



# Web Frameworks and Tooling

What is it?

How do I use it?

And why should I want to?

**Geertjan Wielenga**

**<http://blogs.sun.com/geertjan>**

**Sun Microsystems**

**Source Talk Tage, 27 September, 2006**



# Agenda

- Introduction
- 1. Why?
- 2. What?
- 3. How?
- 4. When?
- Conclusion

# Once upon a time...

- JSP
- Servlet

## But now...

Click Echo JDO Facelets Struts  
Cocoon JSF iBatis  
Beehive Beehive  
iBatis SiteMesh Maverick  
EJB Keel Tapestry  
Hibernate Velocity Aranea  
WebWork Tiles Spring  
Struts WebObjects Wicket  
iBatis

Facelets

JSF

Struts

These are currently  
supported  
in NetBeans IDE...

Wicket

**Plus... many more in-house  
frameworks...**

# How the IDE can help

- NetBeans module a.k.a. “plugin”
- manifest.mf
- layer.xml
- Java classes with functionality
- “Matisse” GUI Builder for design
- Easy to install in NetBeans IDE
- Easy to distribute as binary file to others

# NetBeans module can provide...

- Creating a source structure
- Syntax highlighting
- Code completion
- Refactoring
- Hyperlinking
- Component Palette
- Multiview Editors
- Error annotations



## ...and also these:

- File templates
- Indentation engines
- Internationalization
- Options
- Project samples
- JavaHelp helpsets
- Junit testing

# Web Framework Tooling...

Creating  
a source  
structure

Syntax  
highlighting

Code  
completion

Options

File  
templates

Refactoring

Component  
Palette

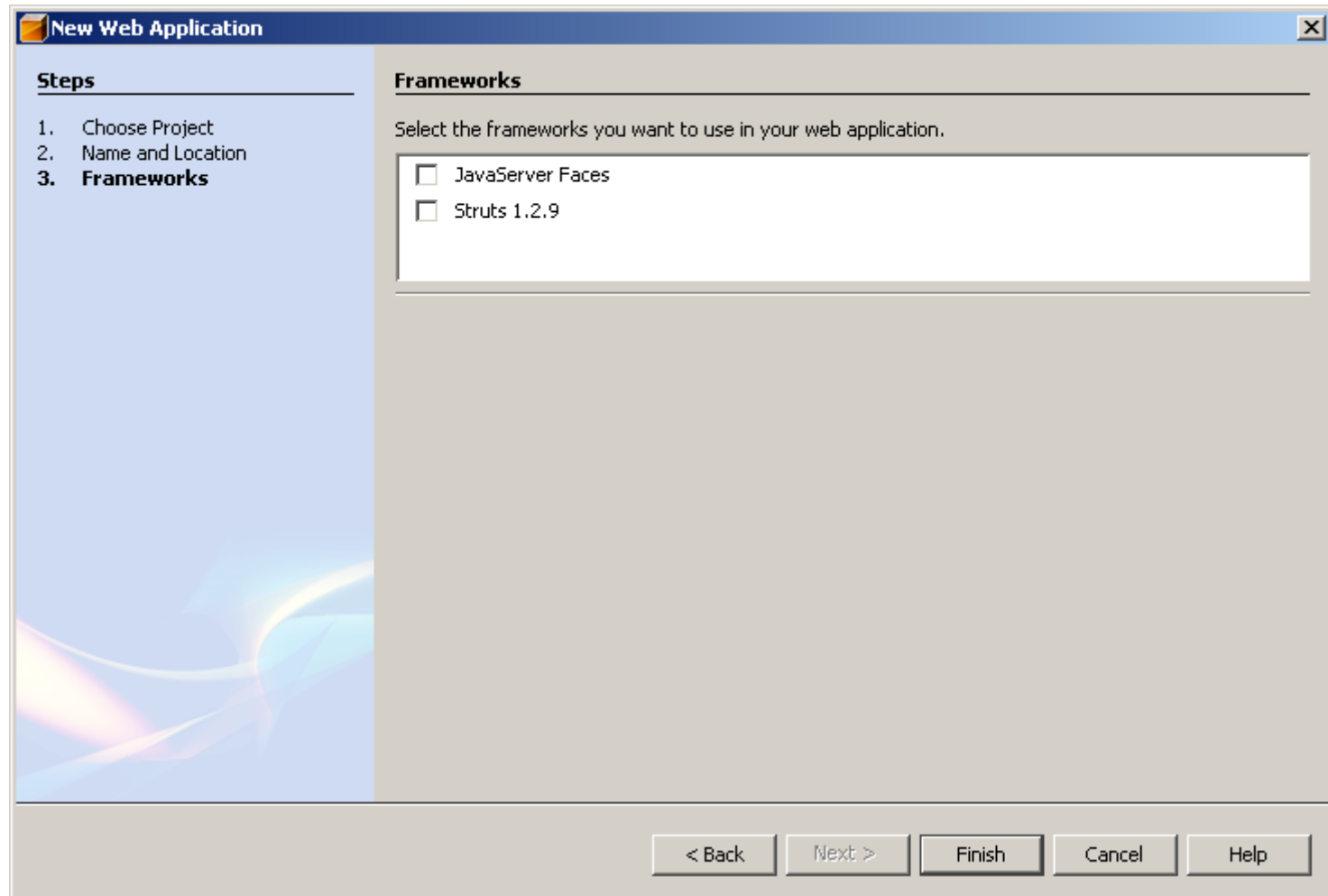
Indentation  
engines

Hyperl  
inking

Multiview  
Editors

JavaHelp  
helpsets

# Where everything begins...



# Configuration Panel...

**New Web Application**

**Steps**

1. Choose Project
2. Name and Location
3. **Frameworks**

**Frameworks**

Select the frameworks you want to use in your web application.

