Design and Implementation of Operational Dashboards on SAS Information Delivery Portal

Grace Jiang, Jeremy Terbush
RCI, Carmel, IN

ABSTRACT

This paper demonstrates examples of the design and implementation of operational dashboards on the SAS information delivery portal at RCI (Resort Condominiums International). Steps detailed include processing of data from different sources using SAS ETL tools, outputting a variety of stylized formats with SAS ODS, publishing packages to the SAS WEBDAV server and customizing portal pages to accommodate different end users. By employing these steps, information is organized to be shared across the company and supports effective decision making on the SAS BI (Business Intelligence) platform. Automated reports, graphs and stored processes are accessible to the whole company via a single website. This paper also provides some practical tips to create html files in Microsoft Visio, publish these non-SAS contents to the SAS WEBDAV server, and link them with SAS contents through the SAS portal.

INTRODUCTION

The SAS information delivery portal serves as a focal point for the revenue management operations team in RCI to access information they need. Portal dashboards present intuitive and graphical views of organization's data to decision makers. It improves the "time-to-market" circulation of the reports. As a centralized starting point for further investigation, it links reports and graphs so that users have the capability to "drill-down" into more detailed reports. It also helps high level and busy executives to monitor the performances of the organization and analysts. *Please Note: All data shown are fake to ensure confidentiality.*

SAS BUSINESS INTELLIGENCE PLATFORM

RCI uses SAS ETL studio to build data mart and data warehouse from multiple systems: mainly IBM mainframe for weeks data and Sun UNIX system for points data. All information assets are stored centrally in the SAS Metadata Repository.

SAS portal server and WebDAV server act together as RCI's self-service reporting system. Figure1 shows the architecture of RCI's business intelligence platform. Web-based Distributed Authoring and Versioning (WebDAV) is a standardized set of extensions to the HTTP protocol that allows users to collaboratively edit and manage files on remote Web servers. RCI uses the WebDAV-based Xythos WebFile Server (WFS) as the file management platform. Users can publish portal contents, such as documents, reports, and images to a WebDAV repository and can view packages that have been published to WebDAV. Basically, WebDAV is acting as a report warehouse. SAS information delivery portal can display these WebDAV contents in an organized fashion. OLAP cubes, information maps, results of SAS Stored Processes, and other information assets can also be accessed via SAS portal. Users can view reports and graphs, launch store process, explore information maps and cubes interactively, and perform further personalization to meet individual needs.
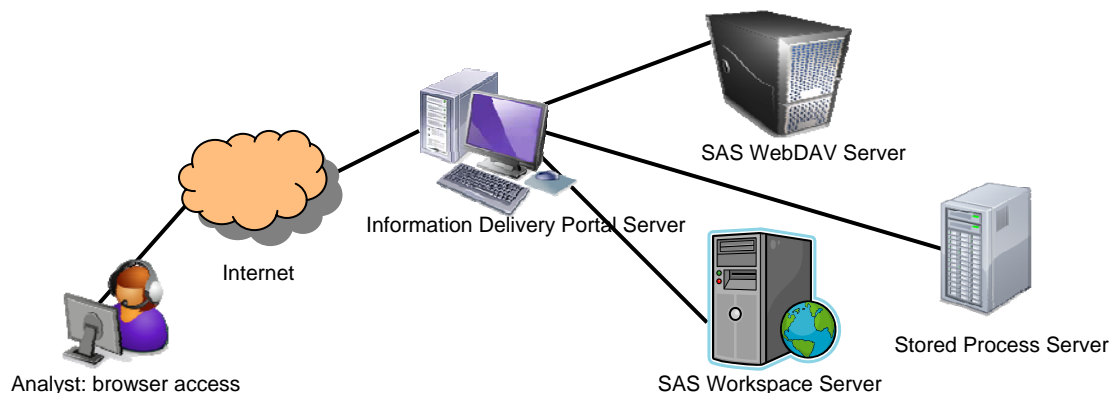

