

Rocky Mountain Software Symposium

Renaissance Suites Flatiron Hotel

May 02 - 04, 2008

<http://www.nofluffjuststuff.com/conference/denver/2008/05/index.html>
(event schedule as of May 3, 2008)

Fri, May. 02, 2008					
	Flatiron Ballroom	Red Rocks	Chautauqua	Flagstaff	Eldorado
12:00 - 1:00 PM	REGISTRATION				
1:00 - 1:15 PM	WELCOME				
1:15 - 2:45 PM	Structuring concurrent applications in JDK 5.0 Brian Goetz	A Thorough Introduction To Groovy Jeff Brown	Evolutionary SOA Neal Ford	Spring+JPA+Hibernate: Standards Meeting Productivity for Java Persistence Ken Sipe	Credit Card Software Development: Recognizing and Repaying Technical Debt Jared Richardson
2:45 - 3:15 PM	BREAK				
3:15 - 4:45 PM	Effective Concurrent Java Brian Goetz	Agile Test Driven Development With Groovy Jeff Brown	Regular Expressions in Java Neal Ford	Spring 2.5 - Spring without XML Ken Sipe	Techniques 2008 Jared Richardson
4:45 - 5:00 PM	BREAK				
5:00 - 6:30 PM	The Java Memory Model Brian Goetz	DSL in Groovy Venkat Subramaniam	Spring 2.5 MVC Portal Development with Annotations Ken Sipe	10 Tips for Getting Your Project Back on Track Jared Richardson	Dependency management Brian Pontarelli
6:30 - 7:15 PM	DINNER				
7:15 - 8:00 PM	Keynote: by Neal Ford				

Sat, May. 03, 2008					
	Flatiron Ballroom	Red Rocks	Chautauqua	Flagstaff	Eldorado
8:00 - 9:00 AM	BREAKFAST				
9:00 - 10:30 AM	Grails - Agile Web 2.0 The Easy Way Jeff Brown	JavaServer Faces: A Whirlwind Tour David Geary	Restoring Agility: Getting Your Team Back on Track Jared Richardson	Java Performance Myths Brian Goetz	Acceptance Testing Application Behavior Venkat Subramaniam
10:30 - 11:00 AM	BREAK				
11:00 - 12:30 PM	Advanced Web Development With Grails Jeff Brown	Facelets David Geary	Caring about your Code Quality Venkat Subramaniam	Architecture and Scaling Ken Sipe	Test Driven Design Neal Ford
12:30 - 1:30 PM	LUNCH				
1:30 - 3:00 PM	Filthy Rich Clients with the Google Web Toolkit, Part I David Geary	Introduction to JRuby Neal Ford	Know your Java? Venkat Subramaniam	The Busy Java Developer's Guide to Debugging Ted Neward	Shippers Unite! Jared Richardson
3:00 - 3:15 PM	BREAK				
3:15 - 4:45 PM	Filthy Rich Clients with the Google Web Toolkit, Part II David Geary	"Design Patterns" in Dynamic Languages Neal Ford	7 Habits of Highly Effective Developers Ken Sipe	The Busy Java Developer's Guide to Monitoring Ted Neward	Guice Dependency Injection Brian Pontarelli
4:45 - 5:30 PM	BIRDS OF A FEATHER SESSION				

Sun, May. 04, 2008					
	Flatiron Ballroom	Red Rocks	Chautauqua	Flagstaff	Eldorado
8:00 - 9:00 AM	BREAKFAST				
9:00 - 10:30 AM	Groovy, the Blue Pill: Writing Next Generation Java Code in Groovy Scott Davis	Viva La Javolution! Brian Sletten	The Busy Java Developer's Guide to ClassLoaders Ted Neward	SOAs Challenges Ken Sipe	Code Metrics & Analysis for Agile Projects Neal Ford
10:30 - 11:00 AM	MORNING BREAK				
11:00 - 12:30 PM	Groovy, The Red Pill: Metaprogramming, the Groovy Way to Blow a Buttoned-Down Java Developer's Mind Scott Davis	What's Going On? : Complex Event Processing w/ Esper Brian Sletten	The Busy Java Developer's Guide to Annotations Ted Neward	Hacking - The Dark Arts Ken Sipe	The 90-Minute Startup Michael Nygard
12:30 - 1:15 PM	LUNCH				
1:15 - 2:15 PM	EXPERT PANEL DISCUSSION				
2:15 - 3:45 PM	Grails for (Recovering) Struts Developers: A Groovy Alternative Scott Davis	Give it a REST Brian Sletten	The Busy Developer's Guide to Scala Ted Neward	Productive Programmer: Acceleration & Automation Neal Ford	Failures Come In Flavors (part 1) Michael Nygard
3:45 - 4:00 PM	BREAK				
4:00 - 5:30 PM	Groovy, Grails and Google Maps: Mashups 101 Scott Davis	REST - Live! Brian Sletten	The Busy Java Developer's Guide to Hacking (on) the JDK Ted Neward	Productive Programmer: Canonicity & Focus Neal Ford	Failures Come In Flavors (part 2) Michael Nygard

