

CONTENTS INCLUDE:

- JSF Overview
- Development Process
- Lifecycle
- The JSF Expression Language
- JSF Core Tags
- JSF HTML Tags and more...

JavaServer Faces 2.0

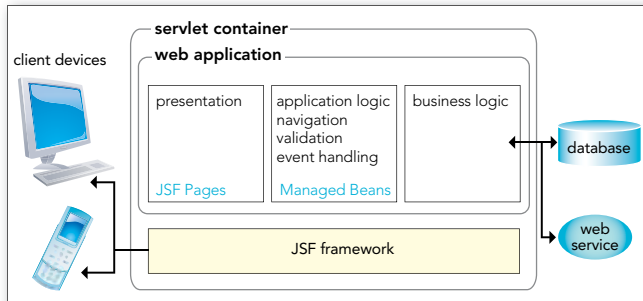
By Cay Horstmann

JSF OVERVIEW

JavaServer Faces (JSF) is the “official” component-based view technology in the Java EE web tier. JSF includes a set of predefined UI components, an event-driven programming model, and the ability to add third-party components. JSF is designed to be extensible, easy to use, and toolable. This refcard describes JSF 2.0.

DEVELOPMENT PROCESS

A developer specifies JSF components in JSF pages, combining JSF component tags with HTML and CSS for styling. Components are linked with managed beans—Java classes that contain presentation logic and connect to business logic and persistence backends.



In JSF 2.0, it is recommended that you use the facelets format for your pages:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml"
xmlns:f="http://java.sun.com/jsf/core"
xmlns:h="http://java.sun.com/jsf/html"
xmlns:ui="http://java.sun.com/jsf/facelets">
<h:head>...</h:head>
<h:body>
<h:form>
...
</h:form>
</h:body>
</html>
```

These common tasks give you a crash course into using JSF.

Text Field
page.xhtml

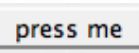
12345678901234567890

```
<h:inputText value="#{bean1.luckyNumber}">
```

WEB-INF/classes/com/corejsf/SampleBean.java

```
@ManagedBean(name="bean1")
@SessionScoped
public class SampleBean {
    public int getLuckyNumber() { ... }
    public void setLuckyNumber(int value) { ... }
    ...
}
```

Button



page.xhtml

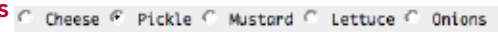
```
<h:commandButton value="press me" action="#{bean1.login}"/>
```

WEB-INF/classes/com/corejsf/SampleBean.java

```
@ManagedBean(name="bean1")
@SessionScoped
public class SampleBean {
    public String login() {
        if (...) return "success"; else return "error";
    }
    ...
}
```

The outcomes success and error can be mapped to pages in faces-config.xml. If no mapping is specified, the page /success.xhtml or /error.xhtml is displayed.

Radio Buttons



page.xhtml

```
<h:selectOneRadio value="#{form.condiment}>
<f:selectItems value="#{form.items}"/>
</h:selectOneRadio>
```

WEB-INF/classes/com/corejsf/SampleBean.java

```
public class SampleBean {
    private static Map<String, Object> items;
    static {
        items = new LinkedHashMap<String, Object>();
        items.put("Cheese", 1); // label, value
        items.put("Pickle", 2);
        ...
    }
    public Map<String, Object> getItems() { return items; }
    public int getCondiment() { ... }
    public void setCondiment(int value) { ... }
    ...
}
```



JBoss Developer Studio 2.0

- Build web apps using rich JSF components
- Choose from 120+ skinnable RichFaces components
- Includes Seam, Hibernate, Drools, JBoss Portal and much more

For more on JBoss and JSF 2.0, goto
jboss.org/jbossrichfaces/

Validation and Conversion

Page-level validation and conversion:

```
<h:inputText value="#{bean1.amount}" required="true">
  <f:validateDoubleRange maximum="1000"/>
</h:inputText>

<h:outputText value="#{bean1.amount}">
  <f:convertNumber type="currency"/>
</h:outputText>
```

The number is displayed with currency symbol and group separator: \$1,000.00

Using the Bean Validation Framework (JSR 303) **2.0**