Figure1: business intelligence architecture

GENERATING OPERATIONAL DASHBOARDS

Batch jobs are scheduled to automatically generate and update dashboards files stored in the WebDAV server. The frequency can be hourly, daily or weekly, which depends on how often the information needs to be updated. End users have access to refreshed information through SAS portal. Steps to generate a WebDAV content portal page include:

- Publish contents to WebDAV server
- Add a WebDAV content portlet to a page
- Enter WebDAV content location
- Share the page to group

PUBLISH SAS OUTPUT TO WEBDAV SERVER

The SAS Output Delivery System (ODS) is capable of producing all procedural output in a formatted style. The output can be HTML, RTF, PCL, PS, XML, Latex or PDF format. HTML pages are the most commonly format published to WebDAV server, since they can be linked from here to there. Following is the code using store process to publish an html package to SAS WebDAV server:

```
%global _PARENT_URL _COLLECTION_URL _IF_EXISTS _HTTP_USER _HTTP_PASSWORD;
%global _EXPIRATION_DATETIME _RESULT _ABSTRACT _DESCRIPTION;
%global _GOPT_DEVICE;
%global _ODSOPTIONS _ODSDEST _ODSSTYLE;
%let _PARENT_URL=;
%let _IF_EXISTS= updateany;
%let _HTTP_USER= usrid;
%let _HTTP_PASSWORD= pwd;
%let _EXPIRATION_DATETIME=%sysfunc(inputn(31DEC2007:12:40:40.0, datetime20.));
%let _RESULT= PACKAGE_TO_WEBDAV;
%let _ABSTRACT= Report;
%let _DESCRIPTION= Report Desc;
%let _GOPT_DEVICE = activex;
%let _ODSDEST = html;
%let _ODSSTYLE = seaside;

%let path= &webdavserver_addr/&destfolder;
%let _COLLECTION_URL= &path;
%let _ODSOPTIONS=%str(file="myfile.html");

%stpbegin;
    /*ODS CODE*/
%stpend;
```
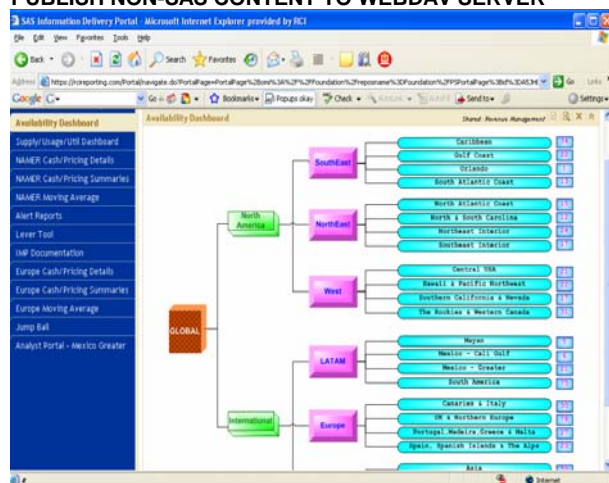
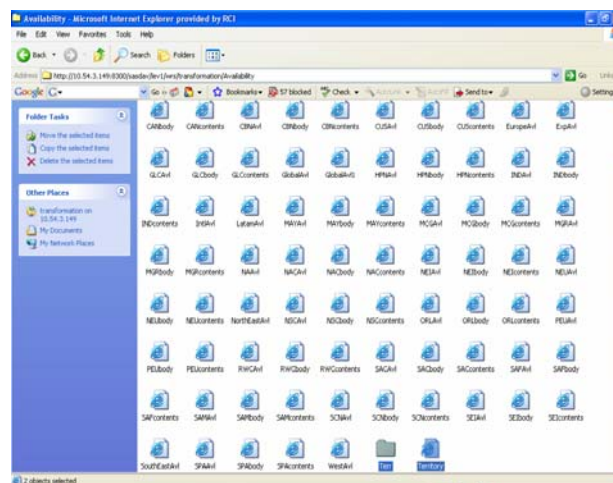### PUBLISH NON-SAS CONTENT TO WEBDAV SERVER

Figure2: territory tree

Figure3: WebDAV contents for available dashboard

When using option _IF_EXISTS=updateany, each file has to be published through SAS. However, more often than less, non SAS output contents, such like images, company logo, spread sheets, and even documents need to be put under the same folder as the SAS outputs. For example, a territory tree (Figure2) serves as an entry to available dashboard. The map was created in Microsoft Visio and saved as terr.html file. And there is also a folder "territory" which contains components of the html page (Figure3), such as .gif files. In this case, status 404 (not found) exceptions will happen if these contents are manually uploaded to WebDAV server. The error information in log would be:

ERROR: WEBDAV transport engine encountered errors while publishing the package.