Rocky Mountain Software Symposium

Renaissance Suites Flatiron Hotel

May 02 - 04, 2008

<http://www.nofluffjuststuff.com/conference/denver/2008/05/index.html>
(event schedule as of May 3, 2008)

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.

Effective Concurrent Java by Brian Goetz

The Java programming language has turned a generation of applications programmers into concurrent programmers through its direct support of multithreading. However, the Java concurrency primitives are just that: primitive. From them you can build many concurrency utilities, but doing so takes great care as concurrent programming poses many traps for the unwary.

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.

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.

Dependency management by Brian Pontarelli

This talk covers the difficult subject of dependency management and uses the Savant open source framework to illustrate how to tackle some of the more difficult problems of dependency management. During this talk we'll cover the basics of dependency management, software versioning, compatibility, upgrading, and much more.

Guice Dependency Injection by Brian Pontarelli

This presentation covers the latest dependency injection framework named Guice. Guice was written by the developers at Google and makes dependency injection lighter, faster and easier to write. Attendees will learn how to dependency inject their classes using Guice annotations and modules.

Viva La Javolution! by Brian Sletten

You're a good Java programmer. You understand the JDK libraries and how to use them. The problem is that many fundamental APIs don't take the bigger performance picture in mind. Garbage collection can end up killing your app if you aren't careful. Concurrency problems and contention can keep your well-intentioned software from leveraging modern hardware architecture that support multi-core and multi-cpu systems. Who knew that simply using the standard library code the way it was designed was opening you up for performance problems in your apps? Don't worry, Javolution has your back.

What's Going On? : Complex Event Processing w/ Esper by Brian Sletten

We write very complicated software, don't we? In our systems, we detect when simple things happen. Customers log in, people buy things, a stock is sold at a particular price, inventory shifts locations... all of these events mean little things, but what about the larger picture? Complex events are particular patterns of simpler events that suggest something deeper is happening. Do you know how you'd discover these bigger picture occurrences? Come hear how the Esper open source software represents a new class of complex event processing (CEP) frameworks that can be added to even high volume, high transaction systems.

Give it a REST by Brian Sletten

As developers, we sometimes get to make choices about the technologies we use, sometimes not. We base these decisions on personal experiences, recommendations from others and a general sense of where the industry is going. Web Services have been all the rage for several years now. We have been told time and

again that we should be building systems around them; as an industry, we've never been more confused. Perhaps it is time to Give it a REST.

REST - Live! by Brian Sletten

You've read the articles, the books, the Ph.D. thesis and all of the meta-commentary about building RESTful APIs, but you're still not sure where to begin. This is an interactive session and has almost no slides. You should come prepared to discuss ideas and maybe pair program with me and everyone else in the room. Bring your ideas for open source projects that we might want to expose through a resource-oriented model. Bring your concerns about your domains that you are convinced don't fit this model.

JavaServer Faces: A Whirlwind Tour by David Geary

In April 2005, annual growth rates for jobs in JavaServer Faces, Struts, and Ruby on Rails were all at about 0%. Today, Struts' growth rate still hovers around 0%, but JSF and Rails have taken off. At the end of 2007, both JSF and Rails were growing at a rate of between 400-500% annually (according to indeed.com). JSF has passed the adoption tipping point, and is now the Java-based framework of choice, as is evidenced by its ecosystem. From vendors such as MyEclipse and RedHat to open source projects such as Seam, Facelets, and Ajax4JSF, JSF is where the action is. Come see why JSF is so popular. In this code- and demo-intensive session, I'll show you the fundamentals of JSF. **Prerequisite:** *Some knowledge of Java-based web applications, such as Struts, is a plus, but is not required. If you have a significant experience with JSF, you probably already know most of what's covered in this session.*