- JavaServer Faces
- Struts 1.2.9

JavaServer Faces Configuration

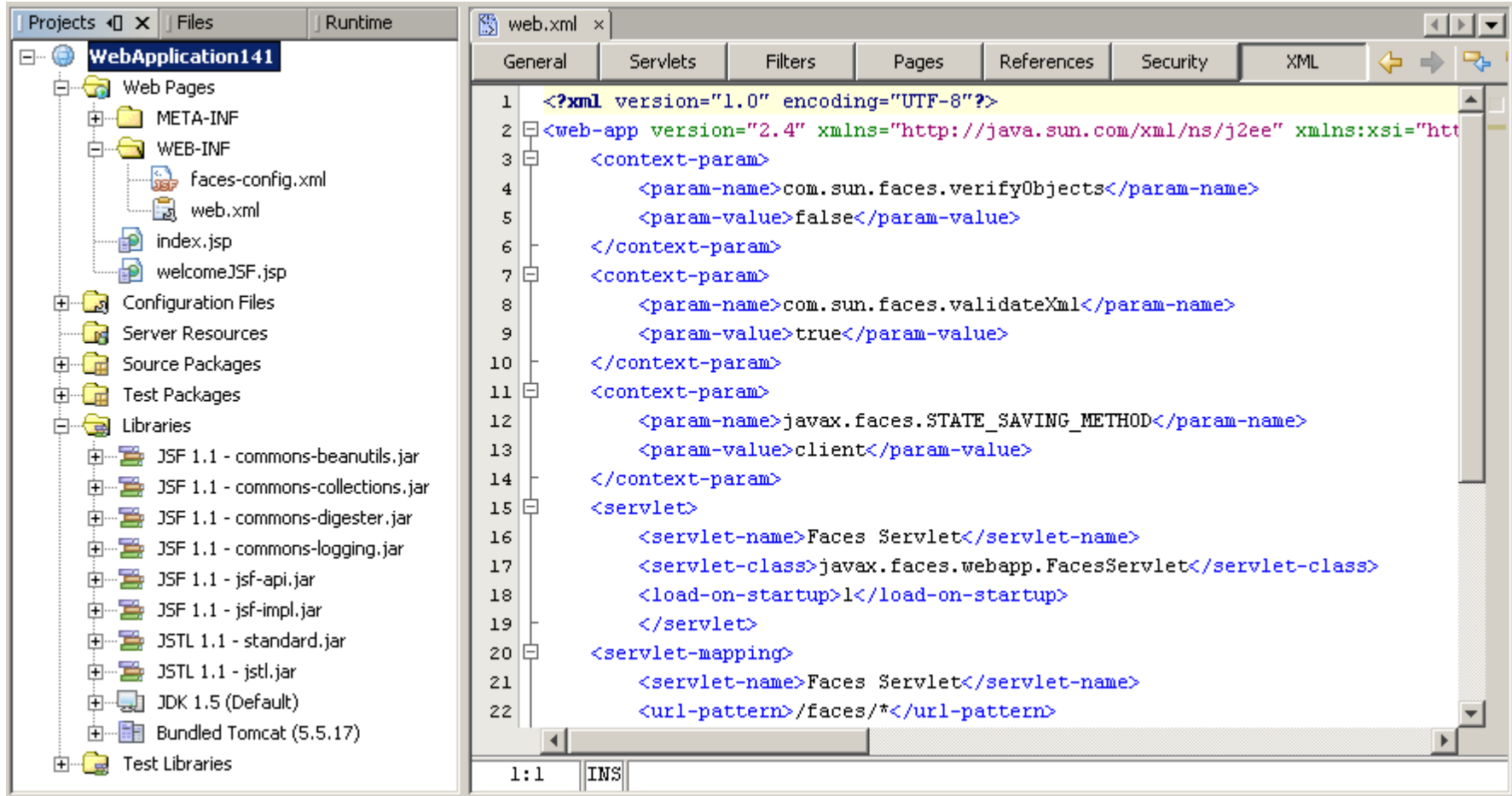
JSF Servlet Name:

Servlet URL Mapping:

Validate XML     Verify Objects

< Back    Next >    Finish    Cancel    Help

# When you click “Finish”...

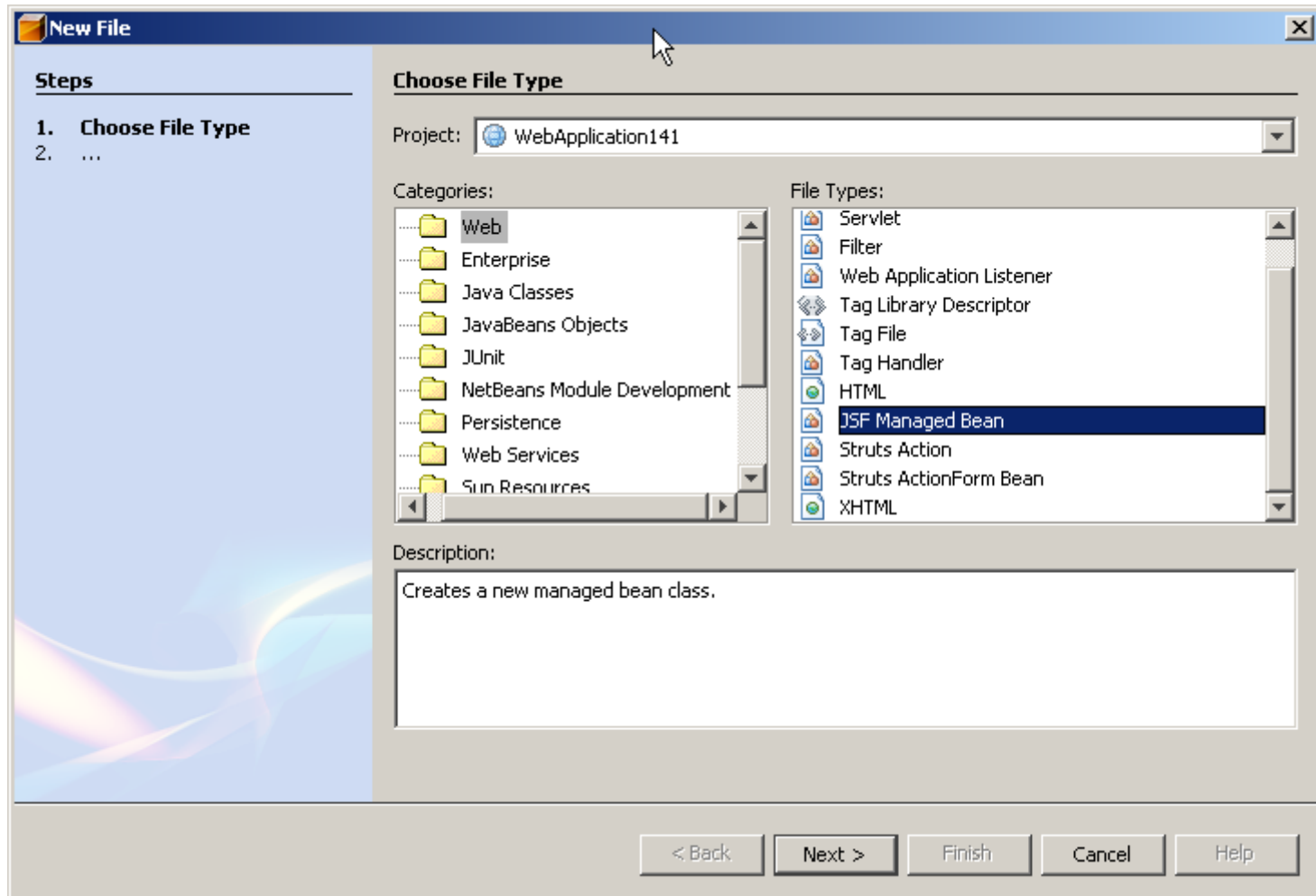


The screenshot shows an IDE window with a project named "WebApplication141". The left pane shows the project structure, including "Web Pages", "Configuration Files", "Server Resources", "Source Packages", "Test Packages", "Libraries", and "Test Libraries". The right pane shows the content of the "web.xml" file, which is an XML configuration file for a web application. The XML content is as follows:

```

1 <?xml version="1.0" encoding="UTF-8"?>
2 <web-app version="2.4" xmlns="http://java.sun.com/xml/ns/j2ee" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://java.sun.com/xml/ns/j2ee http://java.sun.com/xml/ns/j2ee/web-app_2_4.xsd">
3   <context-param>
4     <param-name>com.sun.faces.verifyObjects</param-name>
5     <param-value>>false</param-value>
6   </context-param>
7   <context-param>
8     <param-name>com.sun.faces.validateXml</param-name>
9     <param-value>>true</param-value>
10  </context-param>
11  <context-param>
12    <param-name>javax.faces.STATE_SAVING_METHOD</param-name>
13    <param-value>client</param-value>
14  </context-param>
15  <servlet>
16    <servlet-name>Faces Servlet</servlet-name>
17    <servlet-class>javax.faces.webapp.FacesServlet</servlet-class>
18    <load-on-startup>1</load-on-startup>
19  </servlet>
20  <servlet-mapping>
21    <servlet-name>Faces Servlet</servlet-name>
22    <url-pattern>/faces/*</url-pattern>
  
