

## Introduction to JavaServer Faces (3 Day)

JavaServer Faces, or simply JSF, combines Java Servlets and JavaServer Pages into a server-side implementation of the Model-View-Controller Design Pattern. The JSF framework provides developers with a unified infrastructure upon which Internet applications can be constructed.

The JavaServer Faces framework is establishing itself as the new standard for the development of web applications. Designed under Sun's Java Community Process by many of the same people that developed the Jakarta Struts framework, Faces is proving itself to be the next step in the web application evolutionary process. This course introduces the developer to the JSF architecture and provides the basis for planning, developing, and deploying Web based applications using the JSF framework. After taking this class, the developer will be able to quickly construct dynamic server-side web pages using JSF. They will also be able to integrate the Web application with many of the other Java2 Enterprise Edition application server methodologies such as Enterprise Java Beans, JavaMail, and SOAP. This class combines lecture with a unifying, hands-on experience, and open discussion that will help the developer quickly understand the benefits of JSF and how to use the framework.

### **Audience:**

This course is intended for Systems Programmers, Application Developers, and Multi-tiered client/server developers.

### **Prerequisites:**

Participants should already have a solid understanding of Java programming and understand the basics of XML. The course also assumes a basic understanding of HTML syntax and JavaServer Pages syntax. Understanding of Enterprise Java Beans (EJB) is also a plus.

### **Benefits:**

Upon completing the course, the student will:

- Write web applications that take advantage of the FacesServlet, FacesContext and Action Java classes to control the user experience of the web application.
- Write JSF applications that gather and update information from external application servers such as EJBs, CORBA servers, and database servers.
- Create and use custom Tag Libraries in JavaServer Pages.
- Understand the basics of web security and learn to take advantage of the security features provided by the Web Server.
- Understand the use of the standard JSF Validators and how to write custom Validators.
- Understand the use of the standard JSF Data Conversion classes and how to write custom Data Converters.
- Take advantage of the JSF architecture that supports rendering output in several formats from the same application. Such as: HTML, WML, XML, etc.

## Topics Covered

- Introduction and Overview
- The JSF Architecture
- JSF Request Objects
- Simple JSF User Interface components
- The EL Expression Language and Advanced User Interface components
- Event Handling
- Data Validation
- Data Conversion
- Rendering Custom User Interface Objects

## **Event Handling**

- The Java 2 Event Model
- The JSF Event Model
- The ActionEvent Class
- The ValueChangeEvent Class
- Event Listeners in JSF
- The ActionListener
- The ValueChangeListener

## **Data Validation**

- Validation
- Using Standard Validators
- Handling Error Output
- <f:validateLength>
- <f:validateLongRange> & <f:validateDoubleRange>
- Custom Validators
- Simple Validators
- Custom Validators
- Creating the Validator Class
- Creating the Tag Handler
- Registering the Validator Class
- Using the Validator Class

## **Advanced Data Validation**

- The StateHolder Interface
- Validating Dependent fields
- Creating a multi-field validator framework

## **Data Conversion, Rendering, and Custom Controls**

- Data Conversion vs. Rendering
- Converters
- Custom Converters
- Writing the Converter Class
- Renderers

- Renderers in the “Real World”
- Creating the Renderer Class
- Creating the Tag Handler
- The TLD for the Tag Handler
- Registering the Renderer
- Using the Component

### **Creating a Graphical Custom Control**

- Writing a graphic control
- Creating a custom Java Servlet to generate graphics formats
- Registering the custom Servlet