```
public class SampleBean {
  @Max(1000) private BigDecimal amount;
}
```

Error Messages

Amount Amount: 'too much' is not a number. Example: 99

```
<h:outputText value="Amount"/>
<h:inputText id="amount" label="Amount" value="#{payment.amount}"/>
<h:message for="amount"/>
```

Resources and Styles

page.xhtml

```
<h:outputStylesheet library="css" name="styles.css" target="head"/>
...
<h:outputText value="#{msgs.goodbye}!" styleClass="greeting"/>
```

faces-config.xml

```
<application>
  <resource-bundle>
    <base-name>com.corejsf.messages</base-name>
    <var>msgs</var>
  </resource-bundle>
</application>
```

WEB-INF/classes/com/corejsf/messages.properties

```
goodbye=Goodbye
```

WEB-INF/classes/com/corejsf/messages_de.properties

```
goodbye=Auf Wiedersehen
```

resources/css/styles.css

```
.greeting {
  font-style: italic;
  font-size: 1.5em;
  color: #eee;
}
```

Table with links

Name	
Washington, George	Delete
Jefferson, Thomas	Delete
Lincoln, Abraham	Delete
Roosevelt, Theodore	Delete

page.xhtml

```
<h:dataTable value="#{bean1.entries}" var="row" styleClass="table"
  rowClasses="even,odd">
  <h:column>
    <f:facet name="header">
      <h:outputText value="Name"/>
    </f:facet>
    <h:outputText value="#{row.name}"/>
  </h:column>
  <h:column>
    <h:commandLink value="Delete" action="#{bean1.deleteAction}"
      immediate="true">
      <f:setPropertyActionListener target="#{bean1.idToDelete}"
        value="#{row.id}"/>
    </h:commandLink>
  </h:column>
</h:dataTable>
```

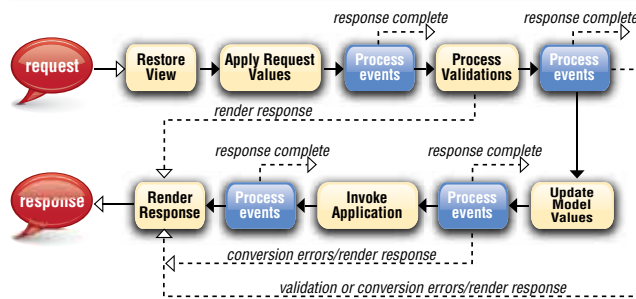
WEB-INF/classes/com/corejsf/SampleBean.java

```
public class SampleBean {
  private int idToDelete;
  public void setIdToDelete(int value) { idToDelete = value; }
  public String deleteAction() {
    // delete the entry whose id is idToDelete
    return null;
  }
  public List<Entry> getEntries() { ... }
  ...
}
```

Ajax 2.0