```

# Creating new artifacts...



**New JSF Managed Bean**

**Steps**

1. Choose File Type
2. **Name and Location**

**Name and Location**

Class Name:

Project:

Location:

Package:

Created File:

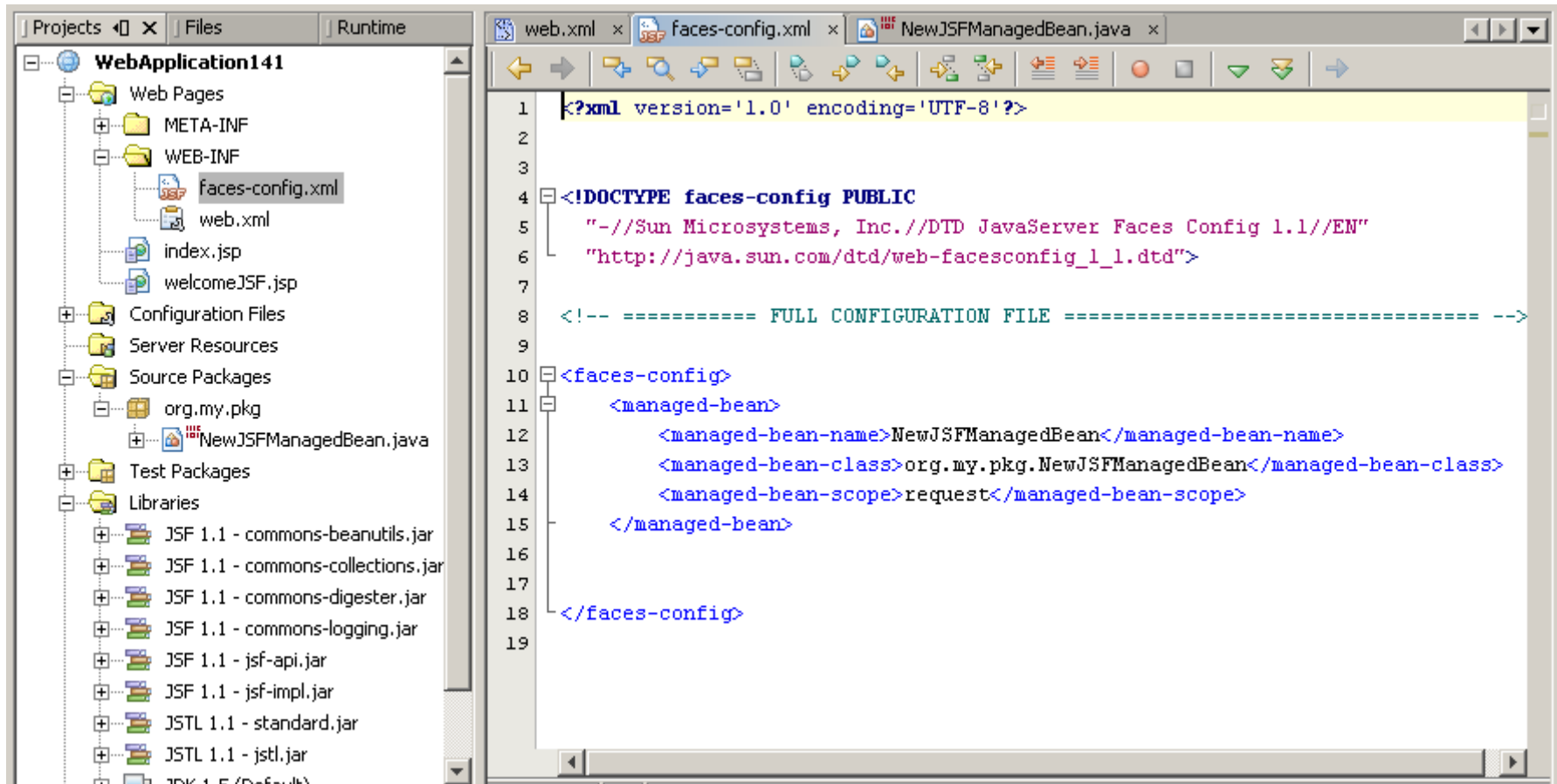
Configuration File:

Scope:

Bean Description:

< Back   Next >   Finish   Cancel   Help

# After you click “Finish” ...



The screenshot shows an IDE window with two panes. The left pane displays the project structure for 'WebApplication141', including folders like 'Web Pages', 'Configuration Files', and 'Source Packages', and various JAR files in the 'Libraries' section. The right pane shows the 'faces-config.xml' file with the following XML content:

