

# Rocky Mountain Software Symposium 2006

Renaissance Suites Flatiron - Denver, CO

November 10 - 12, 2006

(session listing as of 11/12/2006)

The No Fluff Just Stuff Software Symposium 2006 tour is pleased to announce the Rocky Mountain Software Symposium coming to Denver on November 10 - 12, 2006. You will have the opportunity to attend the best technically focused Java / Open Source event anywhere. We make this claim based on the following:

- 1) **Excellent Speakers with unparalleled access**
- 2) **Limited Attendance = 250 Registrants Max**
- 3) **No Vendors, No Sales Pitches, No Marketecture**
- 4) **Unmatched Value - less than 1/3 of the cost of a national conference**
- 5) **Since 2002, we have delivered over 75 conferences throughout North America**

The No Fluff Just Stuff Software Symposium Series caters to individual developers, development teams, project managers, architects and independent consultants. The Rocky Mountain Software Symposium will offer 6 concurrent sessions over three days with over 66 sessions to choose from. The following topics will be featured:

- 1) XML / Web Services
- 2) Architecture
- 3) Client side Java
- 4) Core Java
- 5) Groovy
- 6) ServerSide Java

## Registration Fees

Attendees	Before 10/23/2006	After 10/23/2006
1-4	\$725	\$825
5-9	\$650	\$725
10-14	\$625	\$700
15-24	\$600	\$675
25+	\$575	\$650

## The Registration Fee includes the following:

- 1) All Access Pass to the three day symposium
- 2) Handouts from all sessions attended w/binder
- 3) CD with all presentational content @ registration
- 4) Great NFJS Swag
- 5) Opportunity to win an iPod everyday during the symposium raffle

Go to <http://www.nofluffjuststuff.com> for more details.

Questions/Comments: Contact Jay Zimmerman: [zimmerman@nofluffjuststuff.com](mailto:zimmerman@nofluffjuststuff.com) or (303)469-0486.

# Rocky Mountain Software Symposium 2006

Renaissance Suites Flatiron - Denver, CO

November 10 - 12, 2006

(session listing as of 11/12/2006)

## **Introduction to TestNG, the next generation testing framework for developers by Andrew Glover**

No one will argue that JUnit has positively affected the quality of thousands of Java applications around the world. JUnit's simplicity and ease of use ushered in a whole new era of code quality; however, as many developers have found, its simplicity has also limited its use. TestNG was designed from the ground up to overcome some of JUnit's limitations; moreover, TestNG's features make it a great tool to complement your JUnit tests.

## **Practically Groovy: Real World Groovy for Thrill Seekers by Andrew Glover**

The key to incorporating any tool into your development practice is knowing when to use it and when to leave it in the box. Dynamic languages can be an extremely powerful addition to your toolkit, but only when applied properly to appropriate scenarios.

## **Unit Testing Java Objects with Groovy by Andrew Glover**

What makes Groovy particularly appealing with respect to other scripting platforms is its seamless integration with the Java platform. Because it's based on the Java language (unlike other alternate languages for the JRE, which tend to be based on earlier predecessors), Groovy presents an incredibly short learning curve for the Java developer. And once that learning curve has straightened out, Groovy can offer an unparalleled rapid development platform.

## **Improving Java code quality with code auditing tools by Brian Goetz**

Does your program have bugs, despite unit tests, integration tests, and code reviews? You bet. Fortunately, there are some new code auditing tools that can help spot some bugs missed by other approaches.

## **Introduction to Java threads by Brian Goetz**

The Java language included support for threads and concurrency from day 1, but writing correct multithreaded programs is not easy. This session will cover the how and why of using threads in Java.

## **Java Performance Myths by Brian Goetz**

Performance myths about the Java platform abound, from the general "Java is slow", to the more specific "reflection is slow", "allocation is slow", "synchronization is slow", "garbage collection is slow", etc. Many of these myths have their root in fact (in JDK 1.0, everything was slow); today, not only are many of these statements not true, but Java performance has surpassed that of C in many areas, such as memory management.

## **Structuring concurrent applications in JDK 5.0 by Brian Goetz**

JDK 5.0 is a huge step forward in developing concurrent Java classes and applications, providing a rich set of high-level concurrency building blocks.

