

ICEfaces™ Eclipse Integration V3.5.2	
---------------------------------------------	--

Overview

ICEfaces Eclipse Integration provides ICEfaces and ICEfaces facelets support for developers who use **Eclipse IDE for Java EE Developers (Ganymede, v3.4.2)** for ICEfaces application development. This integration extends Eclipse's Web Tools Platform (WTP) and provides following extra features for ICEfaces support.

- Adds ICEfaces, ICEfaces Facelets, and dependent libraries to your project.
- Configures web.xml and faces-config.xml for ICEfaces support.
- Provides ICEfaces/Facelets friendly JSP/JSPX/xhtml page templates.
- Integrates ICEfaces components into Visual JSF Page Designer.

Requirements

1. Eclipse IDE for Java EE Developers v3.4.2

This bundle includes required Eclipse WTP (Web Tool Platform). It can be downloaded from <http://www.eclipse.org/downloads/>

2. ICEfaces-1.8.0 -Eclipse-JavaEE-3.4.2-plugins.v3.5.2.zip

This is core package of ICEfaces Eclipse Integration. It can be downloaded from <http://www.icefaces.org/main/downloads/os-downloads.iface>. It contains ICEfaces runtime library v1.8.0.

Installation

Remove previous version of ICEfaces Eclipse Integrations if they have already been installed. This can be done by selecting **Help -> Software Updates... -> Installed Software**. Highlight all previous version of ICEfaces integration plugins, and right click on it and select **Uninstall**. This requires you to restart Eclipse.

Remember to remove **JSF Facelets Tools Incubator** as well.

Installation

Download and extract **ICEfaces-1.8.0 -Eclipse-JavaEE-3.4.2-plugins.v3.5.2.zip** to a local directory, for example, C:\icefaces.tools.update. After they are extracted, 2 sub-directories features and plugins are created as shown below.

-> features

```
---> com.icesoft.eclipse.icefaces.v1.8.0.feature_1.0.1.200904020030
---> com.icesoft.eclipse.sunri.feature_1.2.2.200904021710
---> com.icesoft.ide.eclipse.icefaces.core.feature_3.5.2.200904081303
---> com.icesoft.ide.eclipse.icefaces.facelets.feature_3.5.2.200904081303
---> org.eclipse.jst.jsf.facelet.feature_0.1.0.200904081303
```

-> plugins

```
---> com.icesoft.eclipse.icefaces.v180_1.0.1.200904020030
---> com.icesoft.eclipse.sunri_1.2.1.200904020030
---> com.icesoft.ide.eclipse.designtime_1.8.2.200904081303
---> com.icesoft.ide.eclipse.icefaces.core_3.5.1.200904020030
---> com.icesoft.ide.eclipse.icefaces.doc_1.8.0.200904020030.jar
---> com.icesoft.ide.eclipse.icefaces.facelets.core_3.5.2.200904081303.jar
---> org.eclipse.jst.jsf.facelet.core_0.1.0.200904081303.jar
---> org.eclipse.jst.jsf.facelet.doc.user_0.1.0.200904081303.jar
---> org.eclipse.jst.jsf.facelet.tagsupport_0.1.0.200904081303
---> org.eclipse.jst.jsf.facelet.ui_0.1.0.200904081303.jar
```

