

# Debugging Jetty with IntelliJ IDEA

## Running/Debugging Jetty with IntelliJ IDEA

In order to integrate Jetty with, IntelliJ IDEA, you need two scripts.

### StartJetty.bat

```
@echo off
setlocal
if not "%JAVA_HOME%" == "" goto JettyAction

:SearchFromRegistry
echo JAVA_HOME environment variable is not
set.
echo Searching registry...
for /f "usebackq tokens=3" %i in (`reg
query
"HKEY_LOCAL_MACHINE\SOFTWARE\JavaSoft\Java
Development Kit" /s ^2^> null \^|find
"JavaHome"`) do set JAVA_HOME=%~i
if not "%JAVA_HOME%" == "" goto JettyAction

:NoJava
echo Could not find JDK in your system.
Please install JDK
endlocal
exit/b 1

:JettyAction
if not "%CD%" == "%~dp0" echo Please say
'Y' if prompted for "Terminate batch job
```

(Y/N)?"

rem %~dp0 is current dir in Winodws NT

set curpath=%~dp0

pushd %curpath%

set runcmd=%JAVA\_HOME%\bin\java

\-showversion \-DSTOP.PORT=8079

\-DSTOP.KEY=secret %VM\_DEBUG\_VAR% \-jar

start.jar %\\*

echo "Using " %runcmd%

```
%runcmd%  
popd  
endlocal
```

## StopJetty.bat

```
@echo off  
setlocal  
if not "%JAVA_HOME%" == "" goto JettyAction  
  
:SearchFromRegistry  
echo JAVA_HOME environment variable is not  
set.  
echo Searching registry...  
for /f "usebackq tokens=3" %%i in (`reg  
query  
"HKEY_LOCAL_MACHINE\SOFTWARE\JavaSoft\Java  
Development Kit" /s ^2^> null \^|\find  
"JavaHome"`) do set JAVA_HOME=%%~i  
if not "%JAVA_HOME%" == "" goto JettyAction  
  
:NoJava  
echo Could not find JDK in your system.  
Please install JDK  
endlocal  
exit/b 1  
  
:JettyAction  
if not "%CD%" == "%~dp0" echo Please say  
'Y' if prompted for "Terminate batch job
```

(Y/N)?"

rem %~dp0 is current dir in Winodws NT

set curpath=%~dp0

pushd %curpath%

set runcmd=%JAVA\_HOME%\bin\java

\-showversion \-DSTOP.PORT=8079

\-DSTOP.KEY=secret \-jar start.jar \--stop

echo "Using " %runcmd%

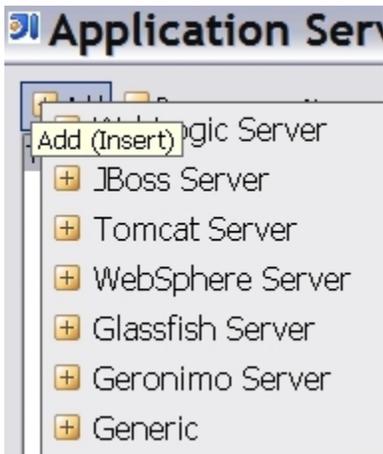
```
%runcmd%  
popd  
endlocal
```

**i** These scripts are not named specifically as IntelliJ scripts because they are also general purpose batch files to start and stop and Jetty.

