

Paper 392-2008

Leveraging SAS Enterprise BI Server for Operational Reporting Workflow

Paul E. Christenson, Blue Cross and Blue Shield of Minnesota, Eagan, MN
Chad Flentje, Blue Cross Blue and Shield of Minnesota, Eagan, MN

ABSTRACT

Blue Cross Blue Shield of Minnesota sends out 2.5 million first class mailings of reports, forms, checks and letters to customers on a monthly basis. The print streams that create these mailings can be thousands of pages long, with different layouts on separate pages. For example, each user can have a different number of pages, one of the pages may require a different kind of paper, and all the pages need to be automatically inserted in an envelope. In addition, all mailed content must be archived for 7 years.

This type of high detail, high volume operational reporting is standard in most customer-service oriented companies. The desire to minimize the number of systems used in an organization is also a standard desire of enterprise architecture. This paper will explore some of the opportunities and pitfalls of integrating SAS 9.1.3 (SP4) Enterprise BI Server, Xythos WebDAV v2.2 and Platform Scheduler v3.0 into an enterprise operational reporting framework.

INTRODUCTION

A healthcare insurance company is a document-generating machine; printing and mailing checks, forms, invoices, and activity reports on a daily basis. A part of providing cost-effective health care is optimizing the usage of system resources and personnel to minimize overhead costs. BCBSM has an Enterprise Architecture governance board that oversees all IT development projects. The board has a number of standards and guidelines that projects have to meet in order to be funded. Several guideline questions apply when developing operational reports:

- **Buy vs Build:** In order to minimize development and maintenance costs for new systems it is frequently better to buy a new application and modify the business processes to fit. The question is how much can the business processes be changed without losing a competitive advantage?
- **Leveraging Existing Assets:** Large organizations standardize on applications platforms to minimize server and support costs. The challenge is whether an existing asset has the capacity to support new business needs?
- **Vendor Stability:** New technology and companies are continually coming and going. You may find a small company that has a perfect fit for an application but will they be around in 5 years?

The SAS Enterprise BI Server framework is uniquely suited to meet these guidelines.

- SAS BI is a framework of development tools and supporting infrastructure. Components like the Information Delivery Portal, Metadata server, WebDAV and Enterprise Guide can be reused from one application to another. Customization for a specific application can be done through Base SAS coding and JAVA.
- SAS licensing is by physical server type not number of CPU's or users. At BCBSM with have our Test, Integration and Production environments running logical partitions on the same UNIX server, with our Quality environment running offsite for disaster recovery. We can divide up this shared environment and manage its capacity in a controlled process.
- SAS is one of the approved vendors for BCBSM. It is a large, stable company with minimal risk of disappearing in 10 years.

Based on the guidelines, the Enterprise Architecture group at BCBSM has tasked the Analytical Tools Team at BCBSM with developing an operational reporting framework, using SAS, for an in-flight business application.

OPERATIONAL REPORTING REQUIREMENTS

BCBSM has multiple applications that process transactional data for Members, Providers and Agencies. These applications are unique, but they share a need for several common business reporting needs:

- Predefined reports with different parameter ranges are generated for multiple users.
- Reports at the member and group level need to be mailed out periodically.
- Reports need to be archived for future retrieval.

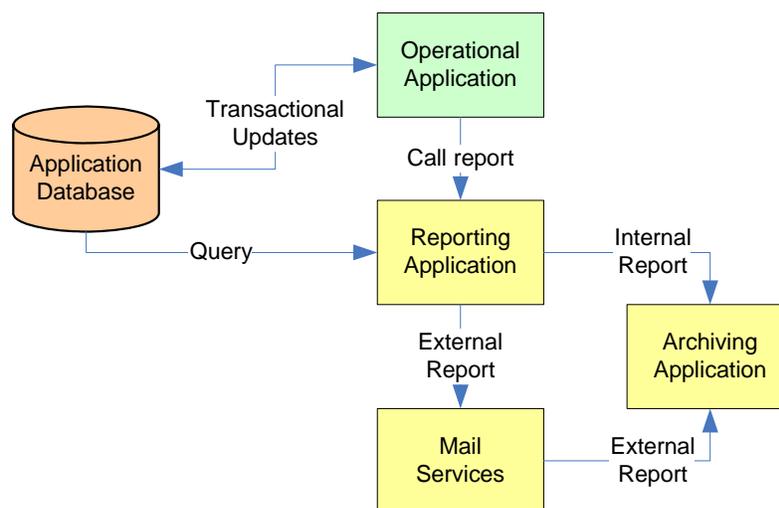


Figure 1. System Interaction

The diagram in Figure 1 shows the interaction between these high-level business needs. The following is a more detailed list of common business and technical requirements in operational reporting:

Support Simple User Interface for running and retrieving reports

- Customer service staff and claims processing clerks need to reporting as a part of their daily workflow.
- Different user groups will be able to access different sets of reports.

Support Role-based access

- In order to keep security manageable, authorization will be based on group roles that the user is a member of and will be managed by a separate security group in a centralized Enterprise directory.

Support for reports with multiple file output types

- The Adobe PDF format has become a document standard in most organizations. It provides the combination of good print quality and support by common web interfaces.
- Reports will also need to be generated for Excel and CSV output types.
- To simplify maintenance, a single code set should support all required format types.