```
<h:commandButton value="Update">
  <f:ajax execute="input1 input2" render="output1"/>
</h:commandButton>
```

LIFECYCLE



THE JSF EXPRESSION LANGUAGE

An EL expression is a sequence of literal strings and expressions of the form `base[expr1][expr2]...`. As in JavaScript, you can write `base.identifier` instead of `base['identifier']` or `base["identifier"]`. The base is one of the names in the table below or a bean name.

header	A Map of HTTP header parameters, containing only the first value for each name
headerValues	A Map of HTTP header parameters, yielding a String[] array of all values for a given name
param	A Map of HTTP request parameters, containing only the first value for each name
paramValues	A Map of HTTP request parameters, yielding a String[] array of all values for a given name
cookie	A Map of the cookie names and values of the current request
initParam	A Map of the initialization parameters of this web application
requestScope, sessionScope, applicationScope, viewScope 2.0	A map of all attributes in the given scope
facesContext	The FacesContext instance of this request
view	The UIViewRoot instance of this request
component 2.0	The current component
cc 2.0	The current composite component
resource 2.0	Use <code>resource['library:name']</code> to access a resource

Value expression: a reference to a bean property or an entry in a map, list or array. Examples:

`userBean.name` calls `getName` or `setName` on the `userBean` object
`pizza.choices[var]` calls `pizza.getChoices().get(var)` or `pizza.getChoices().put(var, ...)`

Method expression: a reference to a method and the object on which it is to be invoked. Example:

`userBean.login` calls the `login` method on the `userBean` object when it is invoked. **2.0:** Method expressions can contain

parameters: `userBean.login('order page')`

In JSF, EL expressions are enclosed in `#{...}` to indicate deferred evaluation. The expression is stored as a string and evaluated when needed. In contrast, JSP uses immediate evaluation, indicated by `${...}` delimiters.

2.0: EL expressions can contain JSTL functions

fn:contains(str, substr) fn:containsIgnoreCase(str, substr) fn:startsWith(str, substr) fn:endsWith(str, substr) fn:length(str) fn:indexOf(str) fn:join(strArray, separator) fn:split(str, separator) fn:substring(str, start, pastEnd)	fn:substringAfter(str, separator) fn:substringBefore(str, separator) fn:replace(str, substr, replacement) fn:toLowerCase(str) fn:toUpperCase(str) fn:trim() fn:escapeXml()
--	--

JSF CORE TAGS

Tag	Description/Attributes
<code>f:facet</code>	Adds a facet to a component - name: the name of this facet
<code>f:attribute</code>	Adds an attribute to a component - name, value: the name and value of the attribute to set
<code>f:param</code>	Constructs a parameter child component - name: An optional name for this parameter component. - value: The value stored in this component.
<code>f:actionListener</code> <code>f:valueChangeListener</code>	Adds an action listener or value change listener to a component - type: The name of the listener class
<code>f:propertyActionListener 1.2</code>	Adds an action listener to a component that sets a bean property to a given value - target: The bean property to set when the action event occurs - value: The value to set it to
<code>f:phaseListener 1.2</code>	Adds a phase listener to this page - type: The name of the listener class
<code>f:event 2.0</code>	Adds a system event listener to a component - name: One of <code>preRenderComponent</code> , <code>postAddToView</code> , <code>preValidate</code> , <code>postValidate</code> - listener: A method expression of the type <code>void (ComponentSystemEvent) throws AbortProcessingException</code>
<code>f:converter</code>	Adds an arbitrary converter to a component - convertedId: The ID of the converter
<code>f:convertDateTime</code>	Adds a datetime converter to a component - type: <code>date</code> (default), <code>time</code> , or <code>both</code> - dateStyle, timeStyle: <code>default</code> , <code>short</code> , <code>medium</code> , <code>long</code> or <code>full</code> - pattern: Formatting pattern, as defined in <code>java.text.SimpleDateFormat</code> - locale: Locale whose preferences are to be used for parsing and formatting - timeZone: Time zone to use for parsing and formatting (Default: UTC)