-> ICEfaces.pdf

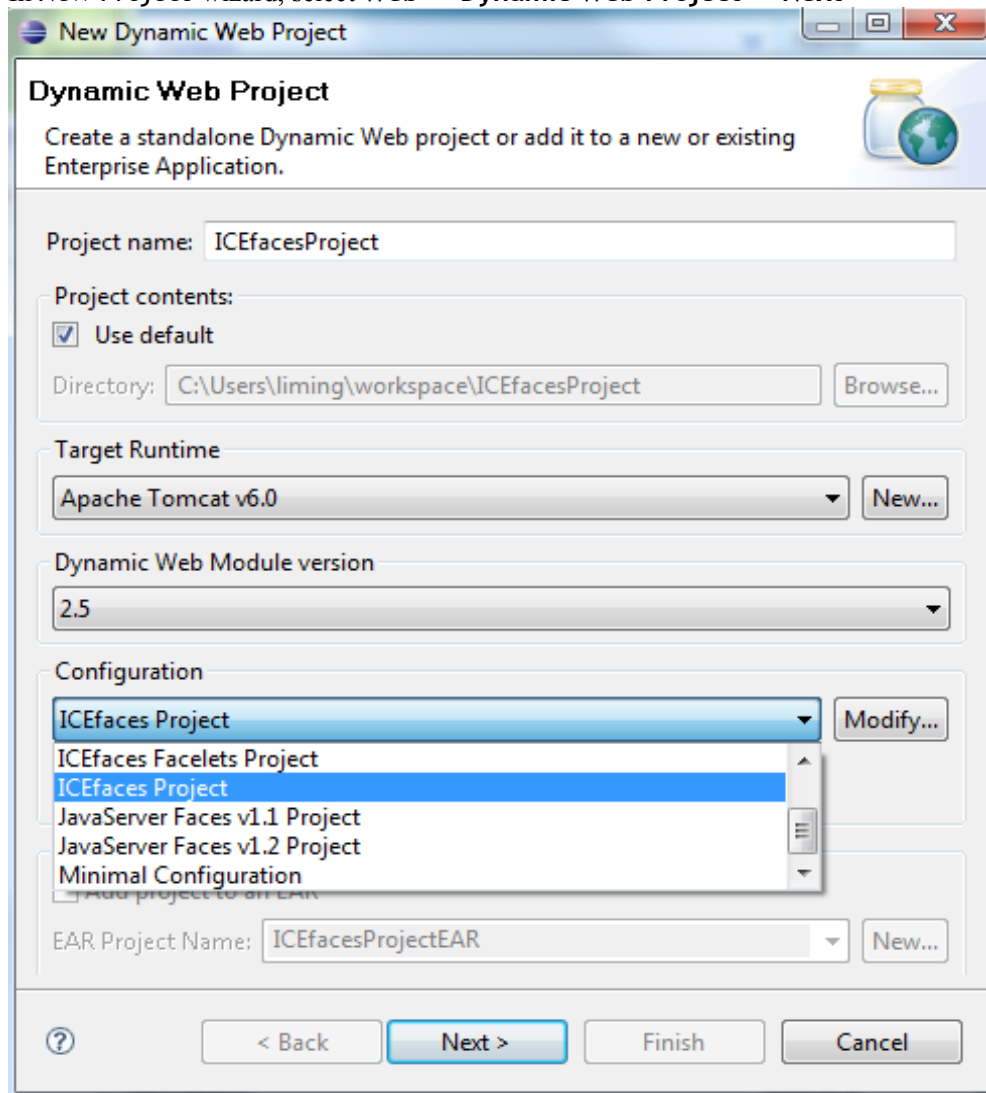
-> ReleaseNotes.html

3. Start Eclipse. From Eclipse main menu bar, select **Help -> Software Updates... -> Available Software...**
-> Add Site...->Local
4. select C:\icefaces.tools.update directory, where the zip files have been extracted in above step. And click **Ok**.
5. Select the previously defined local update site as shown below, and click **Install...**
6. Click Ok once installation finish. This step requires restart Eclipse.

