

Using SAS/IntrNet as a Web-Enabled Platform for Clinical Reporting

Paul Gilbert, DataCeutics, Inc., Pottstown, PA
Steve Light, DataCeutics, Inc., Pottstown, PA
Gregory Weber, DataCeutics, Inc., Pottstown, PA

ABSTRACT

SAS/IntrNet® software provides access to the server-based capabilities of SAS® for enterprise Web-client application development. This software enables pharmaceutical companies to exploit strategic intelligence delivered by SAS software via the Web. For use in clinical reporting applications, SAS/IntrNet allows organizations to maximize their return on investment in SAS. SAS/IntrNet software is designed for SAS users who want to make existing SAS applications available over the Web using the Common Gateway Interface (CGI) program interface.

Currently, most pharmaceutical companies use SAS software to produce statistical analyses, data listings, summary tables and graphics that are used in the submission of clinical study reports to governmental regulatory agencies around the world. The process of producing these reports is commonly known as clinical reporting.

This paper discusses the advantages of using SAS/IntrNet as a platform for Web-enabled clinical reporting and illustrates the use of some of the key features of SAS/IntrNet within a clinical reporting application.

INTRODUCTION

SAS/IntrNet software provides access to the server-based capabilities of SAS for enterprise Web-client application development. This software enables pharmaceutical companies to exploit strategic intelligence delivered by SAS software via the Web. For use in clinical reporting applications, SAS/IntrNet allows organizations to maximize their return on investment in SAS. SAS/IntrNet software is designed for SAS users who want to make existing SAS applications available over the Web using the Common Gateway Interface (CGI) program interface.

Currently, most pharmaceutical companies use SAS software to produce statistical analyses, data listings, summary tables and graphics that are used in the submission of clinical study reports to governmental regulatory agencies around the world. The process of producing these reports is commonly known as clinical reporting.

When selecting a Web platform for clinical reporting, pharmaceutical companies should consider the following significant advantages of choosing SAS/IntrNet over a non SAS-based platform. These advantages are fully discussed in the section titled, Advantages of Using SAS/IntrNet.

- The SAS System is already the industry standard for clinical reporting. This facilitates Information Technology (IT) acceptance of SAS/IntrNet as a tool to extend the clinical reporting environment.
- Ability to leverage the current SAS-based environment and existing SAS expertise.
- SAS/IntrNet software gives developers access to the same SAS procedures that end-users commonly utilize in clinical reporting.
- A typical SAS development team can provide Web-based solutions in a short amount of time with only a few days training.

Some of the key features of SAS/IntrNet that can be used when developing a clinical reporting application are listed below. They are discussed in detail in the section titled, Features of SAS/IntrNet.

- The SAS/IntrNet host-based authentication feature provides end-to-end security and can utilize the existing SAS server security model.
- SAS/IntrNet supports load balancing. The clinical reporting workload can be distributed across multiple server processes and systems allowing for future growth.
- SAS/IntrNet supports saved sessions. During a session, macro-variables and data files can be saved and easily accessed by subsequent Web-client requests.
- SAS/IntrNet provides easy access to server-based files such as SAS programs, SAS Log files and SAS Output files.
- SAS/IntrNet allows for easy access to SAS libraries which can be used to create dynamic HTML pages.
- SAS/IntrNet automatically converts URL and HTML form parameters to SAS Macro variables that can be

used in SAS programs.

ADVANTAGES OF USING SAS/INTRNET

The SAS system is the industry standard for clinical reporting. Most pharmaceutical companies currently use Base SAS®, SAS/Stat® and SAS/Graph® to produce their clinical reports. SAS/IntrNet is an established product that compliments this proven and familiar SAS technology. Since SAS products are already a familiar part of the computing environment, SAS/IntrNet can be readily accepted as the primary tool for developing clinical reporting applications. It is our experience that security is usually the major hurdle to overcome. When implemented correctly, SAS/IntrNet can provide end-to-end security, utilizing the current information technology security infrastructure.

Companies can leverage their current SAS-based environment and existing employee knowledge. Existing clinical reporting software such as standard SAS programs, in-house developed SAS Macro libraries or off-the-shelf SAS Macro libraries can potentially be reused within a SAS/IntrNet application. In addition to their SAS expertise, the current clinical reporting staff is already familiar with these existing tools. This knowledge along with learning some basic HTML and JavaScript is typically all that is required when developing web-based clinical reporting applications with SAS/IntrNet. This is much more cost effective than retraining users on non-SAS web development languages or hiring staff with a different skill set.