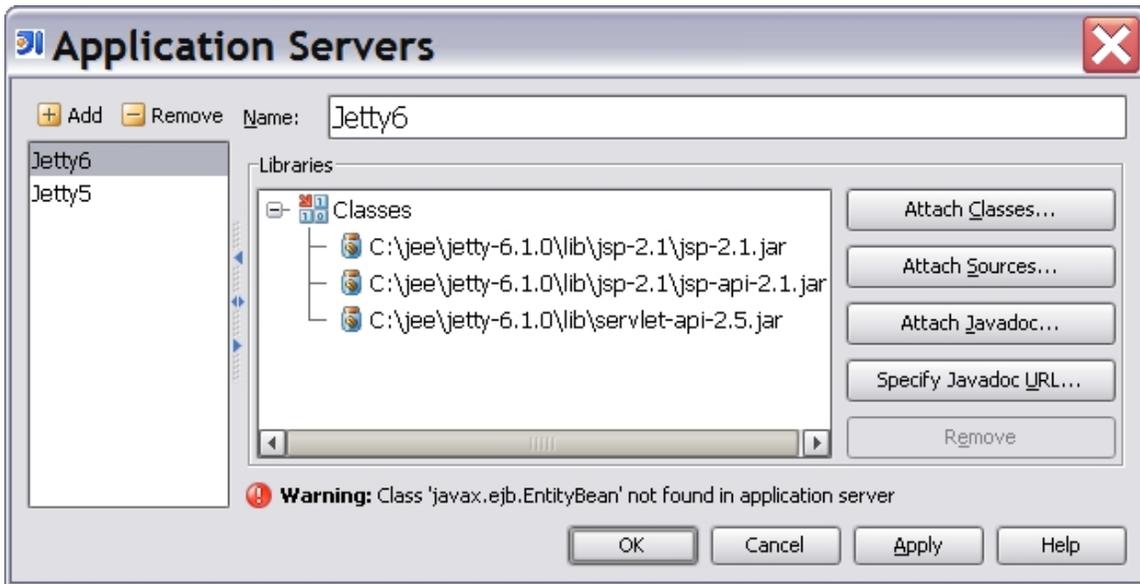
The script works reliably when placed immediately in Jetty dir. The script uses JAVA\_HOME environment variable if set. If it is not set, then the script tries to determine JAVA\_HOME from Windows Registry. The script can be run from any location, however the relative paths are computed with respect to Jetty dir. If you are running script from other dir then you should say 'Y' if prompted for "Terminate batch job (Y/N)?". This situation frequently arises when Ctrl+C is used for terminating a batch program. Otherwise, you will notice that your current directory is still set to Jetty dir. This is generally a good practice for other Batch programs as well.

