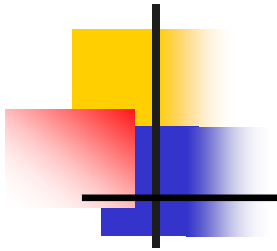
What's New in the Servlet and JavaServer Pages Technologies?

Noel J. Bergman

DevTech®

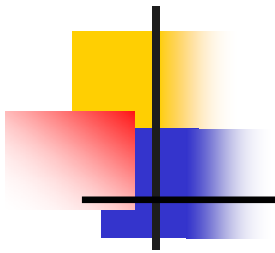


Session Overview

What are all the new features in Servlet, JavaServer Pages and related Java specifications that you are not yet taking advantage of? This talk will cover enhancements in these critical core web application technologies, and how you can benefit from them.

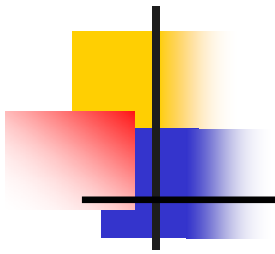
Topics will include Listeners, Filters, Expression Language, sessions, standard tags, and creating custom tags; with additional, lighter, discussion of companion specifications, specifically Portlets and Java Server Faces.

PLEASE ASK QUESTIONS! ☺



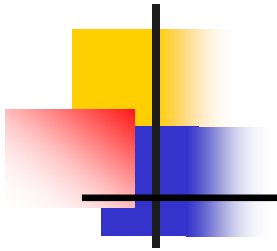
Session Prerequisites

- You will want a background understanding of Servlets and JavaServer Pages, but we'll be talking about what's been happening in these areas.
- You will want to understand the MVC pattern.
- In order to work with the technology covered during the session, you will need a relatively recent Java application server, such as current releases of Apache Tomcat, perhaps with Apache Jetspeed-2 and/or Apache MyFaces, if you want to explore those technology areas. Or commercial products such as IBM WebSphere or BEA WebLogic.



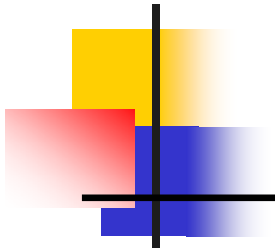
Web Tier Java Technologies

- The Web Tier contains the *Controller* and *View* components from the classic *Model, View, Controller* pattern. We're going to focus on the Java technologies that reside in the web tier, and the (potential) role that each plays.



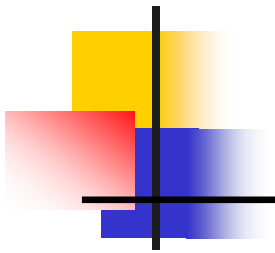
Topics

- Servlets
  - Recent changes are mostly clarifications and relatively minor enhancements. But most web developers aren't using things that are new since Servlet 2.0, so we're going to review key concepts, particularly Listeners, Filters, and security as separate topics.
  - The Servlet Specification is the core specification for the web container, and therefore governs, and is sometimes changed to suit, the related specifications.
- JavaServer Pages
  - Tag Files
  - Unified Expression Language
- JavaServer Faces
  - For most developers, this whole technology is new, plus we have new features in JSF 1.2, such as the Unified Expression Language that is shared with JSP.
- Portlets
  - As with JSF, Portlets are new to most of us, but are an increasingly important area for web developers, particularly as we move to adopt SOA.
  - Although still only in the Early Draft stages, we can already see several important new features planned for the Portlet 2.0 API.



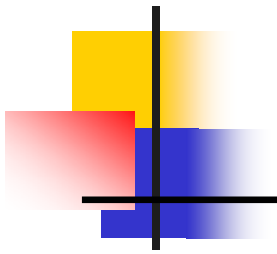
Servlets

- URL addressable end-points for request/response processing in a web container.
- Servlets are the Controller in the MVC pattern.
- They:
  - Take incoming requests;
  - Locate, convert and validate the parameters;
  - Invoke method(s) on the Model as necessary;
  - Select the View;
  - And orchestrate moving data from the Model to the View.



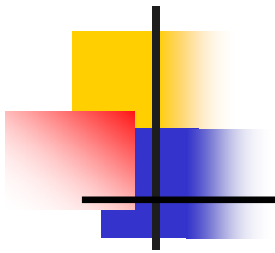
Major HttpServlet Methods

- init()
- service()
- getLastModified()



Portlets

- Portlets are also the Controller in the MVC pattern.
- They:
  - Take incoming requests;
  - Locate, convert and validate the parameters;
  - Invoke method(s) on the Model as necessary;
  - Select the View;
  - And orchestrate moving data from the Model to the View.