SAS/IntrNet provides full access to the capabilities of the SAS system. This gives developers access to the same SAS products used by the SAS clinical reporting specialists. The most used products are Base SAS, SAS/Stat and SAS/Graph.

SAS/IntrNet is well suited for rapid development and prototyping. A typical SAS development team can provide Web-based solutions in a short amount of time with only a few days training. SAS Institute and other vendors provide both beginner and advanced level courses on SAS/IntrNet. Once trained, developers are able to develop sophisticated Web forms that take user-input through the complete set of HTML form controls (e.g. radio buttons, selection lists).

FEATURES OF SAS/INTRNET

There are key features of SAS/IntrNet to be aware of when developing a clinical reporting application. This section discusses some of the key features of SAS/IntrNet that we found most useful in developing our clinical reporting application. The DataCeutics Report Portal™ is used below to illustrate these features.

SAS/IntrNet supports end-to-end security. For security between the Web browser and the Web server, any industry standard and/or firewalls can be used. For security between the Web server and the SAS server, encryption and/or firewalls are supported. In many cases, the existing SAS server security model can remain unchanged by using the SAS/IntrNet built-in host-based authentication feature. The host-based authentication feature of SAS/IntrNet allows you to use the SAS server security application. For example, if you are using a UNIX SAS server, then the user is required to input their UNIX user id and password. The user is then provided with the user access as defined on the UNIX server. This feature is easily implemented in the SAS/IntrNet application startup program (appstart.sas). This program resides on the SAS server and is used to define the SAS/IntrNet session parameters. In the sample file, displayed below, the bolded statements manage the host-based authentication.

SAS/IntrNet Application Startup Program

```
proc appsrv adminpw='password' unsafe='&';%'' &sysparm auth=host;  
  allocate file sample '!SASROOT/samples/intrnet';  
  allocate file drppgm "/app/drpv3.0.0/dev/programs";  
  allocate file drpmac "/app/drpv3.0.0/dev /maclib";  
  allocate library drpmac "/app/drpv3.0.0/dev /maclib";  
  allocate file logfile '/data/sas_intrnet/drpv3.0.0/dev/logs/%a_%p.log';  
  proglibs drpmac drppgm ;  
  log file=logfile symbols=all display=all;  
  session timeout=3600 maxtimeout=3600 verify=( _sessionid _rmtaddr _rmtuser)  
  invsess=drppgm.timeout.sas;  
  request login=sample.weblogin.sas init=drppgm.init.sas timeout=900;
```

SAS/IntrNet supports load balancing. Intelligent load balancing routes client requests to idle servers, ensuring the fastest response times. Servers can be distributed over multiple systems, and they can be started automatically during periods of peak load and stopped during periods of inactivity. This allows the pharmaceutical company to optimize the clinical reporting application during times of peak demand and during times of low demand. This feature is easily implemented in the SAS/IntrNet Broker Configuration file (broker.cfg) that resides on the Web server and manages the SAS/IntrNet CGI broker software. In the sample file, displayed below, the bolded statements manage load balancing.

SAS/IntrNet Broker Configuration File

```
PoolService drpv300dev
ServiceAdmin "Paul Gilbert"
ServiceAdminMail "gilbertp@dataceutics.com"
ServiceLoadManager dciunix1:5008
SasCommand "/usr/bin/sas /app/drp/v3.0.0/dev/utilities/appstart.sas +
-rsasuser -noterminal -SYSPARM -log /data/sas_intrnet/drp/v3.0.0/dev/appstart.log"
IdleTimeout 60
ServiceTimeout 1800
MinRun 3 /* 3 servers minimum */
Server dciunix1
Port 5025-5032 /* 8 servers maximum */
FullDuplex True
```

SAS/IntrNet supports saved sessions. With saved sessions, a user's working library is available over multiple Web-client requests. This allows macro variables, data files and catalogs to be saved as the user is accessing the clinical reporting application. SAS data sets and catalogs can be saved over multiple sessions by placing them in a library named SAVE. SAS macro variables can be saved by naming global macro variables with a "SAVE" prefix. In the sample file, displayed below, the bolded statements manage the saved sessions.

