

# Pacific Northwest Software Symposium 2006

Redmond Marriott Town Center - Redmond, WA

September 15 - 17, 2006

(session listing as of 9/16/2006)

The No Fluff Just Stuff Software Symposium 2006 tour is pleased to announce the Pacific Northwest Software Symposium coming to Redmond on September 15 - 17, 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 Pacific Northwest Software Symposium will offer 5 concurrent sessions over three days with over 55 sessions to choose from. The following topics will be featured:

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

## Registration Fees

Attendees	Before 8/28/2006	After 8/28/2006
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.

# Pacific Northwest Software Symposium 2006

Redmond Marriott Town Center - Redmond, WA

September 15 - 17, 2006

(session listing as of 9/16/2006)

## **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.

## **Applied AOP by Brian Sletten**

Most people new to Aspect-Oriented Programming (AOP) are fed up with separation of concerns zealots explaining how great their techniques are at dealing with... logging. Ok, you get it. Logging is a cross-cutting concern that can be appropriately modularized. What else does AOP have to offer? A lot, it turns out. This talk will give an introduction to the motivations of AOP as well as a series of concrete examples drawn from enterprise and client side Java. Come learn how AspectJ-flavored AOP can begin to benefit you immediately either in development or production environments. Learn how to enforce architectural policies, find Swing threading issues, reduce the invasiveness of the Observer design pattern or even improve the reusability of your domain models.

## **Applied REST by Brian Sletten**

REST sounds like such a simple thing. But, what is it really? How do you convince your boss to let you try it when she has been sold on the equation  $SOAP = SOA + P(\text{rofit})$ ? How do you go about building, deploying, publishing and orchestrating web services without the (Un)Holy Trinity of SOAP, WSDL and UDDI?

## **Introducing the Semantic Web by Brian Sletten**

Just as the world is feeling comfortable with the Web, Tim Berners-Lee et al inform us that what we have seen so far is just the beginning. His original plans at CERN were larger and grander. The Semantic Web is the new vision of machine-processable documents and metadata to improve search, knowledge discovery and data integration and management. While there are many naysayers chiding such grand visions, there are also pragmatic and useful technologies emerging that can be applied today.

## **NetKernel : XML Processing for the 21st Century by Brian Sletten**

A wise man once said, "XML is like lye. It is very useful, but humans shouldn't touch it." If you've had to incorporate XML into your project by hand, you have probably been burned by getting too close. NetKernel turns this wisdom on its head and encourages you to use XML like the liquid data stream you want it to be. Imagine the simplicity of REST married to the power of Unix pipes. Come see how this open source / commercial product built on a compelling modern architecture can be used to create, manipulate and transform XML.

## **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.

### **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**

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.

### **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.

### **#Show Me the Numbers# - Agile Planning Tools and Techniques by David Hussman**

This session will focus on tools and techniques for tracking an agile project plan from creation to project completion.

### **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.

### **Pragmatic Unit Testing with TestNG and EasyMock by Howard Lewis Ship**

You've heard about unit testing but were daunted when it came time to put the pedal to the metal. That's because JUnit is just one tool and there's others you need to learn about, including the wonderful and wierd EasyMock and the easy and powerful TestNG.

### **Tapestry In Action (Part One) by Howard Lewis Ship**

An introduction to the Jakarta Tapestry web application framework, which will explain the concepts and features of the framework while live coding simple applications. Tapestry forms, request cycle, component object mode. The use of several important components (including the powerful Table data grid) will be featured.

### **Tapestry In Action (Part Two) by Howard Lewis Ship**

In Tapestry, components are not an add-on; in fact, anything but! Tapestry components are integral to the entire framework # if something dynamic is going on in a page, there's a component involved.

### **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.

### **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.

### **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 Intro by Justin Ghtland**

The Spring framework is one of the fastest growing open source frameworks. New job postings are gaining rapidly, and many customers are adopting Spring instead of heavier alternatives. In this session, we'll introduce Spring. You'll see how Spring can give you much of the power of EJB, without the complexity or pain. Spring uses concepts like dependency injection and aspect oriented programming to ease standard

enterprise development. Spring developers write plain, ordinary Java objects (POJOs), instead of sophisticated components. In this session, you'll see a basic Spring application. You'll also see some details about some of the enterprise integration strategies, including: # Spring AOP # Transactions # Persistence # Model/view/controller When the session is over, you won't be an expert, but you should have a much clearer understanding of what Spring does, what it doesn't do, and why it's growing so rapidly.

