Paper 012-2010

SAS Data and Macros on iPhone

Sy Truong, Meta-Xceed, Inc., San Jose, CA

ABSTRACT

Apple iPhone® has revolutionized mobile computing, taking smart phones to the next level by obtaining about 70 percent of the application downloaded with 3 billion total downloads of 140,000 applications. All major websites such as Google, eBay and Amazon have their services available as an "iPhone app" rather than having the user go to a web browser. The iPhone applications add a new dimension to the user experience by leveraging on traditional client computing features of a web browser, but new features are added such as the ability to know where the user is, voice communication, multi touch screen interface, accelerometer along with a multitude of custom applications 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 dataset can be viewed on an iPhone. It takes a traditional a SAS macro and its parameters presented through a GUI for user selection on an iPhone and then executes the macro with the output results displayed on the iPhone for review. Imagine how you can access the most up to date and dynamic business information delivered directly to you anywhere where there is a cellular 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 affect on how we access and work with information. The iPhone 3Gs along with the iPad accompanied by the App Store has increased the cadence as competitors such as RIM Blackberry, Google Droid and Palm Pre try to catch up. The industry is reaching a point of critical mass that can no longer be ignored since the iPhone and its related iPod touch and iPad devices have become a powerful force. The revolutionary App Store makes it easy for users to download applications and has surpassed over 140,000 apps with over three billion downloads. It is becoming a ubiquitous mobile computing device going beyond just being a phone and therefore blurring the classification of a netbook and augmenting laptops. It can no longer be berated as a toy with faddish applications that have no use other than for entertainment or amusement.

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 mindset that PCs were just toys was analogous to how skeptics today view the iPhone 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 which changed everything and pushed mainframe computing into relics of the past. The many applications that are being developed for the iPhone demonstrate that there are many killer apps that can be a game changer as analogous to its predecessor in the PC hey day.

There have been many wars waged within the computer industry ranging from hardware to browser wars. What it all boils down to when it comes to the user's experience are the applications. In the smart phone war, the App Store has demonstrated its popularity and its ability to have applications sell other applications. In many instances, the iPhone 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 of 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".

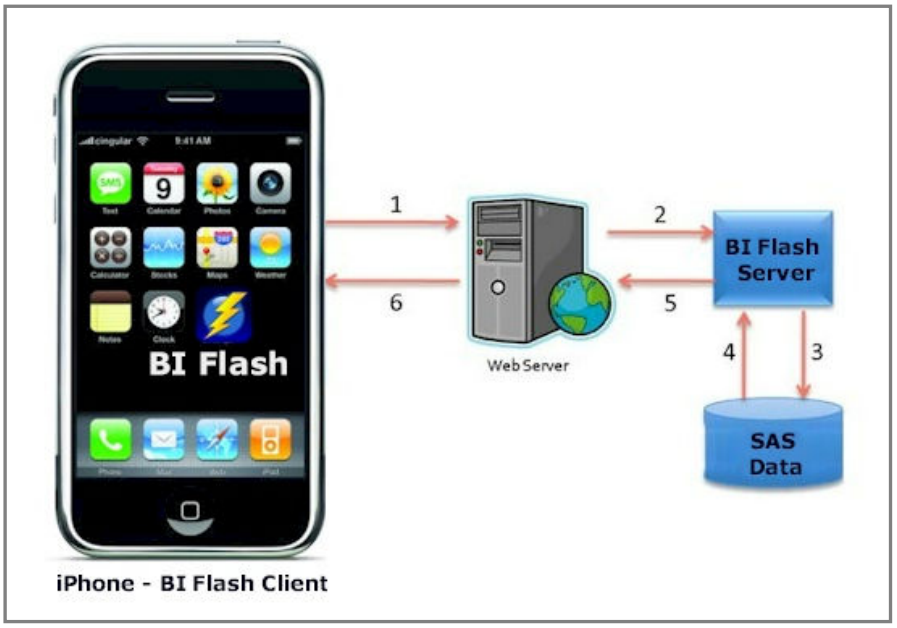
It is not too late for SAS analysts who are glued to their laptops, desktops and servers to benefit from the mobile revolution. This paper will elaborate on a few of the following areas which debunk the notion of iPHone as a mere toy and show that it can deliver serious business applications such as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

- System Architecture – How an iPhone accesses SAS data and macros
- Application Servers – SAS as an application server for iPHone applications
- 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 multi touch 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 an iPhone App. It will demonstrate how real practical and useful SAS macros 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 will illustrate and pave 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 iPhone 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 iPhone application replaces the browser. The diagram below shows the components of this computing architecture including: the iPhone application, a web server and the BI Flash application server.