SAS/IntrNet Application Startup Program

```
proc appsrv adminpw='password' unsafe='&';%'" &sysparm auth=host;
allocate file sample '!SASROOT/samples/intrnet';
allocate file drppgm "/app/drp/v3.0.0/dev/programs";
allocate file drpmac "/app/drp/v3.0.0/dev /maclib";
allocate library drpmac "/app/drp/v3.0.0/dev /maclib";
allocate file logfile '/data/sas_intrnet/drp/v3.0.0/dev/logs/%a_%p.log';
proglibs drpmac drppgm ;
log file=logfile symbols=all display=all;
session timeout=3600 maxtimeout=3600 verify=( _sessionid _rmtaddr _rmtuser)
invsess=drppgm.timeout.sas;
request login=sample.weblogin.sas init=drppgm.init.sas timeout=900;
```

SAS/IntrNet provides tools to allow easy access to static files such as SAS programs, SAS Log files and SAS Output files. There are several application utilities that give developers a head start in Web enablement activities by providing functions that you would otherwise have to generate yourself. These utilities perform data set conversions, serve files, and encode and decode character strings. These application utilities are documented at <http://support.sas.com/rnd/web/intnet/apputil/index.html>.

One of the application utilities is the FILESRV macro. It is used to serve a wide variety of external files and catalog entries, including those that are not defined to a Web server. With the FILESRV macro, you can control which files to serve and you can also control the HTTP and MIME headers that are served with the file. The Sample File View Page below allows users to view and download files from the server.

The user selects the file to view; the file is then downloaded to the user's desktop to a viewing application. This download function is performed by the FILESRV macro. The program that uses the FILESRV macro is displayed in the File Serve Code below.

Sample File View Page

The screenshot shows a web browser window titled "DataCeutics Report Portal version 3.0.0 Development Instance - Microsoft Internet Explorer". The browser's address bar and menu bar are visible. The main content area displays the following information:

- Project:** Sample Project 1
- Study:** Protocol 003 - DEV
- Active:** Sample Project 1/Protocol 003 - DEV
- User ID:** gilbertp
- Active Domain:**
- Active Subset:**

On the left side, there is a navigation menu with the following options:

- Log Out
- Build Report
- Run Report
- View Report
- Manage

The main content area is titled "View Files: Select a File to View" and contains the following text:

View the titles of the associated output by moving your mouse over the Titles below

- ▶ **ae-list-03** Titles
- ▶ **ae-summ-01** Titles
- ▶ **ae-summ-01_test14** Titles
- ▶ **ae-summ-01_test17** Titles
- ▶ **ae-summ-01_test18** Titles

Below this list, there are several links for the file "ae-summ-01_test18":

- [ae-summ-01_test18.sas](#) (2004/09/08 15:43)
- [ae-summ-01_test18.log](#) (2004/09/29 10:34)
- [ae-summ-01_test18.lst](#) (2004/09/08 15:43)
- [ae-summ-01_test18.pdf](#) (2004/09/29 10:34)
- [ae-summ-01_test18.rtf](#) (2004/09/29 10:33)
- [ae-summ-01_test18.stg](#) (2004/09/29 10:34)

At the bottom of the list, there are more items:

- ▶ **ae-summ-01_test19** Titles
- ▶ **ae-summ-01_test2** Titles
- ▶ **ae-summ-01_test20** Titles
- ▶ **ae-summ-01_test3** Titles
- ▶ **ae-summ-01_test5** Titles
- ▶ **ae-summ-01_test6** Titles
- ▶ **ae-summ-01_test7** Titles
- ▶ **ae-summ-01_test8** Titles
- ▶ **ae-summ-01_test9** Titles
- ▶ **ae-summ-03** Titles

The browser's status bar at the bottom indicates "Local intranet".

File Serve Code

```
%filesrv(file=&save_filename., filetype=E);
```

SAS/IntrNet allows for easy access to SAS libraries and to utilize the libraries to create dynamic HTML pages. SAS/IntrNet can surface the attributes of the SAS libraries that reside on your SAS server. This can be done by using Base SAS procedures such as Proc Contents or Proc SQL to query the SAS dictionaries. The example below illustrates creating dynamic HTML code that is generated from a list created by querying a SAS dictionary.

