

Lone Star Software Symposium

Embassy Suites Park Central

October 26 - 28, 2007

<http://www.nofluffjuststuff.com/sh/2007-10-dallas>

(event schedule as of October 24, 2007)

| Fri, Oct. 26, 2007 | | | | |
|--------------------|---|--|---|--|
| | Plano/Richardson | Denton | Addison | Dallas |
| 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 | Domain Driven Design Venkat Subramaniam | Give it a REST Brian Sletten | JavaServer Faces: A Whirlwind Tour David Geary |
| 2:45 - 3:15 PM | BREAK | | | |
| 3:15 - 4:45 PM | Groovy and Java: The Integration Story Scott Davis | Annotation Hammer Venkat Subramaniam | RESTlet for the Weary Brian Sletten | Killer JavaScript Frameworks: Prototype, Scriptaculous, and Rico David Geary |
| 4:45 - 5:00 PM | BREAK | | | |
| 5:00 - 6:30 PM | Real World Grails Scott Davis | get Fit Venkat Subramaniam | NetKernel : XML Processing for the 21st Century Brian Sletten | Ajaxian Faces David Geary |
| 6:30 - 7:15 PM | DINNER | | | |
| 7:15 - 8:00 PM | Keynote: by Scott Davis | | | |

| Sat, Oct. 27, 2007 | | | | |
|--------------------|---|--|---|---|
| | Plano/Richardson | Denton | Addison | Dallas |
| 8:00 - 9:00 AM | BREAKFAST | | | |
| 9:00 - 10:30 AM | The Secrets of GORM Scott Davis | Domain Specific Languages Venkat Subramaniam | Data Integration : Beyond Cutesy Mashups Brian Sletten | RAD JSF with Seam, Facelets, and Ajax4jsf, Part One David Geary |
| 10:30 - 11:00 AM | BREAK | | | |
| 11:00 - 12:30 PM | Spring in Action Craig Walls | Drooling with Groovy and Rules Venkat Subramaniam | Applied AOP Brian Sletten | RAD JSF with Seam, Facelets, and Ajax4jsf, Part Two David Geary |
| 12:30 - 1:30 PM | LUNCH | | | |
| 1:30 - 3:00 PM | Spring Cleaning: Tips for managing XML clutter in your Spring configuration Craig Walls | JRuby Venkat Subramaniam | Monitoring Software Quality with Continuous Integration Andrew Glover | The Google Web Toolkit, Part One David Geary |
| 3:00 - 3:15 PM | BREAK | | | |
| 3:15 - 4:45 PM | Spring-WS: Contract first web-services for Spring Craig Walls | Harnessing the Power of Maven Ryan Breidenbach | Behavior-driven development in Java Andrew Glover | The Google Web Toolkit, Part Two David Geary |
| 4:45 - 5:30 PM | BIRDS OF A FEATHER SESSIONS | | | |

| Sun, Oct. 28, 2007 | | | | |
|--------------------|--|---|--|--|
| | Plano/Richardson | Denton | Addison | Dallas |
| 8:00 - 9:00 AM | BREAKFAST | | | |
| 9:00 - 10:30 AM | Groovy For Java Programmers Jeff Brown | OSGi: A Well Kept Secret Venkat Subramaniam | Enterprise Performance and Scalability Ted Neward | Developing (and testing) with the GWT from end to end Andrew Glover |
| 10:30 - 11:00 AM | BREAK | | | |
| 11:00 - 12:30 PM | Advanced Metaprogramming With Groovy Jeff Brown | Internationalization and Localization in Java David Bock | The Busy Java Developer's Guide to Debugging and Monitoring Ted Neward | When is an Estimate Just an Estimate? Derek Lane |
| 12:30 - 1:15 PM | LUNCH | | | |
| 1:15 - 2:15 PM | EXPERT PANEL DISCUSSION | | | |
| 2:15 - 3:45 PM | Test Driven Development With Groovy And Grails Jeff Brown | Introducing Agility to Large Organizations David Bock | The Busy Java Developer's Guide to Java Platform Security Ted Neward | Refactoring Ant builds with Ivy, Groovy, and good old fashion common sense Andrew Glover |
| 3:45 - 4:00 PM | BREAK | | | |
| 4:00 - 5:30 PM | Introduction To Agile Web Development With Grails Jeff Brown | Maintaining Project Integrity with JDepend, Macker, PMD, Maven, and other open source tools David Bock | Java6: Exploring Mustang Ted Neward | Everything Old Is New Again Glenn Vanderburg |

Lone Star Software Symposium

Embassy Suites Park Central

October 26 - 28, 2007

<http://www.nofluffjuststuff.com/sh/2007-10-dallas>
(event schedule as of October 24, 2007)

Monitoring Software Quality with Continuous Integration by Andrew Glover

The practice of continuous integration facilitates early visibility into the development process by regularly conducting software builds, thus integrating disparate software pieces earlier than later, which often times minimizes the interval between when a defect is coded and when it is discovered. Given the automated nature of continuous integration spawned builds, software teams can now start to look at their build process as something more useful than a simple compile and test process.