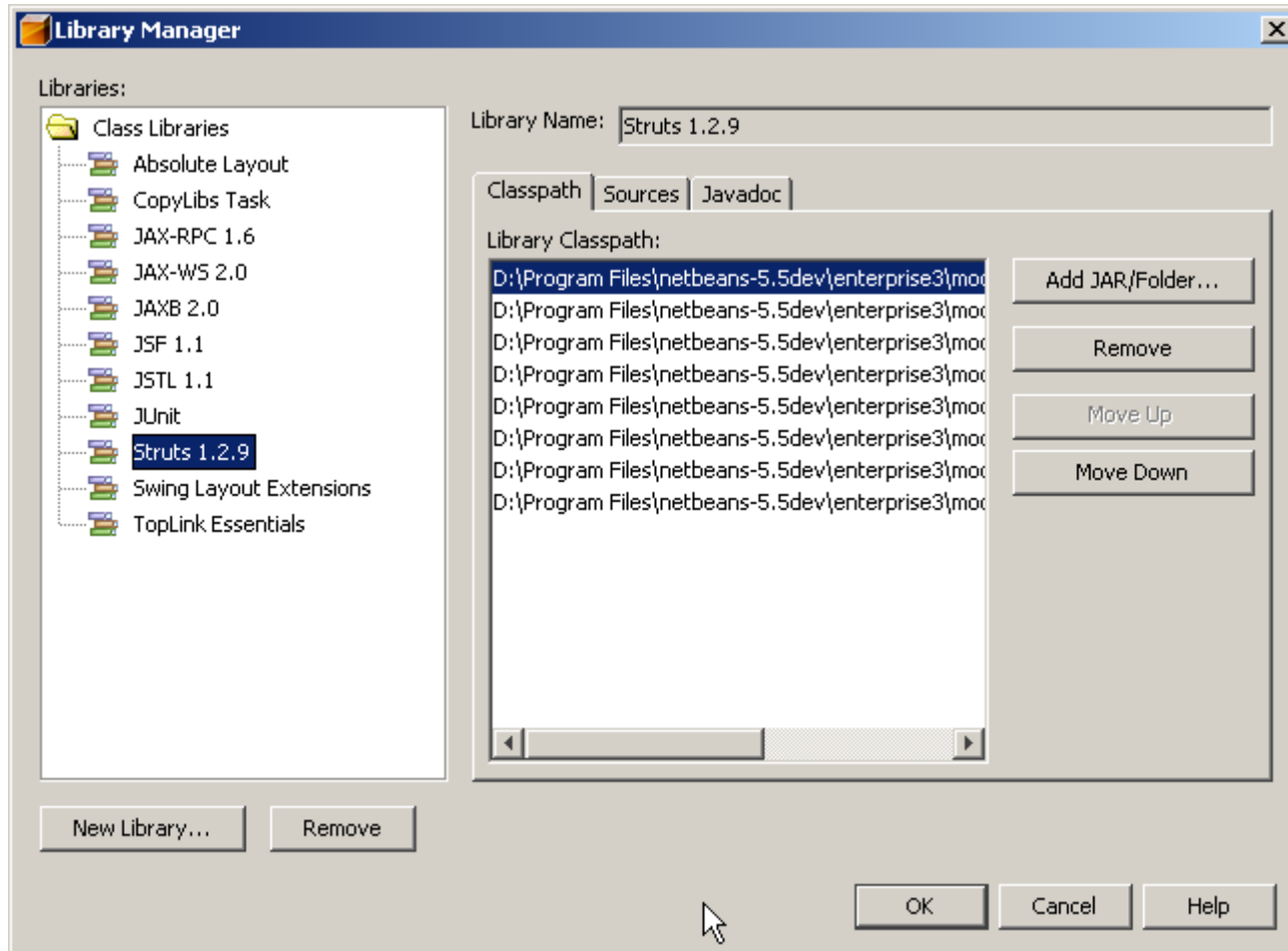
```

1 <?xml version='1.0' encoding='UTF-8'?>
2
3
4 <!DOCTYPE faces-config PUBLIC
5     "-//Sun Microsystems, Inc.//DTD JavaServer Faces Config 1.1//EN"
6     "http://java.sun.com/dtd/web-facesconfig_1_1.dtd">
7
8 <!-- ===== FULL CONFIGURATION FILE ===== -->
9
10 <faces-config>
11     <managed-bean>
12         <managed-bean-name>NewJSFManagedBean</managed-bean-name>
13         <managed-bean-class>org.my.pkg.NewJSFManagedBean</managed-bean-class>
14         <managed-bean-scope>request</managed-bean-scope>
15     </managed-bean>
16
17 </faces-config>
18
19

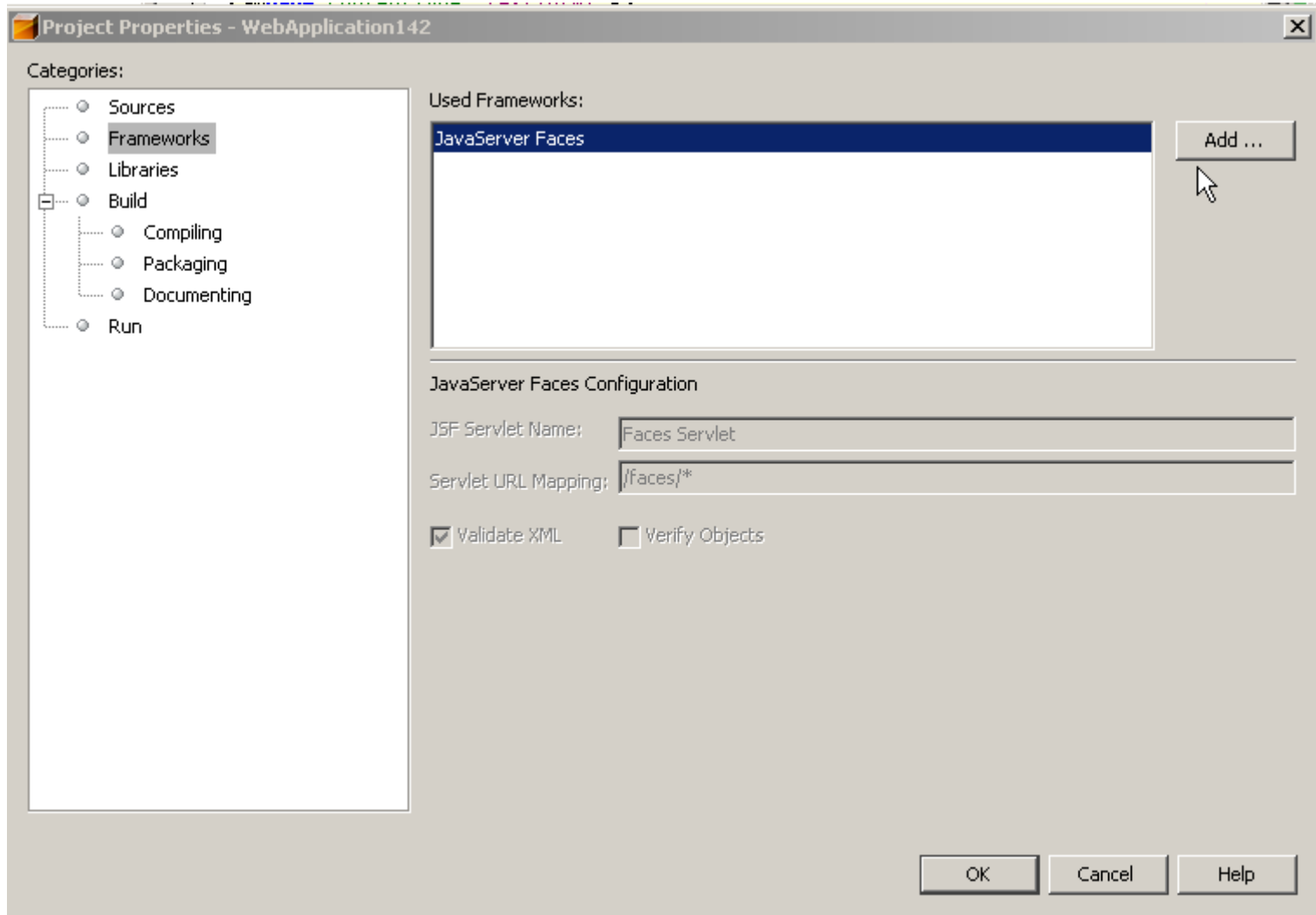
```