## IntelliJ Configuration

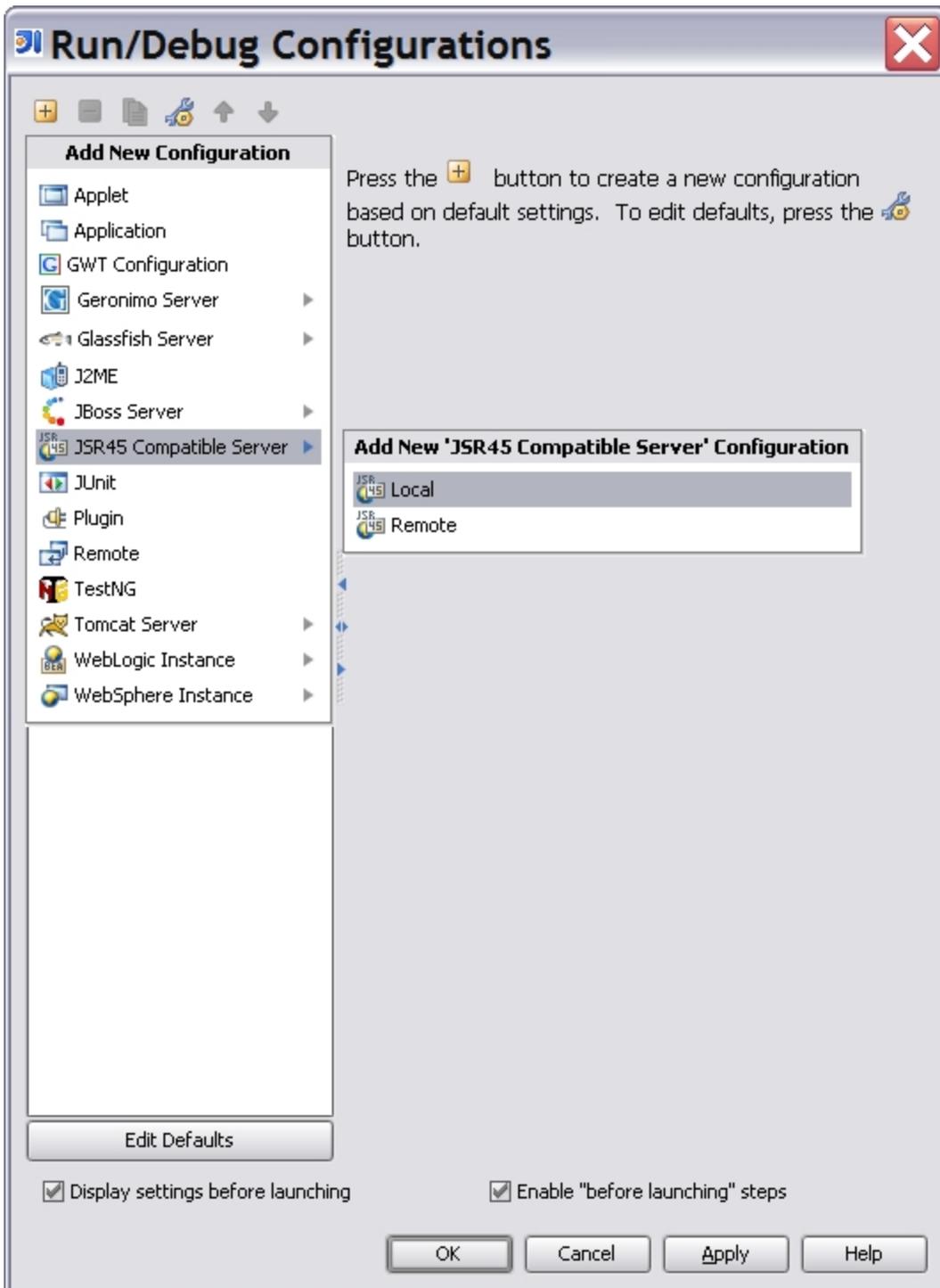
First of all, you would need to setup Jetty as Application Server in IntelliJ Settings. It's usually Ctrl+Alt+S or File->Settings. Navigate to "Application Servers" option, and add a Generic Server.



It is named Jetty here and servlet-api-2.5.jar and other jars from jetty lib dir are attached. You will also notice IntelliJ warning regarding not finding EJB classes. That's ok because Jetty is not an EJB container.



To Debug and Run your web app project from IntelliJ, you need to create a Local "JSR45 Compatible Server" configuration under Run-> Edit Configurations menu. Note that IntelliJ only allows you to configure this for each project.



I named the Local configuration as "Jetty".

