

Research Triangle Software Symposium

Marriott RTP

July 20 - 22, 2007

<http://www.nofluffjuststuff.com/sh/2007-07-raleighdurham>

(event schedule as of July 21, 2007)

Fri, Jul. 20, 2007					
	Salon A&B	Salon C	Salon D	Raleigh	Durham
12:00 - 1:00 PM	REGISTRATION				
1:00 - 1:15 PM	WELCOME				
1:15 - 2:45 PM	Groovy: Greasing the Wheels of Java Scott Davis	The Busy Java Developer's Guide to ClassLoaders Ted Neward	JavaScript for Ajax Programmers Justin Gehtland	Agile Immersion David Hussman	Annotation Hammer Venkat Subramaniam
2:45 - 3:15 PM	BREAK				
3:15 - 4:45 PM	Groovy and Java: The Integration Story Scott Davis	The Busy Java Developer's Guide to Java Platform Security Ted Neward	Prototype: Ajax and JavaScript++ Justin Gehtland	Creating Agile Requirements David Hussman	Java 6 Features, what's in it for you? Venkat Subramaniam
4:45 - 5:00 PM	BREAK				
5:00 - 6:30 PM	JavaServer Faces: A Whirlwind Tour David Geary	The Busy Java Developer's Guide to Debugging and Monitoring Ted Neward	Real World Grails Scott Davis	Executable Documentation David Hussman	OSGi: A Well Kept Secret Venkat Subramaniam
6:30 - 7:15 PM	DINNER				
7:15 - 8:00 PM	Keynote: by Scott Davis				

Sat, Jul. 21, 2007					
	Salon A&B	Salon C	Salon D	Raleigh	Durham
8:00 - 9:00 AM	BREAKFAST				
9:00 - 10:30 AM	Enterprise Applications with Spring: Part 1 Ramnivas Laddad	10 Ways to Improve Your Code Neal Ford	RAD JSF with Seam, Facelets, and Ajax4jsf, Part One David Geary	Drooling with Groovy and Rules Venkat Subramaniam	Getting Agile Planning and Tracking Up and Running David Hussman
10:30 - 11:00 AM	BREAK				
11:00 - 12:30 PM	Enterprise Applications with Spring: Part 2 Ramnivas Laddad	Building DSLs in Static and Dynamic Languages Neal Ford	RAD JSF with Seam, Facelets, and Ajax4jsf, Part Two David Geary	Everything Old Is New Again Glenn Vanderburg	Cutting an Agile Groove: Agile Transition Tips David Hussman
12:30 - 1:30 PM	LUNCH				
1:30 - 3:00 PM	Introduction to Hibernate Scott Leberknight	Domain Driven Design: Simplifying Enterprise Architecture Ramnivas Laddad	The Google Web Toolkit, Part One David Geary	Java Performance Myths Brian Goetz	Distributed Teams: Remote Agility Jared Richardson
3:00 - 3:15 PM	BREAK				
3:15 - 4:45 PM	Advanced Hibernate Scott Leberknight	The Busy Java Developer's Guide to Reflection Ted Neward	The Google Web Toolkit, Part Two David Geary	Leveraging annotations with AOP Ramnivas Laddad	Build Teams, Not Products Jared Richardson
4:45 - 5:30 PM	BIRDS OF A FEATHER SESSIONS				