## **The Java Memory Model by Brian Goetz**

What's the worst thing that can happen when you fail to synchronize in a concurrent Java program? Its probably worse than you think -- modern shared-memory processors can do some pretty weird things when left to their own devices.

## **ACEGI Authentication - The AJAX way by Brian Pontarelli**

Learn how to use ACEGI in an AJAX friendly way so that when you user's sessions expire, they can quickly log back into the application and continue where they left off. This will cover in detail the changes that were made to the ACEGI framework to support AJAX authentication, why ACEGI needed to be changed and how to implement this solution in a new or existing application that uses the ACEGI framework for authentication.

## **Jini - Not just for your toaster anymore by Brian Pontarelli**

This presentation covers all the basics of the Jini platform, which has recently been transitioned from Sun to Apache. This presentation will show how to construct a service based application using Jini as well as how the Jini network is structured and deployed. In addition, a demonstration of the cool Jini features such as dynamic discovery, recovery and provisioning will be given.

## **Ajaxian Faces by David Geary**

JavaServer Faces is a perfect platform for implementing Web 2.0 interfaces with Ajax. This session explores how you can use these two potent technologies--JSF and Ajax--together to create applications that look and behave like desktop applications but run in the browser.

### **Hands-on Rails by David Geary**

Come to this exciting preview of one of the leading web application framework contenders with the potential to be the Next Big Thing: Ruby on Rails. An innovative framework with an eye-popping array of ultra-cool features such as active record and native support for Ajax, Rails greatly simplifies web application development and puts the joy back in software development. Rails is easy, fun, and very productive; in fact, in the throes of Rails-mania, some converts have claimed that developing with Rails is at least 10 times as fast as your favorite Java framework. Could that be? Come see for yourself.

### **JSF: State of the Art by David Geary**

In 2005, JSF hit its stride, as evidenced from overwhelming support from both vendors and the open-source community. JSF 1.0 had plenty of holes, but open-source projects have arisen to address those needs. This session takes a look at three of those projects: 

- Tomahawk (MyFaces component library)
- Facelets
- Seam

### **JavaServer Faces: A Whirlwind Tour by David Geary**

JavaServer Faces (JSF) has arrived. The standard Java-based web application framework based on Struts, JSF really took off in 2005. Embraced by developers, vendors, and open-source projects, JSF has started to hit its stride. If you haven't come up to speed on JSF basics, this is the place to start.

### **Shale: Turbo-charge your JSF Apps by David Geary**

JavaServer Faces is a well designed user interface framework, but it lacks a number of features you might otherwise expect out of the box; for example, JSF does not explicitly provide support for client-side validation. So, from the folks that brought you Struts, comes Shale, a collection of useful enhancements to JSF. A top-level Apache Software Foundation project, Shale adds some really cool features to vanilla JSF, including: 

- Web flow: script dialog flow
- Remote Method Calls: easily call JavaBean methods from JavaScript
- Tapestry-like views: code views in pure HTML
- Use Apache Commons Validator validators on the client or server, or both
- JSF testing framework: mocks for easy JSF testing

 There's a lot of cool stuff in Shale that makes JSF a much more compelling proposition. Come see what it's all about.

### **The Google Web Toolkit by David Geary**

Developing highly interactive web applications, for the most part requires knowledge of a wide array of technologies: HTML, CSS, JavaScript, XMLHttpRequest, JSP, JSF, etc. With the Google Web Toolkit (GWT), Google turns that notion of development on its head. Instead, you implement Ajax applications by writing almost entirely in Java. You use an AWT-like API, which the Google compiler compiles to JavaScript that runs on the client.

### **Automating Business Value with FIT and FitNesse by David Hussman**

The presentation will briefly discuss stories, the origin and authoring of story tests, and a demonstration of how FIT and FitNesse (FIT living within a Wiki) can be used to automate acceptance tests.

### **Creating, Telling, and Tracking User Stories by David Hussman**

The participants of this session will become agile customers and product owners, using personas to create stories for a sample product development.

### **Losing Battles and Winning Wars: Adopting Agile by David Hussman**

Adopting agile is different for each company, but most companies will go through some amount of change during the adoption of agile.

