

How to configure remote JSP debugging under WebSphere Application Servers

Introduction

This article describes how you can create a run and debug configuration in IntelliJ IDEA 5.1 (or higher) to allow your Web modules with JSP files to be remotely debugged under WebSphere 5.1 and WebSphere 6.0 application servers. It also describes how to configure these application servers to enable integration with IntelliJ IDEA, and provides you with step-by-step instructions for application deployment.

Requirements

Before making any settings, make sure that the following requirements are met:

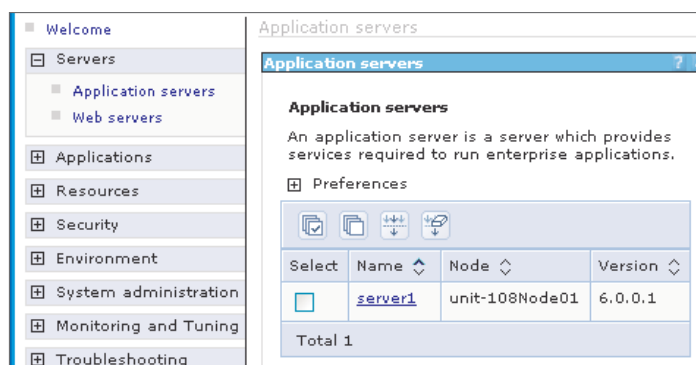
1. WebSphere 5.1 or 6.0 application server is installed (full or express version). IntelliJ IDEA does not have any bundled servers, so you need to install them yourself.
2. JDK (Sun Java2™ Standard Edition SDK) 1.4 or less is installed. Although IntelliJ IDEA supports JDK 5.0, WebSphere application servers require JDK 1.4 or less.
3. WebSphere application server is started.

You can find the full documentation on how to install, start, and initially configure the server at <http://www-306.ibm.com/software/websphere/>.

Configuring WebSphere Application Server

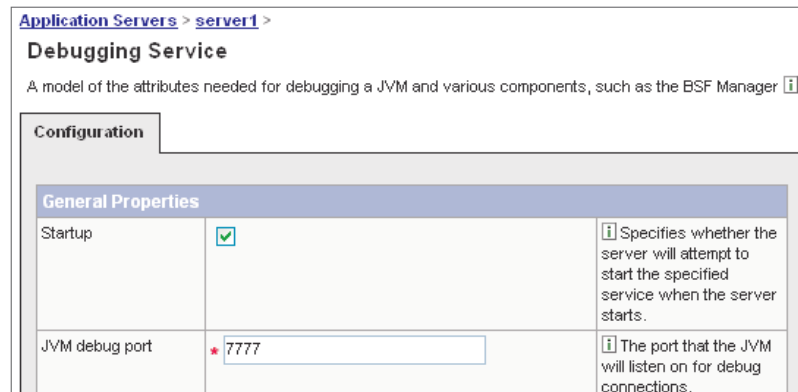
To enable remote debugging on the server, perform the following steps:

1. Start the server's **Administrative Console**.
2. Expand **Servers > Application Servers**.



WebSphere Application Server V6.0: Administrative Console

3. Click the server name and click the **Configuration** tab
4. Navigate to the **Additional Properties** group and click the **Debugging Service** link.
5. Make sure that the debugging service is enabled at server startup.
 - For the **WebSphere 5.1** application server, select the **Startup** check box.



WebSphere Application Server - Express 5.1: enabling debugging

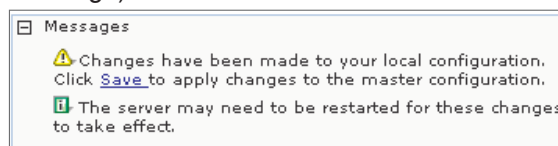
- For the **WebSphere 6.0** application server, select the **Enable service at server startup** check box.

To the **JVM Debug arguments** field add the following line before the existing set of arguments: `-Dwas.debug.mode=true`



WebSphere Application Server 6.0: enabling debugging

6. Click the **Apply** button. Then save the settings (for example, using the notification message)



WebSphere Application Server 6.0: notification message

7. Restart the server using WebSphere's or operating system's administration tools.

Generating the Application Archive File

IntelliJ IDEA can generate the WAR, EAR, or JAR file for your application automatically. For this purpose, you need to configure your module (Web, J2EE Application, or EJB).