Sun, Jul. 22, 2007					
	Salon A&B	Salon C	Salon D	Raleigh	Durham
8:00 - 9:00 AM	BREAKFAST				
9:00 - 10:30 AM	Intro to Java Persistence API (JPA) Mark Richards	Implementing SOA Neal Ford	Structuring concurrent applications in JDK 5.0 Brian Goetz	JRuby Justin Gehtland	Introducing Agility to Large Organizations David Bock
10:30 - 11:00 AM	BREAK				
11:00 - 12:30 PM	Advanced Java Persistence API (JPA) Mark Richards	The Zen of REST Scott Davis	Effective Concurrent Java Brian Goetz	Rails for JRuby Neal Ford	get Fit Venkat Subramaniam
12:30 - 1:15 PM	LUNCH				
1:15 - 2:15 PM	EXPERT PANEL DISCUSSION				
2:15 - 3:45 PM	Getting Started with Grails Jason Rudolph	The Java Memory Model Brian Goetz	Internationalization and Localization in Java David Bock	EJB3 Core Specification (JSR-220) Mark Richards	Metrics-driven Agile Development Neal Ford
3:45 - 4:00 PM	BREAK				
4:00 - 5:30 PM	Advanced Domain Models in Grails: Enterprise Integration Made Easy Jason Rudolph	Making The Right Persistence Framework Choice Mark Richards	Maintaining Project Integrity with JDepend, Macker, PMD, Maven, and other open source tools David Bock	Power Regular Expressions in Java Neal Ford	Shippers Unite! Jared Richardson

Research Triangle Software Symposium

Marriott RTP

July 20 - 22, 2007

<http://www.nofluffjuststuff.com/sh/2007-07-raleighdurham>
(event schedule as of July 21, 2007)

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.

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.

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.

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.

Introducing Agility to Large Organizations by David Bock

For several years, I was a member of a team of people caught in the middle of a 200+ person software development company, with senior management wanting "buzzword compliant process improvement" such as CMMI, and engineers wanting more ?agile? solutions (and people on both sides confusing Agile with ad-hoc). We were responsible for sorting it all out. Reconciling this was a herculean effort, and can be a source of lessons learned for your own process improvement efforts. Are you trying to be more agile in your organization? Are you expecting it to be harder than it needs to be because of political and bureaucratic forces beyond your control? Do you have to "educate" your senior management to protect them from buzzwords? Come learn from my successes... and mistakes.

Internationalization and Localization in Java by David Bock

Internationalization and Localization in Java is easy, right? Everyone knows you just store your strings in some resource bundles, set the locale, wave your hands a little bit, and your application is good-to-go. Right? Maybe not... Java provides some great utilities to get started, but leaves you needing more when it comes to things like screen layout, cultural sensitivities, semantic differences in translation, use of color and iconography, and other issues.

Maintaining Project Integrity with JDepend, Macker, PMD, Maven, and other open source tools by David Bock

How many times have you started a new project only to find that several months into it, you have a big ball of code you have to plod through to try to get anything done? How many times have you been the ?new guy? on an established project where it seems like the code grew more like weeds and brambles than a well-tended garden? With a few good structural guidelines and several tools to help analyze the code, we can keep our project from turning into that big ball of mud, and we can salvage a project that is already headed down that path.

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 2006. Embraced by developers, vendors, and open-source projects, JSF has hit its stride. If you haven't come up to speed on JSF basics, this is the place to start.

RAD JSF with Seam, Facelets, and Ajax4jsf, Part One by David Geary

In this session, see how you can get Ruby On Rails-like productivity on the Java side of the house with this compelling combination of technologies.

RAD JSF with Seam, Facelets, and Ajax4jsf, Part Two by David Geary

A continuation of a 2-session presentation on Seam, Facelets, and Ajax4jsf.

The Google Web Toolkit, Part One 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.

The Google Web Toolkit, Part Two by David Geary

The second part of a 2-session presentation on the Google Web Toolkit.

Agile Immersion by David Hussman

Have you heard about SCRUM or XP but never done it? If you want to give it a try, this session will allow you to participate in planning and executing several agile iterations. A working knowledge of either XP or SCRUM will be helpful but not mandatory.

Creating Agile Requirements by David Hussman

Successful project communities balance written requirements with a healthy amount of discussion. This is at the core of requirements that could be deemed #agile#. Many agile projects choose to use user stories, but others may be using use cases or other forms of written requirements. This session is for anyone wanting to improve their requirements, including the creation of good requirement and the presentation styles that help people focus on creating great software products, and stop focusing on documents.