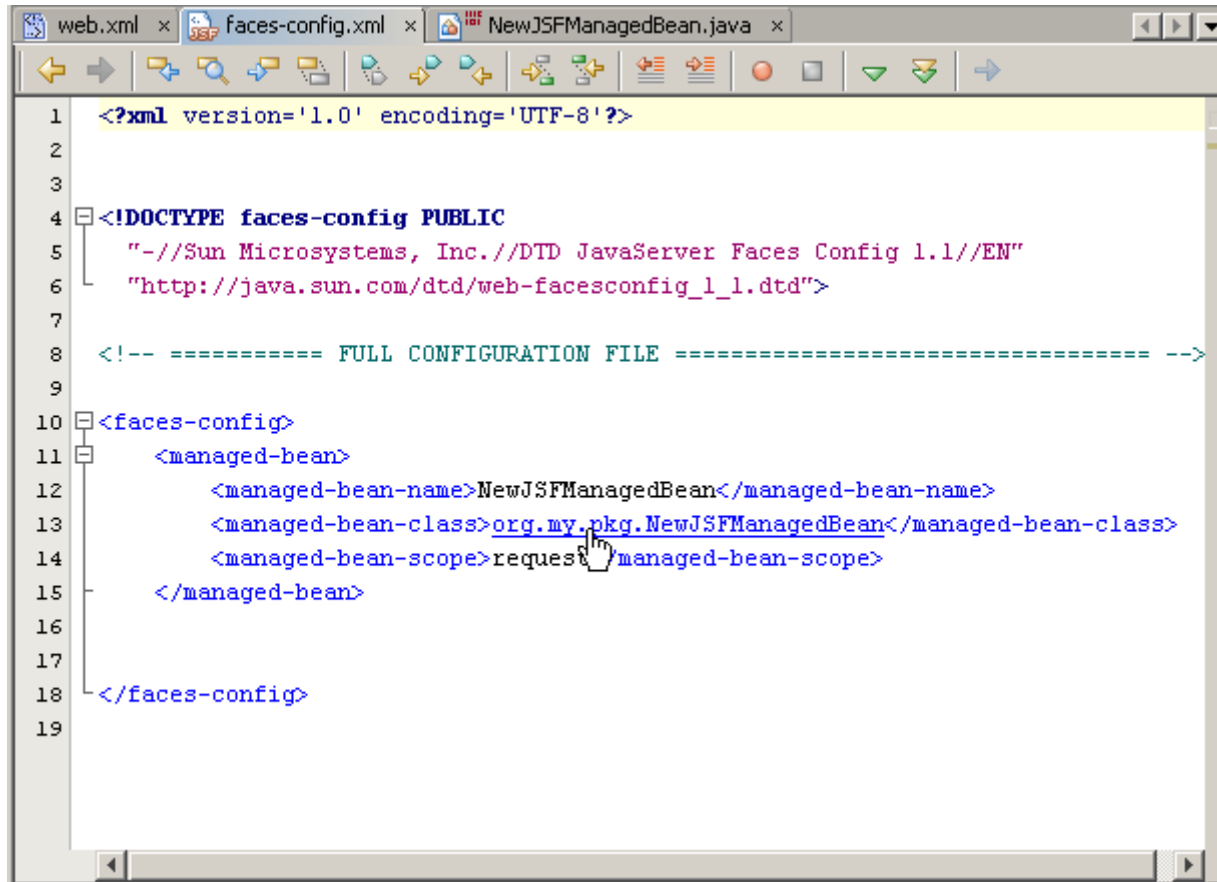
# JAR Files...