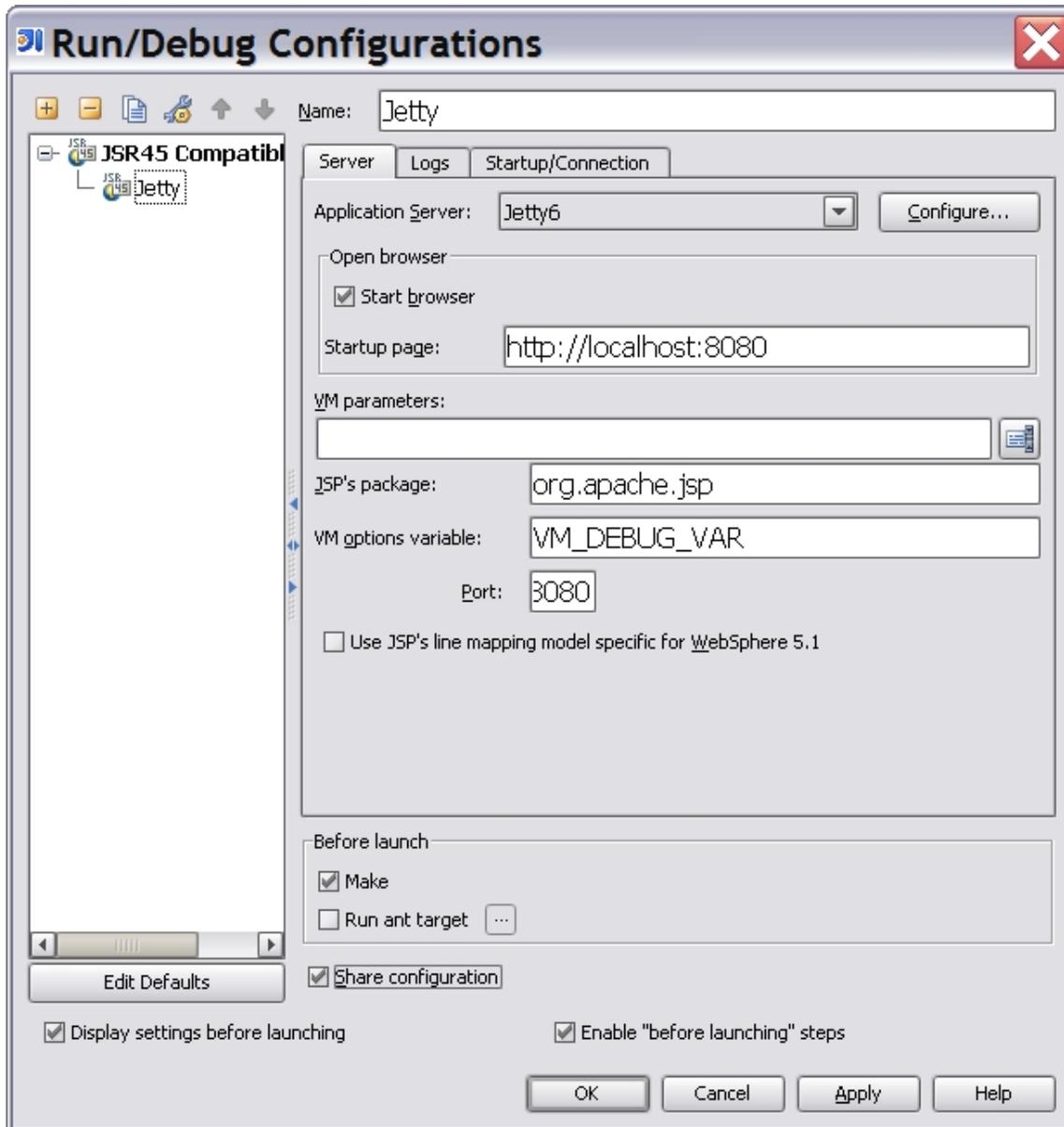
In "Server" tab, select Application Server we just created. Configure port consistent with jetty.xml, or value of jetty.port system property. Specify JSP package as org.apache.jsp. Refer to your webdefault.xml if you have, customized package for JSP generated java classes. Set value of VM options variable VM\_DEBUG\_VAR. This environment variable is used in Batch Scripts.

