JSR344 (JSF 2.2) Status
Update to JCP EC
11 September 2012
Edward Burns @edburns
JCP Spec Lead
The following is intended to outline our general product direction. It is intended for information purposes only, and may not be incorporated into any contract. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions. The development, release, and timing of any features or functionality described for Oracle’s products remains at the sole discretion of Oracle.
My Plan For Your Time Investment

- JSR Summary and Scope
- Big Ticket Features
- Medium Sized Features
- Bug Fixes
My Plan For Your Time Investment

- How I run JSRs
  - What works Well, Puzzlers
- How features are suggested
- Transparency Tools
- Roadmap
Technical Content
SINCE 2004, JSF HAS BEEN THE STANDARD WEB APP FRAMEWORK FOR THE JAVA PLATFORM. JSF WAS DESIGNED TO BE A UI COMPONENT FRAMEWORK THAT SEPARATES UI SEMANTICS FROM RENDERING.
Technical Content

JSR Scope

- “Bring the best ideas in web application development to the Java EE Platform”
  - Maximize Productivity
  - Minimize Complexity

- Categories listed on JSR
  - Ease of Development
  - New Features
  - Fixes
Technical Content

- Faces Flows
- HTML5 Friendly Markup Support
  - Pass through attributes and elements
- Multi-Templating
- Cross Site Request Forgery Protection
- Loading Facelets via ResourceHandler
- File Upload Component
Technical Content

Faces Flows

- Lack of formal way to group pages leads to maintenance problems
  - Hard to understand navigation
  - Like a large program that is a single function with lots of goto statements

- Faces Flows goals
  - Allow grouping related views together
  - Allow building applications as composites of modules of functionality
Technical Content

Faces Flows

- JSF Navigation Story Recap
  - Defines a graph of VDL views
  - XML rule base defines relationship between views
    - Implicit rules added in JSF 2.0

```xml
<navigation-rule>
  <description>global navigation rules</description>
  <from-view-id>*</from-view-id>
  <navigation-case>
    <description>Go to Edit Project Page</description>
    <from-outcome>editProject</from-outcome>
    <to-view-id>/project/edit.jsf</to-view-id>
  </navigation-case>
</navigation-rule>
```
Faces Flows builds on the existing JSF navigation story and adds
- Several node types in addition to VDL views
- Well defined start and end-node concept
- Scope concept
- Ability to package a flow as a module and enclose it in a self-configuring jar

Still described in XML
Technical Content

Faces Flows

- Two main varieties of flows
  - Unbounded (No specific start or end, not within a specific flow)
  - Bounded (with explicit start node and end node(s))
Technical Content

What’s in an HTML5 UI?
Technical Content

What’s in an HTML5 UI?

```html
<html>
  <input />
  <button />
  <audio controls />
  <video />
  <img />
</html>
```
Technical Content

What’s in an HTML5 UI?

```css
border: groove 5px #000000;
-moz-border-radius: 34px;
-webkit-border-radius: 34px;
border-radius: 34px;
```
Technical Content

What’s in an HTML5 UI?

navigator.geolocation.getCurrentPosition();
window.JSON;
window.WebSocket;
Technical Content

What’s in an HTML5 UI?

Web UI

Style

Behavior

Markup

Application Model

Significant Server Side Logic
Technical Content

Defining what goes where

Parts of the UI  
Application Design

JSF Specification
Technical Content

The power is in the developer’s hands

- Markup
  - JSF View Declaration Language

- Style and Behavior
  - JSF Resource Handler

- Application Model
  - The strength of Java EE
View Declaration Language

- An HTML-like syntax for declaring a user interface
- The ability to mix pure HTML with JSF components in the same declaration
  - Some of the elements in the declaration have a server-side representation: a UIComponent + Renderer pairing
  - The rest of the elements in the declaration are either for server-side templating or pure HTML
JSF 2.2 HTML5 Friendly Markup
Let the elegance of HTML shine through

```html
<input type="tel" jsf:id="phone"
    jsf:value="#{bean.mobileNumber}" />
<input type="range" jsf:id="values"
    jsf:value="#{bean.value}" />
<button type="button" data-value="MyData"
    jsf:actionListener="#{bean.buttonClicked}"/>
<progress><h:outputText id="progress"
    value="#{bean.progress}" /></progress>
<h:inputText value="#{bean.input}"
    >
    <f:passThroughAttribute name="data-value"
        value="#{bean.dataValue}" /></h:inputText>
```
JSF 2.2 HTML5 Friendly Markup
Let the elegance of HTML shine through

- **Before JSF 2.2**
  - JSF tags hide complexity of underlying HTML+script+css+images
  - JSF “Renderer”:
    - encode: markup to browser
    - decode: name=value from browser

- **JSF 2.2**
  - Pure HTML+script+css+images in the JSF page
  - JSF Renderer handles decode from browser
    - Leverage the strength of the JSF lifecycle
    - Leverage the expressiveness of HTML5
Style and Behavior
Leverage the JSF ResourceHandler

- To JSF, CSS and JavaScript are just another kind of resource delivered to the browser, along with the markup.
- The JSF ResourceHandler lets you create libraries of localized resources and declare dependencies on them from your UI components.
- Want to use CSS3? Sure, go ahead!
Application Model
Where JSF has always shined

- Data Conversion
- Validation
- Database Integration (Persistence Layer)
- Security
- Expression Language
- Request Processing Lifecycle
- Event System
Technical Content
Other Big Ticket Features

- Multi-Templating
  - Multi Templating : Entire Application :: Facelets : Single Page

- Cross Site Request Forgery Protection
  - POST-back is a virtue. Add protection for GET-based requests