ERROR: The result package could not be created. The error encountered was: ERROR: Web server could not complete requested operation, status = 404.

How to fix it? A trick is to publish a dummy file with the <u>same name</u> as the one you want to store in WebDAV. For the html file, we will publish a dummy .html carrying the same name through SAS, and then replace it with the real one. Code to create a blank dummy file:

```
%let _ODSOPTIONS=%str(file="terr.html");

%stpbegin;
    /*Blank since it is a dummy file*/
%stpend;
```

For the folder, a good thing is that WEBDAV engine doesn't check files under subdirectory. So we don't have to publish dummy files under folder "territory" one by one. Code to create a blank dummy folder:

```
%let _COLLECTION_URL=&path/territory;
%let _ODSOPTIONS=%str(file="dummy.html");

%stpbegin;
    /*Blank since it is a dummy folder*/
%stpend;
```

Replace these dummy files and folder with the real one. And exceptions will not emerge again.

PAGE AND PORTLET
Portlet is the visual component model for the portal. Layout is designed in a page and these portlets are embedded into the grids. Each page is represented by a tab. In preference setting under option menu, you can configure the navigation bar to be either vertically or horizontally. There is a SAS Users Group paper that provides step by step tutorial on SAS information delivery portal:
http://www2.sas.com/proceedings/forum2007/214-2007.pdf

Figure2 a typical personal portal page for an analyst, which has vertical page navigation bar.

HOW PORTAL PAGES ARE CREATED
The portal page looks like an ordinary html page. But there doesn't actually exits such a static page on the portal server. Portal keeps the metadata only: page name, page properties, links to portlets, and key words, etc. For example, the page name is an entry in a metadata table. The key words are entries in another metadata table. The links are entries in 3rd metadata tables. Portlets sequence and format are entry in 4th table. All tables are linked by keys. Java creates dynamic pages of the fly based on the metadata.

This method makes it difficult to replicate a page. RCI is managing inventories based on destinations. Usually pages are created at the territory level for each analyst. Basically page information is pretty much the same except the destination name in the portlet link. For example, we have 20 territories. Page A contains data information about territory A, whose link is located on &webdav_srv/XXXX_terrA/. Page B contains data information about territory B, whose link is located on &webdav_srv/XXXX_terrB. Yet they have the same file name. For a static html page, the only thing you need to do is copy the source code and replace the folder name. However, there is no way to copy a page template or clone a page in portal. What you have to do is creating the 20 portlets one by one, which cost a lot of labor and time.

CENTRALIZED OR DISTRIBUTED

In figure4, Pies, bars, reports, graphs are put together into one page by using proc greplay in SAS code, the whole page is saved as one .html file in WebDAV server. We call it centralized method. In figure5, each report or graph is saved as an individual file. We call it distributed method. In method1, Users only need to create one portlet and link. The layout is flexible for design and space efficient. But the contents and layout are designed in coding time and the code has to be modified if any change needed. And especially, reports have to be reprinted as graph with annotation defined. In method2, contents and layout can be arranged when user creating portlets. End users also have the choices to customize their pages to meet different needs. For example, territory A might be concerning about the confirmation recently so the analyst puts confirmation portlets in the first row grids. While territory B focuses more on deposit information, so the analyst puts deposit information to the most important position.
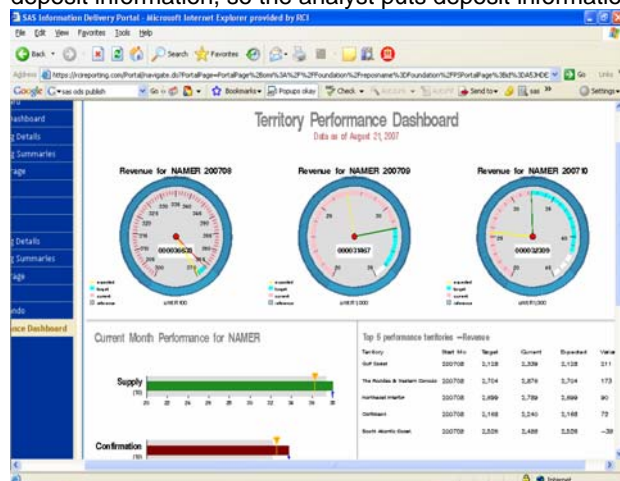

