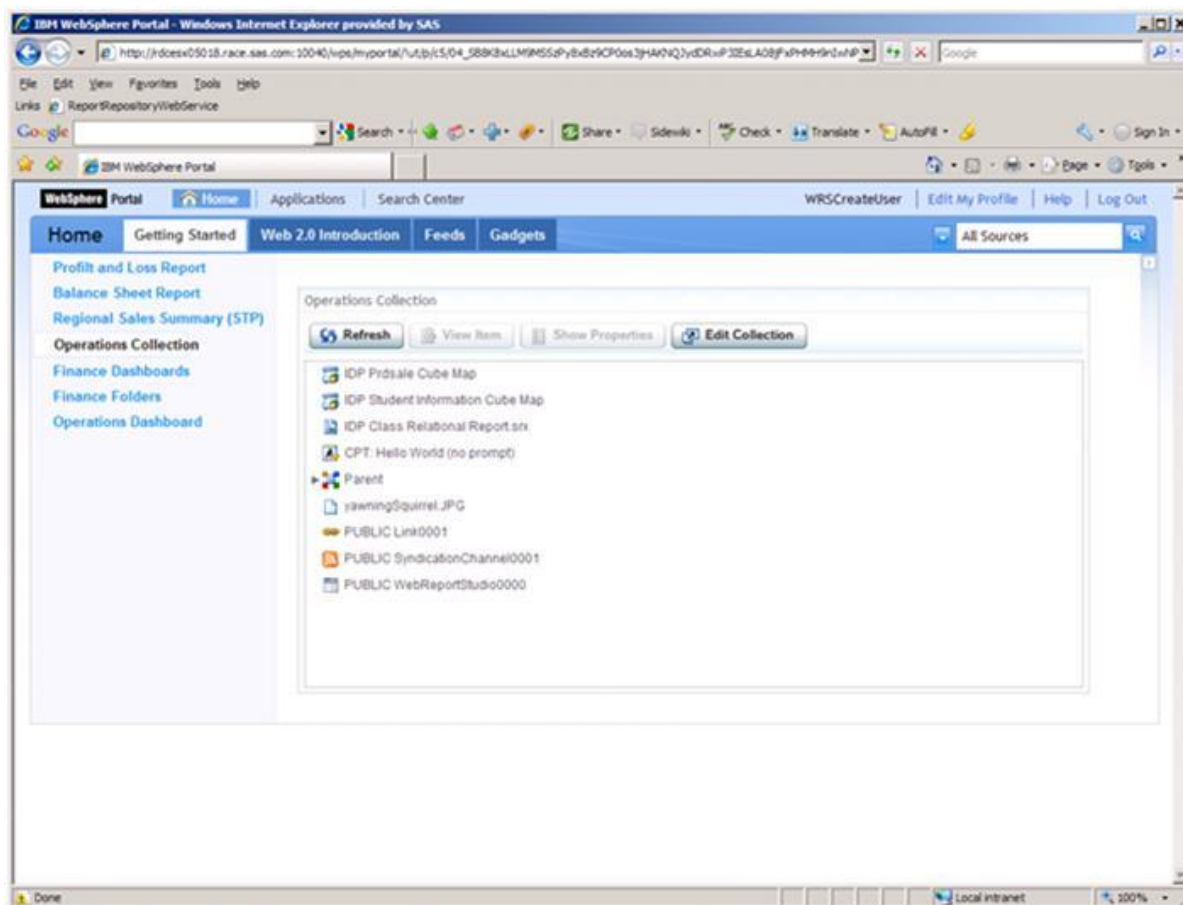
```
selectedFolderItem:
String (like 'SBIP://METASERVER/Products/SAS Intelligence Platform/Samples/Sample:
European Demographic Data(StoredProcess)')
```

It is possible to configure the navigator portlet to just display reports or stored Processes or whatever type you want the folder to display. The name and value pair for the navigator that allows you to set the content types to display is SMART_OBJECT_TYPE.

