254-2011

SAS® Data and Stored Processes on a BlackBerry®

Sy Truong, Meta-Xceed, Inc., Fremont, CA

ABSTRACT

In the competitive smart phone environment, RIM BlackBerry® remains the leading smart phone worldwide changing how we use mobile computing. Although both the iPhone and Android have been making the headlines, BlackBerry is quietly growing in the area of mobile applications for business use with its introduction of App World 2.0 and BlackBerry OS 6.0. BlackBerry applications add an efficient use of local mobile device resources that enhance the user mobile experience; while leveraging traditional client server computing features that are similar to that of a web browser. Models such as BlackBerry Curve, Tour, Storm, Pearl, and Bold all provide feature such as GPS tracking, voice communication, digital video recorder, and multitudes of other functionality that raise the bar beyond web 2.0. SAS® has been an analytical business intelligence powerhouse for many years; yet it is a relative late comer to this mobile computing revolution. This paper demonstrates how a SAS stored process, or a traditional SAS macro, and its parameters can be presented through a native user interface on a BlackBerry. Imagine how you can access the most up-to-date and dynamic business information delivered directly to you anywhere where there is a 3G or Wi-Fi signal. At last, business analytics is no longer placed behind walls guarded by legions of power users, but rather it will be unleashed to users on the go!

INTRODUCTION

Mobile computing with the popularity of smart phones is reaching a tipping point that is going to have a profound effect on how we access and work with information. The BlackBerry Curve and Bold accompanied by the App World has increased the cadence for competitors such as Apple iOS and Google Android. The industry is reaching a point of critical mass that can no longer be ignored since the BlackBerry and related devices have become a powerful force in many corporate environments. The App World makes it easy for users to download applications. It is becoming a ubiquitous mobile computing device that is outpacing the traditional desktop PC. The BlackBerry is no longer viewed as just a mere cell phone, but considered a serious computing device.

When I first started using SAS on mainframe computers, I noticed a similar attitude that system administrators and SAS analysts had towards the IBM PC. These small desktop computers along with its burgeoning Windows operating system was considered a toy and therefore could never really run serious business applications crunching large data with powerful analytics. The closed minded view that PCs were just toys was analogous to how skeptics today view the BlackBerry and other mobile devices. Those same managers closed their eyes to the many uses of the diverse set of applications that run on PCs. They began to see its usefulness when the PCs were connected to the Internet that changed everything and pushed mainframe computing into relics of the past. The many applications that are being developed for mobile devices such as the BlackBerry demonstrate that there are many killer apps that can be game changers which are analogous to its predecessor in the PC heyday.

There have been many wars waged within the computer industry ranging from hardware to browsers. What it all boils down to when it comes to the user's experience is the application. In the smart phone war, the App World has demonstrated its ability to deliver applications. In many instances, the BlackBerry application is competing and replacing mobile web through a web browser. Users are no longer going through a browser, but rather gravitating towards a new access point for a variety of distinct applications. This is a fundamental shift that will seriously alter cloud computing and related Internet services. To put it bluntly, "It is the App Stupid".

It is not too late for SAS analysts who are glued to their desktops and servers to benefit from the mobile revolution. This paper will elaborate on a few of the following areas which debunk the notion of BlackBerry as a mere phone used for email and show that it can deliver serious business applications such as SAS in ways that were not possible

before. Some topics shared include:

- **System Architecture** - How a BlackBerry accesses SAS data, macros and stored processes
- **Application Servers** - SAS as an application server for BlackBerry applications
- **Stored Process** – A SAS program with enhance metadata features
- **Secure Users** - Authenticating users and securing access to data
- **Macro Parameters** - It's no longer limited to check boxes and radio buttons
- **Viewing Data** - Viewing SAS data in a mobile smart phone

This paper will elaborate on these concepts through the example illustrated through BI Flash™ software, which is a combination of SAS programs and a BlackBerry App. It will demonstrate how SAS macros and stored process can deliver dynamic business critical analytics to mobile users. We are about to enter into a new era of computing, leaving behind traditional PCs and web 2.0. This paper paves a new path into how SAS can boldly step into this dynamic world of mobile computing.

SYSTEM ARCHITECTURE

The delivery of SAS data and reports to the BlackBerry requires a different architecture as compared to traditional client server systems. This is similar to web applications in how it delivers information to a browser, but in this case, the BlackBerry application replaces the browser. The diagram below shows the components of this computing architecture including: the BlackBerry application, a web server and the BI Flash application server.