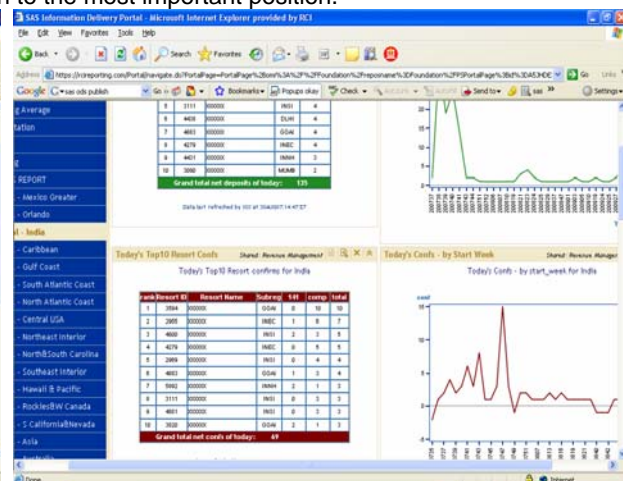Figure4: centralized dashboard   Figure5: distributed dashboard

There are 3 types of SAS users on Portal. Sasdemo, usually testing account, is capable of doing almost anything. Saswebadm, the web administrator, is able to set up new pages and share it with the entire company or one group. Sasguest can only manipulate personal pages.

In RCI, Saswebadm uses centralized method to organize information available for manager or higher level decision makers. While for analysts, they are doing detailed operations and need more flexible information. Saswebadm creates portlets and provides a sheet of keywords. Analysts can search for portlets or pages based on keywords and add those contents to their personal portal.

MORE EXAMPLES

By providing advanced data visualizations, dashboards make an organization's experts more efficient to get on time information and promptly make decisions. In figure5, color is used to identify key measurements: green for deposit, maroon for confirmation, and navy for availability. Users can find information they need promptly. Here are more examples showing the power of SAS BI:
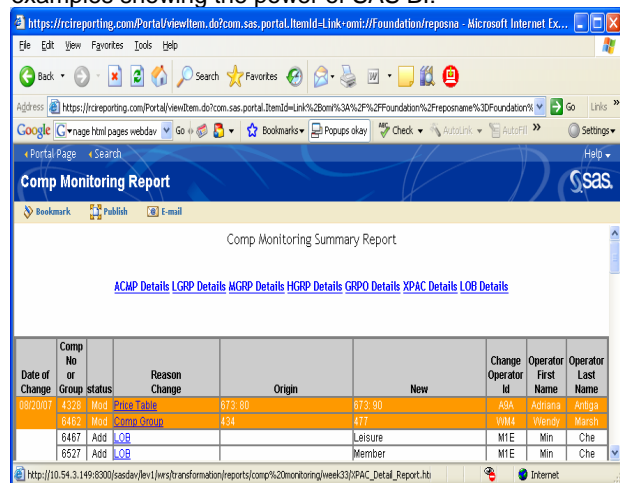

