

# Desert Southwest Software Symposium 2005

Sheraton Phoenix Sky Harbor Airport Hotel - Tempe, AZ

July 29 - 31, 2005

(session listing as of 7/28/2005)

The No Fluff Just Stuff Software Symposium 2005 tour is pleased to announce the Desert Southwest Software Symposium coming to Tempe on July 29 - 31, 2005. You will have the opportunity to attend the best technically focused Java / Open Source event anywhere. We make this claim based on the following:

- 1) **Excellent Speakers with unparalleled access**
- 2) **Limited Attendance = 250 Registrants Max**
- 3) **No Vendors, No Sales Pitches, No Marketecture**
- 4) **Unmatched Value - less than 1/3 of the cost of a national conference**
- 5) **Since 2002, we have delivered over fifty(50) conferences throughout North America**

The No Fluff Just Stuff Software Symposium Series caters to individual developers, development teams, project managers, architects and independent consultants. The Desert Southwest Software Symposium will offer 6 concurrent sessions over three days with over 65 sessions to choose from. The following topics will be featured:

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

## Registration Fees

Attendees	Before 7/4/2005	After 7/4/2005
1-4	\$675	\$775
5-9	\$600	\$675
10-14	\$575	\$650
15-24	\$550	\$625
25+	\$525	\$600

## The Registration Fee includes the following:

- 1) All Access Pass to the three day symposium
- 2) Handouts from all sessions attended w/binder
- 3) CD with all presentational content @ registration
- 4) Custom NFJS Laptop/Backpack
- 5) Opportunity to win an iPod everyday during the symposium raffle

Go to <http://www.nofluffjuststuff.com> for more details.

Questions/Comments: Contact Jay Zimmerman: [jjzimmerman@nofluffjuststuff.com](mailto:jjzimmerman@nofluffjuststuff.com) or (303)469-0486.

# Desert Southwest Software Symposium 2005

Sheraton Phoenix Sky Harbor Airport Hotel - Tempe, AZ

July 29 - 31, 2005

(session listing as of 7/28/2005)

## **Developer Component and System Testing with DbUnit by Andrew Glover**

The open source DbUnit framework provides an elegant solution for controlling a database dependency within applications by allowing developers to manage the state of a database throughout a test. With DbUnit, a database can be seeded with a desired data set before a test; moreover, at the completion of the test, the database can be placed back into its pre-test state.

## **Introduction to TestNG, the next generation testing framework for developers by Andrew Glover**

No one will argue the positive affects JUnit has had on the quality of thousands of Java applications around the world. JUnit's simplicity and ease of use ushered in a whole new era of code quality; however, as many developers have found, its simplicity has also limited its use. TestNG was designed from the ground up to overcome some of JUnit limitations.

## **Using Code Metrics for Targeted Code Refactoring by Andrew Glover**

The knowledge of how to effectively spot smelly code and replace it with proven patterns will ultimately lead to a more stable, maintainable and elegant code base.

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

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

## **SWT Fundamentals by Ben Galbraith**

The Eclipse project's SWT GUI toolkit provides one of the only viable alternatives to Swing for creating so-called rich client applications in Java. Whereas Swing paints its own widgets and has distinguished itself with a complex (and often obtuse) API, SWT relies on the host operating system for widget rendering and sports a simple, clean API. If your goal is to create a Java application that "looks" like a normal Windows application (or OS X, or Linux), SWT will revolutionize your world. In this session, I introduce SWT from the ground up. I start at a high-level, but quickly move into the details of SWT's API. By the presentation's end, attendees will have a solid understanding of SWT.

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

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

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

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

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

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

## **Advanced SWT and JFace by Ben Galbraith**

This session picks up where SWT Fundamentals leaves off. Among the advanced topics I discuss are creating custom SWT widgets and exploring tight native integration. I combine another compelling topic with the advanced SWT material: JFace. SWT is a more akin to AWT than Swing; it's concerned more with wrapping native functionality than providing any high-level abstractions. JFace is an API on top of SWT that provides such abstractions. The combination of SWT and JFace is comparable to Swing. My coverage of JFace includes an introduction to several of its frameworks, such as the Viewer and Window frameworks,

along with many examples. Learning JFace will enable you to write complex SWT applications much faster.

### **AJAX: Creating Next-Generation, Highly Dynamic, Off-line Capable Web Applications with HTML and JavaScript by Ben Galbraith**

As recent high-profile web apps such as Google's GMail have shown, modern browsers are capable of natively rendering web apps with highly dynamic and compelling UIs - fetching server data without page refreshes, animating and manipulating page contents on-the-fly, even offline use. The line between web and "desktop" apps is blurring.

### **XML made easy with XOM by Brian Sam-Bodden**

XML is quickly becoming the common ground for disparate systems to exchange information and most Java developers deal with XML almost on a daily basis, whether is in deployment descriptors and configuration files or as the data format at the center of their applications.

### **Advanced Object-Relational Mapping with Hibernate by Brian Sam-Bodden**

Hibernate is rapidly becoming the tool of choice when it comes to Object-Relational Mapping in Java. For simple applications with fairly simple object models and database schemas, using Hibernate is fairly straight forward. Unfortunately for most of us real applications have complex object-models that need to be wired to sometimes ancient and convoluted database schemas.

### **Business Rules Engines in Java and J2EE- An Introduction to the Drools Rules Engine by Brian Sam-Bodden**

Software development is expensive, when business rules are hard-coded in your application's source code, changes and additions to those rules translate to wasted time and money. Good object-oriented, component-based approaches can alleviate the burden of keeping up with changes in the business world but they still require that expert knowledge of the changes be passed from the decision makers to the business analysts and finally to programmers that need to implement these changes. Business Rule Engines and Business Rule Languages are based on the basic premise of separation of concerns by empowering business domain experts to express the rules of business in a way that it is directly usable by applications.