<code>f:convertNumber</code>	Adds a number converter to a component - type: <code>number</code> (default), <code>currency</code> , or <code>percent</code> - pattern: Formatting pattern, as defined in <code>java.text.DecimalFormat</code> - minIntegerDigits, maxIntegerDigits, minFractionDigits, maxFractionDigits: Minimum, maximum number of digits in the integer and fractional part - integerOnly: True if only the integer part is parsed (default: false) - groupingUsed: True if grouping separators are used (default: true) - locale: Locale whose preferences are to be used for parsing and formatting - currencyCode: ISO 4217 currency code to use when converting currency values - currencySymbol: Currency symbol to use when converting currency values
<code>f:validator</code>	Adds a validator to a component - validatorID: The ID of the validator
<code>f.validateDoubleRange</code> , <code>f.validateLongRange</code> , <code>f.validateLength</code>	Validates a double or long value, or the length of a string - minimum, maximum: the minimum and maximum of the valid range
<code>f.validateRequired 2.0</code>	Sets the required attribute of the enclosing component
<code>f.validateBean 2.0</code>	Specify validation groups for the Bean Validation Framework (JSR 303)
<code>f:loadBundle</code>	Loads a resource bundle, stores properties as a Map - basename: the resource bundle name - value: The name of the variable that is bound to the bundle map

<code>f:selectItem</code>	Specifies an item for a select one or select many component - binding, id: Basic attributes - itemDescription: Description used by tools only - itemDisabled: false (default) to show the value - itemLabel: Text shown by the item - itemValue: Item's value, which is passed to the server as a request parameter - value: Value expression that points to a <code>SelectItem</code> instance - escape: true (default) if special characters should be converted to HTML entities - noSelectionOption 2.0: true if this item is the "no selection option"
<code>f:selectItems</code>	Specifies items for a select one or select many component - value: Value expression that points to a <code>SelectItem</code> , an array or <code>Collection</code> , or a <code>Map</code> mapping labels to values. - var 2.0: Variable name used in value expressions when traversing an array or collection of non- <code>SelectItem</code> elements - itemLabel 2.0, itemValue 2.0, itemDescription 2.0, itemDisabled 2.0, itemLabelEscaped 2.0: Item label, value, description, disabled and escaped flags for each item in an array or collection of non- <code>SelectItem</code> elements. Use the variable name defined in <code>var</code> . - noSelectionOption 2.0: Value expression that yields the "no selection option" item or string that equals the value of the "no selection option" item
<code>f:ajax 2.0</code>	Enables Ajax behavior - execute, render: Lists of component IDs for processing in the "execute" and "render" lifecycle phases - event: JavaScript event that triggers behavior. Default: <code>click</code> for buttons and links, <code>change</code> for input components - immediate: If true, generated events are broadcast during "Apply Request Values" phase instead of "Invoke Application" - listener: Method binding of type <code>void (AjaxBehaviorEvent)</code> - onevent, onerror: JavaScript handlers for events/errors
<code>f:viewParam 2.0</code>	Defines a "view parameter" that can be initialized with a request parameter - name, value: the name of the parameter to set - binding, converter, id, required, value, validator, valueChangeListener: basic attributes
<code>f:metadata 2.0</code>	Holds view parameters. May hold other metadata in the future

JSF HTML TAGS

Tag	Description	
<code>h:head 2.0</code> , <code>h:body 2.0</code> , <code>h:form</code>	HTML head, body, form	
<code>h:outputStylesheet 2.0</code> , <code>h:outputScript 2.0</code>	Produces a style sheet or script	
<code>h:inputText</code>	Single-line text input control.	
<code>h:inputTextArea</code>	Multiline text input control.	
<code>h:inputSecret</code>	Password input control.	
<code>h:inputHidden</code>	Hidden field	
<code>h:outputLabel</code>	Label for another component for accessibility	
<code>h:outputLink</code>	HTML anchor.	
<code>h:outputFormat</code>	Like <code>outputText</code> , but formats compound messages	
<code>h:outputText</code>	Single-line text output.	
<code>h:commandButton</code> , <code>h:button 2.0</code>	Button: submit, reset, or pushbutton.	
<code>h:commandLink</code> , <code>h:link 2.0</code>	Link that acts like a pushbutton.	
<code>h:message</code>	Displays the most recent message for a component	