Enable cost-effective development with reusable report code.

- Use as many “out of the box” components as possible. For the modules that do need to be developed, use best-practice development guidelines.
- Too often, reporting starts growing “organically” and rapidly become difficult to manage.
- Simplify navigation by grouping common reports together and enable them with a parameter-driven interface.
- Choose a reporting framework that will support an open API so it can be called by other systems or custom interfaces.

Support for reports with variable run times.

- Because report run time can vary greatly based on format complexity, volume of data queried and number of users on the database, reports will be run in foreground mode, background mode or scheduled depending on the situation.
- If reports are run in background there will need to be a holding area for the finished reports that the user can access based on their security rights.
- Very large reports will need to be scheduled at night as a batch job that can be picked up by the user the next day.

Support for reports that are thousands of pages in length.

- Report runs for Explanation of Benefits mailings can be for 100,000 members or more.
- A PDF can be made up of multiple users with variable number of pages per user.
- SAS has a current limitation of 999 images in a given report output. This becomes an issue if you have corporate logos needing to be displayed on each page. A method for adding or manipulating images on an existing PDF will simplify report processing.

Support the formatting of a complex report with different pages having different layouts.

- Reports with multiple sections, form letters and tax documents may require a combination of Proc Report or Data Null Step reports.
- A method for interleaving two reports and grouping by a customer would be useful for streamlining process.

Enable Users to review reports before they are sent to Mail services or archived.

- A PDF document will need to be reviewed in temporary storage by authorized users. The user interface will need to recall the document for review, and then give the user the ability to approve for Mail services or recreate the document.
- The version of a document printed and sent to Mail Services is the version to be archived.

Support special control information on each page for automatic Envelope Inserters, or changing printer paper.

- Mail room Envelope Insertion machines fold printed pages and insert them into envelopes for mailing. A special code called an OMR code needs to be placed on each page so the inserter knows how many pages to insert and when the last page for a customer has been inserted.
- If there are more than 7 pages for a single envelope the pages need to be set aside since a different envelope will be used.

Support archiving of customer documents for up to 7 years.

- Document-retention rules at BCBSM require a 7 year retention period.
- Archived documents for a customer must be able to be retrievable by customer service while the customer is on the phone.
- Customer service will query for a section of a document for a single customer, but the document will be generated and archived for all customers.

Support multiple data sources

- Reports frequently need to pull from more than one data source and integrate the data into a single report.

OPERATIONAL REPORTING DESIGN**FUNCTIONAL REQUIREMENTS**

Besides business level requirements the following functional requirements are defined for operational reporting system.

- The system must allow the user to generate hard-copy reports (as per each report spec).
- The system must allow the user to generate electronic-copy reports (as per each report spec).
- The system must allow the user to re-access previously stored electronic-copy reports.
- The system must allow the user to generate hard-copy reports (as per each report spec) as part of the function step in the transactional application.
- The system must allow the user to manually inspect electronic-copy and reject it.
- The system must allow the user to manually inspect electronic-copy and approve it for printing.
- The system must allow the user to flex the run-time parameters (as per each report spec).
- The system must allow the user to delete or archive previously stored electronic-copy reports.
- The system must allow the user the ability to reproduce hard-copy from a previously stored electronic copy.
- The system must be able to automatically generate hard/soft-copy reports (using predefined parameters) based upon a temporal event.
- The system must allow the user the ability to specify a printer.

USE-CASE VIEW

At BCBSM the Unified Modeling Language (UML) is used in design and development stages. The following section is the UML artifact used in condensing requirements into architectural and technical designs. Figure 2 shows a high level mapping of these use cases and their interactions.