# Project Properties Extension...



# Hyperlinking...



```

1  <?xml version='1.0' encoding='UTF-8'?>
2
3
4  <!DOCTYPE faces-config PUBLIC
5     "-//Sun Microsystems, Inc.//DTD JavaServer Faces Config 1.1//EN"
6     "http://java.sun.com/dtd/web-facesconfig_1_1.dtd">
7
8  <!-- ===== FULL CONFIGURATION FILE ===== -->
9
10 <faces-config>
11   <managed-bean>
12     <managed-bean-name>NewJSFManagedBean</managed-bean-name>
13     <managed-bean-class>org.my.pkg.NewJSFManagedBean</managed-bean-class>
14     <managed-bean-scope>request</managed-bean-scope>
15   </managed-bean>
16
17 </faces-config>
18
19

```

# Web Framework Tooling...

Creating  
a source  
structure

Syntax  
highlighting

Code  
completion

Options

File  
templates

Refactoring

Component  
Palette

Indentation  
engines

Hyperl  
inking

Multiview  
Editors

JavaHelp  
helpsets

# Good news: Lots of help!

- 1. Great source code in NetBeans sources.
- 2. Open source projects:
  - > <https://nbwicketssupport.dev.java.net/>
  - > <https://nbfaceletssupport.dev.java.net/>
- 3. [dev@openide.netbeans.org](mailto:dev@openide.netbeans.org)
- 4. Javadoc & Tutorials (let's have a look...)
- 5. <http://blogs.sun.com/geertjan>
- 6. Two very recent developments...

# Pilot Chapter of NetBeans API Book



## Contents

- Registering a Web Framework.....
- Prerequisite Knowledge.....
- 1. Introduction.....
- 1.1 Preparing to Provide Support for a Web Framework.....
- 1.2 The WebFrameworkProvider Class.....
- 1.3 Getting Started Really Quickly.....
- 1.4 Example: Basic Registration.....
- 2. Preparing to Work with the WebFrameworkProvider Cla.....
- 3. Providing a Framework Configuration Panel.....
- 3.1 Creating the Panel.....
- 3.2 Example: Adding a Configuration Panel to the WebF.....
- 3.3 Coding the Panel.....
- 4. Creating a Source Structure and Files.....
- 4.1 Preparing to Use the Extend() Method.....
- 4.2 Example: Defining the Extend() Method.....



