





## Faces Config Structure

This tab is a structured representation of the current `faces-config.xml`. Use it to manage tags in the `faces-config.xml` file more clearly. You do not need to remember the required tags order and the structure of the config file. If you want to add an element to the `faces-config.xml`, you can create it using this tab, and IntelliJ IDEA will automatically register corresponding tags in the correct order within your config file.

In this section:

- [Toolbar](#)
- [Common Faces Settings](#)

### Toolbar

Item	Description
	<p>Use this button to create new items in the <code>faces-config.xml</code>, like components, managed beans, validators, etc. Select the desired node in the tree-view and click this button to add an item.</p> <p>You cannot add another <b>Faces Config</b> here using this button, because this view represents the structure of the current config. If you want to add another config, use <b>Project View</b> or <a href="#">Web</a> tool window.</p>
	<p>Use this button to remove items from the <code>faces-config.xml</code>. Select the desired item in the tree-view and click this button to remove it.</p> <p>You can remove the <b>Faces Config</b> as well using this button.</p>
	Click to expand the tree-view.
	Click to collapse the tree-view.

### Common faces settings

#### Application

The *application* element provides a mechanism to define the various per-application-singleton implementation classes for a particular web application that is utilizing JavaServer Faces.

Item	Description
View Handler	Stands for <code>&lt;view-handler&gt;</code> . Contains the fully qualified class name of the concrete <code>ViewHandler</code> implementation class that will be called during the Restore View and Render Response phases of the request processing lifecycle.
Property Resolver	Stands for <code>&lt;property-resolver&gt;</code> . Contains the fully qualified class name of the concrete <code>PropertyResolver</code> implementation class that will be used during the processing of value reference expressions.
Action Listener	Stands for <code>&lt;action-listener&gt;</code> . Contains the fully qualified class name of the concrete <code>ActionListener</code> implementation class that will be called during the Invoke Application phase of the request processing lifecycle.
State	Stands for <code>&lt;state-manager&gt;</code> .

Manager	
Default Locale	Stands for <default-locale>. Declares the default locale for this application instance.
Navigation Handler	Stands for <navigation-handler>. Contains the fully qualified class name of the concrete NavigationHandler implementation class that will be called during the Invoke Application phase of the request processing lifecycle.
Variable Resolver	Stands for <variable-resolver>. Contains the fully qualified class name of the concrete VariableResolver implementation class that will be used during the processing of value reference expressions.
Message Bundle	Stands for <message-bundle>. The base name of a resource bundle representing the message resources for this application.
Default Render Kit	Select the default Render Kit from the list.
Supported Locales	Use add/remove buttons to configure <supported-locale> elements. The <supported-locale> allows authors to declare which locales are supported in this application instance.

## Factory

The *factory* element provides a mechanism to define the various Factories that comprise parts of the implementation of JavaServer Faces.

Item	Description
Render Kit Factory	Stands for <render-kit-factory>. Contains the fully qualified class name of the concrete RenderKitFactory implementation class that will be called when <code>FactoryFinder.getFactory(RENDER_KIT_FACTORY)</code> is called.
Context Factory	Stands for <faces-context-factory>. Contains the fully qualified class name of the concrete FacesContextFactory implementation class that will be called when <code>FactoryFinder.getFactory(FACES_CONTEXT_FACTORY)</code> is called.
Lifecycle Factory	Stands for <lifecycle-factory>. Contains the fully qualified class name of the concrete LifecycleFactory implementation class that will be called when <code>FactoryFinder.getFactory(LIFECYCLE_FACTORY)</code> is called.
Application Factory	Stands for <application-factory>. Contains the fully qualified class name of the concrete ApplicationFactory implementation class that will be called when <code>FactoryFinder.getFactory(APPLICATION_FACTORY)</code> is called.

## See Also

### Concepts:

- [JSF](#)

### Procedures:

- [Enabling JSF Support](#)
- [Creating and Editing Faces Configuration](#)

### Web Resources:

- [Developer Community](#)