In this example, the BlackBerry application communicates through standard TCP/IP protocol to a web server. The web server then communicates to an application server, which is actually a SAS session processing SAS programs and data. The output resulting from the SAS program is then delivered back to the BlackBerry in a similar way a web browser would access web pages stored on a web server. The distinction, however, is that the BlackBerry application is not a web browser and the SAS session running on the server is more dynamic and creates a more responsive user experience utilizing local resources compared to static HTML. The SAS data, macro programs and stored processes that are requested may be simple and standard yet facilitating the communication takes a little more effort. The request from the BlackBerry application and the delivery of information from the server is handled by BI Flash. This delivers the full power of SAS on the server to the BlackBerry. The following steps are taken for users to access SAS information on the BlackBerry.

STEP 1: Download BlackBerry Application

One of the successful aspects of the BlackBerry is how users can easily download a BlackBerry App directly from the RIM App World. This is a user-friendly way of searching for and downloading applications to your BlackBerry. In this case, you can search for and download the "BI Flash" application that is also referred to as the client component. This enables the communication between the BlackBerry and the SAS server.

STEP 2: BI Flash Application Server
More details on this step will be explained in the "Application Server" section describing how an administrator can manage and utilize the BI Flash server. This server functions as a listener waiting for a request from the BlackBerry. Upon receiving a request, it would process this analogous to how you would submit SAS programs from display manager or a stored process on EG. The server would generate a SAS log and output results in XML, which is then sent back to the BlackBerry to be reviewed.

STEP 3: Connect BlackBerry to Application Server
Most BlackBerry applications have configuration settings. This allows users to easily configure settings during setup and then use the application without further changes unless future configuration changes are needed. In order for the BI Flash BlackBerry app to access SAS, the user would need to configure the following:

- **Host Name** – The name of the server or an IP address of the SAS server
- **Username** – A valid username that has been defined on the server needed during authentication
- **Password** – A user defined password to secure access

There may be other configuration options which will set the default behavior of the application, but the parameters above show the minimum requirements in order to connect to a SAS server.

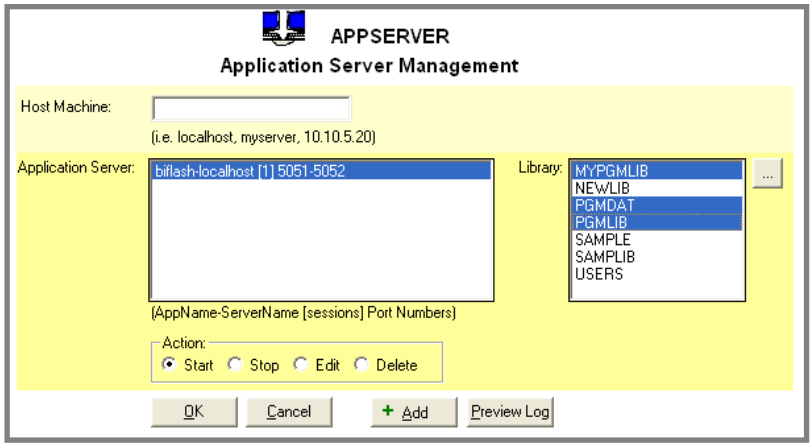
STEP 4: Run Application
The final step taken by the user to access SAS data is to execute the SAS macros from the BlackBerry. This request is initiated from the BlackBerry application and sent directly to the server along with the user selected options. The results are then returned to the BlackBerry displaying the most updated information on the server.

The system architecture in this example is rather simple compared to other systems that require multiple layers of middleware. It is similar to the SAS/IntrNet where users are on a web browser accessing SAS data and programs on the server through the broker and SAS application server. The difference, however, is that the client is not a browser, but rather a dynamic BlackBerry application which can fully take advantage of the user interface and processing of the BlackBerry device and environment.

APPLICATION SERVERS

An application "server" in this context is essentially a SAS session waiting to receive requests from the user. Once a request is received, it would then execute the specified SAS scripts and then deliver its output back to the user. One application server can serve many users at once. The server is able to do this when queuing up multiple requests and then executes it one at a time. Each application server is assigned a set of SAS libraries to easily access predefined data and related SAS programs or stored processes.

