

most significant bits

News and Information For High-Tech Professionals

Published by Pocket Protector Press™

A division of Stout Systems



Fall 2007

In This Issue

.mobi—The New Internet Frontier

A whole new Internet chapter is unfolding.

A Java Web Application...Using Only Java?

Developing Web apps with Java has normally involved a whole lot of non-Java tasks. Are those days numbered?

Recent News

Stout Systems welcomes new employees Steve Clark (Software Engineer Consultant), David Smith and Robert Dennis Jr. (Database Consultants), Matthew Wickey (Project Management Consultant).

Job Openings

Check out current permanent and contract openings on our Web site StoutSystems.com.

Current Candidates

A sampling of job seekers we represent is available on StoutSystems.com.

Subscribe to Our Newsletter

If you would like to receive this newsletter directly at your desk or in-box, send a note to info@stoutsystems.com providing your subscription info (e-mail or ground mail address).

Call us:

(734) 663-0877

Every business is going to need one...

.mobi—The New Internet Frontier

by Brian Skory

It seems that one day I blinked and there was a whole new chapter of the Internet unfolding in the form of .mobi Web sites. I knew about certain sites that had been optimized for viewing on a PDA screen—like a scaled down Google screen on someone's Web-enabled phone—but not having a Mobile Data Device myself I really hadn't paid much attention to it. Then one day I saw my son coding a "fun and games" site designed to be viewed over web-enabled phones, and thus began my education into the .mobi scene.

Why a .mobi site?

The fact is that despite the technical solutions available for compressing and tailoring .com Web sites for small screen display, these sites usually remain time consuming and expensive to download. And although there are several specialized content adaptation solutions, they are rarely consistent and often times ineffective. .mobi eliminates the need for these intermediary steps since the sites have already been specifically designed for viewing on mobile devices.

What can you do with a .mobi site?

Going to get your horoscope or joke of the day can be a welcome distraction while standing in line at the bank. But how about logging into your bank's .mobi site and quickly finding an account balance or a specific transaction without having to horizontally scroll all over the place. Or how about when you're out shopping and you realize that a carry out from your favorite restaurant would be just the thing

to top off the day? Log onto their .mobi site, check out their menu (nicely optimized for your screen), make your selections and click on the "place order" or "dial restaurant's phone number" button. These may not be immediately available for your current bank or restaurant, but they will be. Currently, you can access news at CNNMoney.mobi, check the weather at Accuweather.mobi, check your train status at Amtrack.mobi, and keep your kids entertained with Disney.mobi, just to mention a few.

Cool Code

Let's say you're building a menu site for Ribs-R-Us. Your homepage comes up with links to Appetizers, Main Dishes, etc. Typically you would need to scroll down to the link you're interested in and then select it. This, however, would garner you a lower score on a ".mobi readiness test". A better way to do it would be to utilize Access Keys. Access Keys allow a user to quickly navigate to a link on a page by assigning one of the keypad keys to that link. The HTML code to execute this is as follows:

```
[1]<a accesskey="1" href="link1.html">Link 1</a>
[2]<a accesskey="2" href="link2.html">Link 2</a>
```

Brian Skory is a Technical Staffing Consultant with Stout Systems. He has spent the better part of twenty years in the technical trenches doing everything from network administration to slinging espressos. E-mail brianskory@stoutsystems.com.

Editor's Note: For more cool code and additional info on .mobi, visit our Web site for the extended version of Brian's article.

stoutsystems.mobi coming soon!

A Java Web Application...Using Only Java?

By Bill Heitzeg

Developing Web applications with Java, especially applications that need back-end data sources, has normally involved a whole lot of non-Java tasks—including HTML and XML. In the past year I've had the good fortune to leave it all behind and create Web applications using only Java.

Before getting too deep into this, let me explain a little about the Web applications that I've worked on in the past. I started writing Java Enterprise applications (J2EE) about five years ago. Almost all the work I did up until 2005 involved communications services, especially Web services. J2EE is really nice for this kind of thing and makes Web services a breeze.

Two years ago I was helping one of my clients with Web services, when an emergency pulled me away to help one of my client's clients implement the services we had created. It turned out that the problem wasn't the services, but the Web application. I worked on this project for the next nine months, and I learned quite a bit about the limitations of JSP and many other template-based solutions.

The Issues

For me, the two most difficult things about working with JSP have been implementation decisions and debugging. Even though I never really liked it, it's not very difficult to get used to switching between JSP/HTML pages and Java code.

What caused problems for my customer was the lack of a clear way of implementing its design. This has already been a multi-year project before I started, so several developers had come and gone.

- Some developers put their code right in the page.
- Some developers put their code in plain-text page libraries that they included in the main pages.
- Some developers used tag libraries.
- Many wrote their code inside a Servlet.
- Near the end we started using JSTL because we didn't feel there was enough variation!