Executable Documentation by David Hussman

Why is so much documentation worthless? Wouldn't it be nice if your documentation actually reflected what your system does? One way to do this is through is by creating #executable documentation. If you are struggling with ambiguous requirements, lack of contact with the business, or a chasm between development and testing, this session is for you.

Getting Agile Planning and Tracking Up and Running by David Hussman

If your company is using agile or thinking about it, this session will show you how to plan and tracking an agile project. Examples projects will be discussed, including the glory and horrors. Various planning tools that help distributed teams will be presented as well as a collection of lo-fi tools which truly help find and address the issue that plagues so many projects: #when are we going to complete this project#.

Cutting an Agile Groove: Agile Transition Tips by David Hussman

If you are thinking about, just starting, or in the midst of transitioning to agile, this session is overflowing with practices, tip, techniques, and experiences. Stop talking about whether or not you are doing agile and come learn how to setup and maintain agility that extends beyond a single project.

Everything Old Is New Again by Glenn Vanderburg

The early years of computers -- the '50s and '60s -- were characterized by furious exploration of a huge variety of different ideas. Since then many of the hot topics of those days have moved to the fringe, largely ignored by the mainstream of software development. But some of them are being rediscovered, and a lot of what we think of as "new developments" are really just some old ideas returning to center stage.

Distributed Teams: Remote Agility by Jared Richardson

How do you keep a team scattered across time zones in sync?

Build Teams, Not Products by Jared Richardson

A great team builds great software, but how do you build a great team?

Shippers Unite! by Jared Richardson

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

Getting Started with Grails by Jason Rudolph

Grails is an open-source web application framework that's all about getting things done. Grails combines best-of-breed Java technologies (including Hibernate and Spring), convention over configuration, and the powerful and dynamic Groovy language. Together with these elements and Groovy's ability to seamlessly integrate with your existing Java code, Grails finally legitimizes rapid web application development for the Java platform.

Advanced Domain Models in Grails: Enterprise Integration Made Easy by Jason Rudolph

Have you seen someone develop a Rails or Grails application in a matter of minutes, only to later discover that their domain model and database schema followed conventions that are different from your existing systems? Or perhaps you're interested in using Grails, but you don't want to duplicate your existing Java domain classes in Groovy. In this session, we'll see how Grails makes it easy to hook into your pre-defined schemas or existing entity classes, while still getting all the rapid application development (RAD) goodness that Grails has to offer.

JavaScript for Ajax Programmers by Justin Ghtland

This presentation covers JavaScript from the perspective of an Ajax programmer. We assume that you may be using an Ajax toolkit, but still need to be able to read, modify, and test the JavaScript code in your application. You will learn the common idioms of JavaScript by looking at working code from the Ajax toolkits themselves.

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

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.

Intro to Java Persistence API (JPA) by Mark Richards

In addition to providing a simplified API, the new EJB3 specification (JSR-220) defines a standard ORM Java Persistence API (JPA) that is rapidly gaining in popularity. As you will see in this session, JPA bears a striking resemblance to popular ORM solutions like Hibernate and Toplink. In this session we will explore in detail the new Java Persistence API offered by JSR-220. We will start by discussing the overall design and architecture of the JPA and how the major components within JPA interact. We will then look at defining mapping objects (entities) and how to use the EntityManager to manage these entities. Through interactive coding examples we will investigate the pros and cons of detached entities and merging, how to map and use entity relationships (1-1, 1-N, N-1, and N-N), discuss Lazy Loading, and finally see how to use XML mappings rather than annotations. More advanced features of JPA will be covered in a separate session.

Advanced Java Persistence API (JPA) by Mark Richards

This session picks up where the Intro to JPA session left off and covers some of the more advanced topics in the Java Persistence API. Some of the topics covered in this session include switching persistence providers, versioning, compound keys, entity inheritance, and finally handling both simple and complex stored procedures. Some knowledge of JPA is recommended for this session as I will not be covering the

basics of JPA (that is covered in a separate Intro to JPA session). Through a combination of slides and interactive coding I will demonstrate these advanced topics using both Hibernate and Toplink JPA.