Major Portlet Methods

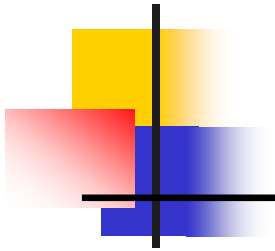
- Portlet
  - init()
  - processAction()
  - render()
- EventPortlet
  - processEvent()
- ResourceServingPortlet
  - serveResource()

Major Changes in Portlet 2.0

- JSR-286 is still in the Early Draft stages
  - **NOW** is your chance to make comments!
- Major new features:
  - Events
  - Filters
    - <lifecycle>
    - Weakly typed – often requires down-casting ☹
  - Resource serving
  - Shared session attributes (***removed!***)

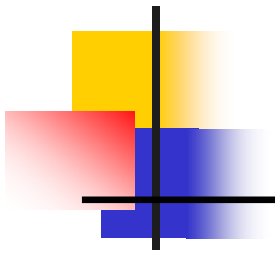
Servlet *vs.* Portlet

- Both fulfill the Controller role, and perform very similar tasks in so doing.
- Addressability
  - A servlet has a defined mapping, and is a URL addressable resource.
  - A portlet's URL is definable only by invoking portlet APIs, and its string representation is entirely vendor specific.
    - Some vendors may expose a non-standard URL mapping, such as that provided by IBM WebSphere Application Server v6.1 (not to be confused with WebSphere Portal Server)
      - ✓ Ref: http://publib.boulder.ibm.com/infocenter/wasinfo/v6r1/index.jsp?topic=/com.ibm.websphere.base.doc/info/aes/ae/cport_url_address.html
- Content
  - A Servlet's response generally represents a page, making edge caching easier.
  - A Portlet's response is a markup fragment that is intended to be aggregated on a page. There is even a specification, part of Web Services for Remote Portlets (WSRP), for the CSS styles to be used in the markup.
- Request/Response Handling
  - A servlet handles a request and generates the response on every round trip to the server.
  - A portlet's request/response lifecycle is more complex, with two discrete phases: an optional event phase, and a render phase.
- Portlets are intentionally placed on equal footing with Servlets in this presentation, rather than as an "after thought" at the end of the session. As network-based applications evolve to use a Service Oriented Architecture, Portlet-based UIs are expected to replace monolithic servlet-based UIs.



JavaServer Pages

- The standard Java technology for the View role.
- Markup language optionally mixed with custom tags, expression language, and scripting elements (discouraged, except within tag files).
- Compiled to, and executed as, servlets.

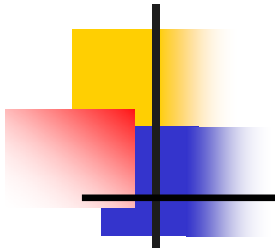


Major Changes in JSP

- JavaServer Pages 2.1
  - Most changes are clarifications and minor usability tweaks, with the exception of a new Unified Expression Language and the API changes necessitated by it.
- JavaServer Pages 2.0
  - Tag Files
  - JSP Fragments
  - The Expression Language is a standard part of JSP, and not tied to the JSTL.

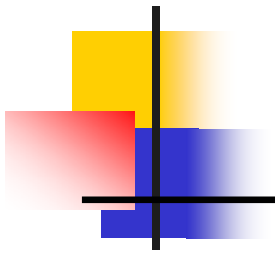
Unified Expression Language

- Now defined as a separate document, although still within the JSP Specification group for the moment.
- Supports both `${expr}` (from JSP) **and** `#{expr}` (from JSF), with accompanying TLD enhancements
  - "JSP-style" `${}` expression evaluation is immediate
  - "JSF-style" `#{}` expression evaluation is deferred
- No need for YAIFT (Yet Another IF Tag)
  -
```
<c:if test="${expr}">
  conditionally executed
</c:if>
```
  - One generic tag fits all needs
- The same applies to many bean access tags.



Unified Expression Language

- Generalized `[]` and `.` Expressions
- New "scope" objects
  - `pageContext`
  - `pageScope`
  - `requestScope`
  - `sessionScope`
  - `applicationScope`
  - `param/paramValues`
  - `header/headerValues`
  - `cookie`
  - `initParam`

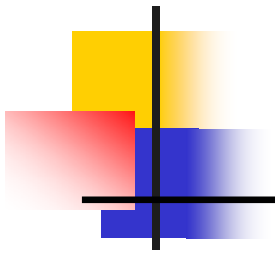


JavaServer Faces

- The Java MVC framework standard.
- For building web apps that have more complex controller needs.
- Combines a core runtime, a generic servlet or portlet, and application specific code.
- Supports pluggable converters, validators and a declarative nagivation mode.
- The View is made from JavaServer Pages with JSF tags and binding expressions.
- User code provides (a bridge to) the model.