As you can imagine, debugging was a nightmare. You may say that these people implemented JSP wrong, but that's just the problem. There are so many ways to implement JSP that doing it wrong isn't all that difficult. Because of this inherent lack of a clear path, the scenario we were facing isn't all that uncommon.

Enter Google Web Toolkit

In May of 2006, I started a new Web application project. The goal was to import mobile phone bills, analyze them, and show customers how to save money. This was a classic example of a Web application that needed to navigate data intelligently and cleanly.

A month before the start of this I was frantically trying to decide what to do. I played with Ruby on Rails, .NET, Spring, Tapestry and a couple of different JSP solutions.

One day I got an email from my friend Dianne Marsh, who told me Google had just released a Java Ajax toolkit that I should check out. I went to the Google Web Toolkit (GWT) site, checked it out, and in two days I had written the first prototype for the new mobile savings application.

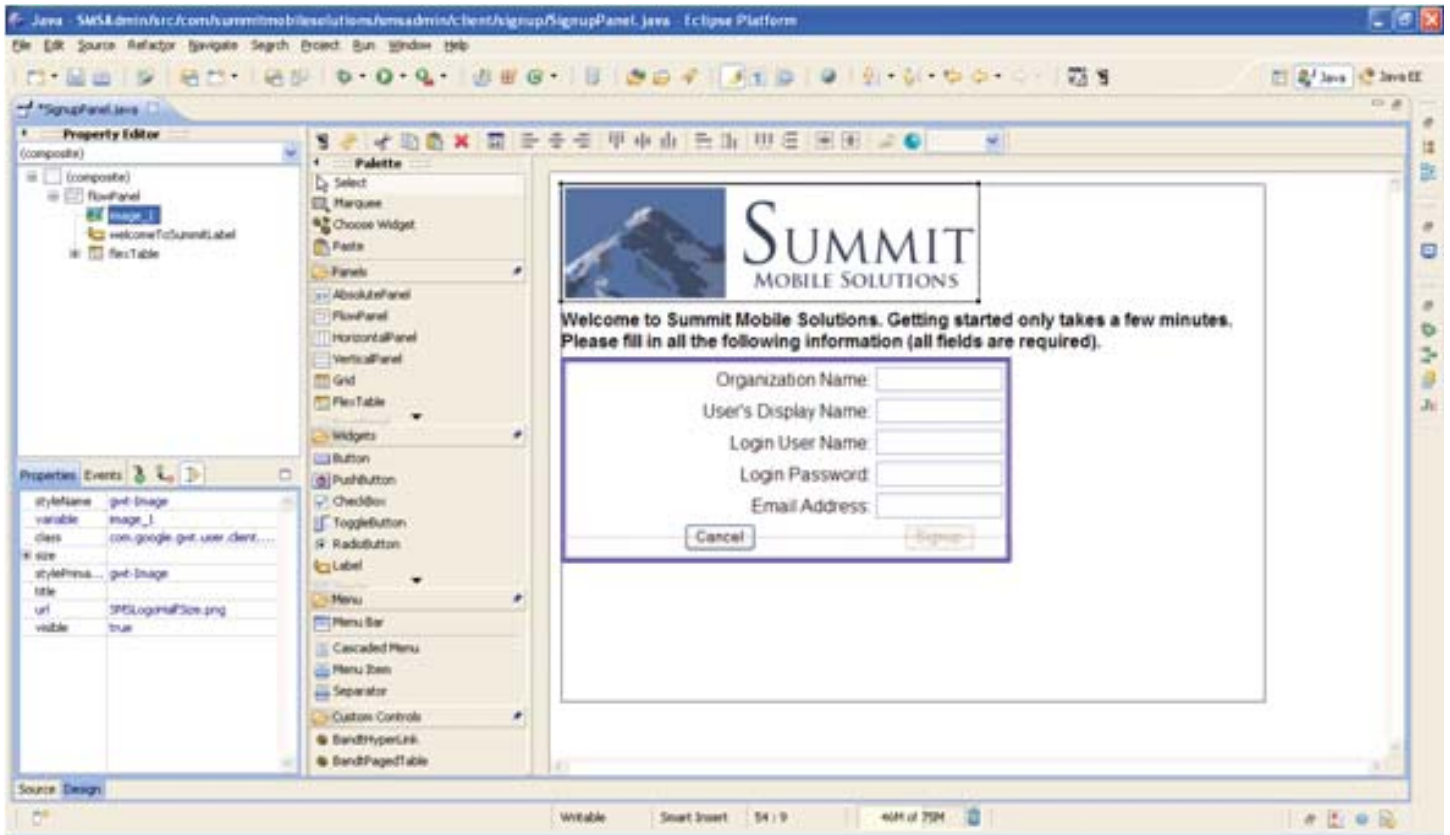
Why did I get hooked on GWT? The answer is simple. You write only Java! There is absolutely no need to write any HTML or XML. The code looks a lot like Swing—or any of the many GUI OO languages. It's clean and debug-able, and unit tests can be created easily against any and all code. In addition, because I chose GWT, I also was able to continue to leverage my Java and J2EE knowledge and to use Hibernate with Annotations for my Object Relational Mapping (ORM). The application is now on the market, works very well, and we spend all our time adding new features instead of rewriting old ones. You can see a demo at summitmobilesolutions.com.