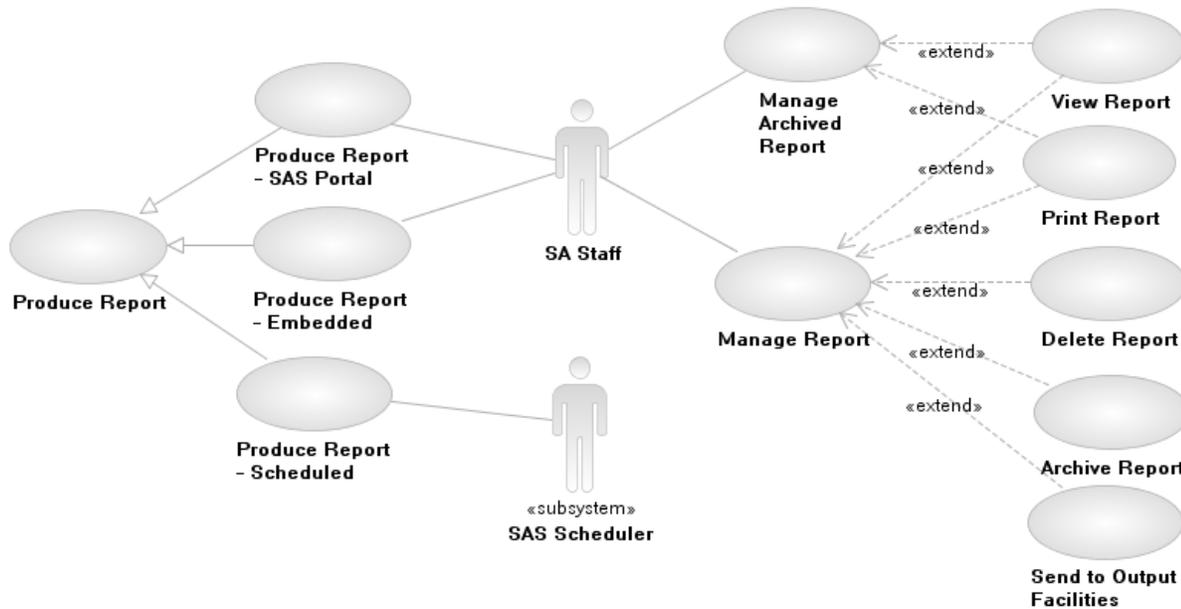


Figure 2. Use Case Context Diagram

Produce Report

The use case describes the three ways that a report can be produced.

- **SAS Portal:** Authorized users will need to manually generate pre-defined reports from the system. The Produce Report – SAS Portal use case will describe the sequence of events executed by the user to create reports.
- **Embedded:** Authorized users will need to generate a report as part of a processing function in the transactional application. The Produce Report – Embedded use case describes the sequence of events for generating 'embedded' reports.
- **Scheduled:** Some reports are generated on a scheduled basis. The Produce Report – Scheduled use case will describe the sequence of events for scheduling.

Manage Report

Authorized users will need to view, review and print generated reports. This use case describes the sequence of events as expected by the business to find, view, print, or delete a report that has already been produced.

Manage Archived Report

Authorized users will occasionally need to view and print an archived report. This use case describes the sequence of events as expected by the business; to find, to view, to print, or to delete a report that has already been produced. Note; Retrieval of content is outside the scope of this paper. Sending content to the archive server is in scope.

ACTIVITY DIAGRAMS

UML Activity Diagrams are used to map functional steps in the reporting workflow to components in the reporting framework. Figure 3 shows the various objects in Activity diagram:

- **Initial Activity:** The starting point of the diagram.
- **Activity Final:** The end point of the diagram. A diagram can have multiple end points.
- **Action:** The logical step(s) being taken in the diagram.
- **Send Signal:** Initializes a message to another activity.
- **Accept Signal:** Receives a message from or within a diagram.



Figure 3. Activity Diagram Notation

Produce Report – SAS Portal

The logical steps for generating reports through a custom remote portlet running in a local service on the SAS BI mid-tier is shown in Figure 4. Key actions executed in the portlet are:

- **Check/remove expired reports:** Reports are temporarily stored in WebDAV for immediate viewing and then moved to the permanent archive server. In order to minimize WebDAV storage, reports are tested for an expiration date and deleted from WebDAV when a user first logs in.
- **Enter report parameters:** When a report is selected, a custom parameter page is generated. The parameter page may contain a simple date-range selection or a combination of pick lists with search capabilities. In order to reduce development costs, support for cascading parameters was minimized.
- **Display generation progress:** The portlet has a queue-manager component that monitors and displays the status of a user's report.

The following actions are common to all stored processes regardless of calling mechanism:

- **Gather report data:** Depending on the report, this may be a multistage process with explicit SQL calls and heterogeneous queries.
- **Generate report and metadata:** Besides the report, a metadata or descriptor file is generated. This file contains instructions for the Archive server and Print Services server.
- **Signal user the report is complete:** When a report is finished running an e-mail message is sent to the user.
- **Stage report if Archive only:** A copy of report files is placed in a staging area. A separate, secure copy process moves the files to the Archive server.

