

# Rocky Mountain Software Symposium 2005

Renaissance Suites Flatiron - Broomfield, CO

November 11 - 13, 2005

(session listing as of 11/9/2005)

The No Fluff Just Stuff Software Symposium 2005 tour is pleased to announce the Rocky Mountain Software Symposium coming to Broomfield on November 11 - 13, 2005. 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 fifty(50) 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 5 concurrent sessions over three days with over 55 sessions to choose from. The following topics will be featured:

- 1) .Net
- 2) ServerSide Java
- 3) XML / Web Services
- 4) Client Side Java
- 5) Architecture
- 6) Core Java

## Registration Fees

Attendees	Before 10/24/2005	After 10/24/2005
1-4	\$675	\$775
5-9	\$600	\$675
10-14	\$575	\$650
15-24	\$550	\$625
25+	\$525	\$600

## 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) Custom NFJS Laptop/Backpack
- 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 2005

Renaissance Suites Flatiron - Broomfield, CO

November 11 - 13, 2005

(session listing as of 11/9/2005)

## **Ajaxian JavaScript Frameworks by Ben Galbraith**

In the "Introduction to Ajax" session, we discuss what Ajax is, how it works, and how others are using it. This session goes deeper into Ajax by reviewing the existing JavaScript frameworks that aim to make it easier.

## **Advanced Swing: Architecture and Frameworks by Ben Galbraith**

Are you spending more time plumbing your Swing applications than solving business problems? Has your Swing application grown out of control? This session is for you.

## **Creating Killer Graphics and Professional PDFs with XML by Ben Galbraith**

You can do some pretty cool things with XML these days (despite what some curmudgeons in the technology world may claim). In the past few years, XML has solidified its place as the lingua franca of data sharing and data manipulation. But XML as a data transfer language is only marginally interesting. Things get really exciting when XML is dynamically transformed into other formats. In this session, I focus on two XML formats which can be readily transformed into high-quality presentation-centric output formats. XSL-FO is a typesetting format for XML that can be readily converted into PDF (or Postscript and some other formats). SVG is a vector graphics language in XML -- a sort of open-source version of the popular Macromedia Flash format. SVG files can be converted into beautiful, completely scalable -- and interactive -- images.

## **Creating Polished Swing Applications by Ben Galbraith**

Too often, Swing applications are slow, ugly, and hard-to-maintain. It turns out that it doesn't have to be this way. Swing can be used to create highly-responsive, beautiful applications that are very maintainable. If this isn't consistent with your own experience, don't feel bad; it's not very obvious how to make Swing sing.

## **Making the Most of XML by Ben Galbraith**

For many of us, XML has become a ubiquitous presence in application development, whether parsing, validating, or manipulating it. For many of us, all that XML is coupled with pain, in the form of tedious APIs (like, say, the W3C DOM API) and confusing technologies (oh, I don't know, W3C XML Schema?).

## **Being Productive with Java in the Enterprise by Ben Galbraith**

It sounded like such a good idea back in the mid-nineties: based the Java platform on a standards-based, open community, and let anyone participate. There is no question that Sun's strategy for Java's stewardship via the JCP and sponsored open-source has yielded some enormous benefits. However, these have not been enjoyed without tremendous cost.

## **Introduction to Ajax by Ben Galbraith**

Ajax -- called DHTML just a few months ago -- has revolutionized (or "radically iterated", if you like) web application development in the short few months since the term was coined. What is it all about? Why are we excited about a set of capabilities that have been sitting in our browser for years? What can you do with it? And, how can you do it?

## **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? It's probably worse than you think -- modern shared-memory processors can do some pretty weird things when left to their own devices.

## **Garbage Collection in the HotSpot JVM by Brian Goetz**

Pop quiz: which is faster, Java or C++? If you are talking about allocation performance, the answer is Java, hands-down.

## **Introduction to concurrency utilities 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.

## **Squashing bugs with FindBugs 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 Spring by Bruce Tate**

This session, for the Spring beginner, helps you: # Understand dependency injection and inversion of control # Know the meaning of lightweight containers and Spring # Understand the basic pieces of Spring # See core Spring modules in action, including Persistence, AOP, transactions. Attendees need not know anything about Spring. This session does talk about integration with core J2EE frameworks like JDBC and transactions.

### **Beyond Java by Bruce Tate**

All programming languages have a limited life span, and Java is no different. This is a philosophical session rather than a programming session. Sooner or later, Java will lose its leadership position. This session will explore Java's strengths and weaknesses. We'll try to understand whether conditions are ripe for alternatives to emerge, and what those alternatives may be.