The SAS program that is requested by the user also resides on the server as defined by a library. This can be pre-assigned similar to how you can assign a LIBNAME or FILENAME in an AUTOEXEC file used during an execution of a SAS program. The assignment of these libraries is established when the application server is setup.



Each application server communicates to its corresponding

BlackBerry application through its own assigned TCP Socket port. This ensures that there is no collision between multiple application sessions for optimal performance. It is optional to increase the number of ports to handle additional requests. In this case, a separate SAS session is started on the server running simultaneously to enable greater bandwidth.

SAS STORED PROCESS

SAS programming has changed in recent years with the introduction of the BI architecture that supports stored processes. The essence of a stored process is still a SAS program, but there are some differences. A SAS program is stored as an ASCII text file residing on a file system. A stored process, however, is managed on a BI metadata server that has more metadata management along with security for each stored process. This paper will describe both SAS programs and stored processes in a similar manner from the BlackBerry's perspective. This approach is more flexible allowing users to access both types of SAS scripts on their mobile devices.

The BlackBerry user interface does not change much between a SAS program and a stored process. However, the access method on the SAS server is quite different. The following steps describe what happens behind the scenes as BI Flash server component performs queries against the metadata server before it can access and then submit the stored process.

STEP 1: Request Stored Process Folder Location

As analogous to a SAS program that resides within a folder on a file system, a stored process is saved on the metadata server within folders. These folders are stored within repositories managed by the metadata server and can only be viewed through a client such as Enterprise Guide. In order for the BlackBerry to access this, the server component must first issue a request in XML format to identify the folders where the stored process is located. An example SAS code for this is shown here:

```
*** Specify the BI Metadata Server Access Information ***;
options metaport=8562;
options metauser="myusername";
options metapass="mypass!@#";

*** Build the XML Request for Stored Process Folder Information ***;
filename xmlReq temp lrecl=1024;
data _null_;
   file xmlReq;
   put '<GetMetadataObjects>';
   put ' <Reposid>$METAREPOSITORY</Reposid>';
   put ' <NS>SAS</NS>';
   put ' <Type>Tree</Type>';
   put ' <Flags>384</Flags>'; /*OMI_GET_METADATA+XMLSELECT*/
   put ' <Options>';
   put ' <XMLSelect search="*[@TreeType=&apos;';
   put "BIP Folder";
   put '&apos;]"/>';
   put ' </Options>';
   put '</GetMetadataObjects>';
run;
```

Although this is in XML format, it is generated from a SAS program within a data step. The result is an XML file that is then sent as a request for the information stored in the "BIP Folder" containing the stored process.

STEP 2: Create XML Map to SAS Dataset

XML files can be parsed programmatically, but this is inefficient for SAS when left as a text file. A better approach is to convert the XML into a SAS dataset allowing for SAS programs to easily process. This is accomplished by creating an XML map describing to SAS how each column within the SAS dataset corresponds to the XML file tag sets. The following code performs this XML map to assist with the parsing of the stored process folder and related paths.

```
*** Create a SAS XML Map Converting the Tree Query Result into SAS dataset ***;
filename xmlMap temp lrecl=1024;
data _null_;
  file xmlMap;
```

```
   put '<?xml version="1.0" ?>';
   put '<SXLEMAP version="1.2" name="TreeMap">';
   put ' <TABLE name="TreePaths">';
   put ' <TABLE-PATH syntax="XPath">/GetMetadataObjects/Objects/Tree</TABLE-PATH>';
   put ' <COLUMN name="treeNodeName" retain="yes">';
   put ' <PATH syntax="XPath">/GetMetadataObjects/Objects/Tree/@Name</PATH>';
   put ' <TYPE>character</TYPE>';
   put ' <DATATYPE>string</DATATYPE>';
   put ' <LENGTH>256</LENGTH>';
   put ' </COLUMN>';
   put ' <COLUMN name="treeNodeid">';
   put ' <PATH syntax="XPath">/GetMetadataObjects/Objects/Tree/@Id</PATH>';
   put ' <TYPE>character</TYPE>';
   put ' <DATATYPE>string</DATATYPE>';
   put ' <LENGTH>17</LENGTH>';
   put ' </COLUMN>';
   put ' </TABLE>';
   put '</SXLEMAP>';
run;

*** Setup Fileref to Receive the Results ***;
filename xmlResp temp lrecl=32767;

*** Assign an XML libname to access the results ***;
libname treeinfo xml xmlfileref=xmlresp xmlmap=xmlmap access=readonly;

libname pgmdat '&curdatpath';
data pgmdat.TreePaths;
   set treeinfo.TreePaths;
run;
```

The XML is created and parsed through traditional SAS data step. The XML map coupled with the XML SAS library engine function as a translator making it effortless moving data between two distinctively different formats.