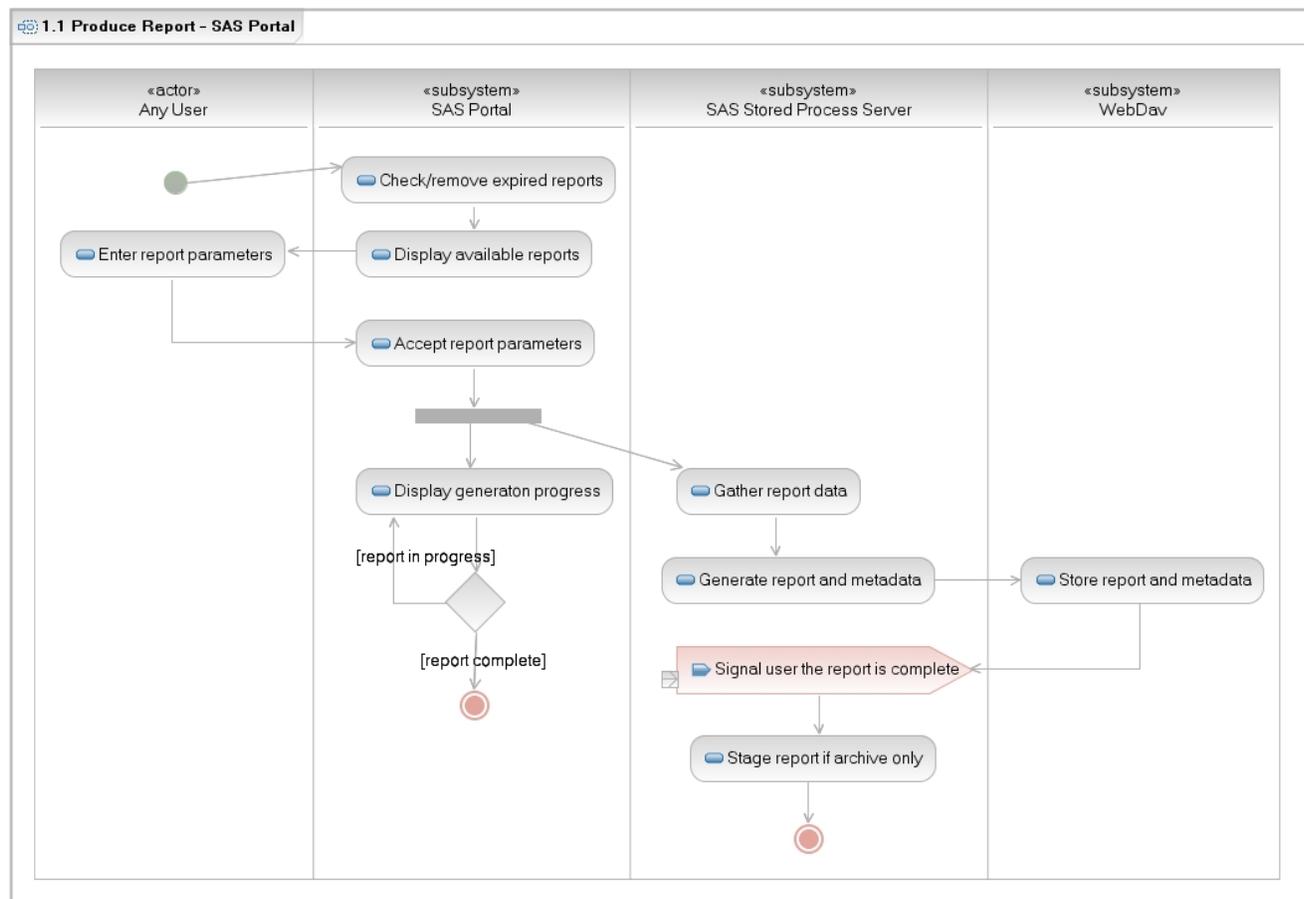


Figure 4. Produce Report – SAS Portal Activity Diagram

Produce Report – Embedded

The logical steps for generating a report from an external application are shown in Figure 5. After the external application stores the process results, it makes a remote service call to the SAS BI Mid-tier. It connects as an application id and passes all pertinent parameters in an XML stream. When the request is received, it launches the stored process following the common action steps.

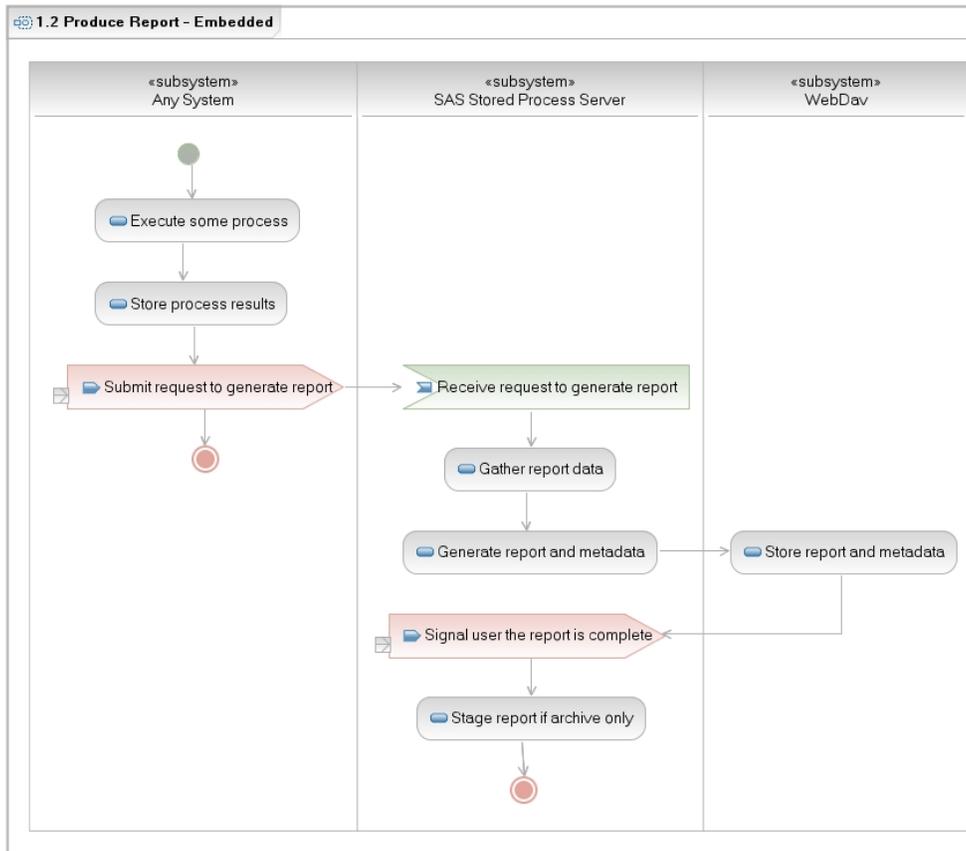


Figure 5. Produce Report – Embedded Activity Diagram

Produce Report - Scheduled

Running reports from the SAS Platform Scheduler is shown in Figure 6. The Platform scheduler cannot directly call a stored process but it can call a Data Step Batch server. The Batch wrapper code mimics the setup variables used by the stored process server and then calls the underlying SAS code that the STP metadata points to.