1. In IntelliJ IDEA, click **File > Settings (Ctrl + Alt + S)**, and then select **Modules**.
2. In the **Modules** list, click the module for which you need to generate the archive file.
3. Click the **J2EE Build Settings** tab.
4. Select the **Create web module war/ear/jar file** check box, and specify the location of the file. This path will be required later, on the application deployment stage.

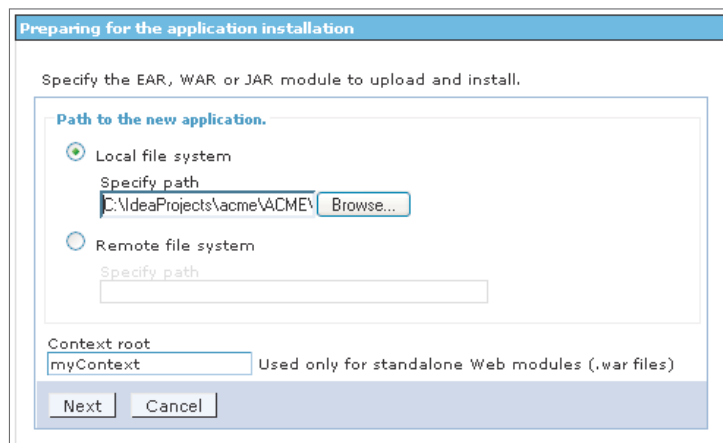
Apply the changes, and make the project (click the **Build > Make Project** menu or press **Ctrl + F9**).

The application is now ready for deployment.

Deploying Application

You deploy the application using the WebSphere application server's tools.

1. Start the server's **Administrative Console**.
2. Click **Applications > Install New Application**, and then click your server.
3. Select **Local file system** or **Remote file system** depending on your server installation, and then specify the path to your WAR, EAR, or JAR file (usually, it corresponds to the path you specified when generating the application archive file in IntelliJ IDEA).
4. In the **Context Root** text box, specify the context (e.g. "myContext").



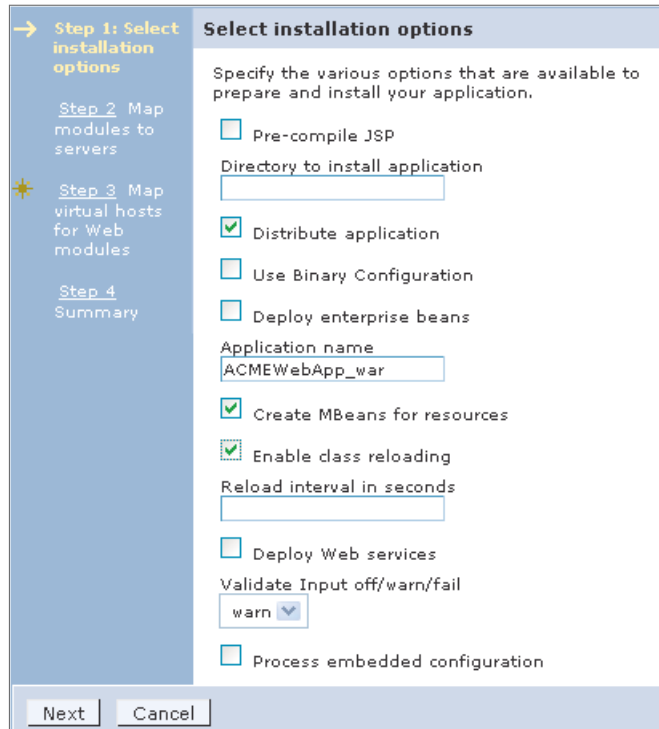
WebSphere Application Server 6.0: enabling debugging

5. Click **Next**. Skip the next screen and click **Next** again.



The Security Warning may appear. It displays the content of the `was.policy` file. You can simply click **Continue**.

- On the first step of the **Install New Application** wizard, select the **Enable class reloading** check box, and then click **Next**. This allows you to redeploy the application without reinstalling it



The screenshot shows the 'Select installation options' dialog box. On the left, there is a sidebar with four steps: Step 1 (selected), Step 2, Step 3, and Step 4. The main area contains the following options:

- Pre-compile JSP
- Directory to install application:
- Distribute application
- Use Binary Configuration
- Deploy enterprise beans
- Application name:
- Create MBeans for resources
- Enable class reloading
- Reload interval in seconds:
- Deploy Web services
- Validate Input off/warn/fail:
- Process embedded configuration

At the bottom, there are 'Next' and 'Cancel' buttons.