Behavior-driven development in Java by Andrew Glover

Behavior-driven development, or BDD, has attracted a lot of attention a la RSpec in the Ruby community, but BDD's roots stem from JBehave, a Java based framework. In this session, we'll look at what BDD is and how it shifts the traditional testing vocabulary from being test-based to behavior-based.

Developing (and testing) with the GWT from end to end by Andrew Glover

Because GWT applications are written in plain old Java, they can be easily incorporated into a traditional build process and your favorite IDE; plus, you can test aspects of these applications with JUnit, not to mention, everyone's favorite functional testing framework: Selenium.

Refactoring Ant builds with Ivy, Groovy, and good old fashion common sense by Andrew Glover

Are your Ant builds giant XML files that scream for attention? Why not enhance your build process to act like a quality gate, much like a test suite would?

Give it a REST by Brian Sletten

As developers, we sometimes get to make choices about the technologies we use, sometimes not. We base these decisions on personal experiences, recommendations from others and a general sense of where the industry is going. Web Services have been all the rage for several years now. We have been told time and again that we should be building systems around them; as an industry, we've never been more confused. Perhaps it is time to Give it a REST.

RESTlet for the Weary by Brian Sletten

If you have started to take a look at REST as way of exposing web services or managing information spaces, you may be frustrated by the support offered by legacy containers. There is no direct support for REST concepts in the J2EE specs (yet). XML-based configurations are so 1990's. Come learn about Restlets, a little API that has caught the attention of many in the RESTafarian community.

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.

Data Integration : Beyond Cutesy Mashups by Brian Sletten

Ever since we started doing relational joins, we've looked for ways to tie data together. The web has given us no end of new data sources to integrate but it seems like the best we can come up with is locating Starbucks on Google Maps. The problem with browser-based mashups is that they don't survive the session, we have no way of referring to the results in future queries and ultimately we don't maintain ownership or control of the process. We want control of our data and our mashup results. We want ever more ways to view, explore and requery them in multi-faceted ways. Do you know what your data integration strategy is for the next few years? Are you sure? You owe it to yourself to come find out.

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. Now that Spring 2.0 provides support for AspectJ, the time has never been better to learn about these new (but backwards compatible) ways of thinking about building software.

Spring in Action by Craig Walls

Spring has been one of the most exciting frameworks to emerge in the past few years. With Spring you can decouple your application's objects, enrich them with AOP, and apply transactional boundaries and security to them declaratively. It simplifies data access, remoting, web services, and JMS. It comes with its own web framework. And, even though Spring eliminates much of the need for EJBs, it will still integrate nicely with any EJBs you may have lying around. What's not to love?

Spring Cleaning: Tips for managing XML clutter in your Spring configuration by Craig Walls

The biggest complaint about Spring is the vast amount of XML required to configure an application. In this presentation, I'll show you ways to reduce or even eliminate much of the XML required to configure Spring.

Spring-WS: Contract first web-services for Spring by Craig Walls

Many web-service platforms make web-services easy by simply SOAP-ifying an object's interface. That's certainly a quick way to get started with web-services, but what happens when the object's interface changes?

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.

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.

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.

Killer JavaScript Frameworks: Prototype, Scriptaculous, and Rico by David Geary

An introduction to the popular Prototype JavaScript framework, and two frameworks built on top of Prototype: Scriptaculous and Rico.

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.

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.

When is an Estimate Just an Estimate? by Derek Lane

All software projects have to provide estimates. Webster defines estimate as, "to determine roughly the size, extent, or nature of". Whens the last time you provided an estimate that wasn't taken as a guarantee? The movement toward Agile software development has been to put estimates back in their place, as rough sizing techniques. Estimating should be quick, yet based on something concrete so that it has meaning. This session will introduce some simple techniques for estimating things on your software projects (requirements, tasks, etc.), and provide hints on variations that have been applied on real world projects. These techniques are aimed at lowering the bar to adopting and integrating true estimation into your current 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.