STEP 3: Query and Capture Stored Process

Once the XML input and output XML filenames are defined with their respective XML maps, a query can be applied to the BI metadata server where the stored processes resides using PROC METEDATA. This is the step that actually submits the request taking all the XML information defined within the input to formulate the request and then returning the output result. The output is initially in XML format, but with the use of the XML libname and XML maps defined, will result in being a SAS dataset. This format allows for easy processing for subsequent SAS code.

```
*** Query the BI Metadata Server ***;
proc metadata in=xmlReq out=xmlResp;
run;
```

STEP 4: Parsing Resulting Data

The results from PROC METADATA can then be read as input into another SAS step for additional processing. In this example, the information contains the folder and path locations where the stored processes are stored. This information will be used to later submit the stored process.

```
*** Assign a XML libname to Access the Result for Processing ***;
libname filedir xml xmlfileref=xmlResp xmlmap=xmlMap access=readonly;
libname pgmdat '&curdatpath';
data pgmdat.FileDirectory;
   set filedir.FileDirectory;
run;
```

STEP 5: Submit the Stored Process

Upon request form the BlackBerry, the stored process is then executed on the BI server. This is applied in a similar way that you would submit a SAS program in batch mode. The difference is that the location of the file and name of SAS program has been captured and specified through the PROC METADATA as shown in the previous steps.

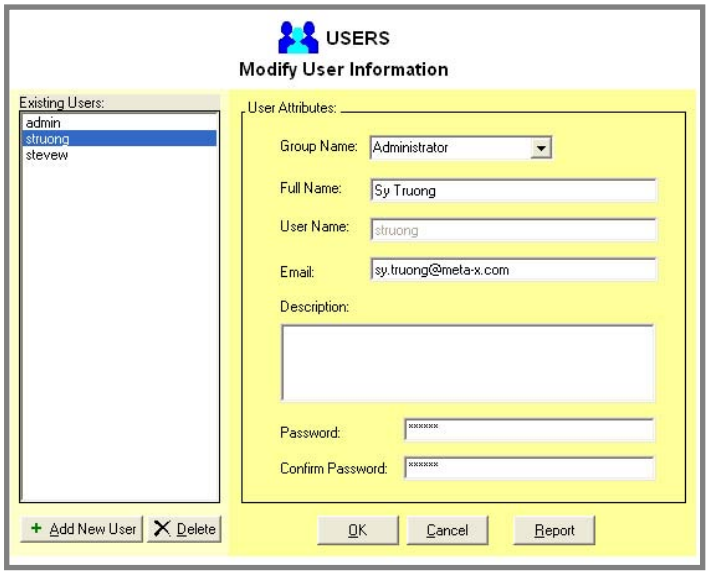
```
*** Define SAS Configuration Commands ***;
sasroot = pathname('sasroot');
cur_os=symget('sysscpl');
command="SAS.EXE";

*** Generate the SAS Submit Command depending on Operating System ***;
if index(cur_os,'WIN_') > 0 or index(cur_os,'XP_') > 0 or index(cur_os,'NET_') > 0
   or index(cur_os,'X64_SRV')>0 then do;
   submit continue;
   systask command '"&sasroot&slash&command" -sysin "&sasfile" -log "&logfile"
   nosplash' WAIT;
   endsubmit;
end;
else do;
   submit continue;
   systask command '&sasroot&slash&command -log &logfile &sasfile  nosplash' WAIT;
   endsubmit;
end;
```

This example uses SCL and the submit block to create the command code since it can efficiently generate and execute the command with minimal coding.

SAS stored processes are stored on a BI Server that contains additional layers of metadata. This provides more management features as compared to a traditional SAS program. The extra layers of middleware, however, require more effort to access and process. It requires a unique set of skills combining knowledge of XML and SAS that is a steep learning curve for some SAS programmers. The investment is worthwhile since it opens up possibilities such as in this example illustrating how a BlackBerry can access and submit stored processes.