### **Ready, Set, Agile? by David Hussman**

As with many methodologies, moving agile into an organizations poses larger challenges. Before jumping in, it helps to ask a few questions before "racing toward agility". This session will provide 3 tactical steps that can help your adoption of agile.

### **Java Collections Power Techniques by Glenn Vanderburg**

The Java Collections framework is a cornerstone of Java development. It's been a part of J2SE for six years now. Every Java developer knows it#how to create Lists, Maps, and Sets, how to put things into them and take things out, and how to iterate over the contents. But there's a lot more to the collections framework than

that -- and very few programmers really know how to exploit the power that's just under the surface.

### **JavaScript Exposed: There's a Real Programming Language in There! (Part 1) by Glenn Vanderburg**

With the sudden importance of Ajax, it's time to take JavaScript seriously. That means learning it the right way: looking at the fundamentals of the language and surveying its strengths and weaknesses, instead of just copying other people's poorly written examples.

### **JavaScript Exposed: There's a Real Programming Language in There! (Part 2) by Glenn Vanderburg**

Building on part 1, this talk dives deep into JavaScript's object model. We'll see how it differs from more mainstream object-oriented languages, and why. We'll explore how to hide some of those differences, as well as the reasons you might not want to. Additionally, we'll cover useful tools for JavaScript testing, debugging, and profiling.

### **Modern Project Infrastructures by Glenn Vanderburg**

The support infrastructure for your software project is a crucial factor for success. A new generation of tools offers significant benefits over their predecessors. This talk discusses how to choose the right mix of tools for a top-shelf project infrastructure.

### **Advanced Hibernate by Justin Ghtland**

Hibernate is easy to get started with, but can sometimes be hard to make efficient or secure. In fact, the default settings for Hibernate create applications that will run slowly, cause unwanted round trips to the database, and may be more restrictive and/or permissive from a security standpoint than you would otherwise want.

### **Ajax Architecture by Justin Ghtland**

Ajax applications have unique architectural challenges and opportunities. This presentation will show you how to take advantage of the Ajax's strengths, and work around its quirks.

### **Introduction to Hibernate by Justin Ghtland**

O/RM (Object/Relational Mapping) seeks to eliminate repetitive or tedious work enabling the CRUD (create, read, update, delete) that underlies most applications. Hibernate is a popular, open-source O/RM tool that uses reflection (instead of code generation, like EJB, or bytecode injection, like JDO) to manage your persistence layer. This session will introduce you to Hibernate. After an overview of common usage scenarios, including web and enterprise applications, we'll examine the basics of getting Hibernate running. We'll cover the mapping file format and syntax, including common relational mapping structures. Then, we'll examine the Hibernate API for interacting with the framework. Finally, we'll cover the common architectural decisions you'll have to make as you include this (or any other) O/RM framework.

### **JRuby by Justin Ghtland**

JRuby is not one, but two great technologies: the Ruby language, and the Java Virtual Machine and libraries. In this talk you will learn the basics of programming JRuby, and how to integrate JRuby code into existing Java projects.

### **Prototype: Ajax and JavaScript++ by Justin Ghtland**

Learn to simplify Ajax development with Prototype through a series of real-world examples. Along the way, learn to code in Prototype's modern JavaScript style, taking advantage of Prototype's extensions to JavaScript's object model

### **Spring Dependency Injection by Justin Ghtland**

Dependency Injection (DI) is the cornerstone of Spring. The core concept is quite simple, but (surprise!) actual practice can become complex. To take full advantage of Spring DI, you need to understand not only the basics on configuration, but also the container lifecycle model and the various hooks provided by the framework.

### **Spring Security with ACEGI by Justin Ghtland**

Spring offers developers a simpler, more robust method for configuring applications. These benefits extend to security through the ACEGI framework. ACEGI makes the otherwise daunting task of securing your application logical and straightforward. More importantly, through its support for single sign-on provision through Yale's CAS system and its ability to provide instance-level authorization, Spring extends the common security model of most J2EE apps beyond what they are traditionally capable of.

### **Migrating from Struts Action 1 to Struts Action 2 by Matt Raible**

Struts has outgrown its reputation as a simple web framework and has become more of a brand. Because of this, two next generation frameworks are being developed within the project: Shale and Action 2.0. Action 2.0 is based on WebWork, and though its backing beans are similar to JSF, its architecture is much simpler, and easier to use.