### **Where Agile meets Argyle: New processes in established companies by Bruce Tate**

Agile programming is a collection of core principles and techniques that allow software developers to create lighter, more responsive applications, and to have fun doing it. Many established organizations are either openly or sub-consciously hostile to many of the principles of Agile development.

### **Stretching Java by Bruce Tate**

In Stretching Java, we'll look at some of Java's limitations, see how other programming languages solve those problems, and look at how Java developers can implement those ideas in Java using open source frameworks, design strategies, and tools.

### **Introduction to Hibernate by Bruce Tate**

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.

### **Dealing with Enterprise Database Challenges In an Object Oriented Application by Clinton Begin**

This presentation will focus on database challenges that object oriented developers often face in an enterprise environment. Object Relational Mapping (ORM) tools are not ideal for all databases, and therefore alternatives must be sought.

### **Ruby for Java Programmers by Dave Thomas**

Ruby recently enjoyed its tenth birthday. Instead of cake and candles, the community celebrated by releasing a wave of new libraries and frameworks that make Ruby programming even easier. This talk features some of the best of these, as we explore Ruby.

### **Herding Racehorses and Racing Sheep by Dave Thomas**

Are you frustrated by experts who can't tell you what to do, or by junior team members who refuse to see the big picture? How can you best develop careers: both yours and those of your teammates and managers? How can we learn to apply experience more effectively, and why do the many approaches designed to tame complexity actually end up increasing it?

### **OpenSource Ecosystems by Dave Thomas**

Open Source communities produce high quality software with little management and (typically) no pay. Most people looking at open source focus on using this software in their projects.

### **Ruby on Rails by Dave Thomas**

The Ruby on Rails framework has exploded onto the scene over the last few months. Propelled by some genuine benefits, and fueled by a whole lot of controversy, Rails seems here to stay. So, is it a Java killer? (No.) Is it a great way to develop certain classes of web application? (Yes.) Does it really deliver the 10-fold increase in developer productivity that some have claimed? (It depends...)

### **Shale: The Next Struts? by David Geary**

Struts is the most popular Java-based Web application framework today, but that's rapidly changing. There's a newcomer on the block, a leaner, meaner, better-designed framework loosely based on Struts that's poised to dethrone Struts as the reigning king of Java-based web application frameworks. That framework, of

course, is JavaServer Faces. Craig McClanahan, the father of Struts and the co-spec lead for JSF 1.0, has proposed reinventing Struts for Struts 2.0 as a set of services for JSF applications. That new framework, which has no direct ties to Struts as we know it, is called Shale.

### **Felix: A bag of Tricks for Java Server Faces by David Geary**

Okay, so you know a little about JSF. You understand managed beans, action outcomes and how to attach standard JSF validators to components in a JSP page. But there is a great deal of functionality that the average web application supports that JSF doesn't provide out of the box. For example, wouldn't you like to have JSF automatically place asterisks in front of labels for required fields? You are going to implement client-side validation, which JSF does not support out of the box, aren't you? Of course, you're going to test your application, right? And don't forget to trap unauthorized use of the back button.

### **An Introduction to JavaServer Faces by David Geary**

There are a lot of Java-based web application frameworks, but how many of them are: 1. Based on the most popular open-source framework (Struts), and 2. The standard that must be supported by every J2EE 1.5 container? JavaServer Faces (JSF) debuted in the Spring of 2004. Throughout the rest of 2004, JSF gained momentum with a handful of books and a growing user community that includes the popular MyFaces open-source JSF implementation which has recently moved from SourceForge to Apache. Perhaps the most telling sign of the times is Craig McClanahan's proposal for Struts 2.0, code named Shale, which reinvents Struts as a set of services for JSF applications.

### **Killer Web UIs by David Geary**

User interfaces are usually the most turbulent aspect of an application during development. Constant tinkering with the UI means constant changes to your code, so as a UI developer, you want to minimize the scope and effects of those code changes. Open-source Java provides two powerful software packages that help you manage UI complexity: Tiles and Sitemesh. Tiles composes webpages from discrete regions of your user interface known as tiles. A tile contains a JSP page for layout and one or more JSP pages for content. Sitemesh decorates webpages with decorators that can be associated with URL patterns. Once you set up your decorators, you can decorate pages that match a decorator's URL pattern.

### **New Features in Java 5 by Jason Hunter**

The new Java 5 release introduces a number of significant Java language enhancements: generics, typesafe enums, autoboxing, an enhanced "for" loop, a static import facility, and a general-purpose metadata facility. This talk gives an overview of the changes and helps you understand what all the funny new syntax means.