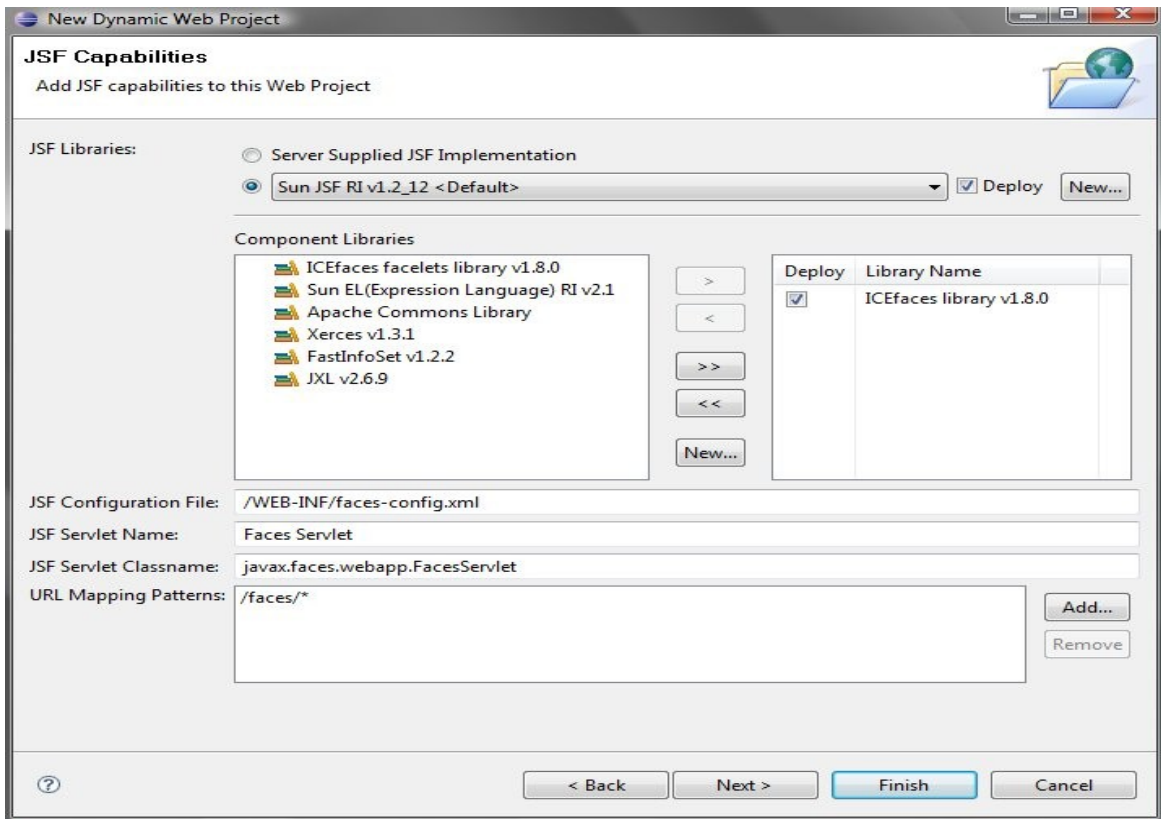
Create new project with ICEfaces capabilities

The following instructions describe how to use how to create a new project with ICEfaces capabilities.

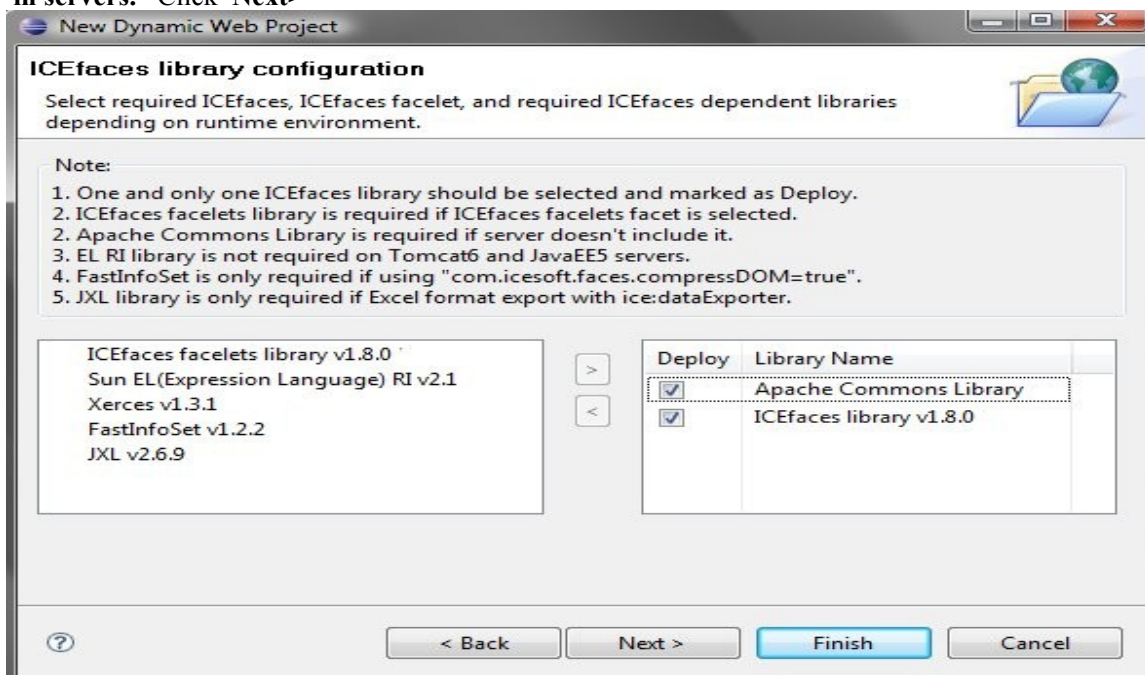
1. From Eclipse main menu bar, select **File -> New -> Project...**,
2. In *New Project* wizard, select **Web -> Dynamic Web Project -> Next>**