SECURE USERS

The management of security for a stored process is accomplished through the BI Server. On the other hand, traditional SAS programs, require additional tools to be developed that assist BlackBerry users with methods of access. There are two distinct roles that users play within the framework of delivering SAS data and programs to the BlackBerry; including an administrator and end-user. The administrator is usually the SAS analyst or statistician that developed and manages the data, SAS program, or stored process on the server. The BlackBerry user receives the reports and data onto their BlackBerry. Before a user can access SAS reports and data, the following steps are taken to ensure proper security.



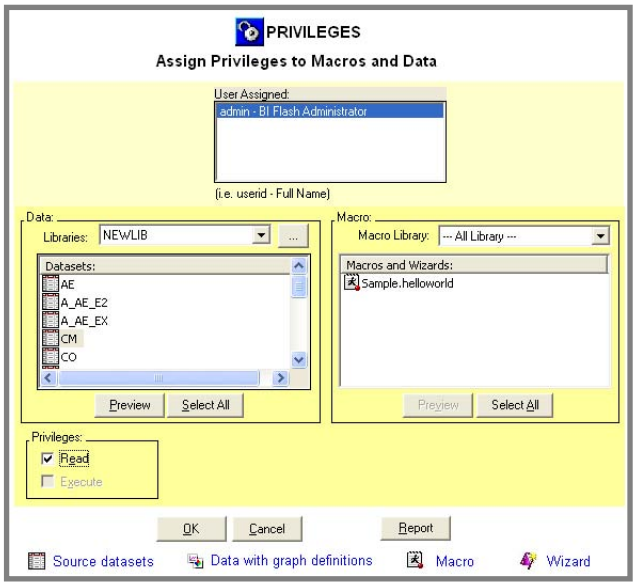
STEP 1: Account Setup

An account is set up on the server with the proper credentials in order to identify and authorize the user. The key attributes needed to authenticate the user include a unique user ID and user defined password.

STEP 2: User Privileges

By default, the user only has access to a set of sample SAS macro programs, stored processes and datasets that come with the system. In order to deliver real information, additional SAS macros and data need to be registered and have read and execute permissions granted.

The permissions model is simple compared to an operating system layer of security since there are only two types of privileges needed including "read" permission for SAS data and "execute" permission for SAS macros. It is implied that if the user can execute a macro that they also have read permission to the macro and its related output.



STEP 3: Login and View

Once the administrator applies the correct permissions, the BlackBerry user can see the programs and stored processes on their selection lists. They can then view the data in a viewer or execute the macro and view its resulting output report.

It is useful to have users access the data that they have been granted access. The management of each program and data, however, adds administrative overhead. In this example, the effort is kept to a minimum so that both the administrator and BlackBerry end-user can get to the information that is needed with little effort.

All of the user permissions are handled on the server and stored centrally within a SAS dataset. Each dataset and SAS macro is managed centrally on the server within a dataset with each item assigned to a unique identifier. The dataset structure that stores this is shown here.

Variable	Type	Label
Objid	Num	Object ID
Libname	Char	Library Name
Items	Char	Items
Type	Char	Object Type
Userinter	Char	User Interaction Name
Datetime	Num	Date Time of User Interaction
Privileges	Char	Privileges

In this case, the privileges variable stores a simple value "read" or "write". This dataset is then used to document and manage all privileges associated with macros and data stored in the system.

MACRO PARAMETERS

In the traditional batch environment, users specify the options for a macro by typing the selected values in a SAS program script when invoking the macro. The entered values would then be processed by the macro by inserting them into the specified parameter within the code. An example macro call is:

```
*** Generate Report of data by specific subset ***;
%dataview (indata = mylib.demog,
           sortby = subjid startdt,
           reptitle = Demographic data sorted by subject ID and start date);
```

In this example, there are three parameters including: INDATA, SORTBY, and REPTITLE. The BlackBerry application captures these parameters with a user-friendly interface as compared to having the user write a SAS macro call. In this case, the BlackBerry user selects the values for the macro parameters through the native interface of the BlackBerry device. The user would then access and execute the `%dataview` macro by performing the following steps.

STEP 1: Selecting Macro

The user would navigate to the macro section by clicking on the macro button on the navigational bar at the top. This brings the user to a list of all the SAS program macros that the user has privileges to in the current library.