- Loading Facelets Via ResourceHandler

- File Upload Component
  - Because we can depend on Servlet 3.0!
Technical Content


- Components/Renderers
  - Component Managed EL Context
  - WindowId

- Facelets
  - HTML5 DOCTYPE is the default
  - Programmatic Composite Component creation
  - FaceletFactory API

- Lifecycle
  - View Actions
Technical Content

- Managed Beans
  - Everything is CDI injectable
  - CDI ViewScoped

- Resources
  - Relative ResourceHandler
Technical Content

- Components/Renderers
- Ajax
- Specification errors and clarifications
- EL
- Facelets
- Lifecycle
- Resources
Transparency Content
JSR-344 is now on JCP 2.8
My Plan For Your Time Investment

- JSR Summary and Scope
- Big Ticket Features
- Medium Sized Features
- Bug Fixes
My Plan For Your Time Investment

- How I run JSRs
  - What works Well, Puzzlers
- How features are suggested
- Transparency Tools
- Roadmap
Transparency Content
How I run JSRs

What works well

– Spec lead drives all aspects of development, pulling expertise from EG for specific targeted questions.
– Minimize time commitment from EG members by being very informal in what they put on the EG list.
  ▪ Simple emails turn into formal feature requests
  ▪ Discussion digressions generate new ideas
– Public Issue Tracker
  ▪ JSF has had one since 2004
  ▪ Still a good idea
Transparency Content

How I run JSRs

- What works well
  - What worked well in iteration N may not work at all in N+1
  - Example, JSR 314, JSF 2.0
    - Ajax sub-group
    - Composite component sub-group
  - JSR-344 JSF 2.2
    - Hard to get traction even on single-thread development effort
Transparency Content
How I run JSRs

- **Puzzlers**
  - How to get more done by delegating to EG members?
    - Who has the time?
    - Does it really allow parallel work?
    - Things fall off a cliff.
  - How to stay on schedule?
    - JSR schedule stated PFD in Q4CY2011
    - Managing the tension between stated JSR goals and what seems important to the EG at any given moment
  - Inconsistently responsive EG
  - Inconsistently available Spec Lead
Transparency Content
How New Features Are Suggested

 Held public vote at beginning of JSR
 Public Issue Tracker
 Email from to EG List
 Need to fit into existing JSR charter
 Needs to fit into existing committed schedule
Transparency Content

Tools

- Public java.net projects
  - spec http://jsf-spec.java.net/
  - impl http://jsf.java.net/ Open Source: GPL+Classpath Exception

- Mailing Lists
  - jsr344-experts@javaserverfaces-spec-public.java.net
    - Public readable archive, JSPA signed member read/write
  - users@javaserverfaces-spec-public.java.net
    - Public readable archive, any java.net member read/write
    - All mail sent to jsr344-experts is sent to users
Transparency Content

Tools

- Issue Tracker
  - spec http://jsf-spec.java.net/issues/
  - impl http://jsf.java.net/issues/
- Wiki: Used in 2.0, not using in 2.2.
- Public Gantt Chart
  - http://jsf-spec.java.net/planning/
- Nightly spec snapshot from Hudson
- Google+ Hangouts
# Transparency Content

Tools [http://jsf-spec.java.net/planning/](http://jsf-spec.java.net/planning/)
Transparency Content

Tools

- Let’s take a look at the API

http://maven.java.net

javax.faces-api
Transparency Content

Tools

- Let’s take a look at the API

2.2-SNAPSHOT

Archive Browser

-javadoc.jar

index.html
Where is JSF in its Lifecycle?

**JSR Timeline**

- **JSR 127**
  - JSF 1.0 11 March 2004
  - JSF 1.1 27 May 2004

- **JSR 252**
  - JSF 1.2 11 May 2006
  - JSF 1.2 Maintenance Release 1 19 December 2006
  - JSF 1.2 Maintenance Release 2 13 June 2008
  - JSF 1.2 Maintenance Release 3 25 August 2008

- **JSR 314**
  - JSF 2.0 1 July 2009
  - JSF 2.1 16 July 2010
  - JSF 2.1 Maintenance Release 2 22 November 2010

- **JSR 344**
  - Started 14 April 2011
  - Early Draft Review released 8 December 2011
  - Working toward Public Review Draft
Where is JSF in its Lifecycle?

Achievements

- JSF 2.0 was the last BIG FEATURE release for JSF
- From now on, JSF will be maintained using smaller, more frequent releases.
  - JSF 2.1, released Q4 2010, JCP Minor Revision
  - JSF 2.2, slated for release in 2012, full JSR
    - Requires JavaEE 6, will release before JavaEE 7
  - JSF 2.3, will target JavaEE 7
Where is JSF in its Lifecycle?

Roadmap

- JSF 2.2
  - Public Review Draft by December 2012
  - No need for Renewal Ballot
    - Early Draft Review published on 8 December 2011!
Where is JSF in its Lifecycle?

Roadmap

- Java EE 7 Roadmap
  - [https://blogs.oracle.com/theaquarium/entry/java_ee_7_roadmap](https://blogs.oracle.com/theaquarium/entry/java_ee_7_roadmap)
  - JavaEE 7 complete in *Spring 2013*

- JSF 2.2
  - Releasing on Java EE 7 schedule
  - Will still run on Java EE 6 containers
    - Any features that happen to depend on EE 7 will gracefully degrade.
    - Example: JSR-353 JSON parsing
Questions & Answers
The preceding was intended to outline our general product direction. It is intended for information purposes only, and may not be incorporated into any contract. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions. The development, release, and timing of any features or functionality described for Oracle’s products remains at the sole discretion of Oracle.