EJB3 Core Specification (JSR-220) by Mark Richards

The new EJB 3 specification (JSR-220) offers some great improvements over the prior EJB specs in terms of development simplicity and new features. In this session we will discuss why EJB is still important, and explore in detail some of the new features of the core EJB 3 specification. Included in this session will be defining and accessing session beans, JTA transaction management, declarative security, and interceptors. For those of you who still like to code in XML, I will also discuss and show how we can use XML rather than annotations within EJB3. During the session I will demonstrate the new features of EJB 3 through interactive coding examples. Note: this session does not cover the new Java Persistence API (JPA) - that topic is covered in separate JPA sessions.

Making The Right Persistence Framework Choice by Mark Richards

Java Persistence has come a long way in the past 4 years. We have many viable options available now, including JPA, Hibernate, iBATIS, Toplink, and OpenJPA. With so many options available now it is difficult to know when to use which framework. In this session we will focus on native Hibernate, JPA, and the iBATIS framework, and discuss the main strengths and weaknesses of each approach and what the decision criteria is for using each of these frameworks. Knowing that it is not a one-size-fits-all situation when it comes to Java Persistence, through interactive coding we will take a look at how to use iBATIS together with native Hibernate or JPA and when this makes sense. By attending this session you will gain the knowledge necessary to make informed decisions about which Java Persistence Framework to choose for your current or next Java-based application.

10 Ways to Improve Your Code by Neal Ford

No one writes perfect code, and every developer eventually falls into a slump where they just crank out the same code day after day. This session illustrates 10 different ways to improve your code, covering sacred cows, good citizens, smells, and more.

Building DSLs in Static and Dynamic Languages by Neal Ford

This session discusses building Domain Specific Languages and DSL-style code in Java, Groovy, and Ruby. It discusses the different types of DSLs, details on how to implement them in Java, Groovy, and Ruby, and example problem domains where DSLs make sense.

Implementing SOA by Neal Ford

This talk avoids SOA hype and gets to the meat of the matter: how do you implement a Service-Oriented Architecture, what are the technological pitfalls, how do you test it, and what traps should you avoid. No marketecture: just implementation details.

Rails for JRuby by Neal Ford

This session explains all the hype surrounding Ruby on Rails, in a context familiar to Java developers. It covers convention over configuration, ActiveRecord, controllers, views, Ajax, scaffolding, testing, and deployment...on the JVM, using JRuby.

Metrics-driven Agile Development by Neal Ford

Agile software development is a highly disciplined way to build software, and one of the side effects of this discipline is the ability to gather meaningful metrics. This session describes what makes agility perfectly suited to metrics gathering and what kinds of real-world metrics you can generate.

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.

Enterprise Applications with Spring: Part 1 by Ramnivas Laddad

This session (part 1 of the two-part session) shows the core concepts in the Spring Framework -- the most popular lightweight container that recently crossed 1 million downloads.

Enterprise Applications with Spring: Part 2 by Ramnivas Laddad

This session (the second part of the 2-part session) will cover advanced concepts in the Spring framework. While the core concepts in the first session will get you started with Spring, the advanced concepts in this session will make you effective at developing Spring-based applications.

Domain Driven Design: Simplifying Enterprise Architecture by Ramnivas Laddad

Domain-Driven Design (DDD) recommends dealing with complex software system using a domain model and preserving the model through implementation. A direct mapping between domain model and software artifacts creates simple to understand, inexpensive to implement, and easy to evolve systems. The DDD approach suggests ways to distill domain knowledge into a model and offers patterns to design and implement that model.

Leveraging annotations with AOP by Ramnivas Laddad

Specifying metadata using annotations has gained huge popularity since its introduction in Java 5. However, the story on consuming annotations isn't as clear. Reading and processing annotation is still a complex process often requiring you to understand byte-code manipulation tools and their low-level API. As a result, most developers shy away from using custom annotations, limiting their usages of annotations only those prescribed by frameworks. The result is missed opportunities for programming simplification. In this session, we explore how AOP can make it a simple task to consume annotation in a powerful manner.