### **Complex Builds with Ant by Brian Sam-Bodden**

Ant has revolutionized the way we build applications in Java and it has become a de facto standard in the Java world. As applications grow in complexity some developers are finding themselves dealing with ever growing and complex builds. Complex builds have to deal with Multiple Operating System, multiple Application Servers, multiple APIs and multiple stages of development.

### **Politics of Persistence by Bruce Tate**

This session will help a Java developer choose a persistence framework. After the session, you will # Understand the core strengths and weaknesses of the main persistence frameworks in the Java space # Understand where marketing influences can impact persistence # Know what's going on behind the scenes to impact the persistence pictures # Answer questions about persistence frameworks that might not be mainstream

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

As dynamic languages become more popular, it pays to look at frameworks based in other languages.

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

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

### **Introduction to Spring by Bruce Tate**

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

### **Lightweight Development Strategies by Bruce Tate**

Based on the book Better, Faster, Lighter Java, this beginner to intermediate session will dive into philosophies for lightweight development. It's not a hardcore programming session, but we will talk about

process, technologies like Spring, and design patterns like AOP and Dependency Injection. This philosophical session will talk about architectural philosophies rather than low-level programming issues.

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

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

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

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

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

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

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

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

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

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

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

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

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

Struts is the most popular Java-based Web application framework today, but that's rapidly changing. There's a newcomer on the block, a leaner, meaner, better-designed framework loosely based on Struts that's poised to dethrone Struts as the reigning king of Java-based web application frameworks. That framework, of course, is JavaServer Faces. Craig McClanahan, the father of Struts and the co-spec lead for JSF 1.0, has proposed reinventing Struts for Struts 2.0 as a set of services for JSF applications. That new framework, which has no direct ties to Struts as we know it, is called Shale.

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

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

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

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

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

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

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

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

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

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

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

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

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

### **Introduction to Aspect-oriented programming with AspectJ by Ramnivas Laddad**

Aspect Oriented Programming (AOP) enables modularizing implementation of crosscutting concerns that abound in practice: logging, tracing, dynamic profiling, error handling, service-level agreement, policy enforcement, pooling, caching, concurrency control, security, transaction management, business rules, and so forth. Traditional implementation of these concerns requires you to fuse their implementation with the core concern of a module. With AOP, you can implement each of the concerns in a separate module called aspect. The result of such modular implementation is simplified design, improved understandability, improved quality, reduced time to market, and expedited response to system requirement changes. Come to this session and learn all about how AOP can help you simplify developing complex systems.

### **Performance monitoring in J2EE applications by Ramnivas Laddad**

J2EE has become the main new platform for enterprise application deployment. Good performance is an important requirement from the business viewpoint. Supporting this requirement needs application profiling during the development phases and performance monitoring after deploying the application. Come to this session to understand challenges and choice in monitoring J2EE applications.

### **Design Pattern Modularization with AOP by Ramnivas Laddad**

Design patterns # object oriented, concurrency control, and J2EE # all have certain crosscutting elements present. The obvious result of conventional implementation is unclear implementation that is tedious to implement and tough to change. Aspect-oriented programming (AOP) offers a way to simplify implementation of these design patterns. Further, AOP offers new design patterns of its own that allow for new ways of implementing functionalities.

### **Aspect-oriented refactoring: Taking refactoring to a new level by Ramnivas Laddad**

Refactoring allows reorganizing code while preserving the external behavior, while AOP facilitates modularizing crosscutting concerns in a system through use of a new unit of modularity called aspect. Aspect-oriented refactoring synergistically combines these two techniques to refactor crosscutting elements. Individually, refactoring and AOP both share the high-level goal of creating systems that are easier to understand and maintain without requiring huge upfront design effort. A combination of the two # aspect-oriented refactoring # helps in reorganizing code corresponding to crosscutting concerns to further

improve modularization that is easy to understand, highly consistent, and simple to change.

### **Java Generics in Depth by Ramnivas Laddad**

The Java generics facility in Java 5.0, similar in spirit to C++ templates, enables implementing parameterized types. Using this facility, you can get more help from the compiler to write type-safe code and avoid many ugly casts. While Java generic facility is simpler than C++, make no mistake; it brings its own set of intricacies! Find out all about this important feature in Java 5.0.

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

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

### **Class Loading in Java: Building Dynamic Systems Without Pain by Stuart Halloway**

(3 Hour Session) One of Java's greatest strengths is its flexible deployment model. In this session you will learn how Class Loaders facilitate deployment, and how to troubleshoot Java and J2EE Class Loading problems.

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

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

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

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

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

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

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

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

### **Effective Enterprise Architecture by Ted Neward**

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

### **The Fallacies of Enterprise Systems (Architecture) by Ted Neward**

There's a set of fallacies that every enterprise developer has fallen for at some point in their enterprise development lives, and unless they've come to realize it early enough, all cause big trouble and painful learning experiences in the long run.

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

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

### **Working with Java Metadata by Ted Neward**

As part of JDK 1.5, Java has introduced a facility for developers to create and use custom metadata annotations, as developed by the JSR 175 committee. This represents a radical new shift for the Java programming language, quite possibly larger and farther-reaching than generics or any other language feature.

### **Effective Enterprise Java: Security by Ted Neward**

Security's become a hot topic among enterprise developers in recent years, but to many developers, security is still the white elephant in the middle of the room. Discussions about security usually begin with, "Uh, we'll worry about that later", or, "Start with two really large prime numbers.....". Security isn't as hard as

developers make it out to be, but it is something that developers need to face and recognize.