Groovy For Java Programmers by Jeff Brown

Groovy is an agile dynamic language for the Java platform. Groovy has a Java like syntax along with many features inspired by languages like Python, Ruby and Smalltalk. This session covers a lot of ground including many interactive examples to hilite the powerful language features that make Groovy compelling. A lot of momentum is building in the Groovy and Grails communities right now and this session is aimed at Java developers who want to leverage the power of Groovy.

Advanced Metaprogramming With Groovy by Jeff Brown

The dynamic nature of Groovy makes it a fantastic language for building dynamic applications for the Java Platform. The metaprogramming capabilities offered by the language provide everything that an application development team needs to build systems that are far more capable than their all Java counterparts. Taking advantage of Groovy's metaprogramming capabilities brings great new possibilities that would be very difficult or just plain impossible to write with Java alone. Building Domain Specific Languages in Groovy is easy to do once a team has a good understanding of the Metaobject-Protocol (MOP) and the method dispatch mechanisms used by the Groovy runtime environment.

Test Driven Development With Groovy And Grails by Jeff Brown

The value of Test Driven Development (TDD) has become widely accepted. The practice has extended beyond just XP teams. Good TDD practices yield high quality software and help teams maintain confidence in their software as complexity grows. The dynamic nature of Groovy makes TDD easy and fun. Groovy may be used to unit test not only Groovy code but other code as well. Testing Java code with Groovy is a snap. Learn to use the power of Groovy to test your systems.

Introduction To Agile Web Development With Grails by Jeff Brown

Grails brings the powerful "coding by convention" paradigm to Groovy and Java. Grails is not just another flavor in the pool of web development frameworks for Java. Grails leverages the powerful dynamic features of Groovy while taking advantage of best of breed technologies like Hibernate, Spring, Sitemesh and Quartz to make web application development both fun and easy.

Harnessing the Power of Maven by Ryan Breidenbach

2006 appeared to be the year that Maven achieved the momentum it needed to overtake Ant as the build tool of choice for Java developers. A lot of that has to do with the vastly improved Maven 2. But a lot of it has to do with the simplicity, organization and power that Maven brings to projects. The session will bring developers new to Maven with everything it has to offer. This includes creating your very first Maven project, learning the significance of the POM file, how to let Maven and its repositories manage your dependencies, and how to let Maven report of the health of your own projects. And for the Ant users in the audience, you will get to see a side-by-side comparison of the two build tools' build philosophies.

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 Secrets of GORM by Scott Davis

GORM (the Grails Object/Relational Mapper) is one of the many high points of the Grails web framework. GORM is a thin Groovy wrapper over Hibernate, but that doesn't begin to capture excitement of what GORM brings to the party. Imagine being able to call `book.save()` and `book.delete()` on your `Book` class; calling `Book.get(1)` to retrieve your book from the database by primary key; using `Book.list()` to pull an `ArrayList` of `Book` objects into your application. Now imagine getting all of that functionality (and more) for free with each new class you define. No interfaces to implement. No abstract classes to extend. Persistence that is transparent, automatic, and simple to use: GORM.

Enterprise Performance and Scalability by Ted Neward

Wondering why your enterprise Java app just? sucks? Trying to figure out why you can't get more than 10 concurrent users online at the same time? Looking for ways to try and spot the slowdowns and ways to fix them?

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 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.

Java6: Exploring Mustang by Ted Neward

Mustang, the Java6 release, is out, 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.

Domain Driven Design by Venkat Subramaniam

Domain Driven Design (DDD) is an approach that places emphasis on the domain model and carrying it into implementation. DDD is mostly repackaging of fundamental OO Design. It brings new emphasis to what we should be already doing, but often find it hard and confusing given the realities and complexities of our real world. In this presentation we will take a close look at what DDD is and how to use it for agile development. We will discuss several design options, and also look at some examples of good modeling and layering.

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.

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.

Domain Specific Languages by Venkat Subramaniam

Domain Specific Languages or DSLs are languages that target a specific kind of problem or domain. We've had various degree of success with DSLs, over the past several years, in narrow areas. However, DSLs are not widely used in general purpose application partly because the popular widely used languages today do not make it easy.

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.

JRuby by Venkat Subramaniam

Object-oriented scripting languages, or agile dynamic languages, as some like to call those, are gaining programmers' attention. Several dynamic languages are on the JVM. Groovy and JRuby are two languages

that are drawing developers' interest. Sun has shown support for these two, and especially JRuby by hiring the core developers.

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.