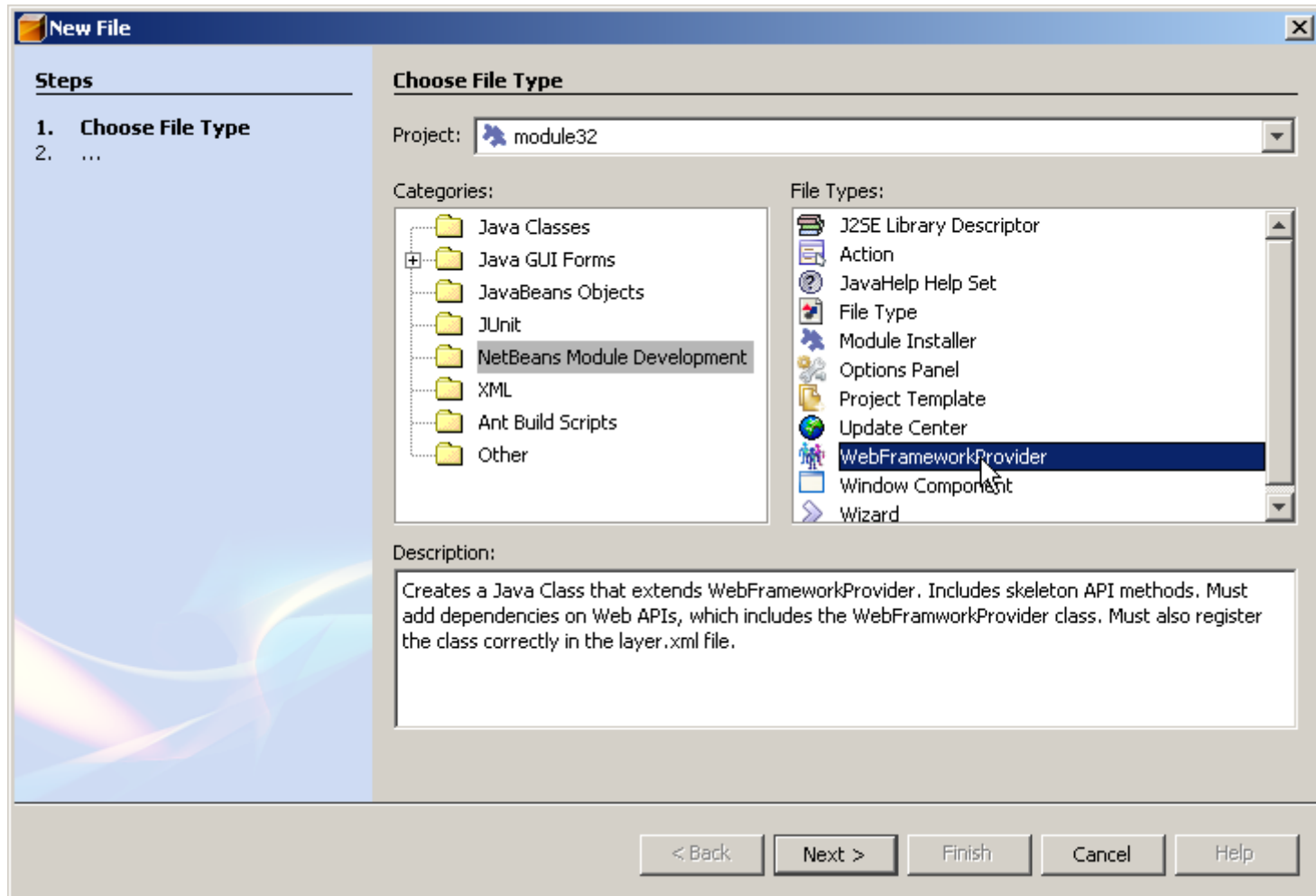
# Let's browse through the chapter...

# Let's put it into practise...

- [platform.netbeans.org](http://platform.netbeans.org) (*not* [www.netbeans.org](http://www.netbeans.org))
- Look in Javadoc
- WebFrameworkProvider class (NetBeans Web API)
  - > `extend()`
  - > `isInWebModule()`
  - > `getConfigurationFiles()`
  - > `getConfigurationPanel()`



# Wizard for Getting Started



# Wizard for Getting Started (ctd.)

**New WebFrameworkProvider**

**Steps**

1. Choose File Type
2. **Name and Location**

**Name and Location**

Class Name:

Project:

Location:

Package:

Created File:

Framework name:

- Tapestry
- Cocoon
- RIFE
- Beehive
- Echo
- Keel

< Back    Next >    Finish    Cancel    Help

```
public class TapestryWebFrameworkProvider extends WebFrameworkProvider {

    /** Creates a new instance of __NAME__ */
    public TapestryWebFrameworkProvider() {
        super("Tapestry", "Description");
    }

    //Empty for now, but this is where all the artifacts are created:
    public Set extend(WebModule webModule) {
        return null;
    }

    //If true, the Project Properties dialog box will show that the framework is selected.
    public boolean isInWebModule(WebModule webModule) {
        return true;
    }

    //Works with the framework's configuration files
    public File[] getConfigurationFiles(WebModule webModule) {
        return null;
    }

    //Returns the lower part of the New Project wizard or the Project Properties dialog
    //box, where the user can specify settings that are specific to your framework.
    //Here, nothing is returned:
    public FrameworkConfigurationPanel getConfigurationPanel(WebModule webModule) {
        return null;
    }
}
```

## ...and three small little steps:

- Declare dependency on the module that provides the `WebFrameworkProvider` class
- Register the `WebFrameworkProvider` class in the `layer.xml` file
- Install the module.

**...and then the framework is registered!**

# Demo!

1. Create module (plugin) project
2. Declare dependency on Web API
3. Use WebFrameworkProvider wizard
4. Register the framework provider class
5. Install the module

# Summary

- Integrating support for a web framework
  - > Wizards and templates
  - > Pilot chapter for upcoming module development book
  - > NetBeans API JavaDoc
  - > Samples and Tutorials
- Need help?
  - > [dev@openide.netbeans.org](mailto:dev@openide.netbeans.org)
  - > <http://blogs.sun.com/geertjan>
- Which frameworks do you want support for?

# Questions & Answers