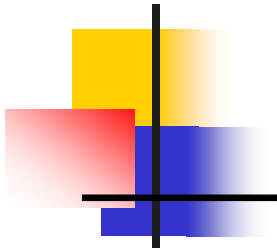
Major Changes in JSF

- JavaServer Faces 1.2
  - The Unified Expression Language
    - Various changes to the API to accommodate the EL
  - Improved ability to use JSP, JSTL and other JSP custom tags with JSF
  - "The usual clarifications"
  - Incremental enhancements



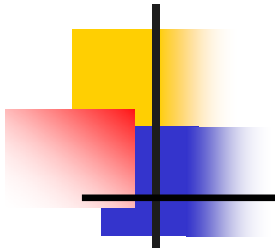
Why JSF?

- Ease of use due to EL-based bindings
  - Value bindings automatically synchronize UI values and JavaBean properties.
  - Method bindings automatically invoke JavaBean methods when the UI command component is clicked.
- Flexible mechanisms for automatic data conversion and validation.
- Strong separation of responsibilities between Java Developers and UI Designer, with ease of use for the UI Designers.
- Major focus by vendors and independent developers.



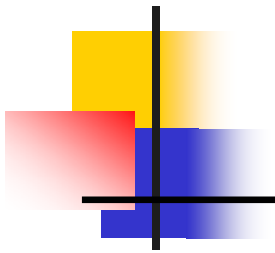
Listeners

- Receive notification of lifecycle events generated by the Servlet/Portlet container.
- Events
  - Application Lifecycle (v2.3)
    - Application Context created/destroyed
    - Application scope attribute changes (add, remove, replace)
  - Session Lifecycle (v2.3)
    - Session created/destroyed
    - Session scope attribute changes (add, remove, replace)
    - Session attributes can receive events for:
      - ✓ Session activating/passivating
      - ✓ Session binding
  - Request Lifecycle (v2.4)
    - Request created/destroyed
    - Request scope attribute changes (add, remove, replace)
- Let's talk about some use cases ...



Filters

- Filters have an interface similar to servlets and portlets (will be introduced in Portlet API 2.0).
- In addition to a request and response, the method interface receives a chain, and is able to pass (optionally modified) requests and responses further down the chain, or to abort the chain before it reaches the servlet or portlet resource at its terminus.
- Filters are now able to process both requests from the outside as well as those internally dispatched.
  - <dispatcher> on <filter-mapping>
- Let's talk about some use cases...



Error Handling

- You can provide your own custom handling for specific HTTP codes and Java Exceptions by specifying resources to handle each one.
- You can even put a filter in front of the error handling resource.
- Let's talk about some use cases...

Container Managed Security

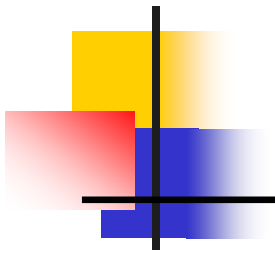
- J2EE containers, of which the servlet container is one, implement container managed security based upon roles.
- The servlet container is required to support several methods for authentication.
  - HTTP Basic/Digest authentication
  - HTTP Form-based authentication
  - HTTPS (SSL) client certificate based authentication
- The servlet container implements role based security based upon a series of entries in web.xml:
  - Security Constraint
    - Web Resource Collection
    - Authorization Constraint
    - Method Constraint
    - Transport Security Constraint
  - Login Configuration
  - Security Roles
- Let's talk about some use cases...

Programmatic Security

- Role-based
  - Request objects for both Servlets and Portlets provide methods for inquiring into the identity of the authenticated user making the request, and for verifying whether or not that user is in a given role. You can use this check to supplement container-managed security with more flexible role-based checks.
- Non-Role-based
  - As we did before the advent of container-managed security, you can use filters and/or custom code to supplement the security model, as per your needs.
  - For example, you can implement time-based and other non-role based access control to resources.
  - Filters would be the preferred mechanism to implement this sort of security, since they can be declaratively attached to web resources without code change.
- Please note: IBM WebSphere Portal 6.0 (current) and earlier provide a fixed set of Security Roles, and do not implement programmatic security as defined by JSR-168, although WebSphere Portal v6.0 does offer a form of programmatic security based upon IBM's personalization model.
  - Ref: http://publib.boulder.ibm.com/infocenter/wpdoc/v6r0/topic/com.ibm.wp.ent.doc/wps/jsrcpr.html#jsrcpr__limitations
  - Ref: http://publib.boulder.ibm.com/infocenter/wpdoc/v6r0/topic/com.ibm.wp.ent.doc/pzn/pzn_attadm.html
- Let's talk about some use cases...