Facelets by David Geary

Facelets is a combination of Tiles and Tapestry, and it's the hottest JSF-related open source project on the planet. It's popularity is well deserved, and in fact, much of what is in Facelets today will make its way into the JSF 2.0 spec due out in 2008. So not only can you come to this session and see some really cool demos that you can put to use in the real world, but you'll also be learning JSF 2.0 before it's even been defined! How's that for a ROI? **Prerequisite:** *Some knowledge of JSF is essential. If you're familiar with a templating framework, such as Velocity or Tiles, that's a plus, but not required.*

Filthy Rich Clients with the Google Web Toolkit, Part I by David Geary

The Google Web Toolkit (GWT) is truly a revolutionary framework that lets you develop Ajaxified web applications without knowing anything about Ajax or JavaScript. But the GWT goes way beyond basic Ajax by letting you implement desktop-like applications that run in the ubiquitous browser.

Filthy Rich Clients with the Google Web Toolkit, Part II by David Geary

In the second part of this talk, you will learn how to extend the GWT by implementing custom widgets, including a scrolling viewport and a drag and drop framework. After discussing custom widgets, you will see how to integrate database access into your GWT applications, and how to deploy your GWT applications to external servers.

Credit Card Software Development: Recognizing and Repaying Technical Debt by Jared Richardson

Technical debt has long been recognized in technical circles for years, but convincing your manager to budget time to repay "technical debt" has always been problematic. Let's couch the term technical debt concept in language more familiar to our managers: credit card debt.

Techniques 2008 by Jared Richardson

There are a number of great techniques you can use across technologies and projects. Come hear some of my favorites and contribute a few of your own. We'll discuss topics from DRY to creating a zone defense for your product.

10 Tips for Getting Your Project Back on Track by Jared Richardson

Software projects fail over and over for many of the same reasons. We'll look at some of the more avoidable problems and some solid ways to fix them, or avoid them in the first place.

Restoring Agility: Getting Your Team Back on Track by Jared Richardson

An agile team is first and foremost "a team". When that gets lost in the rush to get a product out the door, the people suffer as well as the products. It's bad for the company, but even worse for the team members. We'll learn how to defuse some of the more common problems you'll run into on dysfunctional teams.

Shippers Unite! by Jared Richardson

An overview of the Agile software approach from the book Ship It! A Practical Guide to Successful Software Projects.

A Thorough Introduction To Groovy by Jeff Brown

Groovy is an agile dynamic language for the Java platform. The language and its libraries bring many things to the table to ease the process of building applications for the Java platform. This session provides a detailed run through Groovy with lots of code samples to drive home the power of the language.

Agile Test Driven Development With Groovy by Jeff Brown

Dynamic languages bring a lot of interesting elements to the table for teams interested in doing Test Driven Development (TDD). Groovy lends itself very well to TDD and this session demonstrates many features of the language and its libraries that help teams build more testable systems and build better tests.

Grails - Agile Web 2.0 The Easy Way by Jeff Brown

Grails is a full stack MVC framework for building web applications for the Java platform. Grails makes web application development both fun and easy. This session covers all of the fundamentals of building web applications with Grails.

Advanced Web Development With Grails by Jeff Brown

Grails makes web application development both fun and easy. This session dives beyond the basics to cover advanced details of Grails that bring the really exciting features to your applications. **Prerequisite:** *Grails - Agile Web 2.0 The Easy Way*

Spring+JPA+Hibernate: Standards Meeting Productivity for Java Persistence by Ken Sipe

Well the standards created EntityBeans.... yea. and the community created Hibernate. Fortunately the standards body learned some lessons and created JPA. JPA requires a vendor implementation and none make a better choice than Hibernate. Combined with Spring this trio is a powerhouse when it comes to developer productivity on applications requiring persistence.

Spring 2.5 - Spring without XML by Ken Sipe

Spring 2.5 is brand spanking new, with a number of fantastic features. With growth of large and complex Spring applications which struggle with xml manageability and with the added pressure of Guice and SEAM there is a push for less XML, with solution leaning towards annotations. Spring 2.5 adds to the toolset provided in Spring 2.0 to provide a development environment where XML is greatly reduced... or eliminated if you so choose.

Spring 2.5 MVC Portal Development with Annotations by Ken Sipe

Looking to get into the portal development space. Looking to accelerate your development and leverage your existing Spring knowledge. This session will combine the 2 in one presentation showing several demos along the way. **Prerequisite:** *Spring*

Architecture and Scaling by Ken Sipe

Scale... what is scale... how do you applications which are scalable. How do you know if the application scales?