### **Seven Simple Reasons to Use AppFuse by Matt Raible**

AppFuse is an open source project/application that uses best-of-breed Java open source tools to help you develop web applications quickly and efficiently. Not only does it provide documentation on how to develop light-weight POJO-based applications, it includes features that many applications need out-of-the-box: authentication and authorization, remember me, password hint, skinnability, file upload, Ajax libraries, signup and SSL switching. This is one of the main features in AppFuse that separates it from the other "CRUD Generation" frameworks like Ruby on Rails, Trails and Grails. AppFuse is already an application when you start using it, which means code examples are already in your project. Furthermore, because features already exist, the amount of boiler-plate code that most projects need will be eliminated.

### **Advanced Enterprise Debugging Techniques by Neal Ford**

This session discusses techniques and tools for debugging enterprise applications (without using `System.out.println()`!)

### **Power Regular Expressions in Java by Neal Ford**

Regular expressions should be an integral part of every developer's toolbox, but most don't realize how important it is. Regular expressions have existed for decades, but many developers don't understand how to take full advantage of this powerful mechanism, either through command line tools and editors or in their development.

### **Ruby for Java Developers by Neal Ford**

This session introduces Ruby, aimed specifically at Java developers.

### **Testing with Selenium by Neal Ford**

This session describes the use and workings of Selenium, the open source web user interface testing tool.

### **The Productive Programmer by Neal Ford**

This session shows you how to become a more productive programmer every day by using tools that you didn't know you already had.

### **Web Application Security Vulnerabilities by Neal Ford**

This session highlights common mistakes made by web programmers, stating the problems and avoidance techniques.

### **Agile Estimating, Planning and Tracking by Pete Behrens**

Business leaders and stakeholders require accountability and accuracy in our software release projections and yet, as an industry, we have failed. However, many of these same leaders are not convinced that agile is any more than an excuse to avoid projections at all. While it is true that agility provides the framework to support change, it doesn't mean you can't provide accurate projections. In fact, a well-executed agile process actually provides more accurate results with less time investment than traditional methods. This session will demonstrate these agile project management techniques to manage 6-12 month projects.

### **Agile Organizational Patterns by Pete Behrens**

What makes agile processes work? Why do they sometimes fail? This session will go behind the agile processes to see how your organizational structures (roles, responsibilities and collaborations) enable or inhibit your process. Just as design patterns help to create well-formed code and applications, agile organizational patterns to enable well-formed and successful agile teams - a key success factor to implementing an agile process.

### **AJAX Performance and Scalability by Ron Bodkin**

You know AJAX can improve Web application usability, but only if designed properly. How do you deploy and manage responsive rich client networked applications that scale? If you are using services from multiple sources, how can you identify bottlenecks?

### **Effective AOP by Ron Bodkin**

Aspect-Oriented Programming (AOP) allows better application architectures by centralizing scattered and tangled code for problems like security, error handling, testing, policy enforcement and feature variations across a product line. The recent releases of Spring 2.0 and AspectJ 5 illustrate the maturing options for Java projects.

### **Easing into Agile by Scott Davis**

How do you get started with an Agile development methodology? Everyone has been talking about eXtreme Programming for years, but how do you get it introduced to your team? Many times, you're not simply transitioning from one methodology to another -- you're introducing a methodology for the first time. Adding structure to a previously unstructured endeavor. Adding a touch of discipline where programmers once roamed free.

### **Groovy: Greasing the Wheels of Java by Scott Davis**

This is the year of the dynamic scripting language. Ruby (and Rails) has won the hearts and minds of many independent software developers. JavaScript is experiencing a renaissance thanks to the wild success of AJAX and websites like Google Maps. And Groovy (JSR-241) brings the same level of excitement and "scripting goodness" to the Java platform.

### **Keynote: Why Mapping? Why Not? by Scott Davis**

The release of Google Maps was a "Wizard of Oz / Technicolor" moment for web developers everywhere. It didn't just change the way we look at mapping sites; it forever changed the way we look at all web sites. It put AJAX on the map, both figuratively and literally.