The Sample Report Generation Page below allows users to select frequency variables that are used in a Proc Freq procedure to create the output. In this example, we query the DICTIONARY.COLUMNS table to create a list of variables on a dataset. This list is then used to create the HTML selection list on the page. The code is listed below the sample page.

Sample Report Generation Page

The screenshot shows the DataCeutics Report Portal interface. The browser title is "DataCeutics Report Portal version 3.0.0 Development Instance - Microsoft Internet Explorer". The page header displays "Project: Sample Project 1", "Study: Protocol 003 - DEV", and "Active: Sample Project 1/Protocol 003 - DEV". The user ID is "gilbertp" and the active domain is "DEMO - (31 subjects, 23 columns, 31 rows)".

The main content area is titled "General Frequency Listing" and contains the following instructions and options:

- Select Options then press the Submit button
- Select the **Page BY** variables to be included in the report, limit to 5 selections
- Data File Variable List (Select to include in the list - Order by selection):
 - AGE -Age
 - BIRTHDT -Date of Birth
 - DMACTDY -Actual Study Day of Visit
 - DMDT -Date of Visit
 - DMREFDT -Subject Reference Date
- Page BY Variables to Include (Select to remove from list):
- Select Frequency Variables:
 - AGE -Age
 - BIRTHDT -Date of Birth
 - DMACTDY -Actual Study Day of Visit
 - DMDT -Date of Visit
 - DMREFDT -Subject Reference Date
 - ENROLLED -Enrolled Population

The interface also includes a "Log Out" button and a "Build Report" section with options for "Data Domain" (DEMO), "Subset", "Report", "Run Report", "View Report", and "Manage".

Dynamic HTML Creation Code

```

** get the variable metadata **;
proc sql noprint;
  create table save._dsvars as
  select name, label, type, format, length
  from dictionary.columns
  where libname="SAVE" and upcase(memname)="&save_ds."
  order by name;
quit;

** create the first of four HTML selection lists **;
data _null_; set save._dsvars end=eof;
  file _webout;
  if _n_=1 then put
    '<BR>Select Frequency Variables<BR>'
    '<SELECT NAME=save_freq1>'
    '<OPTION VALUE=""></OPTION>';
  put '<OPTION VALUE="" name "">' name '-' label '</OPTION>';
  if eof then put '</SELECT>';
run;

```

SAS/IntrNet automatically translates HTML selections to SAS Macro variables that can be used in SAS programs. The SAS/IntrNet Dispatcher uses a standard method for translating HTML objects to SAS macro variables. These HTML objects include hidden input, text input, single selection lists, multiple selection list, check boxes and radio boxes. The Dispatcher uses macro variables to pass name/value pair data to your programs. Usually, both the macro variable names and list-item names match the names supplied in the HTML code. The HTML names that are used to create the macro variable names must be valid SAS names.

In the example above, we create a HTML single selection list name **save_freq1**. The macro variable named **&SAVE_FREQ1**, containing the value from the user selection, is passed to the next request. In this case, the request is a SAS program containing Proc Freq code. The Report Generation Code is displayed below.

Report Generation Code

```
%let frqs=%cmpres(&save_freq1 &save_freq2 &save_freq3 &save_freq4);  
  
proc freq data=&save_ds.;  
  &bys;  
  table &frqs /list missing;  
run;
```

CONCLUSION

This paper listed several advantages for selecting using SAS/IntrNet for implementing Web-enabled clinical reporting. Additionally, it has illustrated several features of SAS/IntrNet that can be critical in implementing Web-enabled clinical reporting. In our view, the advantages can be distilled down to two primary issues. First, since the SAS System is the industry standard for clinical reporting, this should pave the way for management and user acceptance of SAS/IntrNet to extend the clinical reporting environment. Second, staying with a SAS-based environment allows the company to leverage their current reporting software and to use employee's knowledge and expertise of SAS.

REFERENCES

SAS Institute Web Site: http://www.sas.com/technologies/bi/content_delivery/intrnet/index.html.

CONTACT INFORMATION

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

Steve Light
DataCeutics Inc.
Pottstown, PA 19464
610-970-2333
lights@dataceutics.com

DataCeutics, Inc. (<http://www.dataceutics.com>) is a SAS Alliance Partner and a CDISC member.



CR Toolkit and DataCeutics Report Portal (DRP) are registered trademarks or trademarks of DataCeutics, Inc.

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.