3. Type in project name, select **Target Runtime** and **Configurations**. And click **Next>**
4. In Project Facets wizard dialog, select either ICEfaces Project or ICEfaces Facelets Project.



5. In JSF Capabilities wizard dialog, pre-built Sun JSF RI v1.1_02, Sun JSF RI v1.2_09, Sun EL RI v2.1, and Xerces v1.3.1 libraries, etc are provided. In this step, It is required to choose JSF implementation library. For component libraries, you can add ICEfaces, ICEfaces facelet libraries, and other required libraries. You can also create new libraries to meet your requirement. Component libraries can also be added in the following ICEfaces library wizard. **Note that implementation library Sun JSF RI may need marked as Deployed in servers.** Click **Next>**



6. Following is ICEfaces library configuration wizard. In this wizard you can required pre-defined ICEfaces dependent libraries by clicking > button. Be sure to check the Notes section. Click **Next>**

7. Following is the ICEfaces project configuration wizard, which allow you to enable sample page, change default settings, etc. Note that do not add extension to your sample page name as it is determined by `javax.faces.DEFAULT_SUFFIX`

ICEfaces configurations
ICEfaces project configurations

☒ Create a ICEfaces sample page from template
ICEfaces sample page file name (Extension is not required)
☒ Copy default CSS resources into WebContent.(Required by design-time only)

web.xml

Context Parameter

`javax.faces.DEFAULT_SUFFIX`
`com.icesoft.faces.concurrentDOMViews` ☐
`com.icesoft.faces.synchronousUpdate` ☐
`com.icesoft.faces.uploadDirectory`
`com.icesoft.faces.uploadMaxFileSize`

Servlet

☒ Persistent Faces Servlet `com.icesoft.faces.webapp.xmlhttp.PersistentFacesServlet`
☒ Blocking Servlet `com.icesoft.faces.webapp.xmlhttp.BlockingServlet`
☒ uploadServlet `com.icesoft.faces.component.inputfile.FileUploadServlet`

Listener

☒ `com.icesoft.faces.util.event.servlet.ContextEventRepeater`
☐ `com.sun.faces.config.ConfigureListener`

8. Now you can click **Finish** button if ICEface facet is not selected. Otherwise if Next> is enabled, click Next> button to continue configure Facelets/ICEfaces facet.