In this example, the iPHone 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 iPhone in a similar way a web browser would access web pages stored on a web server. The distinction however is that the iPhone application is not a web browser and the SAS session running on the server is more dynamic compare to a static HTML page. The SAS data and macro programs requested may be simple and standard but facilitating the communication takes a little more effort. The request from the iPhone application and the delivery of information from the server is handled by BI Flash. This makes the experience more dynamic and delivers the full power of SAS on the server to the iPhone. The following steps are taken by the user in order to access SAS information on the iPhone.

STEP 1 – DOWNLOAD IPHONE APPLICATION

One of the most unique and successful aspects of the iPhone is how users can easily download an iPhone App directly from the Apple App Store. This is a user friendly way of searching for and downloading applications to your iPHone. In this case, you can search for and download the "BI Flash" application which is referred to as the client component which enables the communication between the iPhone and the SAS server.

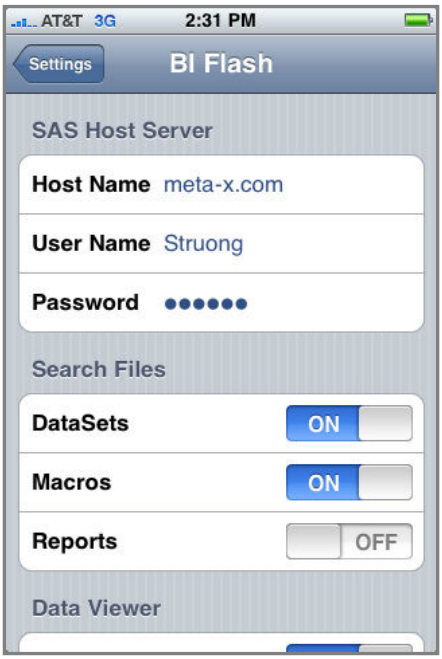
STEP 2 – BI FLASH APPLICATION SERVER

More details on this step will be explained in the "Application Server" section but an administrator would manage and utilize the BI Flash server. This server functions as a listener waiting for a request from the iPhone. Upon receiving

a request, it would process this similar to how you would submit SAS programs from display manager. The server would generate a SAS log and output results in XML which is then sent back to the iPhone to be viewed.

STEP 3 – CONNECT IPHONE TO APPLICATION SERVER

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



1. Host Name – This the name of the server or an IP address of the SAS server
2. User Name – A valid user name that has been defined on the server needed during authentification
3. 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 requirement 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 iPhone. This request is initiated from the iPhone app and sent directly to the server along with the user selected options. The results are then returned to the iPhone displaying the most updated information on the server.

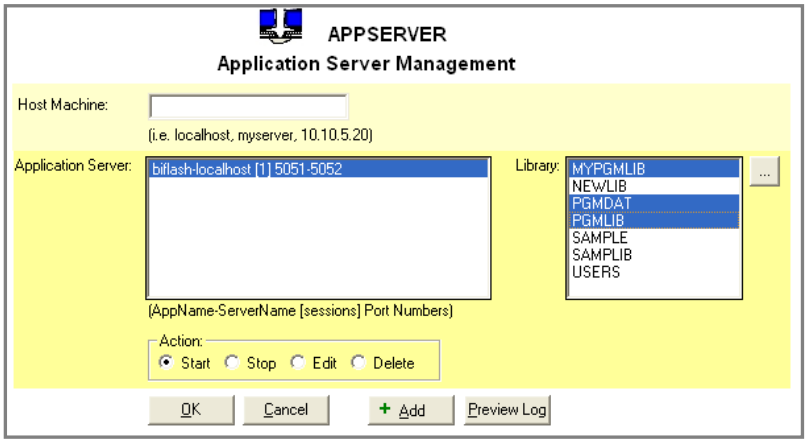
The system architecture in this example is rather simple compared to other systems that require multiple layers of middleware. This is similar to the SAS/IntrNet where users are on a web browser accessing SAS data and programs on the server through the broker and SAS application server. The difference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

