

Delivering Information Everywhere using JSP and SAS®

Bryan Boone, SAS® Institute Inc., Cary, NC
Pat Herbert, SAS® Institute Inc., Cary, NC

ABSTRACT

With the advent of the Internet and the increasing mobility of its users, it has become necessary to allow immediate access to data and reports from non-desktop (wireless) devices such as palm pilots and phones. In this paper we will show how SAS technology in conjunction with JSP and Servlets will keep your users connected to their business information on demand.

INTRODUCTION

The face of business is changing and corporate executives and managers must be mobile in order to meet the fast-paced changes in customer and market demands. To meet these requirements, executives are turning to wireless devices, such as the cell phone and Personal Digital Assistant (PDA), to remain in touch with the office while attending to customer needs.

The Cahners In-Stat Group forecasts that the number of non-desktop (wireless) users will surpass 1.3 billion by 2004. Additionally, it is projected that more than 1.5 billion handsets, PDAs, and Internet appliances will have wireless capability by the end of 2004.

Wireless communications and the Internet are becoming increasingly intertwined. Using the combination of SAS and Java technology, the mobile executive can have important sales and customer data delivered wirelessly to his or her cell phone, PDA or other non-desktop device.

WIRELESS TECHNOLOGY

Wireless Application Protocol (WAP) is a global standard for applications over wireless networks. It is just one of the wireless and handheld device standards that SAS is supporting to enable the mobile user to access his or her data more efficiently. In this paper we will be using WAP as the basis for many examples but the reader should keep in mind that other standards are supported equally. Within the WAP standard, Wireless Markup Language (WML) is used to create pages that can be displayed in a WAP browser. The WML markup language contains hierarchies of screens ("decks") and links between those screens ("cards"). WML is designed to display content on wireless devices such as phones, pagers, and PDAs.

A typical WAP request for a cell phone would entail:

1. The user presses a phone key that initiates a request.
2. The phone browser sends the request to a WAP gateway using the WAP protocol.
3. The WAP gateway creates an HTTP request for the specified URL and sends the request to the designated web server.
4. The web server then processes the HTTP request as it would any other HTTP request.
5. The web server returns the requested WML deck using either a static WML deck or a dynamically generated WML deck.
6. The WAP gateway verifies the HTTP header and WML content and encodes them to binary form. The gateway then creates a WAP response containing the WML and sends it to the phone browser.
7. The phone browser receives the WAP response, processes the WML response and displays the first card of the WML deck to the user.

JAVA TECHNOLOGY

Java is a software language that is platform neutral. It is said that Java is "Write Once, Run Anywhere™". That is, an application developed in Java can be deployed to any machine that supports the Java™ Virtual Machine. Two important technologies that build on Java are Java™ Servlet and JavaServer Pages™ (JSP™). When used in conjunction with wireless technology, developers can use their knowledge of these technologies to provide dynamic delivery of data such as WML. Java's component-based technology, JavaBeans™, makes it easier to build web pages using JSPs and Servlets. JavaBeans separate the user interface from the application logic. This enables the page designer to focus on writing the presentation layer while allowing the application developer to generate the dynamic content portion of the page using Java and JSP. Additionally, JavaBeans provides an integration standard. This is important for developing solutions in heterogeneous environments within the enterprise or across the Internet. SAS Institute Inc. adheres to this standard and offers component-based JavaBeans that allow easy access to complex SAS resources.

SAS TECHNOLOGY

SAS Institute Inc embraces Java technology. AppDev Studio™ the first Java-based development solution to be tailored for the information delivery environment, is a complete suite of application development tools for building thin-client Java applications. Applications written using AppDev Studio™ can tap into SAS resources through webAF™.

SAS's component technology, webAF™, makes it easy to use a standards-based approach to access SAS from the Web. webAF software is a Java framework that enables access to SAS/AF® objects, tables (data sets), multidimensional databases (MDDBs), and other SAS computing resources. webAF® provides data models that enable developers using JavaServer™ Pages (JSP™) to create dynamic content that maximizes the capabilities of the SAS System.

The power of webAF's JSP support lies in its InformationBeans™ and TransformationBeans. InformationBeans encapsulate SAS data by presenting it as webAF data models. The webAF data models are then, in turn, consumed by TransformationBeans, which transform the model into appropriate presentations. A key to integrating SAS into wireless technology lies in the server-side processing of the HTTP request. Here the TransformationBeans display SAS data for the appropriate wireless enabled device or PDA.

CONCLUSION

Changes in today's business market require corporate executives and managers to be more mobile. At the same time, cell phones and PDAs are becoming more common in the workforce and have a tremendous potential to provide more information to those mobile professionals. Wireless technologies facilitate the use of JSPs and Servlets on the web server, where the user's request is ultimately evaluated. SAS technologies, particularly InformationBeans™ and TransformationBeans found in webAF, deliver SAS data wirelessly to cell phones, PDAs and other non-desktop devices, keeping the user connected to his or her business at all times – providing "The Power to Know™".

The complete copy of this paper can be found at
<http://www.sas.com/usergroups/sugi/sugi26/sipapers>

CONTACT INFORMATION

Your comments and questions are valued and encouraged.

Contact the authors at:

Bryan Boone
SAS Institute Inc.
SAS Campus Dr.
Cary, NC 27513
(919) 531-6039 (Voice)
(919) 677-4444 (Fax)
Bryan.Boone@SAS.com

Pat Herbert
SAS Institute Inc.
SAS Campus Dr.
Cary, NC 27513
(919) 531-7618 (Voice)
(919) 677-4444 (Fax)
Pat.Herbert@SAS.com

SAS is a registered trademark or trademark of SAS Institute Inc.
in the USA and other countries. ® indicates USA registration.