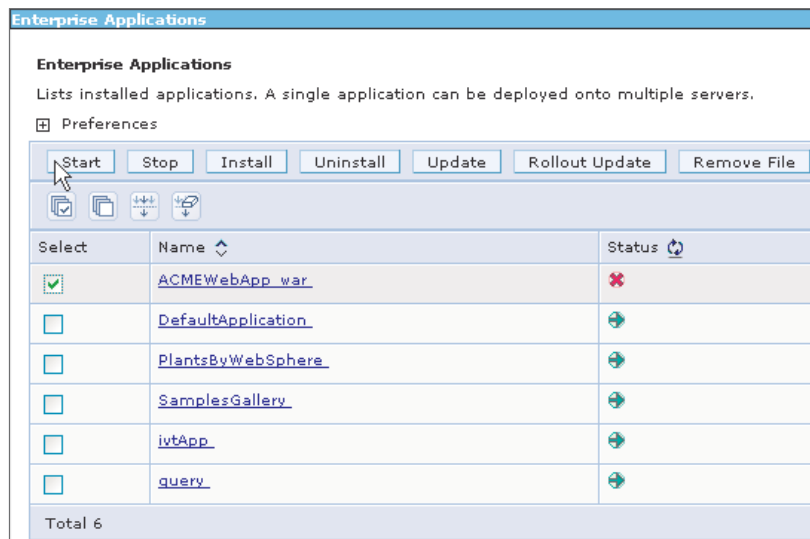
WebSphere Application Server 6.0: selecting installation options.

- On the second step, select the check box next to your Web module to map it to the server. Then click **Next**.
- On the third step, select the check box next to your Web module to map the virtual host to it, and then click **Next**.
- On the last step, just click **Finish**. The deployment will start.



The server may not always notify you when the deployment is complete. If the operation takes too long, click **Enterprise Applications**. If your application is present in the enterprise applications list with the "Stopped" status (red cross icon), the deployment is complete, and you can move to the next step.

- When the deployment is finished, the server console will show you the installation results page. Save the configuration, for example using the **Save to Master Configuration** link on this page.
- Click **Applications > Enterprise Applications**. In the list of applications your application should be displayed with the "Stopped" status (red cross icon).
- Select the check box next to your application, and then click **Start**.

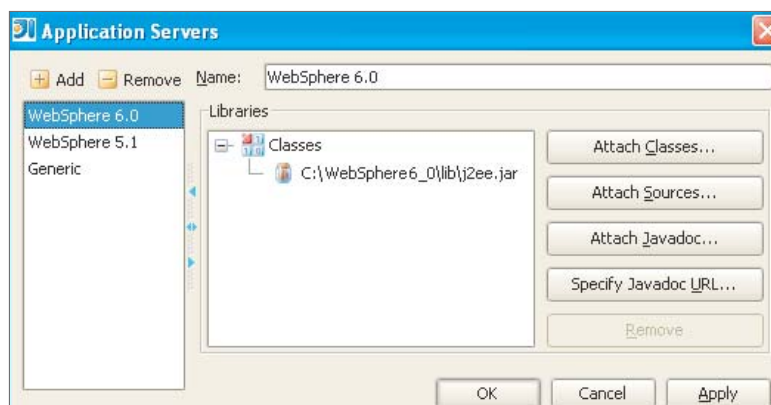


WebSphere Application Server 6.0: starting the application

Creating Run/Debug Configuration

WebSphere 5.1/6.0 application servers are supported by means of the general IntelliJ IDEA integration with the JSR compatible servers and do not have a server-specific type of Run/Debug configuration. You need to create a configuration for a JSR45 compatible server and then modify it according to WebSphere application server's needs.

1. In IntelliJ IDEA, click the **Run > Edit Configurations** menu, and then click the **JSR45 Compatible Server** tab.
2. Click the plus sign or press **Insert** to add a new configuration, and then click **Remote**. Specify some configuration name.
3. Click the **Configure** button to add an application server.
4. In the **Application Servers** dialog box, click **Add**, and then specify some server name.
5. Click the **Attach Classes** button, and find the `j2ee.jar` file for your version of WebSphere Application server (`<installation folder>\lib\j2ee.jar`).