7 Habits of Highly Effective Developers by Ken Sipe

Thoughts lead to words, words lead to action, actions lead to habits. In this session we'll sharpen the development saw in the process of understanding what makes a hyper-productive programmer. The focus will consist of developer habits and development processes.

SOAs Challenges by Ken Sipe

SOA... Is it hype? What's real... and what's not? What is the right abstraction level?

Hacking - The Dark Arts by Ken Sipe

A live Hacking demonstration exposing the tools and techniques used by Hackers.

The 90-Minute Startup by Michael Nygard

What do you get when you add agile programming, automated deployment, self-describing systems, and virtualization? In this session, Michael will create and deploy a fully-functional web site. By the end of 90 minutes, you will be able to access the site live on the 'Net.

Failures Come In Flavors (part 1) by Michael Nygard

The typical JEE application does not reach the fabled "five nines" of availability. Far from it. It's more like "double eights". Come see why enterprise applications and web sites are only serving users 88% of the time instead of 99.999%. Part 1 of 2

Failures Come In Flavors (part 2) by Michael Nygard

What can we do about the dismal uptime of typical applications? We are asked to provide "five nines", but only reach 88%, on average. Come learn how to prevent the Stability Antipatterns from biting you. Apply these Stability Patterns to contain damage, recover from shocks, and survive disasters. Part 2 of 2

Evolutionary SOA by Neal Ford

This session demonstrates that "Agility" and "SOA" complement each other quite well. Just because SOA is buzz-word compliant doesn't mean that you should throw good practices out the window. This session demonstrates how you can apply the principles of agility to building highly complex distributed enterprises.

Regular Expressions in Java by Neal Ford

Regular expressions should be an integral part of every developer's toolbox, but most don't realize what an important topic 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.

Keynote: Ancient Philosophers & Blowhard Jamborees by Neal Ford

It turns out that ancient philosophers knew a lot about software -- did you know that Plato defined object-oriented programming? This keynote applies old lessons to new problems and old problems to new lessons. It describes why SOA is so hard, and why people in your company make bone-headed decisions. What other keynote includes Rube Goldberg, Aristotle, Dave Thomas, and Chindia?

Test Driven Design by Neal Ford

Most developers think that "TDD" stands for Test-driven Development. But it really should stand for "Test-driven Design". Rigorously using TDD makes your code much better in multiple ways.

Introduction to JRuby by Neal Ford

This session describes JRuby, the 100% pure-Java implementation of the Ruby programming language. It covers the basics of programming with JRuby and examples of how to integrate it into existing Java projects.

"Design Patterns" in Dynamic Languages by Neal Ford

The Gang of Four book should have been entitled "Palliatives for Statically Typed Languages", because the recipes it provides are cumbersome solutions to the problems it poses. Using powerful languages makes the solutions in the GoF book look hopelessly complicated. This session shows how to solve the same problems concisely, elegantly, and with far fewer lines of code using the facilities of dynamic languages.

Code Metrics & Analysis for Agile Projects by Neal Ford

What does code + methodology have to do with one another? Everything! Agile projects focus on delivering working code, and tools exist to allow you to verify some quality metrics for your code. This session is a survey of tools and metrics that allow you to determine the quality of your code and strategies to "wire it" into your agile project.

Productive Programmer: Acceleration & Automation by Neal Ford

Developers from the 1980s would be shocked at how inefficiently developers use their computers because of the advent of graphical operating systems. This talk describes how to reclaim productivity afforded by intelligent use of command lines and other ways of accelerating your interaction with the computer and bending computers to do your bidding. Stop working so hard for your computer!

Productive Programmer: Canonicity & Focus by Neal Ford

Getting work done in modern office environments is a daunting task. This session tackles 2 of the things that drag down developer productivity: lack of focus and creeping repetition.

Groovy, the Blue Pill: Writing Next Generation Java Code in Groovy by Scott Davis

There are wild-eyed radicals out there telling you that Java is dead, statically-typed languages are passe, and your skills are hopelessly out-of-date. Those extremists are the same ones who don't bat an eye at throwing out years of experience to learn a new language from scratch, pushing aside a familiar IDE for a new one, and deploying to a whole new set of production servers with little regard to legacy integration. While this "burn the boats" approach to software development might sound exciting to some folks, it's giving your manager the cold shakes right now. What if I told you that there was a way that you could integrate seamlessly with your legacy Java code, continue to use your trusty IDE and stable production servers, and yet take advantage of many of the exciting new dynamic language features that those fanatics keep prattling on about? You'd probably say, "Groovy!" I would, too...