### **Extreme Web Caching by Jason Hunter**

Web Caching is very important for high traffic, high performance web site but few people know all the professional-level strategies. In this talk I'll share some of the tricks of the trade, including advanced tips from Yahoo's Mike Radwin. We'll start with the basics: using client-side caches, conditional get, and proxies. Then we'll talk about more advanced features: how best to handle personalized content, setting up an image caching server, using a cookie-free domain for static content, and using randomization in URLs for accurate hit metering or sensitive content.

### **Java Metadata by Jason Hunter**

Java's new Metadata facility introduced in J2SE 5.0 defines a way to attach decorations to classes, fields, methods, and even packages that can be extracted by the compiler or runtime tools to provide advanced functionality. Think of metadata as an extended @deprecated flag, or think of XDoclet++. In this tutorial session you'll learn how Metadata fits in the Java platform (and how it compares to the C# platform). We'll cover how to use the metadata attributes provided in the core J2SE libraries and how to write your own. We'll also show a bit of what's coming in JSR-181, tasked to define standard metadata attributes for web services.

### **Forgotten Algorithms by Jason Hunter**

There are many interesting and useful algorithms that people just don't remember or never learned. The Boyer-Moore string search algorithm is one prime example. The randomized skip list is another. Both solve common problems with wonderful flair and finesse -- and performance-wise they blow the pants off brute force solutions. This session covers these two algorithms plus several others. It's like your college algorithms course but with a practical bent and absolutely zero proofs. Extra bonus: The Google PageRank algorithm.

### **An Introduction to XQuery by Jason Hunter**

XQuery is a new language from the W3C that lets you query and manipulate XML -- or anything that can be represented as XML, such as relational databases. As a Java developer -- especially a server-side Java developer -- XQuery is key to searching and manipulating large XML repositories or performing any

XML-centric task. This talk introduces XQuery. I'll explain the XQuery language; I'll show how to call XQuery from Java; and as the creator of JDOM, I'll also explain when to use XQuery instead of JDOM, and when to use both.

### **Web Services Applied by Jeff Barr**

This talk centers around Amazon Web Services (AWS), including these aspects: the AWS timeline, components, and data flow; REST vs. SOAP; E-Commerce Service (ECS) data types, functions and demos; Alexa Web Information Service (AWIS) 1.0; using AWS with Java(tm) technology; AWS business opportunities; and more.

### **Test-Driven Development with Spring and Hibernate by Matt Raible**

One of the hardest parts about J2EE development is getting started. There is an immense amount of open source tools for web app development. Making a decision on which technologies to use can be tough--actually beginning to use them can be even more difficult. Once you've decided to use Struts and Hibernate, how do you go about implementing them? If you look on the Hibernate site or the Struts site, you'll probably have a hard time finding any information on integrating the two. What if you want to throw Spring into the mix? For developers, one of the best ways to learn is by viewing sample apps and tutorials that explain how to extend those applications. In order to learn (and remember) how to integrate open source technologies such as Hibernate, Spring, Struts, and Ant/XDoclet, Raible created AppFuse.

### **Spring's AOP and Transaction Frameworks by Matt Raible**

Aspect-Oriented Programming is a great way to implement cross-cutting concerns (i.e. caching, logging, security, transactions) into your applications. Transactions are an essential piece in enterprise applications for ensuring data integrity. This session covers two topics: Spring's AOP and Transactions Framework.

### **Developing Web Applications with Spring and Ajax by Matt Raible**

In this session, you will learn what Ajax is and how it is revolutionizing the way that web applications are developed. In a similar way, the Spring Frameworks is greatly simplifying the code that we write Java web applications with.

### **SOA and ESB: Next Wave of Enterprise Development or Return of the Son of CORBA? by Neal Ford**

Are Service Oriented Architecture and Enterprise Service Buses the next wave of distributed computing or just the same old crap in a shiny new package? This session provides an overview of what most people agree is the definition of SOA and some of the characteristics of ESBs. I talk about EAI, your MOM, SOA, ESB, and all the other acronyms I can come up with.

### **Language Oriented Programming Part 1: Theory by Neal Ford**

This session shows how to use Java as the building block for domain-specific languages. It discusses the next revolution in programming: language-oriented programming and the nascent tools that support it.

### **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.

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

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

### **Enhance Design Patterns with AOP by Nick Lesiecki**

Design patterns have long been part of the experienced developer's tool chest. However, design patterns can affect multiple classes and this makes them invasive and hard to (re)use. This presentation will discuss how AOP solves this problem by fundamentally transforming pattern implementation. The class will examine examples of various traditional design patterns (including some of the famous GoF patterns) and discuss the practical and design benefits of implementing them with aspect-oriented techniques. This session will be of interest to anyone who has struggled with design patterns. It is also the perfect session for a programmer interested non-trivial applications of AOP, or who wishes to see aspect-oriented design in action.