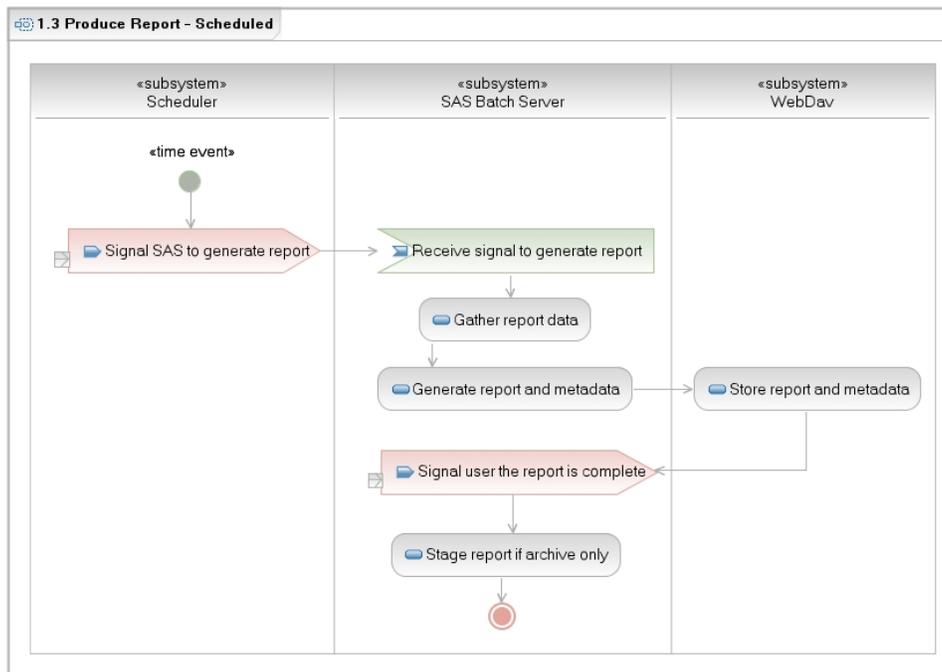


Figure 6. Produce Report – Scheduled Activity Diagram

Manage Report

Reports that are mailed out to members are sampled and inspected before being sent. Figure 7 shows the steps taken to review and approve these reports. Two key action steps are:

- **Display Available generated reports:** The user can select only the reports they are authorized to see. After the user has selected and reviewed the report they can approve it through the UI.
- **Send report and metadata to Content Management:** This message is initiated when the user accepts the report. It kicks off two processes, moving the files to the Print services staging area and removing the report from WebDAV so it cannot be resent.

