**Introduction to Facelets**

### Advantages of Facelets

Facelets advantages include the following:

* Support for code reuse through templating and composite components
* Functional extensibility of components and other server-side objects through customization
* Faster compilation time
* Compile time EL validation
* High performance rendering

### What's Facelets ?

Facelets is a powerful but lightweight page declaration language that is used to build JavaServer Faces views using HTML style templates and to build component trees.

Facelets features include the following:

* Use of XHTML for creating web pages
* Support for Facelets Tag libraries in addition to JavaServer Faces and JSTL tag libraries
* Support for unified expression language
* Templating for components and pages

#### Web Pages

Facelets views are usually created as XHTML pages. JavaServer Faces implementations support XHTML pages created in conformance with the XHTML Transitional DTD, as listed at <http://www.w3.org/TR/xhtml1/#a_dtd_XHTML-1.0-Transitional>.

#### Tag Library Support

JavaServer Faces technology supports different tag libraries to add components to a web page. To support the JavaServer Faces tag library mechanism, Facelets uses XML namespace declarations.

Tag Library

URI

Prefix

Example

Contents

JavaServer Faces Facelets Tag Library

http://java.sun.com/jsf/facelets

ui:

ui:component

ui:insert

Tags for templating

JavaServer Faces HTML Tag Library

http://java.sun.com/jsf/html

h:

h:head

h:body

h:outputText

h:inputText

JavaServer Faces component tags for all UIComponents.

JavaServer Faces Core Tag Library

http://java.sun.com/jsf/core

f:

f:actionListener

f:attribute

Tags for JavaServer Faces custom actions that are independent of any particular RenderKit.

JSTL Core Tag Library

http://java.sun.com/jsp/jstl/core

c:

c:forEach

c:catch

JSTL 1.1 Core Tags

JSTL Functions Tag Library

http://java.sun.com/jsp/jstl/functions

fn:

fn:toUpperCase

fn:toLowerCase

JSTL 1.1 Functions Tags

#### Unified Expression Language Support

Based on the JavaServer Faces support for unified expression language (EL) syntax defined by JSP 2.1, Facelets uses EL expressions to reference properties and methods of backing beans. EL expressions can be used to bind component objects or values to managed-bean methods or managed-bean properties.

### Templating

### JavaServer Faces 2.0 provides the tools to implement user interfaces that are easy to extend and reuse. Templating is a useful feature available with Facelets that allows you to create a page that will act as the base or template for the other pages in a application. By using templates, you can reuse code and avoid recreating similarly constructed pages. Templating also helps in maintaining a standard look and feel in an application with a large number of pages.

### The following table lists Facelets tags that are used for templating and their respective functionality:

Tag

Function

ui:component

Defines a component that is created and added to the component tree.

ui:composition

Defines a page composition that optionally uses a template. Content outside of this tag is ignored.

ui:debug

Defines a debug component that is created and added to the component tree.

ui:define

Defines content that is inserted into a page by a template

ui:decorate

Similar to composition tag but does not disregard content outside this tag.

ui:fragment

Similar to component tag but does not disregard content outside this tag.

ui:include

Encapsulate and reuse content for multiple pages.

ui:insert

Inserts content into a template.

ui:param

Used to pass parameters to an included file.

ui:repeat

Used as an alternative for loop tags such as c:forEach or h:dataTable.

ui:remove

Removes content from a page.

### The Facelets tag library includes the main templating tag <ui:insert>. Atemplate page is created with this tag, it allows defining a default structure for a page. A template page can be reused as a template for other pages, usually referred to as a client pages.

### A client page allows content to be inserted with the help of the <ui:define> tag.

### Composite Components

The JavaServer Faces offers the concept of composite components with Facelets. A composite component can be considered a a special type of template that acts as a component.

Any component essentially is a piece of reusable code that is capable of a certain functionality

### A composite component is a component that consists of a collection of markups and other existing components. It is a reusable, user-created component that is capable of a customized, defined functionality and can have validators, converters and listeners attached to it like a any other JavaServer Faces component. With Facelets, any XHTML page that is inserted with markups and other components, can be converted into a composite component.

###### Composite Component Tags

Tag

Function

composite:interface

Declares the usage contract for a composite component. The composite component can be used as a single component whose feature set is the union of the features declared in the usage contract.

composite:implementation

Defines the implementation of the composite component. If a <composite:interface> element appears, there must be a corresponding <composite:implementation>.

composite:attribute

Declares an attribute that may be given to an instance of the composite component, in which this tag is declared.

composite:insertChildren

Any child components or template text within the composite component tag in the using page will be re-parented into the composite component at the point indicated by this tag's placement within the composite:implementation section.

composite:valueHolder

Declares that the composite component whose contract is declared by the composite:interface in which this element is nested exposes an implementation of ValueHolder suitable for use as the target of attached objects in the using page.

composite:editableValueHolder

Declares that the composite component whose contract is declared by the composite:interface in which this element is nested exposes an implementation of EditableValueHolder suitable for use as the target of attached objects in the using page.

composite:actionSource

Declares that the composite component whose contract is declared by the composite:interface in which this element is nested exposes an implementation of ActionSource2 suitable for use as the target of attached objects in the using page.

### Resources

Resources refers to any software artifacts that the application requires for proper rendering. They include images, script files and any user-created component libraries. As of JavaServer Faces 2.0, resources must be collected in a standard location, which can be one of the following:

* A resource packaged in the web application root must be in a subdirectory of a resources directory at the web application root: resources/<resource-identifier>.
* A resource packaged in the web application's classpath must be in a subdirectory of the META-INF/resources directory within a web application: META-INF/resources/<resource-identifier>.

The JavaServer Faces runtime will look for the resources in the above listed locations, in that order.

Resource identifiers are unique strings that conform to the following format:

[localePrefix/][libraryName/][libraryVersion/]resource name[/resourceVersion]