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.

Groovy and Java: The Integration Story by Scott Davis

I'm attracted to Groovy because of its spirit of inclusiveness. Because it extends my platform of choice, not replaces it -- include a single JAR in your classpath and you are Groovy-enabled. Because it offers full bidirectional integration with Java. Because it offers a nearly flat learning curve for experienced Java developers. Come see how you can use Groovy to augment your existing Java codebase.

Real World Grails by Scott Davis

Scott Davis is the Editor in Chief of aboutGroovy.com. The website, in addition to being, umm, about Groovy, is implemented in Grails. This talk shows you how to get started with Grails, but also talks about the experience of using it in a live, production web site.

KEYNOTE: No, I Won't Tell You Which Web Framework to Use: or The Truth (with Jokes) by Scott Davis

"Which framework should I use?" is the question most often heard on the No Fluff, Just Stuff tour. It's well worth asking. Unfortunately, there is no simple answer. After years on the tour, most speakers have crafted a response that would make any Washington politician proud -- long on style, but essentially, "Well, it depends..."

The Zen of REST by Scott Davis

Google quietly deprecated their SOAP search API at the end of 2006. While this doesn't mean that you should abandon SOAP, it does reflect a growing trend towards simpler dialects of web services. Google joins a number of popular websites (Yahoo, Flickr, YouTube, del.icio.us) that offer all of the benefits of web services without all of the complexity of SOAP.

Introduction to Hibernate by Scott Leberknight

This session introduces the Hibernate Object/Relational Mapping (ORM) framework, showing the basics of persisting Java objects to relational databases. No prior knowledge of Hibernate or ORM is assumed.

Advanced Hibernate by Scott Leberknight

This session covers advanced Hibernate topics beyond simple object persistence including session management, object locking, detachment and versioning, lazy loading performance issues and query tuning, advanced O/R mapping support, legacy database considerations, and the Hibernate cache architecture.

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 Java Platform Security by Ted Neward

Permissions, policy, `SecurityExceptions`, oh my! The Java platform is a rich and powerful platform, complete with a rich and powerful security mechanism, but sometimes understanding it and how it works can be daunting and intimidating, and leave developers with the basic impression that it's mysterious and dark and incomprehensible. Nothing could be further from the truth, and in this presentation, we'll take a pragmatic, code-first look at the Java security platform, including `Permissions`, the `SecurityManager` and its successor, `AccessController`, the `Policy` class and policy file syntax, `JAAS`, and more.

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

If you've never used Reflection (`java.lang.reflect`), you don't know what you're missing. In this presentation, we'll take a code-first, soup-to-nuts look at the Java Reflection APIs, from how to examine the class metadata that Reflection provides, to using annotations to enhance that metadata with your own information, even through the use of Java Dynamic Proxies to create flexible object "interceptors" that can layer services in front of ordinary method calls with nothing more complicated than an interface and a factory.

Annotation Hammer by Venkat Subramaniam

Annotation is an interesting feature in Java. However, like any features, there are good uses and bad uses. When should you use Annotation? This presentation will answer that question for you.

Java 6 Features, what's in it for you? by Venkat Subramaniam

What benefit do new Java 6 features offer you. Are there issues with using these features. The objective of this presentation is not simply to introduce you to the features, but to the effective use of these as well.

OSGi: A Well Kept Secret by Venkat Subramaniam

In this presentation we will introduce OSGi and discuss how it can help modularize and version your enterprise Java applications.

Drooling with Groovy and Rules by Venkat Subramaniam

Rule based programming allows us to develop applications using declarative rules. These can simplify development in applications where such rules based knowledge is used for decision making.

get Fit by Venkat Subramaniam

Unit testing tells you, the programmer, that your code (and the change) meets your expectations. How do you know if you are meeting your customers' expectations? Agile development is all about feedback and doing what's relevant to the customers, isn't it? Framework for Integration testing or Fit helps you to automate tests for customer expectations.