The user then navigates through the libraries by clicking on the library choice as shown in the current "Sample Macros" example. In this case, they would click on the "DATAVIEW" to drill down to its parameters.

STEP 2: Selecting Macro Parameters

Upon selecting of the `%dataview` macro, all the macro parameters will be presented with standard BlackBerry user interface elements. Each parameter will be listed in the order in which it is defined.

In the traditional macro approach, the user would input the values of the parameter by typing the text value within the SAS script file according to the named parameters. This is difficult for most users since they are not familiar with the correct spelling of the library or dataset name. This commonly leads to erroneous entries and errors in the macro execution. On the other hand, the BlackBerry interface is much more user-friendly. In this example, three distinct entry types are displayed as described below:

- **Input Data** – The input data parameter uses a standard SAS two levels dot notation such as LIBNAME.DATASET. For example, the value of "mylib.demog" refers to a library "MYLIB" and the dataset DEMOG. The user can edit this as open text or there is the option of selecting a list for the selection of libraries and associated datasets.
- **Sort Variables** – The sort variables is another standard multiple option selection list. There are several different types of selection lists. In this case, a simplified multiple checkbox selection list is presented.
- **Report Title** – The report title requires a text entry field which the user can type any text value.

There are many other types of controllers which macro parameters can be associated with to make the user entry more intuitive and less prone to errors.

Some of the graphical user interface elements are similar to those found on desktop applications. The text entry and check list are similar to a text entry or list box on desktop applications. They only differ in their layout to make it easier for users on a smaller screen. However, some user interface elements diverge from the desktop interface to fully take advantage of the smaller screen of a BlackBerry. The author of the macro has the option to configure their macros and select which controller to best suited their needs. The default list of values will be displayed based on values of a dataset or the SAS system views such as SASHELP.VSLIB for a list of available libraries. Once the macro author has configured this on the server; the BlackBerry user benefits from having a user-friendly method for selecting and executing the macros.

Upon completion of parameter selection, the user would click on the "Run" button on the upper right to have the program

execute on the server. Depending on the macro ODS options, the result can be generated in: HTML, PDF, RTF or many other file formats that SAS supports. The output viewer supports all of these formats which allow users to navigate to and zoom in on any particular data point.

If the user finds the particular report important to share, they can click the send button and compose an email. In this case, the output will be attached to the email so the recipient can receive the full output along with the email describing the meaning of the report. For large output report files, the email can be optionally sent from the server with the attached. This will avoid having to download a large set of data onto the BlackBerry and provide communication with other team members.

The added ability to share the information through email extends the collaboration effort that is crucial in projects that require large teams. This provides capabilities that were once only available to laptops for mobile users. The ability to do this on the BlackBerry will provide important data and analytics components to more mobile users.

VIEWING DATA

SAS macros are very effective at allowing users to perform analysis by dynamically requesting for reports or views of the data. However, there are times when all the user needs to do is to view the raw data. On the Windows desktop environment, you would use the SAS data viewer. However, the BlackBerry has different capabilities such as magnify tool to zoom in on data views. These additional features can be used to view the data more effectively. The following steps illustrate how to view SAS on a BlackBerry.

STEP 1: Selecting Library

SAS datasets are stored in libraries which point to physical folders on the server. This is also referred to as SAS LIBNAMEs. The main datasets screen on the BlackBerry contains the latest list of datasets. The first item on this list is the library that is currently being viewed.

When the user clicks on the library labeled "DATAEXM", it drills down to a list of libraries that the user has permissions to. The current selected library is identified with a check mark and colored with standard selected blue tint. The user can then select any other library by clicking on the desired library.

STEP 2: Select Data

The main dataset list displays all dataset names along with the associated label available within the selected library. This can be efficiently scrolled through by using trackball that is standard for the BlackBerry interface. The text size and layout of the dataset list is optimized for the user selection within the BlackBerry interface.

STEP 3: View Data

After you have selected a specific dataset, a data viewer presents the initial the entire dataset similar to a HTML report produced from PROC REPORT from SAS. The user can scroll through the data using the trackball and zoom option to view a specific data point.

The interface has been modeled after a standard HTML viewer from the existing standard BlackBerry viewer. RIM has devoted extensive research into developing a user interface that is user-friendly for the mobile device optimizing

use of scrolling and zooming through one finger rolling over or pressing the trackball. Rather than re-inventing the wheel, this data viewer leverages upon RIM's standard to enhance the user's experience.