GWT—What is it?

GWT is a pretty simple concept when you think about it. It's a Java to JavaScript compiler. You write code in Java, and GWT compiles it into JavaScript that works in a standard browser. One of Google's biggest issues seems to have been to create a common way of writing code that would be browser agnostic. They're not there yet, but I like to think that write-once-run-anywhere has been moved a couple of steps forward.

```
signupButton = new Button();
flexTable.setWidget(6, 1, signupButton);
signupButton.setTabIndex(7);
signupButton.setText("Signup");

signupButton.addClickListener(new ClickListener()
{
    public void onClick(Widget sender)
    {
        if (signupButton.isEnabled())
        {
            if (validating)
                validateSignup();
            else
                beginSignup();
        }
    }
});
```



Most people might think that a Java to JavaScript compilation would mean a debugging nightmare. Google solved that by creating an emulator that allows you to run the code, set breakpoints and even do on-the-fly changes. Coming from a JSP world, this completely blew my mind. In over eighteen months of using GWT I have yet to look at the JavaScript it creates.

In addition to debugging, unit testing works like any other Java application. Since all of the code is standard Java, you can simply write unit tests against any of the code and run it using the standard Java Runtime Environment. If you need the ability for your unit tests to run against the hosted code, Google made that possible using the emulator and a special test case class from GWT.

GWT works great with Eclipse. It ships with a tool to create an Eclipse GWT project and for less than \$50 you can get GWT Designer, which among other things allows you to create Eclipse code using a WYSIWYG designer. I've written about twenty different applications now with GWT and I find the Designer to be well worth the money.

If you need server side code, and I can't imagine not needing it, GWT ships with two solutions. You can either use standard JavaScript Notation Object (JSON), something I've never done, or use GWT Remote Procedure Calls (RPC), which works with Servlets. This means that you end up with two code bases—a client code base and a server code base. The client code base becomes

JavaScript and the server code base is deployed as a Servlet. In practicality, you simply wrap it all up in a Java Web Archive (WAR) and deploy it on Tomcat or another standard Servlet container. I use JBoss, but that's because I want to take advantage of some other J2EE features on the server side.

Since the GWT server side is just standard Java deployed on a Servlet container, you can use all your favorite Java libraries. For me, this was huge. My application needed strong database support and I desperately wanted to use Hibernate with Annotations. It also meant that I could use Spring, Log4J, Jackcess (to import access files) and so many other mature Java libraries on the server side.

Obviously I can't speak more highly of GWT. If you're already familiar with Eclipse and Java, it will probably take you less than an hour to take GWT for a test drive. If you're faced with writing a Web application and you already love Java, this will be time well spent.

Bill Heitzeg is Vice President of Technology for Summit Mobile Solutions. He is also the founder of Emerald Software, a consulting firm that specializes in Web service and Web application design and development. Email him at bill@summitmobilesolutions.com or visit his google group at <http://groups.google.com/group/summitgeeks>.

Going on eleven years...

2008 is Looking Great

by John W. Stout

Remember it's DEMAND that fuels an economy—not the stock market, not the Federal Reserve, not the news media. Any time or place where demand for goods or services exists, opportunistic individuals and companies find a way to service that demand.

Right now there is no less demand for talented high tech people and services than before. In fact, there are plenty of well-paying, stimulating opportunities in the Midwest. So the most important thing you can do as a job seeker is to not agree that things are tough. Decide what you want, go after it and you will find it.

As Stout Systems concludes its 10th year in business, I want to thank all of our customers for your long commitment to our services and for helping us make this the most successful year in our history. You have my promise that we are committed to helping you in any way possible, and that we will continue to refine and improve what we do for you.

See you all on the flip side in 2008!

John W. Stout is the founder and president of Stout Systems Development. He has nearly thirty years' experience in the software industry. He is also sought after as a technology speaker, presenting sessions at developer conferences and user groups. E-mail john@stoutsystems.com.



ADDRESS SERVICE REQUESTED

Presorted Standard
US Postage Paid
Permit #187
Ann Arbor MI

In This Issue:

- .mobi—The New Internet Frontier
- A Java Web Application...Using Only Java?