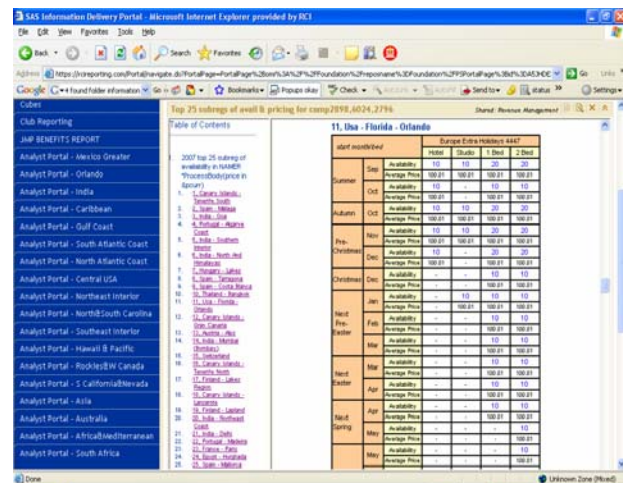Figure6: linkable reports   Figure7: table of contents navigation bar

Figure6 shows linkable summary report and detailed reports are all linkable. It is used to trace and monitor all the changes made by analyst.

Figure7 is an example to use proc template and report style and create table of contents in navigation bar.

There are varieties of portlets types in Figure8. Collection portlet is a place to list multimedia items. It can be an entry to launch store process, OLAP cubes, information maps, webreport studio reports. Users can drill up and down to dig detailed information with harness of these SAS BI tools. At the bottom of Figure8 is an example of WebDAV repository navigator. Figure9 shows the navigation of cube in the portal.
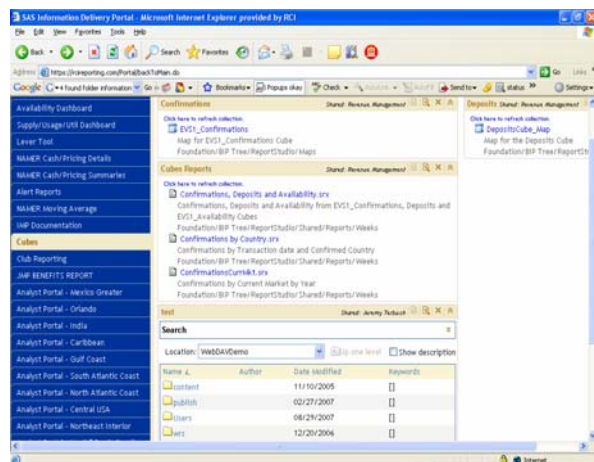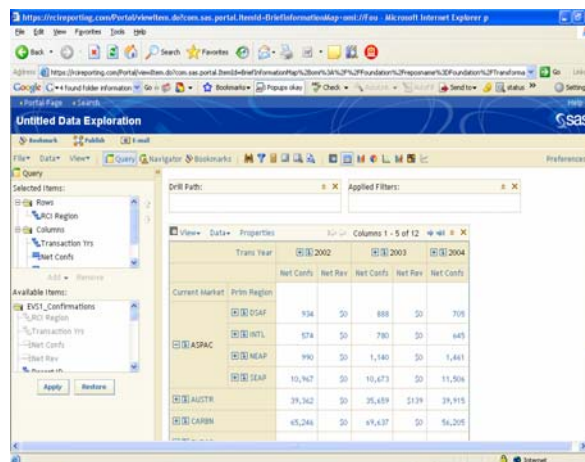

Figure8: collection portlet   *Figure9*: cube navigation in portal

CONCLUSION
In the next version of SAS portal, we expect following capabilities on the SAS portal: there is functionality to clone page and create pages dynamically; and users are able to design the page layout like in spreadsheet, having more flexibility to merge or split grids for portlets.

RECOMMENDED READINGS

http://www2.sas.com/proceedings/forum2007/214-2007.pdf

http://support.sas.com/ctx/samples/index.jsp?sxf=s&mdType=grouping&c1=dashboards&grouping=dashboards

Few, Stephen (2006), *Information Dashboard Design*, Sebastopol, CA. O'Reilly Media, Inc

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

Grace Jiang   Jeremy Terbush
Resort Condominiums International   Resort Condominiums International
9998 Michigan Rd   9998 Michigan Rd
Carmel, IN 47906   Carmel, IN 47906
Phone (317)805-9839   Phone (317)805-8219
FAX (317) 805-8173   FAX (317) 805-8173
Grace.jiang@ric.com   Jeremy.terbush@rci.com