<code>h:messages</code>	Displays all messages	
<code>h:graphicImage</code>	Displays an image	
<code>h:selectOneListbox</code>	Single-select listbox	

h:selectOneMenu	Single-select menu	
h:selectOneRadio	Set of radio buttons	<input type="radio"/> Cheese <input checked="" type="radio"/> Pickle
h:selectBooleanCheckbox	Checkbox	Receive email: <input type="checkbox"/>
h:selectManyCheckbox	Set of checkboxes	<input checked="" type="checkbox"/> Cheese <input type="checkbox"/> Pickle <input type="checkbox"/> Mustard
h:selectManyListbox	Multiselect listbox	
h:selectManyMenu	Multiselect menu	
h:panelGrid	HTML table	
h:panelGroup	Two or more components that are laid out as one	
h:dataTable	A feature-rich table component	
h:column	Column in a data table	

Basic Attributes

id	Identifier for a component
binding	Reference to the component that can be used in a backing bean
rendered	A boolean; false suppresses rendering
value	A component's value, typically a value binding
valueChangeListener	A method expression of the type void (ValueChangeEvent)
converter, validator	Converter or validator class name
required	A boolean; if true, requires a value to be entered in the associated field

Attributes for h:body and h:form

Attribute	Description
binding, id, rendered	Basic attributes
dir, lang, style, styleClass, target, title h:form only: accept, acceptcharset, enctype	HTML 4.0 attributes (acceptcharset corresponds to HTML accept-charset, styleClass corresponds to HTML class)
onclick, ondblclick, onkeydown, onkeypress, onkeyup, onmousedown, onmousemove, onmouseover, onmouseout h:body only: onload, onunload h:form only: onblur, onchange, onfocus, onreset, onsubmit	DHTML events

Attributes for h:inputText, h:inputSecret, h:inputTextarea, and h:inputHidden

Attribute	Description
cols	For h:inputTextarea only—number of columns
immediate	Process validation early in the life cycle
redisplay	For h:inputSecret only—when true, the input field's value is redisplayed when the web page is reloaded
required	Require input in the component when the form is submitted
rows	For h:inputTextarea only—number of rows
binding, converter, id, rendered, required, value, validator, valueChangeListener	Basic attributes
accesskey, alt, dir, disabled, lang, maxLength, readonly, size, style, styleClass, tabIndex, title	HTML 4.0 pass-through attributes—alt, maxLength, and size do not apply to h:inputTextarea. None apply to h:inputHidden
onblur, onchange, onclick, ondblclick, onfocus, onkeydown, onkeypress, onkeyup, onmousedown, onmousemove, onmouseover, onmouseout, onselect	DHTML events. None apply to h:inputHidden

Attributes for h:outputText and h:outputFormat

Attribute	Description
escape	If set to true, escapes <, >, and & characters. Default value is true
binding, converter, id, rendered, styleClass, value	Basic attributes
style, title	HTML 4.0

Attributes for h:outputLabel

Attribute	Description
for	The ID of the component to be labeled.
binding, converter, id, rendered, value	Basic attributes

Attributes for h:graphicImage

Attribute	Description
library 2.0, name 2.0	The resource library (subdirectory of resources) and file name (in that subdirectory)
binding, id, rendered, value	Basic attributes
alt, dir, height, ismap, lang, longdesc, style, styleClass, title, url, usemap, width	HTML 4.0
onblur, onchange, onclick, ondblclick, onfocus, onkeydown, onkeypress, onkeyup, onmousedown, onmousemove, onmouseout, onmouseover, onmouseup	DHTML events

Attributes for h:commandButton and h:commandLink

Attribute	Description
action (command tags)	Navigation outcome string or method expression of type String ()
outcome 2.0 (non-command tags)	Value expression yielding the navigation outcome
fragment 2.0 (non-command tags)	Fragment to be appended to URL. Don't include the # separator
actionListener	Method expression of type void (ActionEvent)
charset	For h:commandLink only—The character encoding of the linked reference
image (button tags)	For h:commandButton only—A context-relative path to an image displayed in a button. If you specify this attribute, the HTML input's type will be image
immediate	A boolean. If false (the default), actions and action listeners are invoked at the end of the request life cycle; if true, actions and action listeners are invoked at the beginning of the life cycle
type	For h:commandButton: The type of the generated input element: button, submit, or reset. The default, unless you specify the image attribute, is submit. For h:commandLink and h:link: The content type of the linked resource; for example, text/html, image/gif, or audio/basic