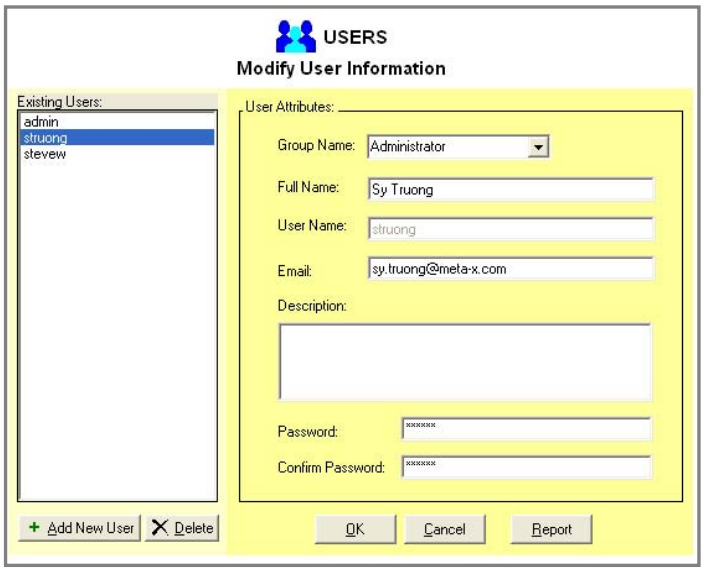
APPLICATION SERVERS

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.

The SAS program that is requested by the user also resides on the server as defined to 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 iPhone application through its own assigned TCP Socket port. This ensures that there is no collision between multiple applications for optimal performance. If needed, it is optional to increase the number of ports to handle many requests. In this case, a separate SAS session is started on the server running simultaneously to enable greater bandwidth.





SECURE USERS

There are two distinct roles that users play to enable the delivery of SAS data and programs to the iPhone including an administrator and user. The administrator is usually the SAS analyst or statistician that developed and manages the data and SAS program on the server. The iPhone user receives the reports and data onto their iPhone. Before a user can access SAS reports and data, the following steps are taken to ensure proper security.

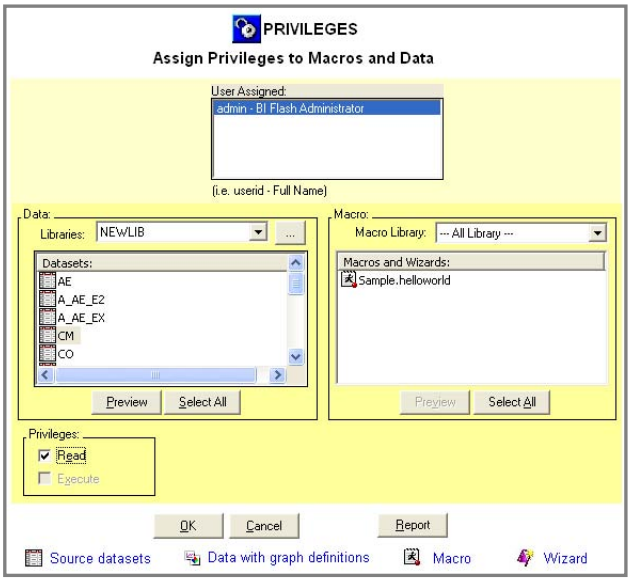
STEP 1 – ACCOUNT SETUP

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 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 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 iPhone user can see the programs appear 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 to which 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 minimal so that both the administrator and iPhone user can get to the information that is needed efficiently.