SMART_OBJECT_TYPE values are InformationMap, Report, StoredProcess, Package, and Content.

To display just reports, the value is:

```
SMART_OBJECT_TYPE=Report
```



SAS DASHBOARD PORTLET

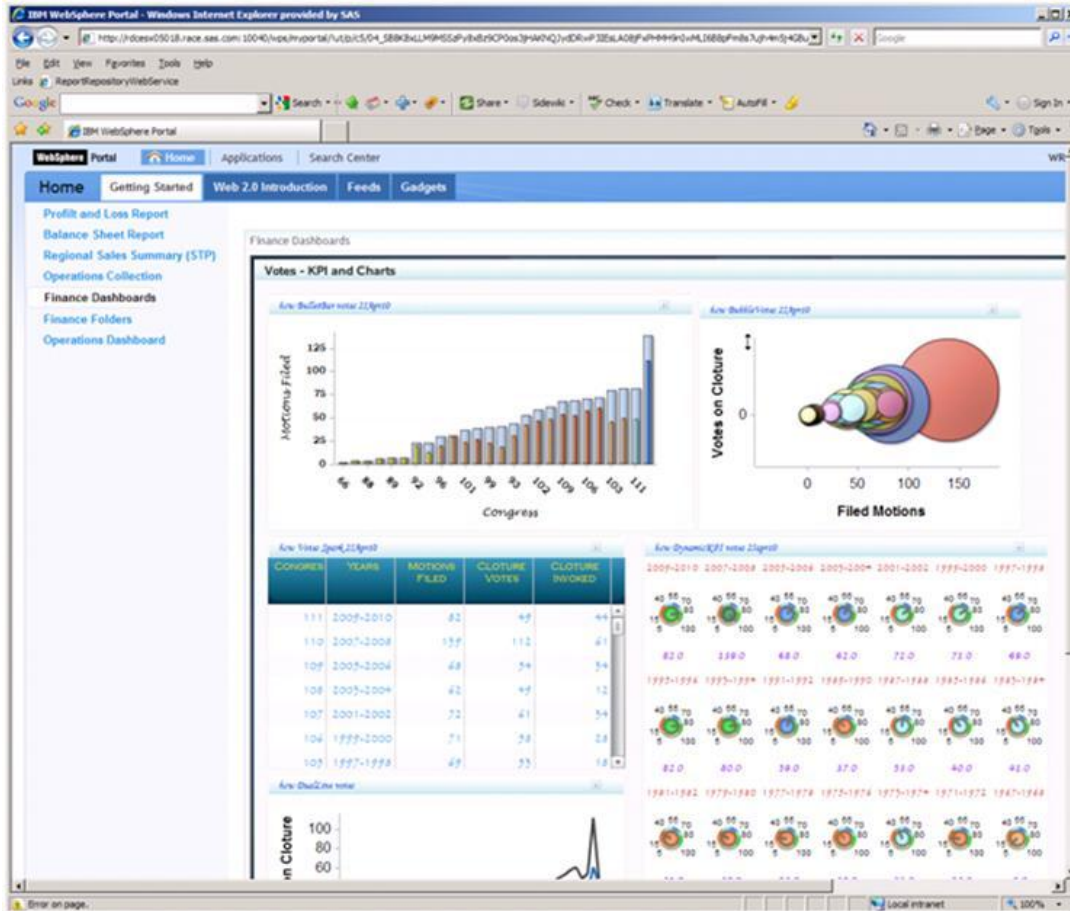
selectedDashboard:

SBIP://METASERVER/Products/BI Dashboard 4.3/DashboardConfigs/KPI/Votes - KPI and Charts.dcx (Dashboard)

Width:
1000

Height:
1200

autoRefreshInterval:
string



SAS DIAGNOSTICS PORTLET

portletTitle:
String (like "My stp portlet")

portletHeight:
String (like "350")