### **Rolling Your Own Google Maps, part I by Scott Davis**

The release of Google Maps was a "Wizard of Oz / Technicolor" moment for web developers everywhere. It didn't just change the way we look at mapping sites; it forever changed the way we look at all web sites. It put AJAX on the map, both figuratively and literally.

### **Java5: The Language, The Libraries, The VM by Ted Neward**

Java5 introduced a whole slew of new features, including annotations (JSR 175), new language features (the enhanced for loop, generics, static imports, and more), new library support (java.lang.instrument, among others), and some interesting enhancements to the virtual machine itself.

### **Java6: Exploring Mustang by Ted Neward**

Mustang, the forthcoming Java6 release, is just around the corner, and even if you're not looking to adopt the new platform right away, it's important to know what's there so you can start to plan for it. In this presentation, we'll go over the major new features of the Java6 platform, including the new integrated XML services capabilities (JAX-WS and JAXB), dynamic/scripting language support (javax.script), new JVM "attach" capabilities, new annotations supported by the javac compiler, and more.

### **Pragmatic XML Services by Ted Neward**

There's a lot of talk about web services, and most of it falls into one of two categories: lots of low-level talk about vendor-specific tools and extensions, or lots of high-level talk that never shows you a line of code. XML services aren't that hard, and in this talk, we'll see how, why and when to do one.

### **The Busy Java Developer's Guide to ClassLoaders by Ted Neward**

If you've ever gotten a ClassCastException and just knew the runtime was wrong about it, or found yourself copying .jar files all over your production server just to get your code to run, then you probably find the Java ClassLoader mechanism to be deep, dark, mysterious, and incomprehensible. Take a deep breath, and relax--ClassLoaders aren't as bad as they seem at first, once you understand a few basic rules regarding their operation, and have a bit more tools in your belt to diagnose ClassLoader problems. And once you've got that, and hear about ClassLoaders' ability to run multiple versions of the same code at the same time, and to provide isolation barriers inside your application, or even compile code on the fly from source form, you might just find that you like ClassLoaders after all... maybe.

### **Java/EE Web Services and SOA @ Work: Architecture & Development by Tom Marrs**

Have you tried to deploy J2EE Web Services and thrown up your hands in frustration at the lack of tool support? Do you want to know how to develop and deploy Java EE-compliant Web Services so that they work every time? Would you like to see how to develop/deploy Web Services in Spring with XFire? Are you wondering if SOA is just hype and fluff? Do you think SOA is just marketing's re-packaging of Web Services? Would you like to know how Web Services and SOA (Service-Oriented Architecture) fit together? If so, then this talk is for you.

### **Java/J2EE Architecture @ Work: EJB 3 vs Spring and Hibernate by Tom Marrs**

You've used EJB in the past and been disappointed - it was too heavy and difficult to use. Like Bruce Tate, maybe you've gone from "Bitter" to "Better, Faster, Lighter". With EJB 3 shipping in early 2006, maybe it's time to take another look. We'll compare EJB 3 with alternative frameworks - Spring and Hibernate - to see if EJB 3 has closed the gap.

### **Get Groovier with Grails by Venkat Subramaniam**

Inspired by the Ruby on Rails project, Grails brings the ease of web development and "convention over configuration" to the Java platform. We will learn how to create web applications using Grails, how to integrate it with Hibernate, and how to Ajax it, all using the built in features of Grails. This section assumes that you are familiar with Groovy or you have attended the #Groovy for Java Programmers# session. The session will be example driven with live coding where we will build a web application from scratch.

### **Practices of an Agile Developer by Venkat Subramaniam**

You have worked on software projects with varying degree of success. What were the reasons for the success of your last project? What were the reasons for those that failed? A number of issues contribute to project success - some non-technical in nature. In this presentation the speaker will share with you practices in a number of areas including coding, developer attitude, debugging, and feedback. The discussions are based on the book with the same title as the talk.

### **Spring into Unit Testing by Venkat Subramaniam**

Test Driving Development is a valuable technique that has several benefits. However, unit testing is hard when your code has dependencies-this often becomes a quick excuse to give up on TDD. This example driven Zero Powerpoint (ZePo) presentation will help you spring into unit test your Spring applications. We will look at techniques to realize good unit testing, and focus on effective use of mock objects and frameworks to help you toss mock objects for your spring application.