All the user permission is 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

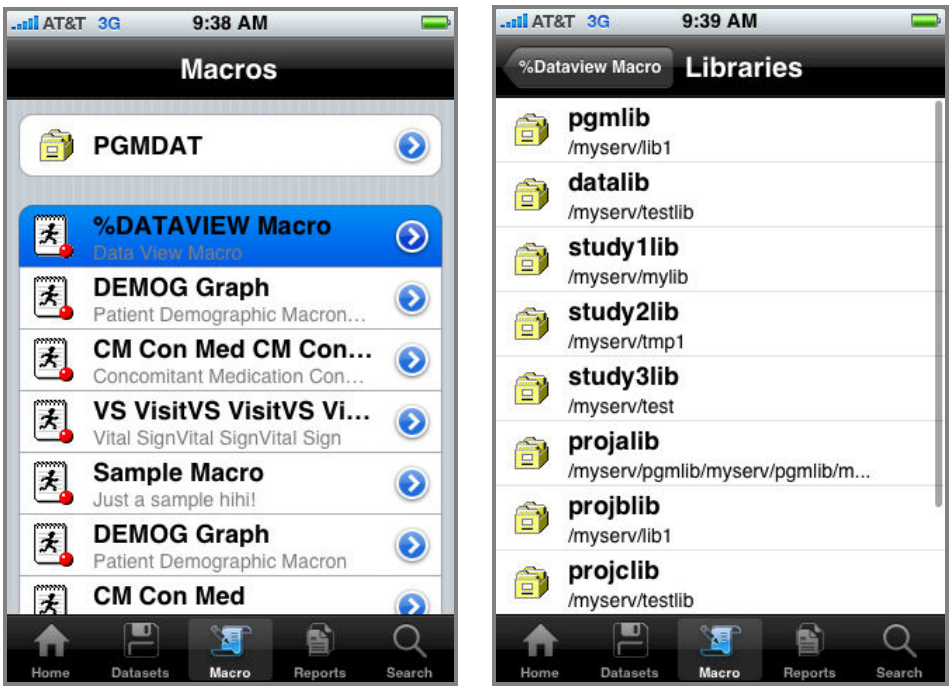
In the traditional batch environment, users specify the options for a macro by typing the selected values in a SAS program script when invoking a macro. The entered values would then be processed by the macro by inserting them into the specified parameter in the code. An example macro call is:

```
*** 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 iPhone application captures these parameters with a user friendly interface as compared to having the user write a SAS macro call. In this case, the iPhone user selects the values for the macro parameters through the multi touch interface of the iPhone user interface. The user would therefore access and execute the %dataview macro by simply performing the following steps.

STEP 1 – SELECTING MACRO

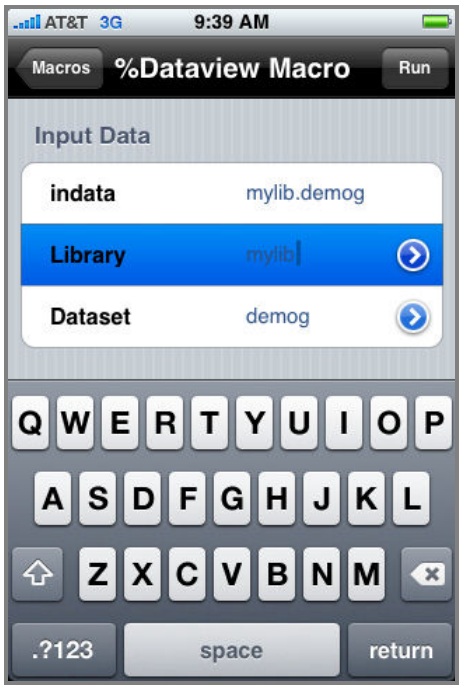
The user would navigate to the macro section by tapping on the macro button on the navigational bar at the bottom. This brings the user to a list of all the SAS program macros that the user has privileges to in the current library.



The user then navigates through the libraries by taping on the library choice as shown in the current “PGMDAT” example. In this case, they would tap on the “%DATAVIEW Macro” to drill down to its parameters.

STEP 2 – SELECTING MACRO PARAMETERS

Upon the selection of the %datataview macro, all the macro parameters will be presented with standard iPhone user interface elements. Each parameter will be listed as in the order in which it is defined.

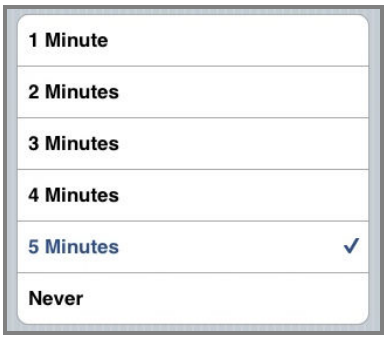


In the traditional macro approach, the user would input the values of the parameter by typing the text upon a SAS script file according to the named parameters. This is difficult for users since they are not familiar with the correct spelling of the library or dataset name. This commonly leads to erroneous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

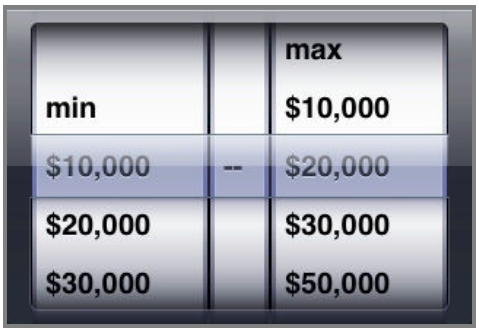
1. **Input Data –** The input data parameter uses a standard SAS two level dot notation which is LIBNAME.DATASET. In this example, it is "mylib.demog" which refers to a library "MYLIB" and the dataset DEMOG. The user can edit this as open text or in addition, there is the option of drilling down to a list for the selection of libraries and datasets in selection.
2. **Sort Variables –** The sort variables is a standard multiple option selection list. There are several different types of selection lists. In this case, a simplified multiple selection list is presented as an example.
3. **Report Title –** The report title requires a standard text entry field which the user can type any text value with the aid of an onscreen keyboard.