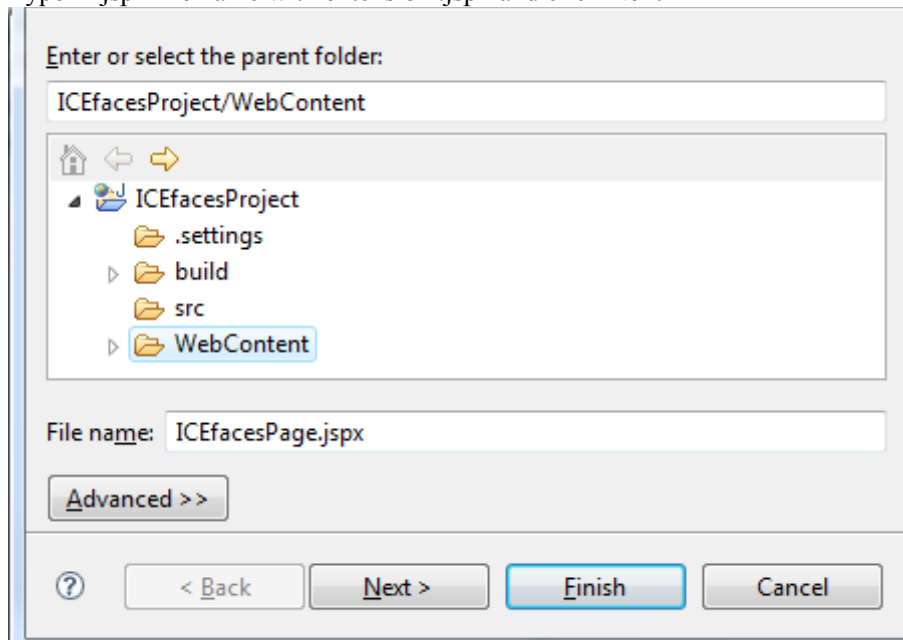
9. **Due to the long-run newly created project validation process, newly created project may disappear for a short while. Manually refresh workspace may be required. Due to the same reason as above, newly created startup sample page ICEfacesPage1.jspx (configurable) may not appear right after project is created in some cases. A manual refresh may be required.** To manually refresh project, right click on selected project and select Refresh from context menu items. This step will ensure new created resources are recognized. These issues will be resolved in next release.

Create ICEfaces friendly page with provided template

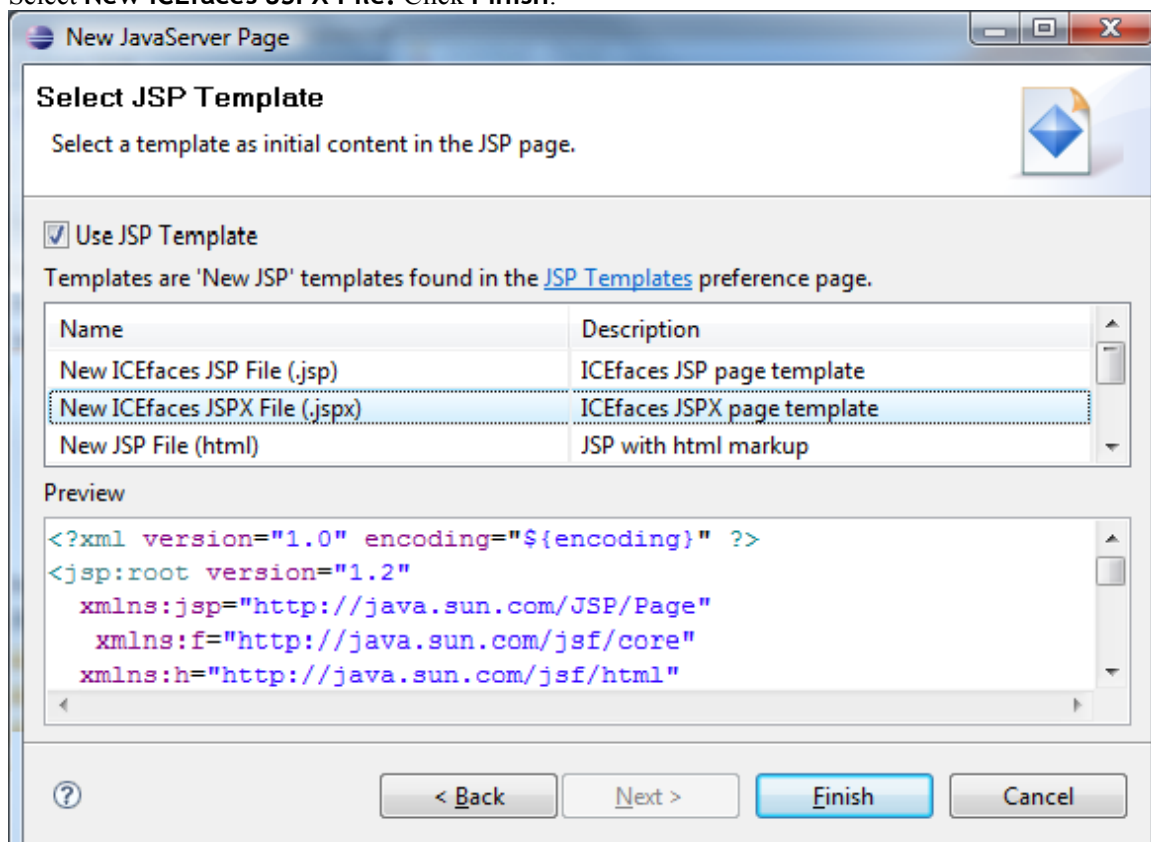
ICEfaces friendly JSP page templates are also provided in this integration to make your ICEfaces application development easier. The following instructions describe how to use the provided template.

