

Preface

I've always had an interest in the nontechnical side of software development: the user experience. It started back when I was working on teams building the core of application servers in C++. We admired the beauty of the C++ language and its expressiveness. We made large, complex systems run seamlessly with elegant code. We marveled at our templating techniques, which made the C++ compiler churn out code just like a code generator would. Then I would leave work and was not able to mention a word of it without receiving blank stares in return.

I decided to find time to write a client-side application that would be as elegant to the user as well-written code can be for a developer. I chose to build an instant messenger application, mostly with C++, that combined the four major networks into one interface. At the time, instant messengers were becoming bloated with features—there were too many buttons distracting users from sending a simple text message. The instant messenger application I developed resulted in a much better user experience for instant messaging: instead of users downloading a 10MB application with a five-step installation process, I optimized the messenger to be 200K with a clean interface (much like the Google Talk messenger is today). As a result, it was downloaded over a million times.

While developing interfaces in C++ I was always impressed by the ease of creating a nice-looking interface on a web page. If you compare the code required to set a font in C++ to cascading style sheets, you'll see what I mean. Then Ajax started to become popular, producing web interface behavior similar to desktop interface behavior. Combine this with the ease

of making things look better with CSS, and you have a much better platform for interface development.

I was really impressed when I saw Google Maps for the first time. The user experience was perfect. I simply typed `maps.google.com` into my browser and I was instantly provided with a fully functional map application. I could drag the map around in different directions, traveling around the world, zooming in and out without waiting for a page refresh. I had a brief look at the technology needed to do this, specifically JavaScript, and was disappointed. I knew there were limits to what you can build with JavaScript. It would be nearly impossible to build large complex client-side applications with it.

Then the Google Web Toolkit (GWT) was released, and I decided to try writing an application using it. In only three weeks I had built the client and server side for a poker application. I put it up at <http://gpokr.com>. You could simply type the URL into your browser and be instantly presented with a live poker game. No downloads, no installations, and the interface could be styled nicely and easily with CSS. Scott Blum, Bruce Johnson, and Joel Webber from the GWT team came by to do some “testing,” and I had the opportunity to thank them for building an incredible tool. I marveled at being able to write elegant Java code that could be transformed into JavaScript by the GWT compiler. I was really impressed by how GWT so solidly let anyone create applications that delivered great user experiences.

After GWT’s initial release, I found that its great abilities weren’t clear to many and that it would take a book with several real examples to illustrate this. I had never written a book before, and to write one on a technology that was not my specialty didn’t seem quite right. But then again, nobody specialized in GWT at this point. I believed enough in the technology to give it a shot. To make up for my lack of experience and before writing any of the chapters, I spent several months exclusively developing GWT applications to explore every part of GWT as well as every part of web technology that GWT could touch. Part II of this book presents five of these applications.

What Is This Book About?

This book is about writing nontrivial Ajax applications to create great user experiences using web technologies and Java development tools, with

GWT bridging the two. The book focuses primarily on the Google Web Toolkit, with an in-depth look at its library and tools. As a secondary focus, it covers software development techniques and patterns using Java, and how to apply Ajax application development with GWT. A tertiary focus is on web technologies, including web standards and other Ajax libraries and APIs.

Who Should Read This Book?

I'm a developer who wrote this book for other developers. Software developers who need to create user-facing applications should read this book. Most of the code in the book is based on Java, but care is taken so that the book is accessible to a beginner with the language. If you don't know Java, you should familiarize yourself with the language before starting this book. Sun has great tutorials to get you started: <http://java.sun.com/docs/books/tutorial/java/index.html>.

GWT is not just an Ajax tool for Java developers. I think this view severely undercuts its true strength. Java developers will find using it easy; however, the technology is for any software developer who needs to build nontrivial Ajax applications. You could be a .NET, PHP, Ruby, or C++ developer. If you're one of these developers you would need to learn another language to build an Ajax application whether you use GWT or not. I recommend that you learn Java—starting with the previously mentioned tutorials from Sun, and GWT through this book and the GWT documentation at <http://code.google.com/webtoolkit/documentation/>—instead of JavaScript. As a result, you will save a substantial amount of time debugging and maintaining the application while creating a much better user experience.

Organization of This Book

This book has two parts. Part I gives you an in-depth introduction to using the Google Web Toolkit. You can use it as a reference for the GWT library or as a guide to using effective development techniques with GWT. Part II provides a thorough look at five nontrivial applications built with GWT. In this part you'll find development patterns, techniques, and subtleties used through application design and development. Each application in this part is designed to be a balance of GWT library usage, web service and

technology interoperation, application design and architecture, and user interface design. As you read through these chapters, you can follow along and construct the applications on your machine. The chapters include most of the code, but you'll need to refer to the source code at www.gwtapps.com in certain instances that are identified.

Part I: Understanding the Google Web Toolkit

- Chapter 1, *First Steps with the Google Web Toolkit*, introduces web technologies, skill sets, and GWT, and includes a short tutorial on creating an Ajax game application.
- Chapter 2, *User Interface Library Overview*, details the user interface library that comes with GWT. This material consists mainly of notes and examples based on the usage of each widget.
- Chapter 3, *Server Integration Techniques*, describes several methods for integrating with server-side applications.
- Chapter 4, *Software Engineering for Ajax*, looks at Java tools for software development and how they apply to GWT development.
- Chapter 5, *Using the Toolkit Effectively*, covers some of the more advanced techniques of development with GWT, including CSS, code generation, internationalization, and performance.

Part II: Rich Web Applications by Example

- Chapter 6, *Gadget Desktop Application*, presents a gadget application with a rich drag-and-drop interface, persistence with cookies and Gears, along with using JavaScript APIs with GWT.
- Chapter 7, *Multi-Search Application*, shows how to create a search application that makes requests to many search engines. The application uses JavaScript Object Notation with Padding (JSONP) to communicate with Google, Yahoo!, Amazon, and Flickr.
- Chapter 8, *Blog Editor Application*, walks you through an application to manage blog entries across many blogs. This application integrates with the Blogger REST API using an HTTP proxy.
- Chapter 9, *Instant Messenger Application*, details a web page instant messenger based on GWT-RPC. It covers how to use an event-based protocol along with optimizing with Comet on Tomcat and Continuations on Jetty.

- Chapter 10, Database Editor Application, looks at a database manager for a traditional web page. The application explores advanced topics such as reading complex data structures from the server using Data Access Objects, code generation for easy XML and JSON, and integrating with PHP, Ruby on Rails, and Java with Hibernate.

Web Support

The web site for this book is located at www.gwtapps.com. It contains the source code and live demos for the sample applications, a forum for questions and error reports, and other useful reference material.

Acknowledgments

I thank the GWT team for creating such an interesting technology to write about. Also, thanks to the great people at Prentice Hall including John Wait, who had an inspirational interest in design and doing something different; Rebecca Greenberg for her excellent copyediting work; Chris Zahn for his attention to detail and enthusiasm; Raina Chrobak for keeping things organized and not freaking out over my constant TOC changes; and Julie Nahil for managing production.

Thanks to all of the reviewers—Sandy McArthur, Jason Essington, Bruce Johnson, Massimo Nardone, and Mark Volkmann—who found mistakes and made suggestions that made this a much better book.

Finally, thanks to the thousands of members of Gpokr and Kdice who tolerated a lack of updates and bug fixes for six months.