GENERAL PORTLET SETTINGS FOR ALL PORTLETS

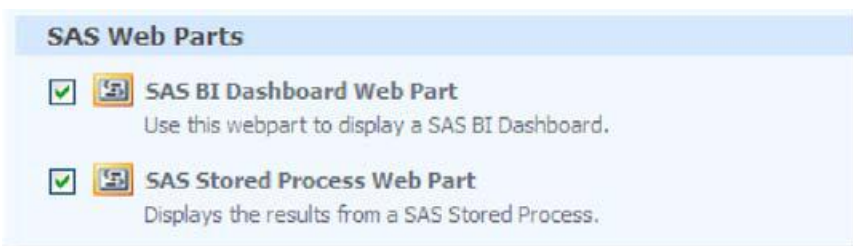
portletTitle:
String (like "My stp portlet")

portletHeight:
String (like "350")

INTEGRATING SAS BI WITH MICROSOFT SHAREPOINT

Another approach to integration is to pull content from SAS into a SharePoint page. Placing SAS content on a SharePoint page can be accomplished by placing a Page Viewer Web Part on the page, and configuring it to pull an HTML page from a SAS stored process. (A Web part is a building block that enables you to place content and control the content's appearance on a SharePoint page.)

In addition to the standard Page Viewer Web Part that is available in SharePoint, SAS provides two custom Web parts—the SAS BI Dashboard Web Part and the SAS Stored Process Web Part. These Web parts enable you to pull SAS content into a SharePoint page. Both Web parts can be configured to include their corresponding content types on a SharePoint page.



Once installed, the SAS Web Parts are listed in the Web Parts Gallery, and they can be added to any Web parts page.

The SAS Web Parts are configured with the SharePoint site's web.config file, which specifies a default connection to the SAS® BI Server. Because these Web parts are pulling SAS content in real time, the user is seeing up-to-date reports, not reports that were created at some point in the past. This integration approach works well for content that be generated quickly.

The SAS Web Parts also handle authentication for the user. When a user opens a SharePoint page, the user is authenticated by SharePoint. (In Internet Explorer, this authentication typically occurs through Windows Integrated Authentication. In other browsers, the user is sometimes prompted for a user name and password.) Because the SharePoint site has verified the user's identity, the SAS Web Parts use that identity to authenticate with the SAS server.

For more information, see *Introduction to SAS Web Parts for Microsoft SharePoint* at <http://support.sas.com/documentation/cdl/en/biwaag/63149/HTML/default/viewer.htm#a003369115.htm>.

REFERENCES

Sun Microsystems, Inc. 2003. "Introduction to JSR 168—The Java Portlet Specification." <http://www.sas.com/apps/whitepapers/whitepaper.jsp>.

IBM WebSphere Portal 6.1.0. IBM.

SAS Institute Inc. 2010. *SAS® 9.2 Intelligence Platform: Web Application Administration Guide, Fourth Edition*. Cary, NC: SAS Institute Inc.

Henderson, David R., and Sean Alexandre. 2009. "Integrating SAS® Business Intelligence with Microsoft SharePoint." *Proceedings of the SAS Global Forum 2009 Conference*. Cary, NC: SAS Institute Inc. Available at <http://support.sas.com/resources/papers/proceedings09/390-2009.pdf>.

ACKNOWLEDGMENTS

The authors would like to thank Scott Sweetland (SAS R&D) for his contributions to this paper.

ABOUT THE AUTHORS

Our integration with third-party portals is an ongoing effort at SAS. If you have questions or comments about topics covered in this paper or other areas of integration, we would love to hear from you. Your comments and questions are valued and encouraged.

Meera Venkataramani
BI Development Support Consultant
SAS R&D
Meera.Venkataramani@sas.com

Sam Atassi
Technical Architect
SAS Professional Services and Delivery
Sam.Atassi@sas.com

Gaurav Agrawal
Principal Technology Architect—SAS Practice
GCE Solutions Inc, Alpharetta, GA
gaurav_agrawal@yahoo.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.