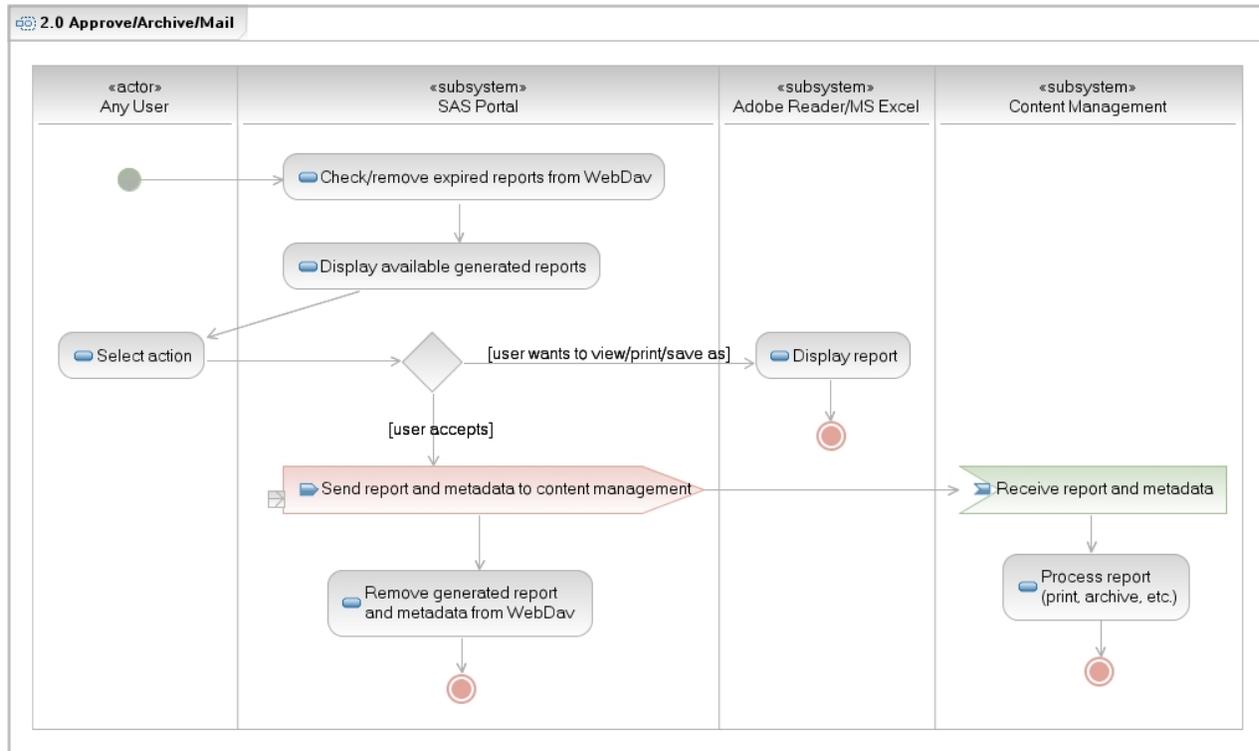


Figure 7. Manage Report Activity Diagram

Move to Content Management

Figure 8 show logical steps for moving files from the staging to Content Management archive server or print services server.

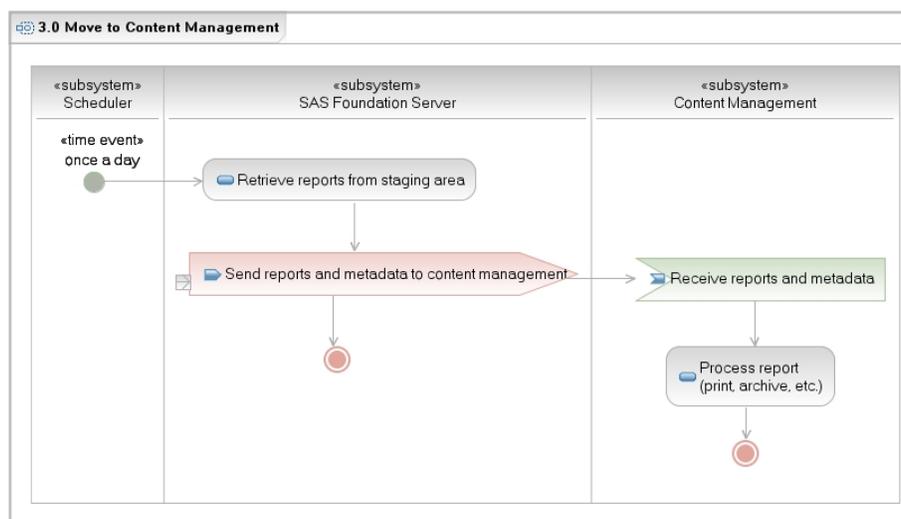


Figure 8. Move to Content Management Activity Diagram

OPERATIONAL REPORTING COMPONENTS

The following table maps the components and services used to fulfill the different process steps.

	Produce Report Portal	Produce Report Embedded	Product Report Scheduled	Manage Report	Move to Content Management
SAS Portal	✓			✓	
Mid-Tier Server		✓			
Metadata Server	✓	✓	✓	✓	✓
Stored Process	✓	✓			
Batch server			✓		✓
WebDAV	✓	✓	✓	✓	
Platform Scheduler			✓		✓
Base SAS and Staging Area	✓	✓			
Print Services					✓
Archive Services					✓

These components and their interactions make up the Operational Reporting Framework as shown in Figure 9. Yellow items are part of the Enterprise BI, Green items are the transaction applications and Enterprise services (not in scope of this paper). The items in Orange are the key third-party components not provided by SAS.