IntelliJ IDEA 5.1: configuring application servers

6. In the **Application Servers** dialog box, click **Apply** to save the changes, and then in the configuration, in the **Application Server** list, select the added server.
7. In the **Startup page** text box, specify the home page of your web application in the following format:

- **WebSphere Application Server 5.1**

```
http://localhost:7080/<ContextRoot>/<home page>
```

- **WebSphere Application Server 6.0**

```
http://localhost:9080/<ContextRoot>/<home page>
```

“7080” and “9080” are default ports for the WebSphere 5.1 and 6.0 servers, respectively. If you have changed the default port, specify the actual port number instead.

`<ContextRoot>` is name of web context used during application deployment, e.g. “myContext”.

`<home page>` can be any page from your Web module.

8. Optionally, you can configure the application server log to display its content in the IntelliJ IDEA's **Run** tool window, in the console.

Click the **Add** button, and then in the **Edit Log Files Aliases** dialog specify some alias name (e.g. “activity”). In the **Log File Location** text box, specify the location of the `activity.log` file. By default, this file is located in the `<home>\logs` folder.

9. In the **JSP package** text box, enter one of the following:

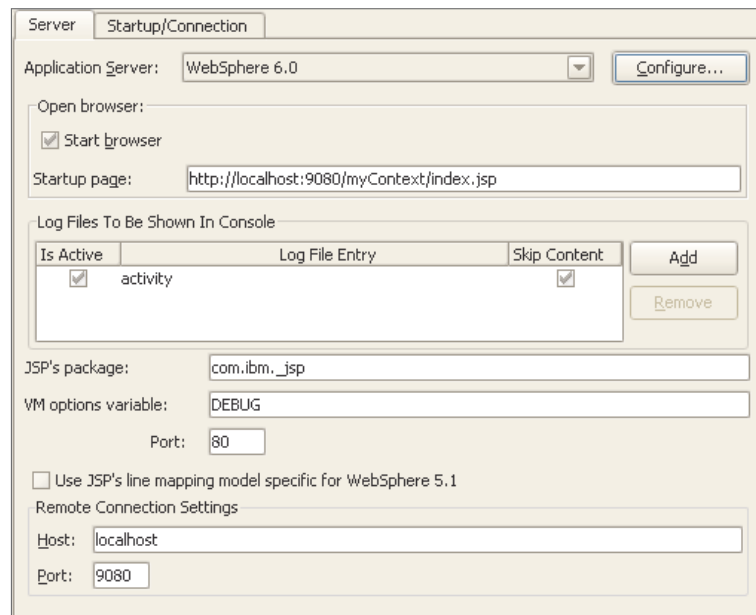
- **WebSphere Application Server 5.1**

```
org.apache.jsp
```

- **WebSphere Application Server 6.0**

```
com.ibm._jsp
```

10. In the **VM options variable** text box, type `DEBUG`.
11. If you use **WebSphere Application Server 5.1**, select to use the specific JSP's line mapping model. For the **WebSphere Application Server 6.0** this check box is ignored, but it is recommended to clear it.
12. In the **Port** text box, specify the port number of the corresponding application server (it should be the same as specified for the startup page).



IntelliJ IDEA 5.0: Run/Debug configuration for WebSphere Application Server 6.0

- Click the **Startup/Connection** tab, and then click **Debug**. Check that the **Socket** option is selected, and the **Port** number is 7777.

Debugging Application

To start debugging the application in IntelliJ IDEA, just click **Run > Debug** or press **Shift + F9**. You can use all the power of the IntelliJ IDEA debugger for the JSP files running on the WebSphere Application Server, including breakpoints, stepping commands, watches, etc.



To apply any changes in the web module, you will need to redeploy the application as described earlier in this article. The “Build on frame deactivation” feature is not supported for remote debugging.

Troubleshooting

The server crashes during the debug session

Such problems may appear if you use the IBM's virtual machine supplied with the WebSphere server. Possible solutions in this case are:

- Run IntelliJ IDEA with the command line option:


```
-Didea.debugger.keep.temp.objects=false
```

 In this case IntelliJ IDEA will avoid calling certain methods from debug API.
- Use Sun's JDK 1.4 or less

IntelliJ IDEA is unable to open the debugger port

In most cases, it is an internal WebSphere server error. Try to restart the WebSphere application server.

Conclusion

We hope that this article helped you to configure remote debugging successfully. If you have encountered any problems that are not described in the **Troubleshooting** section, don't hesitate to contact our technical support: support@jetbrains.com.