The code and algorithm logic that goes into developing the SAS application server and BlackBerry client application is extensive. This section will select a subset to illustrate how a developer can accomplish functions on the BlackBerry device in conjunction with the server running SAS. The following steps are applied to display the list of library on the Blackberry deriving the selections from SAS.

STEP 1: Capture Libraries

A SAS program is executed on the server to capture all libraries available. It is then compared to the user permissions before the final list is sent to the client BlackBerry. The following example code is used to capture this:

```
do while(fetchobs(dsid,cnt) = 0);
   libname1 = upcase(getvarc(dsid,varnum(dsid,'libname')));
   if libname1 not in ('WORK','SASUSER','SASHELP','SASMSG','MAPS',
      'PGMLIB','_MXI','SERLIB','DATLIB','USERS') then do;
      if searchc(liblst, libname1, 1, 1, 'Y') = 0 then do;
         rc = insertc(liblst,libname1,-1);
         path = lowcase(getvarc(dsid, varnum(dsid, 'libpath')));
         idx = index(path, '\');
         path = substr(path, idx);
         path = tranwrd(path, '\', '/');
         rc = insertc(pathlst, path, -1);
      end;
   end;
   cnt = cnt + 1;
end;
dsid = close(dsid);
```

This algorithm utilizes a SAS view SASHELP.VSLIB capturing all the libraries available from the SAS session. The BI Flash server component then inserts its findings into a list to be parsed to be delivered to the BlackBerry.

STEP 2: Subset User Privileges

After the libraries are captured, it is compared to the user permissions access control list. BI Flash stores all the user permissions as described in the permission section above.

STEP 3: XML Library List

Once the final list of libraries has been determined, an XML file is created capturing containing the appropriate libraries as shown here.

```
<Libraries>
   <Library id="1">
      <shortTitle>Sample Library</shortTitle>
      <longTitle></longTitle>
      <descriptionTitle>/myserv/lib1</descriptionTitle>
   </Library>
   <Library id="2">
      <shortTitle>Sydata</shortTitle>
      <longTitle></longTitle>
      <descriptionTitle>/myserv/testlib</descriptionTitle>
   </Library>
   <Library id="3">
      <shortTitle>Sample Macros</shortTitle>
      <longTitle></longTitle>
      <descriptionTitle>/myserv/testlib</descriptionTitle>
   </Library>
...
</Libraries>
```

This is information is delivered to the BlackBerry via a web server. The BI Flash client application parses through the XML file and then displays it in the user selection list. This is displayed using the native BlackBerry selection list providing a better user experience compared to non standard web browser interfaces.

CONCLUSION

The distribution of products and services has radically changed in recent years as for companies like Google, eBay and Amazon take advantage of efficiencies of Internet distribution. Rather than concentrating on the top few blockbuster products, these companies are finding that the many niche products such as user specific advertisement on Google and esoteric books on Amazon add up to be just as significant in total sales as their super sellers. The cost for them to distribute a large array of solutions has become profitable when delivered across the internet as compared to shrink-wrap boxes sold in stores.

This shift in the market is also taking shape in the software. A new marketplace is emerging for software as evident in the App World distribution of BlackBerry Apps. This is proving to be a new and better way to distribute specialized niche software while taking advantages of the efficient distribution model. The mobile computing platform such as the BlackBerry has become an important platform for the delivery useful and specialized analytical software to users. At the current moment, however, there is little in this space for business intelligence applications. It is unquestionable that SAS can deliver sophisticated business intelligence information by delivering SAS macros and stored processes coupled with its powerful programming language resulting in tremendous analytical capabilities to mobile users. In the past, these tools have been limited to power users who can write SAS code or utilize BI tools on SAS servers. This paper has presented examples where the power of traditional SAS programs and stored processes can be delivered to users on the BlackBerry. This bridges the gap by delivering endless possible views of the data to users without requiring them to write SAS code. This will truly liberate business intelligence to be accessed by any user allowing them to get the latest information any time directly from a device in their pocket anywhere.

CONTACT INFORMATION

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

Sy Truong
MetaXceed, Inc. (MXI)
42978 Osgood Rd
Fremont, CA 94539-5627
tel: 510.979.9333
fax: 510.440.8301
E-mail: sy.truong@meta-x.com
Web: www.meta-x.com