

# Western Canada Java Software Symposium 2006

Sheraton Cavalier - Calgary, Alberta

September 22 - 24, 2006

(session listing as of 9/22/2006)

The No Fluff Just Stuff Software Symposium 2006 tour is pleased to announce the Western Canada Java Software Symposium coming to Calgary on September 22 - 24, 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 Western Canada Java Software Symposium will offer 4 concurrent sessions over three days with over 44 sessions to choose from. The following topics will be featured:

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

## Registration Fees

Attendees	Before 9/4/2006	After 9/4/2006
1-4	\$775	\$875
5-9	\$700	\$775
10-14	\$675	\$750
15-24	\$650	\$725
25+	\$625	\$700

## 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.

# Western Canada Java Software Symposium 2006

Sheraton Cavalier - Calgary, Alberta

September 22 - 24, 2006

(session listing as of 9/22/2006)

## "Bottom 10" Reasons that Agile Teams Fail by Clinton Begin

Agile methodologies such as eXtreme Programming and SCRUM are hot topics today -- and they are also hot targets. When things go wrong on an agile project, it's far too convenient to blame the methodology.

## Ruby Rebuttal: A case for Java by Clinton Begin

By jumping from Java to Ruby, are we throwing the baby out with the bathwater?

## iBATIS and the Enterprise Database 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 such as iBATIS must be sought.

## 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.

## Java Performance Myths by Glenn Vanderburg

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.

## 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.

### **Under the Hood of Java Memory Management by Glenn Vanderburg**

Most of the time, Java's automatic memory management works really well—it's one of the things that makes programming in Java a pleasant and productive experience, and it's nice that we don't have to worry about managing memory manually. However, although it's usually nice to ignore memory management, occasionally we have to pay close attention. Sometimes we need to take control of certain aspects of memory management. Sometimes Java programs do exhibit memory leaks, or unacceptably long garbage collection pauses, or very poor overall performance. But because Java's memory management is supposed to be "fully automatic," it can be difficult to find out what's really going on inside the VM.

### **Advanced Hibernate by Justin Gehrtland**

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 Gehrtland**

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 Gehrtland**

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.

### **Programming Java Concurrency by Justin Gehrtland**

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.

### **Spring Dependency Injection by Justin Gehrtland**

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 Gehrtland**

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.

### **Continuous Database Integration by Paul Duvall**

Performing daily or continuous builds is essential for ensuring working software. Yet, most consider only the source, not the database, as a part of these builds. What's good for the source code is also good for your database.

### **Continuous Integration using CruiseControl and Subversion by Paul Duvall**

Continuous Integration (CI) is the process of continually building and testing your software under development. It is identified as a core XP practice, although it works with many software development processes.

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

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

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

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

### **Holistic Testing by Scott Davis**

Mark Twain once said, "Everyone talks about the weather, but nobody does anything about it." Do you feel the same way about Unit Testing? Are you actively testing your code, or are you just thinking about testing your code... some day... once you get some more free time...

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

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

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

In this talk, we'll survey the web services exposed by leading websites (Google, Yahoo, Amazon, eBay) and discuss how they are driving the AJAX revolution. You'll see examples of RESTful, SOAP, and JSON web services, as well as the strengths and weaknesses of each.

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

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

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

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

### **Java Annotations: From Definition to Consumption 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).

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

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

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

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

### **J2EE Security @ Work: J2EE Meets JAAS by Tom Marrs**

Have you wasted time writing lots of security-based code and ever wondered if there's a better way to add security to your application? Are you confused by declarative security? Have you read about JAAS (Java Authentication and Authorization Service) but wondered where it fits? Have you ever said, "Can I just see a working example"? If so, then this talk is for you.

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

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

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

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

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

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

### **Good, Bad and Ugly of Java Generics by Venkat Subramaniam**

Java introduced Generics in the 1.5 version (Java 5). What are the capabilities of Generics? How do you use it? Are there some gotchas in using it? In this example driven presentation, we will start at the basics of generics and look at its capabilities. We will then look at some of the under the hood details on generics implementation. We will then delve into the details of some of the changes to Java libraries to accommodate generics. Finally we will take a look at some restrictions and pitfalls that we need to be familiar with when it comes to practical and prudent use of generics.

### **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 brings 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.

### **Prudent OO Design by Venkat Subramaniam**

Is your code object-oriented? Developing with objects involves more than using languages like Java, C#, C++ or Smalltalk for that matter. From time to time, the OO paradigm stumps even expert developers. Agile programming becomes a mere act of hack if we code without knowing the OO principles. What are these principles # the ones that influence your design? In this presentation the speaker will present some of the challenges that are fundamental in nature. Then he will present OO Design principles and good practices for prudent development of OO code.

### **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?

### **Test First Development by Venkat Subramaniam**

Do you know that unit testing is more of an act of design than verification? What are its benefits? How do we write effective tests? How does unit testing relate to evolutionary design? How does it help you with refactoring? When should you write your tests? What are the types of tests you could write? These are some of the questions that you would ask if you are interested in Unit Testing. What is a better way to learn than practicing it? In this session the attendees will participate in designing and developing a small yet full application. Instead of PowerPoint slides, you will learn from example. The code you help develop will be available for free download on the speaker's web site.

### **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.