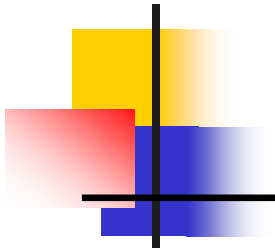
What about “Web 2.0”/AJAX?

- Web Application MVC Frameworks such as JSF provide developers with an easy means to integrate AJAX into applications, but unless security is taken seriously, and major structural changes occur, AJAX technology has the potential to create some of the worse cyber-security disasters since Microsoft Windows became capable of connecting to the Internet.
  - “Just as in the early days of desktop software, [security experts] say, the development momentum is all about features – and protections are being neglected.” CNET News, July 28, 2006
- Alex Russell, co-founder and project lead of The Dojo Toolkit, told an audience invited to discuss AJAX security issues that, "It's worth noting that the fundamental problems with browser security and Web application security haven't changed in five years [and] AJAX doesn't change that.“
- Are you really saying no to this amazing technology?! No, just a caution to adopt a security-first policy on adoption.
  - Will your users turn JavaScript off or take the risk? The NoScript plug-in for FireFox aims to provide a solution, allowing selective enablement of browser-side scripting.



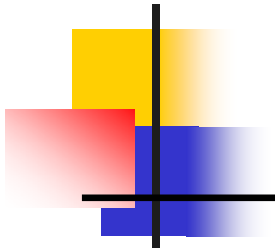
Resources

- Ed Burns' (JSF Specification co-lead) blog on the revisions
  http://weblogs.java.net/blog/edburns/archive/2006/02/new_drafts_of_j_1.html
- JSP 2.1 and JSF 1.2 Chat Transcript
  http://java.sun.com/developer/community/chat/JavaLive/2004/jl1207.html
- JSR 154 (Servlet 2.4/2.5)
  http://jcp.org/aboutJava/communityprocess/mrel/jsr154/index.html
  - Change Log
    http://jcp.org/aboutJava/communityprocess/maintenance/jsr154/JSR154ChangeLog.html
- JSR 245 (JSP 2.1)
  http://jcp.org/en/jsr/detail?id=245
- JSR 252 (JSF 1.2)
  http://jcp.org/en/jsr/detail?id=252
- JSR 286 (Portlet 2.0)
  http://jcp.org/en/jsr/detail?id=286



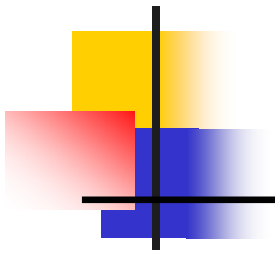
Resources

- AJAX Security Links
  http://www.cgisecurity.com/ajax/
- "The Security Risk in Web 2.0"
  http://news.com.com/The+security+risk+in+Web+2.0/2100-1002_3-6099228.html
- "Chapter and Verse on AJAX Security"
  http://search400.techtarget.com/qna/0,289202,sid3_gci1198365,00.html
- A Few Collected Tutorials Related To AJAX Security
  http://www.maxkiesler.com/index.php/weblog/comments/what_you_should_know_about_ajax_security_24_tutorials/



Resources

- Easy Custom Tags with Tag Files
  http://today.java.net/pub/a/today/2003/11/14/tagfiles.html
- Creating JSP 2.0 Tag Files
  http://www.oracle.com/technology/pub/articles/cioroianu_tagfiles.html
- Playing Tag with JSP 2.0
  http://www.vsj.co.uk/articles/display.asp?id=408



Related Sessions

- Web 2.0/AJAX – Enhancing the VC domain
  - "Building AJAX Enabled JSF Components" – Bill Dudney
  - "AJAX Performance and Monitoring" – Ron Bodkin
  - "Taming AJAX with GWT" – Scott Blum
  - "Real World AJAX" – Scott Davis

Related Sessions

- SOA – We're in the VC space, these are Models
  - "SOA: You Heard the Hype — Now See How It Is Really Done!" – Denise Hatzidakis
  - "The Evolution of Service-oriented Development" – Bill Jones
  - "Next Generation SOA Development" – Bill Jones
  - "Service Component Architecture" – Jean-Sebastien Delfino
  - "Open Source SCA — The Apache Tuscany Project" – Jean-Sebastien Delfino
  - "Service Data Objects" – Steve Brodsky
  - "Patterns and Anti-patterns in SOA" – David Moskowitz

Related Sessions

- Web Services – The Service in SOA
  - "Understanding the Web Services Architecture" – Rajith Attapattu
  - "Building Enterprise Applications with Apache Axis2" – Rajith Attapattu
  - "Why Axis2: The Future of Web Services" – Paul Fremantle