1. Right click on WebContent folder in *Package Explorer* view and select following menu items **New -> JSP**. (For ICEfaces facet project, select menu New -> html to create xhtml file)

2. Type in jsp file name with extension .jspx and click Next>



3. Select **New ICEfaces JSPX File**. Click **Finish**.



4. Current Eclipse release version v3.4 doesn't set JSP-JSP page editor (also called Visual JSF Page Designer, Web Page Editor) as editor forjspx page. Forcing JSP-JSP page editor to loadjspx file is required . See next section "Use ICEfaces components in Visual JSF Page Designer".

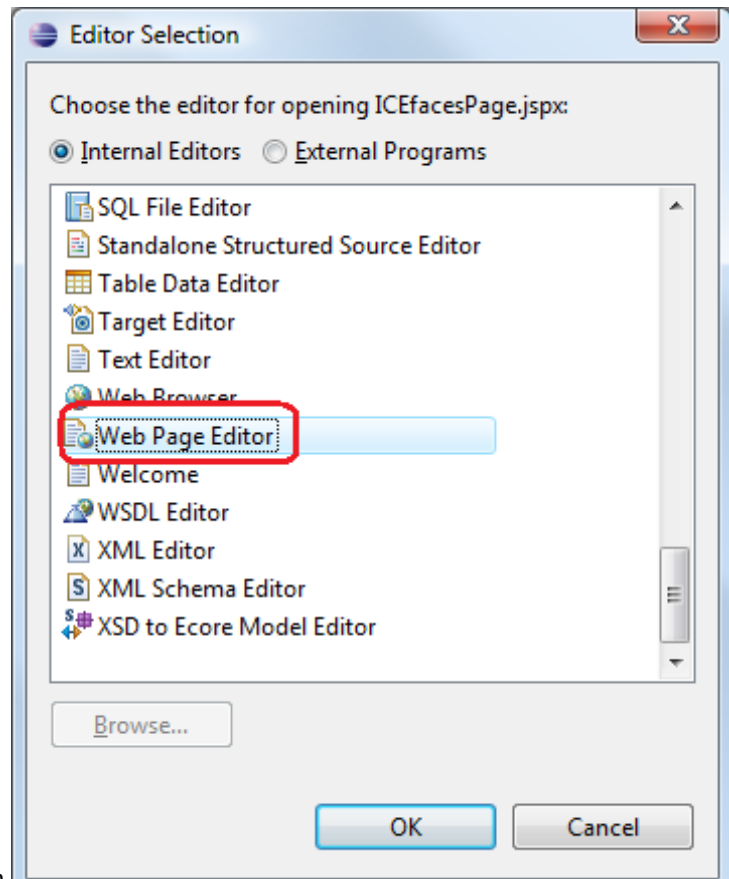
Use ICEfaces components in Visual JSF Page Designer

New Eclipse Visual JSF Page Designer has been added since Eclipse 3.3 as part of WTP 2.0. It provides an intuitive close-to-WYSIWIG editing experience. It allows visually JSF page layout and components drag and drop. See