**⚠ Don't use VM\_OPTS variable name**

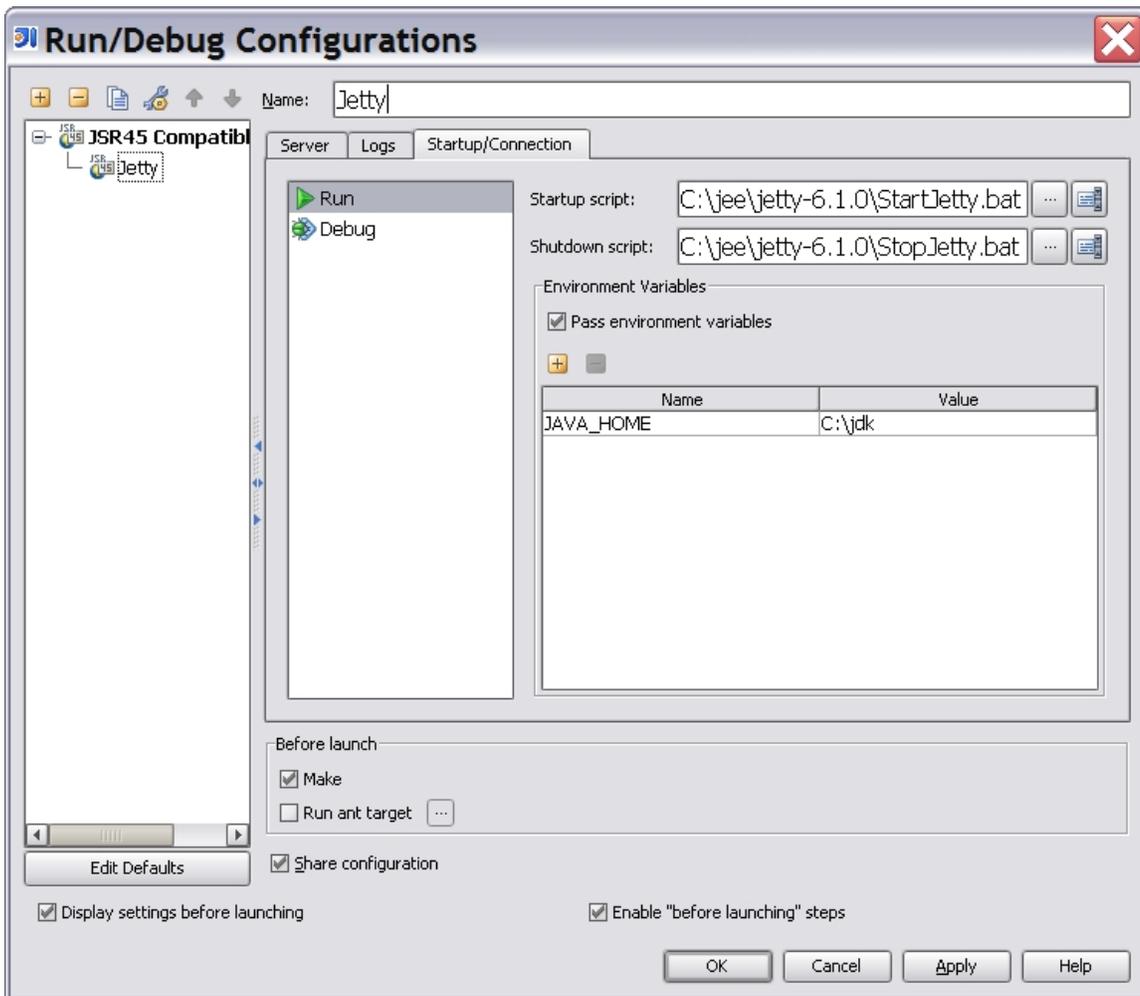
Due to certain bug in IntelliJ plugin,,if you name,this variable VM\_OPTS, it is not passed to Batch Scripts. So avoid using VM\_OPTS name,as environment variable. It's quite easy to use this variable as it is used in IntelliJ examples.

<http://www.jetbrains.com/idea/docs/help/project/rundebugjsr45.html>

You can also check/uncheck Start Browser,depending on what you want.