value	The label displayed by the button or link
binding, id, rendered	Basic attributes
accesskey, dir, disabled (h:commandButton only), lang, readonly, style, styleClass, tabIndex, title	HTML 4.0
link tags only: charset, coords, hreflang, rel, rev, shape, target	
onblur, onclick, ondblclick, onfocus, onkeydown, onkeypress, onkeyup, onmousedown, onmousemove, onmouseout, onmouseover, onmouseup	DHTML events

Attributes for h:outputLink

Attribute	Description
accesskey, binding, converter, id, lang, rendered, value	Basic attributes

charset, coords, dir, hreflang, lang, rel, rev, shape, style, styleClass, tabindex, target, title, type	HTML 4.0
onblur, onchange, onclick, ondblclick, onfocus, onkeydown, onkeypress, onkeyup, onmousedown, onmousemove, onmouseout, onmouseover, onmouseup	DHTML events

Attributes for: h:selectBooleanCheckbox, h:selectManyCheckbox, h:selectOneRadio, h:selectOneListbox, h:selectManyListbox, h:selectOneMenu, h:selectManyMenu

Cheese Pickle Mustard Lettuce Onions

Attribute	Description
enabledClass, disabledClass	CSS class for enabled/disabled elements—h:selectOneRadio and h:selectManyCheckbox only
selectedClass 2.0, unselectedClass 2.0	CSS class for selected/unselected elements—h:selectManyCheckbox only
layout	Specification for how elements are laid out: <code>lineDirection</code> (horizontal) or <code>pageDirection</code> (vertical)—h:selectOneRadio and h:selectManyCheckbox only
collectionType 2.0	selectMany tags only: the name of a collection class such as <code>java.util.TreeSet</code>
hideNoSelectionOption 2.0	Hide item marked as “no selection option”
binding, converter, id, immediate, required, rendered, validator, value, valueChangeListener	Basic attributes
accesskey, border, dir, disabled, lang, readonly, style, styleClass, size, tabindex, title	HTML 4.0—border only for h:selectOneRadio and h:selectManyCheckbox, size only for h:selectOneListbox and h:selectManyListbox.
onblur, onchange, onclick, ondblclick, onfocus, onkeydown, onkeypress, onkeyup, onmousedown, onmousemove, onmouseout, onmouseover, onmouseup, onselect	DHTML events

Attributes for h:message and h:messages

Amount Amount: 'too much' is not a number. Example: 99

Attribute	Description
for	The ID of the component whose message is displayed—applicable only to h:message
errorClass, fatalClass, infoClass, warnClass	CSS class applied to error/fatal/information/warning messages
errorStyle, fatalStyle, infoStyle, warnStyle	CSS style applied to error/fatal/information/warning messages
globalOnly	Instruction to display only global messages—h:messages only. Default: false
layout	Specification for message layout: table or list—h:messages only
showDetail	A boolean that determines whether message details are shown. Defaults are false for h:messages, true for h:message.
showSummary	A boolean that determines whether message summaries are shown. Defaults are true for h:messages, false for h:message.
tooltip	A boolean that determines whether message details are rendered in a tooltip; the tooltip is only rendered if showDetail and showSummary are true
binding, id, rendered	Basic attributes
style, styleClass, title	HTML 4.0

Attributes for h:panelGrid

Attribute	Description
bgcolor	Background color for the table

border	Width of the table's border
cellpadding	Padding around table cells
cellspacing	Spacing between table cells
columns	Number of columns in the table
frame	frame Specification for sides of the frame surrounding the table that are to be drawn; valid values: none, above, below, hsid, vsides, lrs, rrs, box, border
headerClass, footerClass	CSS class for the table header/footer
rowClasses, columnClasses	Comma-separated list of CSS classes for rows/columns
rules	Specification for lines drawn between cells; valid values: groups, rows, columns, all
summary	Summary of the table's purpose and structure used for non-visual feedback such as speech
binding, id, rendered, value	Basic attributes
dir, lang, style, styleClass, title, width	HTML 4.0
onclick, ondblclick, onkeydown, onkeypress, onkeyup, onmousedown, onmousemove, onmouseout, onmouseover, onmouseup	DHTML events

Attributes for h:panelGroup

Attribute	Description
binding, id, rendered	Basic attributes
style, styleClass	HTML 4.0

Attributes for h:dataTable

Attribute	Description
bgcolor	Background color for the table
border	Width of the table's border
cellpadding	Padding around table cells
cellspacing	Spacing between table cells