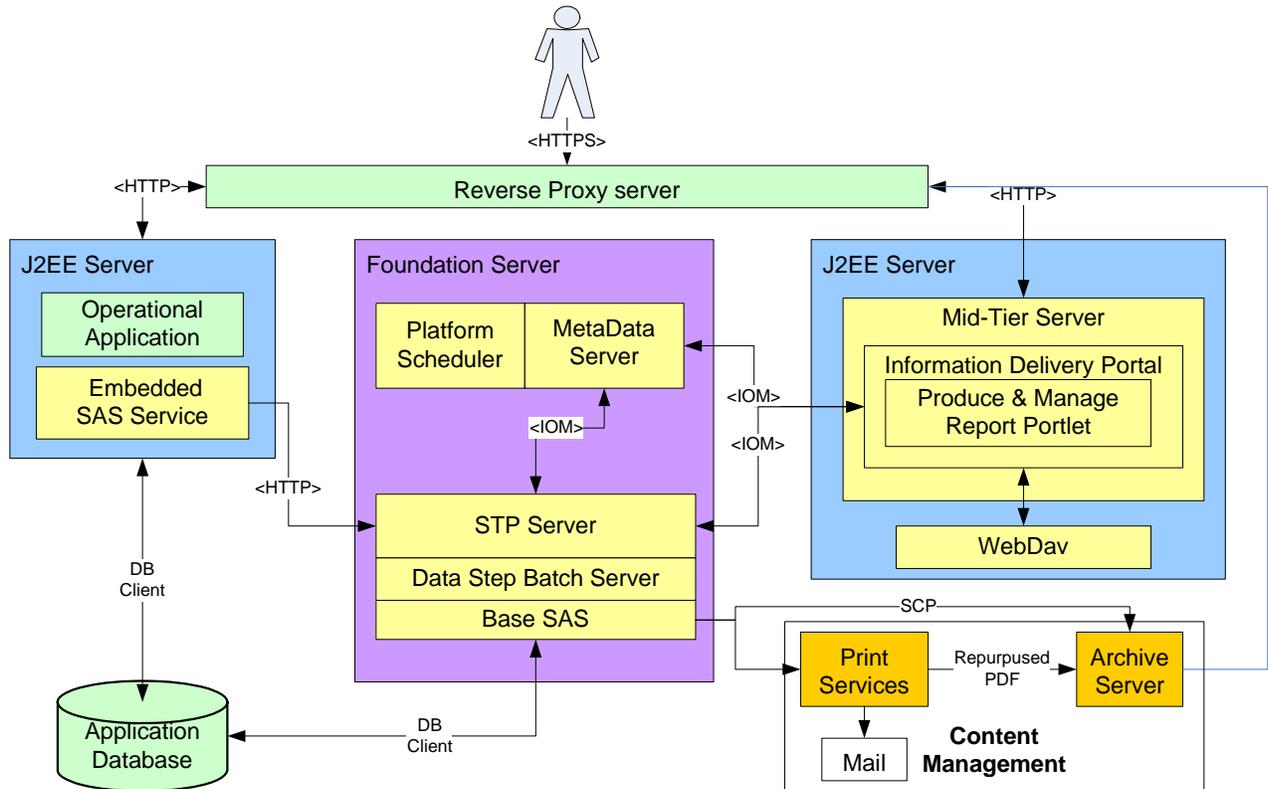


Figure 9. Operation Reporting Framework Overview

IMPLEMENTATION LESSONS LEARNED

The following section covers key issues, system limitations and design considerations that were learned in the process of developing this operational reporting framework.

SAS Portal

The Information delivery portal (IDP) does not meet the requirement as a simple user interface but it does have all the integration points for Trusted User authentication, SAS Metadata server and Stored Process servers. It also allows SAS administration to centralize management of reporting functionality for multiple transaction applications. By creating portlets that launch from within the IDP, interface requirements for a specific user group can be created and still leverage the integration provided by the IDP.

Produce Report Portlet

This portlet is written using Eclipse, Java 1.4.2 and the SAS API bundled with SAS 9.1.3 SP4. The portlet is written as a remote portlet using local SAS services with the following characteristics:

- The portlet enables the functionality covered in the Produce Report Portal diagram in Figure 4 and Figure 7.
- The portlet controls screen presentation and workflow logic. Any query logic or data-processing requests are executed with a stored process with an XML data exchange.
- The portlet is attached to a sticky page that open as the first page. The users do not see the IDP interface.
- The report definitions and parameters are stored in datasets. This is more efficient than querying stored process parameters out of metadata.
- The Portlet is divided into three pages controlled by tabs
 1. A Report select page with custom parameters pages per report. Parameter pages may contain paginated pick lists and cascading parameter lists.
 2. A Queue Status page that displays what order a user's particular report is scheduled to execute in, with the ability to cancel the report if the queues are full. The queue manager was custom built to manage calls to the Object Spawner to give a SAS Administrator an extra level of control in a shared environment.
 3. A Retrieve Report page that lets a user open and view any report that was run by themselves or any other user in their security group. If the report is flagged for review, an approve icon is placed next to the report name. When selected, it initializes the content being sent to Print Services.

Embedded Reports and SAS Mid-tier

Under some circumstances a user needs to launch a report as a part of a workflow within the transactional interface. The SAS JAVA API is added to the interface to create remote service calls to the SAS Mid-tier. The Mid-Tier authorizes the service call based on a specific application ID. Once authorized, the User ID, report parameters and stored process name are passed to the Stored Process server and the report is executed.

It is possible to return a PDF to the application but to enable later retrieval and archiving the stored process was configured to return content to WebDAV as a permanent package.

Stored Processes

There are several features of stored processes that need to be considered for operational reporting.

- Parameters can be passed into the stored process from the default IDP interface, from a customer Portlet, and from an external API. With modification the Stored Process code can be run directly from the Platform Scheduler. This allows the same code to be used in multiple ways and simplifies maintenance. A key advantage is reports can be tested independently of the user interface by configuring the stored process as transient or streaming packages.
- The default Transient Package or Permanent Page output type for stored processes allows the user to see content immediately or run in background mode and save content to WebDAV. Because all web applications run under a Reverse Proxy Server with an automatic 2 minute time out, reports will be launched in background mode.
- Once a stored process is started it can not be canceled or monitored.
- A stored process server has a limited number of open services at any given time. If you are running in a shared environment care should be given in determining what reports should be available to run during the day and what should be scheduled at night. This was the reason an interface specific queue manager was added to the Produce Report portlet.

Base SAS

Base SAS has a long history of report writing. Proc Report and NULL Data Step reports cover most reporting needs. There are a few issues to take note of when writing the SAS code for an operational report:

- Because of the 999 image limit, remove logos (but reserve the space) from reports that are large mass mailings. Logos can be added later in the process by Print Services application.
- Use macros judiciously. Reusable code speeds development and simplifies maintenance.
- Have a coding standard defined with unit test rules in place before you start.
- Perform a level of acceptance testing before the report is converted to a stored process.
- SAS cannot support Envelope Inserter (OMR) codes. There are several third party Print Service applications that will repurpose a PDF and add these codes.
- The ODS escape character can be used to generate x of y page numbering.
- Use Environmental migration and configuration datasets to simplify the code development and testing.