There are many other types of controllers which macro parameters can be associated with to make the user entry more intuitive. Keep in mind that the user is using their finger or thumb on a multi-touch screen. This is therefore different than a pointer device such as a mouse on a large desktop monitor. The following example selection options illustrate how the iPhone interface is optimized to allow users to select their parameters effectively from a mobile device.

1. **CHECK LIST**

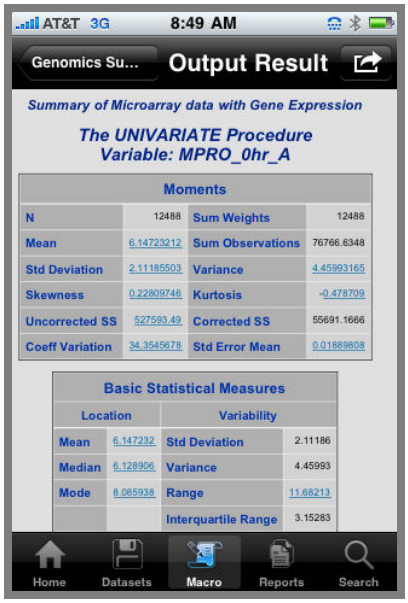


2. SPIN CONTROLLER



Some of the graphical user interface elements are similar to those found on desktop windows applications. The text entry and check list is similar to a text entry or list box on desktop applications. They only differ in their layout so it is easy for users to tap with their fingers. However, some user interface elements diverge from the desktop interface to fully take advantage of the mutli-touch and smaller screen of an iPhone. This is particularly apparent in the spinning control which looks very different compared to any controller on a desktop application. The author of the macro would configure their macros and select which controller is best suited for each parameter. 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 configures this on the server, the user can benefit from having a user friendly method of selecting and executing the macros.

Upon completion of parameter selection, the user would tap on the "Run" button on the upper right to have it execute on the server. Depending on the macro ODS option; the result can be generated in HTML, PDF, RTF or many other file formats. The output viewer supports all of these formats within a multi-touch screen which allows users to jump to and zoom in on any particular output data point.



If the user finds the particular report important to share, they can tap on the upper right corner 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 is handled on the server so the attached report does not have to be downloaded to the iPhone and then sent to the recipient. For optimization, the handling of the large report attachments and email are handled on the server.

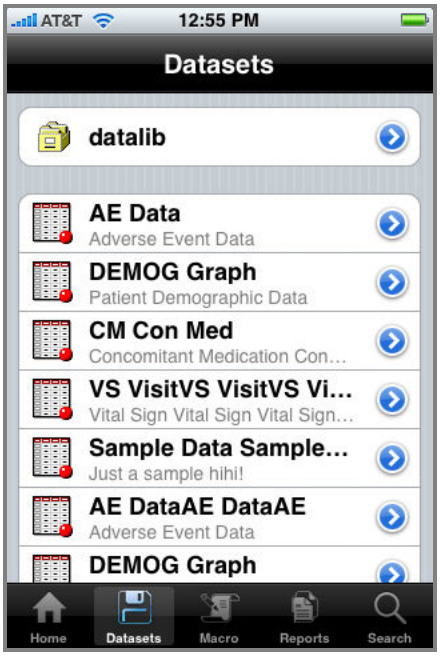
The added ability to share the information through email extends the collaboration effort which is crucial in projects that require large teams. This provides capabilities that were once only available to laptops for mobile users. The ability to do this on the iPhone will provide important data and analytics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

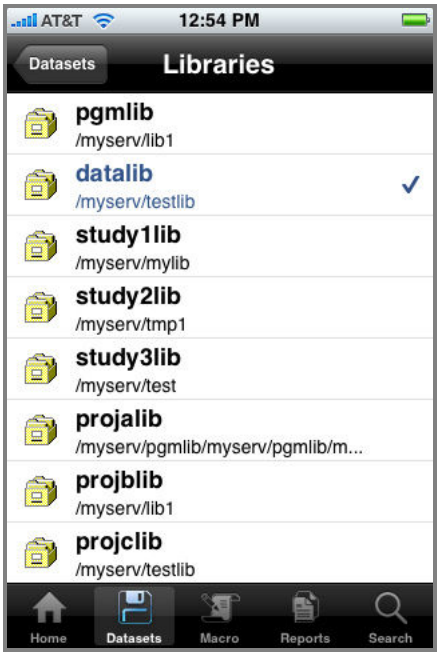
VIEWING DATA

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 view is the actual raw data. On the Windows desktop environment, you would use the SAS data viewer. However, the iPhone has different capabilities such as multi touch pinch to zoom and swipe to slide the views. These additional features can be used to view the data more effectively. The following illustrates the steps taken to view SAS data on the iPhone.

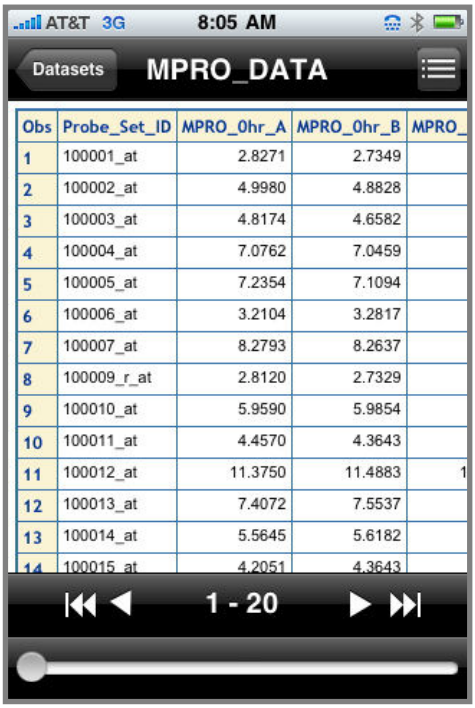
STEP 1 – SELECTING LIBRARY

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