first	index of the first row shown in the table
frame	Specification for sides of the frame surrounding the table should be drawn; valid values: none, above, below, hsid, vsides, lrs, rrs, box, border
headerClass, footerClass	CSS class for the table header/footer
rowClasses, columnClasses	comma-separated list of CSS classes for rows/columns
rules	Specification for lines drawn between cells; valid values: groups, rows, columns, all
summary	summary of the table's purpose and structure used for non-visual feedback such as speech
var	The name of the variable created by the data table that represents the current item in the value
binding, id, rendered, value	Basic attributes
dir, lang, style, styleClass, title, width	HTML 4.0
onclick, ondblclick, onkeydown, onkeypress, onkeyup, onmousedown, onmousemove, onmouseout, onmouseover, onmouseup	DHTML events

Attributes for h:column

Attribute	Description
headerClass 1.2, footerClass 1.2	CSS class for the column's header/footer
binding, id, rendered	Basic attributes

Facelets Tags 2.0

Attribute	Description
ui:define	Give a name to content for use in a template -name: the name of the content

ui:insert	If a name is given, insert named content if defined or use the child elements otherwise. If no name is given, insert the content of the tag invoking the template -name: the name of the content	ui:param	Define a parameter to be used in an included file or template -name: parameter name -value: a value expression (can yield an arbitrary object)
ui:composition	Produces content from a template after processing child elements (typically ui:define tags) Everything outside the ui:composition tag is ignored -template: the template file, relative to the current page	ui:repeat	Repeats the enclosed elements -value: a List, array, ResultSet, or object -offset, step, size: starting index, step size, ending index of the iteration -var: variable name to access the current element -varStatus: variable name to access the iteration status, with integer properties begin, end, index, step and Boolean properties even, odd, first, last
ui:component	Like ui:composition, but makes a JSF component -binding, id: basic attributes	ui:debug	Shows debug info when CTRL+SHIFT+a key is pressed -hotkey: the key to press (default d) -rendered: true (default) to activate
ui:decorate, ui:fragment	Like ui:composition, ui:component, but does not ignore the content outside the tag	ui:remove	Do not include the contents (useful for comments or temporarily deactivating a part of a page)
ui:include	Include plain XHTML, or a file with a ui:composition or ui:component tag -src: the file to include, relative to the current page		

ABOUT THE AUTHOR



Cay S. Horstmann has written many books on C++, Java and object-oriented development, is the series editor for Core Books at Prentice-Hall and a frequent speaker at computer industry conferences. For four years, Cay was VP and CTO of an Internet startup that went from 3 people in a tiny office to a public company. He is now a computer science professor at San Jose State University. He was elected Java Champion in 2005.

Cay Horstmann's Java Blog- <http://weblogs.java.net/blog/cayhorstmann/>

Cay Horstmann's Website- <http://www.horstmann.com/>

RECOMMENDED BOOK

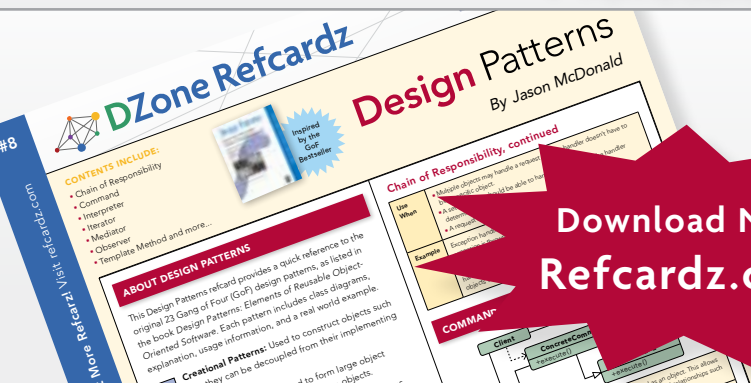


Core JavaServer Faces delves into all facets of JSF development, offering systematic best practices for building robust applications and maximizing developer productivity.

BUY NOW

books.dzone.com/books/jsf

Professional Cheat Sheets You Can Trust



Download Now
Refcardz.com

"Exactly what busy developers need: simple, short, and to the point."

James Ward, Adobe Systems

Upcoming Titles

- Drupal
- Grails
- Core Java Concurrency
- Java Performance Tuning
- Eclipse RCP
- HTML
- Wicket

Most Popular

- Spring Configuration
- jQuery Selectors
- Windows Powershell
- Dependency Injection with EJB 3
- Netbeans IDE JavaEditor
- Getting Started with Eclipse
- Very First Steps in Flex



DZone communities deliver over 6 million pages each month to more than 3.3 million software developers, architects and decision makers. DZone offers something for everyone, including news, tutorials, cheatsheets, blogs, feature articles, source code and more. **"DZone is a developer's dream,"** says PC Magazine.

DZone, Inc.
1251 NW Maynard
Cary, NC 27513
888.678.0399
919.678.0300

Refcardz Feedback Welcome
refcardz@dzone.com

Sponsorship Opportunities
sales@dzone.com

ISBN-13: 978-1-934238-61-5
ISBN-10: 1-934238-61-9

50795

\$7.95