### **AOP Applied, lessons from a J2EE project by Nick Lesiecki**

Aspect Oriented Programming offers enhanced modularity and cleaner separation of crosscutting concerns. That's all fine and well for architecture geeks. But can it help your project today? Has anyone applied it in the

real world? The answer is "yes," and in this session, an AOP expert and early adopter will demonstrate how his team used aspect oriented programming to implement non-trivial business concerns. Along the way attendees will learn about advantages of AOP and understand some of the problems encountered adopting it.

### **Groovy = Java + Ruby + Python for the JVM by Rod Cope**

Groovy is a new dynamic, object-oriented scripting language for the Java Virtual Machine. It has the expressive power of Ruby, the simplicity of Python, and can use all existing Java code. See how to build a Swing GUI, execute Ant scripts in-line, access a database, read and write XML, and more in a few lines of code.

### **Real World Web Mapping by Scott Davis**

In this presentation, we'll explore the top four mapping sites and show you how to take advantage of their free services. MapQuest, Yahoo Maps, Google Maps, and MSN Virtual Earth all bring slightly different capabilities to the table. These sites allow you to create your own interactive maps with minimum effort and no previous mapping experience. They take care of hosting the mapping data and making it easy to manipulate -- all you have to do is bring a little bit of know-how to the party.

### **Guerrilla Web Techniques by Scott Davis**

Frameworks? We don't need no stinkin' web frameworks. OK, so maybe that's overstating the case. Web frameworks do plenty of good things, but sometimes they can also be golden handcuffs. Too many web developers fall into the trap of thinking, "If it can't be done by my web framework, then it simply can't be done."

### **Advanced Hibernate by Stuart Halloway**

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.

### **Java Platform Security and JAAS by Stuart Halloway**

The Java platform is built from the ground up with security in mind. This talk will introduce the security features of the J2SE, building quickly from the basic classes to realistic examples.

### **Programming Java Concurrency by Stuart Halloway**

Java has always provided a model for concurrency and threads. With Java 1.5, this model received a major facelift. Learn how to use the new concurrency utilities to build responsive, scalable, and correct concurrent applications.

### **Cryptography for Programmers by Stuart Halloway**

For centuries people have used crypto to build (and break) secure systems. Computers have only raised the pitch of conflict, providing enormous cryptographic power at commodity prices. Most programmers do not write their own crypto libraries, instead relying on the services of an operating system or virtual machine. But even with all this support, building secure systems is a daunting task.

### **Unit Testing Java with Jython and JRuby by Stuart Halloway**

JUnit is great. Jython and JRuby are even better. Unit testing libraries look the same everywhere, so why not use the one that lets you get your job done faster?

### **Introduction to Java Reflection by Stuart Halloway**

Reflection is writing code that manipulates itself. Well-written reflective code automates a broad class of repetitive, error-prone programming tasks. Poorly-written reflective code obfuscates programs and destroys the benefits of the type system. We'll focus on the former.

### **Introduction to Web services, 2005 edition by Ted Neward**

WSDL, and Schema and SOAP, oh my! It's 2005, and the Web services landscape looks even more confusing than it did two years ago, despite all sorts of promises to the contrary. What's it all mean, and how the heck did we get here when the original goal was to try and keep it all simple?

### **The Fallacies of Enterprise Systems by Ted Neward**

There's a set of fallacies that every enterprise developer has fallen for at some point in their enterprise

development lives, and unless they've come to realize it early enough, all cause big trouble and painful learning experiences in the long run.

**Passing Messages: A Flexible, Powerful and Extensible Communication Model by Ted Neward**

Over the last decade, focus in inter-process communication has centered on Remote Procedure Calls (RPC) and its object-oriented equivalents.

**Effective Enterprise Architecture by Ted Neward**

Bring all of your enterprise Java questions to this open forum discussion hosted by the author of #Effective Enterprise Java#, Ted Neward.

**J2EE Web Services @ Work by Tom Marrs**

Do you want to know how to develop and deploy J2EE-compliant Web Services? Have you been confused by the tangled web of new deployment descriptors? Have you tried to deploy J2EE Web Services and thrown up your hands in frustration at the lack of tool support? If so, then this talk is for you. The new J2EE Web Services standard was designed for interoperability, but the tools (such as XDoclet) are broken. We'll show how to work around the problems and deploy an EJB as a Web Service.