Groovy, The Red Pill: Metaprogramming, the Groovy Way to Blow a Buttoned-Down Java Developer's Mind by Scott Davis

This talk focuses on the ways that Groovy can turn a traditional Java developer's world-view upside down. We'll start by talking about how you can thumb your nose at The Man by leaving out many of the main syntactic hallmarks of Java: semicolons, parentheses, return statements, type declarations (aka Duck-typing), and the ever-present try/catch block. Then we'll look at features like operator overloading and method pointers that Groovy welcomes back into the language with open arms.

Grails for (Recovering) Struts Developers: A Groovy Alternative by Scott Davis

Struts enjoys an unprecedented marketshare in the Java web development space -- 60%-70% according to most surveys. As newer, modern web frameworks come to the scene, very little attention is paid to the real costs of migrating an existing Struts application. This talk shows you ways to mix Groovy into a legacy Struts application, dramatically reducing both the lines of code and the complexity. We'll also introduce you to Grails (a Groovy-based web framework) whose URL-mapping capabilities allow it to replace your Struts application without breaking legacy URLs.

Groovy, Grails and Google Maps: Mashups 101 by Scott Davis

Groovy is a new dynamic language that dramatically speeds up Java development. Grails is a complete web framework in a box, including a web server and a database. Google Maps allows you to add maps to your webpage in a few lines of code. Put all three together and you are built for speed.

The Busy Java Developer's Guide to Debugging by Ted Neward

Bugs? We all know your code has no bugs, but someday, you're going to find yourself tracking down a bug in somebody else's code, and that's when it's going to be helpful to have some basic ideas about bug-tracking in your toolbox. Learn to make use of the wealth of tools that the Java Standard Platform makes available to you--tools that your IDE may not know exist, tools that you can make use of even within a production environment.

The Busy Java Developer's Guide to Monitoring by Ted Neward

Crashes? Outages? Slow response? We all know that it's never your code that causes these things, but for some reason those pesky system administrators still insist on paging you at 4AM to come in and fix those problems, anyway. For some reason, they just keep expecting you to support this thing, even after QA said it was OK!

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.

The Busy Java Developer's Guide to Annotations by Ted Neward

Want to get the soup-to-nuts story on Java annotations? In this presentation, we'll first talk about what annotations provide to the Java language. After setting ourselves a conceptual basis to operate from, we'll look at the language definition for Java annotations, from how to use them to how to define them. Finally, we'll take a look at the other side of annotations, consuming them at source-level (using "apt", the annotation processing tool), class-level (using a bytecode toolkit such as BCEL), and at runtime (using enhancements to the Reflection API made in Java5).

The Busy Developer's Guide to Scala by Ted Neward

Scala is a new programming language incorporating the most important concepts of object-oriented and functional languages and running on top of the Java Virtual Machine as standard "dot-class" files.

The Busy Java Developer's Guide to Hacking (on) the JDK by Ted Neward

Ever since its 1.1 release, the Java Virtual Machine steadily becomes a more and more "hackable" (configurable, pluggable, customizable, choose your own adjective here) platform for Java developers, yet few, if any, Java developers take advantage of it. Time to take the kid gloves off, crack open the platform, and see what's there. Time to play.

DSL in Groovy by Venkat Subramaniam

DSL or Domain Specific Languages focus on a domain or problem at hand. They're expressive, but their restricted scope keeps them simple and small from the user point of view. However, designing them is not easy. In this presentation we will explore the features of Groovy and show how they can be used to create DSLs.

Acceptance Testing Application Behavior by Venkat Subramaniam

How do you ensure your applications meet the expectations of your key customers? In this session we will explore using the FIT tool and Behavior Driven Design tools to do exactly this.

Caring about your Code Quality by Venkat Subramaniam

We all have seen our share of bad code. We certainly have come across some good code as well. What are the characteristics of good code? How can we identify those? What practices can promote us to write and maintain more of those good quality code. This presentation will focus on this topic that has a major impact on our ability to be agile and succeed.

Know your Java? by Venkat Subramaniam

Java has been around for well over a decade now. It started out with the goal of being simple. Over the years, its picked up quite a bit of features and along comes complexity. In this presentation we will take a look at some tricky features of Java, those that can trip you over, and also look at some ways to improve your Java code.