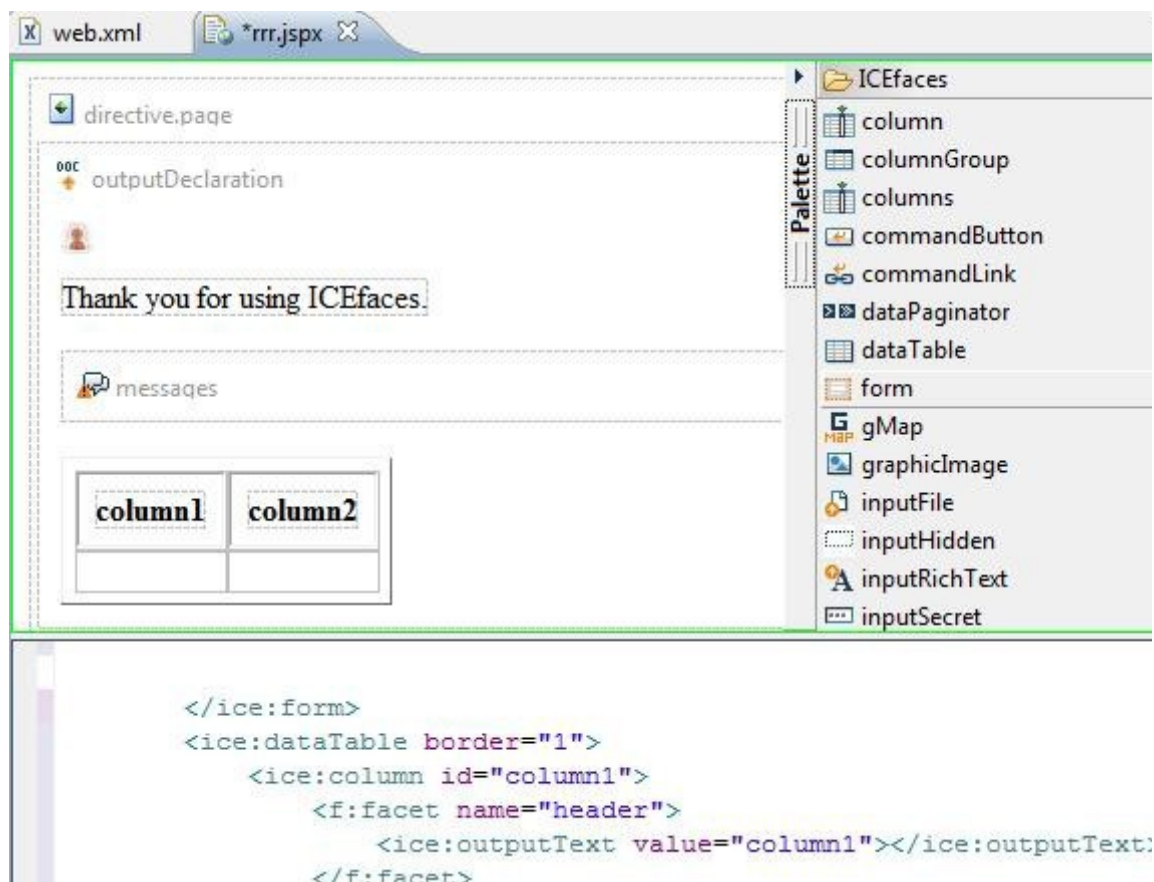
JavaServer Faces Tooling Developer Guide for detail. This integration includes ICEfaces components design time support for Eclipse Visual JSF Page Designer. ICEfaces components palette view, property view for ICEfaces components' attributes are both supported.

1. Use JSF-JSP editor for ICEfaces page. Current Eclipse release version v3.4 doesn't set JSP-JSP page editor as editor for jsp page. Following the following instructions to open ICEfaces page in JSF-JSP editor.

1. Right click on the ICEfaces jsp page
2. Select **Open with -> Other...**



3. Select Web Page Editor, and click OK button
2. Select ICEfaces components in Palette view



3 Edit ICEfaces component attributes in Property editor