WebDAV

The WebDAV server provides excellent temporary file storage for report output before it is archived or sent to Mail services.

- SAS has created the security components so access to WebDAV folders is managed by the SAS Metadata Server.
- The Permanent Package setting for stored processes sends all content (foreground or background) to a WebDAV folder. This requires no code being added to the report, simplifying maintenance.
- WebDAV is owned by Xythos and has an open set for API for other applications to access.
- SAS has provided multiple access components to WebDAV including a Libname engine and a set of package-publishing and retrieve macros.
- Package names must be unique if you do not want to overwrite the previous report run. A unique package name can be created on the fly by concatenating together the Stored Process name, _METAUSER and a Date/Time Stamp.
- There needs to be a master ID that owns the WebDAV root directory. Separate branches are created below that root directory, one per security group in the reporting application. The master ID is used by the batch job that moves code from the staging area to the Content Management.

Platform Scheduler

Reports that are to be printed out for mass mailings may run for several hours and may place a heavy load on a transactional system. These reports will be launched at night by the Platform scheduler to keep from overloading the transactional database and better balance workload. The jobs and flows are configured by SAS Administrators to optimize system usage.

Data Step Batch Server

The scheduler cannot directly run a stored process so the following changes are made in order to maintain only one version of code:

- A Data Step Batch server needs to be setup in the environment for Jobs
- A batch job wrapper is created to mimic parameters called by %STPBEGIN and %STPEND.
- The last line of the wrapper calls the SAS file that the metadata stored process points at.
- The scheduled job is pointed at the batch job wrapper.

SAS Metadata

Security is often a secondary consideration when developing a reporting application. The SAS Metadata server provides an integrated security hub for all the other components in the SAS BI Server. This simplifies both development and long term maintenance of code

- Centralized authorization to Portal Pages, Reports, and WebDAV Folders.
- Synchronization of the Metadata server to an enterprise directory server using batch loads will simplify overall system maintenance.

Content Management

There are multiple third party products that provide enterprise level content management. The following two systems provided significant functionality in areas where the SAS BI Server is not strong.

Print Services

Print Services software is designed to read and repurpose a PDF file. The software creates a map of the PDF format that it can then use to manipulate the original PDF. Sections can be added, deleted or moved. Font type and size can be changed and Graphics can be added. A frequent use of this technology is to add OMR print codes to existing documents. Other functions Print Services software can perform are:

- Adding, resizing or moving corporate logos and other images.
- Long PDF runs can be broken up in smaller files using the same technique used by OMR codes for identifying when a section of a report for one member ends and another begins.
- Mailing Addresses on the page do not always line up correctly with the envelope window when a report is generated. The Print Services software can reformat the address text on the fly so a cheaper stock envelope can be used.
- Presorting the PDF's by zip code so they can be mailed in zip code bundles, reducing postage.

Archive Services

At BCBSM any report that is sent to external customers and many internal reports must comply with a document retention policy. Several third party software packages are available for providing enterprise-wide content management. When interfacing to one of these packages here are some things to consider:

- Information on length of retention, size of documents and number being added on a monthly basis will be needed to provide sizing calculations for disk space growth.
- The grain of the PDF being stored, one big document for all users or one document per user, will impact how the system will be configured for retrieval.
- If a large document is stored for all members, then full-text searching may be useful for simplifying retrieval of content. When indexing the text avoid indexing on numeric data in the report. This will reduce index storage.
- Many applications will have a batch loader provided. Frequently, this will need a descriptor file with commands for the batch loader.
- Parameters should be selected carefully to optimize indexing for content searching.

CONCLUSION

This Framework does provide significant value to BCBSM by centralizing reporting functionality that can then be reused to provide operational reporting for other transactional systems. There are several additions to the SAS Enterprise BI Server release 9.2 that would improve operational reporting.

- Remove 999 image limit
- List report wizard for Proc Report in EG 9.2
- Proc Report enhancements like ByPageNo and SPANROWS

There are enhancements in future releases that would be very helpful.

- Platform scheduler being able to launch stored processes directly
- Production release of ODS Object for DATA STEP _NULL_ reports
- The ability to Stop or monitor the progress of a Stored Process

REFERENCES

Publish package interface

http://support.sas.com/rnd/itech/doc9/dev_guide/app/pkgintf/index.html

226-2007 **Beyond the Basics: Advanced PROC REPORT Tips and Tricks**

Allison McMahill, SAS Institute Inc., Cary, NC

ACKNOWLEDGMENTS

We would like to thank the Analytical Tools Team at BCBSM for their extensive work in converting this idea into a reality.

CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the authors at:

Paul Christenson
Blue Cross Blue Shield of Minnesota
3535 Blue Cross Road
Eagan, MN, 55122-1154
(651) 662-4422
Paul_e_Christenson@BlueCrossMN.com

Chad Flentje
Blue Cross Blue Shield of Minnesota
3535 Blue Cross Road
Eagan, MN, 55122-1154
(651) 662-6414
Chad_k_Flentje@BlueCrossMN.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.