By taping on the library right arrow labeled “datlib”, the user 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 taping 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 the vertical swipe action which is standard for the iPhone interface. The user can then select the data to be viewed by taping on the right arrow or anywhere in the row of the dataset. The text size and layout of the dataset list is optimized for the user selection within the iPhone interface.

STEP 3 – VIEW DATA

Upon the specific data selection, a data viewer is presented with all the rows of the data. Chunks of the data of delivered to the iPhone which can be configured. In this example, 20 rows at a time are displayed. The user can then jump to specific blocks of data using the next and previous buttons or the horizontal slider control to navigate to specific chunks of the data.

In the event that the data table being viewed is very large, it is broken down to smaller chunks to optimize download and viewing experience. In this example, it is shown in small chunks of 20 rows as indicated at

the bottom navigation bar of "1-20". This represents the display of the first 20 rows of data in the current dataset.

If the user were to tap on this observation count index, a "Data Chunk" selection screen is presented providing the user the ability to jump directly to any other portion of the dataset without having to scroll through a large number of rows. This has been optimized to match the multi-touch navigational interface of the iPhone allowing the user to get to their specific data point of interest quickly.

The interface has been modeled after other existing standard iPhone applications such as iPod music player which has standard buttons and layout. Apple has devoted extensive research into developing a user interface that is user friendly for the mobile device optimizing object size and placement for optimal functionality. Many of the same user elements are implemented in this data viewer leveraging upon Apple's standard to enhance the user's experience. Since users who own an iPhone are most likely familiar with the interface of an iPod to select music, they do not have to learn a new interface for selecting and viewing datasets. In the list view, the selection and viewing of datasets is analogous to playing and selecting options of each song being played on the iPod.

The code and algorithm logic that goes into developing the SAS application server and iPhone client application is extensive. This section will select one section in step 1 on libraries and describe how this is accomplished. The following steps are applied to present the list of library screens.

STEP 1 – CAPTURE LIBRARIES

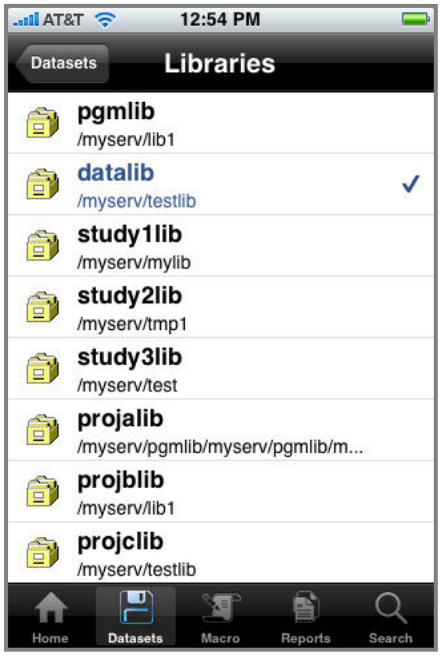
A SAS program is initiated on the server to capture all libraries available. It is then compared to the user permissions before the final list is sent to the client iPhone. The following example code is used to capture this:

```
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);
```

In this case, it captures libraries from the SASHELP.VSLIB view and inserts its findings into a list to be used later.

STEP 2 – SUBSET USER PRIVILEGES

A SAS program is initiated on the server to capture all libraries available. 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 is captured, an XML file is created capturing all the libraries as shown here.

```
<Libraries>
   <Library id="1">
      <shortTitle>pgmlib</shortTitle>
      <longTitle></longTitle>
      <descriptionTitle>/myserv/lib1</descriptionTitle>
   </Library>
   <Library id="2">
      <shortTitle>datlib</shortTitle>
      <longTitle></longTitle>
      <descriptionTitle>/myserv/testlib</descriptionTitle>
   </Library>
   <Library id="3">
      <shortTitle>study</shortTitle>
      <longTitle></longTitle>
      <descriptionTitle>/myserv/testlib</descriptionTitle>
   </Library>
...
</Libraries>
```

This is information is then delivered to the iPhone via a web server used to for the user selection.

CONCLUSION

The distribution of products and services has radically changed in recent years as it becomes more efficient for companies like Google, eBay and Amazon to take advantage of efficiencies of Internet distribution. Rather than concentrating on the top few block buster products, these companies are finding that the many niche products such as specialized 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 the large array of solutions has become profitable when efficiently delivered across the internet as compared to shrink wrap boxes sold in stores. This shift in the market is also taking shape in how software is being delivered. Apple has been at the forefront with this type of distribution with smaller unknown musical artists on its ITunes store. It is now extending that making the AppStore a new way in distributing more specialized software within this hyper efficient distribution model. This is apparent when over 50,000 applications have been made available and growing. The mobile computing platform such as the iPhone has become a potent platform to deliver useful and specialized analytics software to users. At the current moment, however, there is very little in this space for business intelligence applications on the Appstore and none for SAS. It is unquestionable that SAS can deliver sophisticated business intelligence information by delivering SAS macros coupled with its programming language to deliver tremendous analytical power. In the past, the access to these tools has 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 can be delivered to users on the iPhone. This bridges the gap by delivering endless possible aggregate views of the data to users without requiring them to understand SAS programming or the details of the SAS system. This will truly liberate business intelligence to be accessed by any user allowing them to get the latest information anytime directly from a device in their pocket.

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