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

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.

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

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

### **Pragmatic Extreme Programming Part 1: Planning & Design by Neal Ford**

This session begins a detailed discussion about how to actually get XP done in the real world (and what to tell your boss). This session includes artifacts (project tracking sheets, code coverage reports, etc.) from real XP projects.

### **Pragmatic Extreme Programming Part 2: Architecture, Coding, and Testing by Neal Ford**

Continues the discussion from Part 1, focusing on how to keep the benefits of XP without sacrificing its effectiveness. This session shows real artifacts of XP in action.

### **Real-world Agile Development by Neal Ford**

Lots of developers want to use Agile development technique but don't know where to start. This session discusses how to get started with Agility, the key benefits you can expect, and the pitfalls to avoid.

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

Is Service Oriented Architecture 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. I talk about SOA, ESB, CORBA, your MOM, and a bunch of other acronyms.

### **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.

### **Ajax, Flash, and Java - Choosing The Right Rich Client Technology for Your Next Project by Scott Delap**

Today's users are beginning to demand richer and richer application experiences. Plain html pages simply don't cut it anymore. Applications like Google Maps (Ajax) and Yahoo Maps (Flash) show how the UI experience can be pushed to the next level. As an IT manager, how do you decide which route to take however? Should you use Ajax because it is the new "it" technology. Is Flash a viable option with its 95%+ browser availability? Perhaps Java deployed through web start is really the best choice in contrast to what the buzz would lead you to believe. This presentation takes a look at these three core rich client technologies from both deployment/user experience and ease of development perspectives.

### **Creating Polished Swing Applications by Scott Delap**

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.

### **Introducing the Eclipse Rich Client Platform by Scott Delap**

Rich client application development using Java can be intimidating giving the vast flexibility in application design and structure. It also can be frustrating to create the large number of support services (persistence, menus, event and job frameworks) that a large scale rich client applications needs. The Eclipse Rich Client Platform is one project attempting to solve these issues by providing a core infrastructure that not only provides the day to day services a rich client application developer needs, but also providing a suggested path to guide you down the road of designing your application. This presentation introduces both the Eclipse RCP and the tools provided by the Eclipse IDE that assist developers in writing RCP apps.

### **Extend the Customization Possibilities of Your Java App with Scripts by Ted Neward**

Ever wished you could just put parts of your program in end-users' hands and let them build the infinite little changes they want? Ever thought about how you might make your application more robust by writing less code, not more? Embed a scripting engine into your application--complete with the safeguards necessary to ensure that users can't do anything they shouldn't be able to--and release yourself from the Principle of Perpetual Enslavement.

### **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.

### **Effective Enterprise Java: State Management by Ted Neward**

Managing state--both transient state (like your shopping cart) and your durable state (like your order placements, your inventory management forms, and so on)--is tricky in an enterprise application. In this talk, we'll examine some of the trickiness, both high-level and low-

### **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.

### **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.

### **Groovy for Java Programmers by Venkat Subramaniam**

Object-oriented scripting languages, or agile dynamic languages, as some like to call those, are gaining programmers' attention. Groovy bring this excitement to the Java platform with its ability to generate byte code. You can use Groovy instead of Java for some parts of your application. By learning it, you can switch between the languages where you consider fit.

### **Open Source Tools for Agile Development by Venkat Subramaniam**

As a Java developer, you have taken the time to learn the basics of the language and relevant parts of its rich API. However, you need more than that to develop serious industrial strength applications. In this presentation, the speaker will introduce you to a number of open source tools which you can use to improve your application quality and your development process.

### **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.

### **Programming with Mock objects by Venkat Subramaniam**

You are convinced that Test Driven Development is good for you and your project. You realize the benefits it has to offer. What's holding you back? All the code and components that your code so heavily depends on is most likely making you wonder if TDD is really for you. We will start out by looking at dependency and dependency inversion. Then we will discuss how mock objects can help separate our code from its

dependencies.

### **Refactoring your code - a key step in agility by Venkat Subramaniam**

Refactoring is one of the core practices in Agile Software Development. Refactoring is based on some core principles that apply to more than writing good code. But, what's refactoring? Why should you do it? How do you go about doing that? What tools are available to successfully refactor your App?

### **Working with Rules Engines 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.