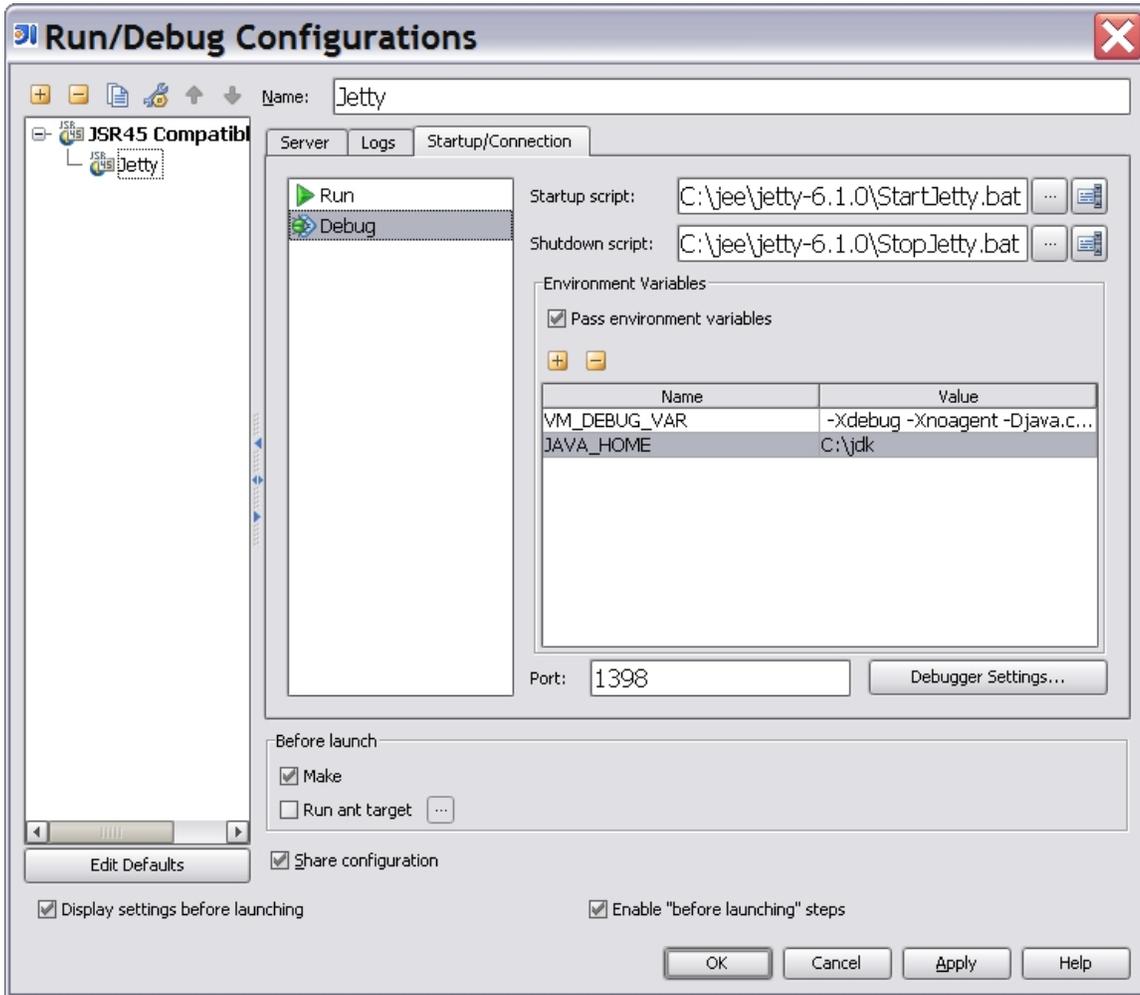


In "Startup/Connection" tab,,you need to select the start and stop script for,running and debugging script.



I have also specified JAVA\_HOME environment variable in this case. The StartJetty.bat and StopJetty.bat script work for both running and debugging.

IntelliJ passes debug parameters in form of an environment variable to the script. See the value of VM\_DEBUG\_VAR created by IntelliJ for Debug mode.



Other VM parameters and program arguments can be passed by selecting  icon next to each script.



Now create a jsp in your web app and try a breakpoint in a jsp.

## Jetty with,Java Service Wrapper by Tanuki Software

Another approach,of running and,debugging is using,Java Service Wrapper itself in console mode using -c option. The advantage of using Java Service using is that you don't have to specify JVM options and program arguments in multiple places. For debugging purpose, you can simply add one of the following two options in your Java Service Wrapper conf file.

```
-agentlib:jdwp=transport=dt_socket,server=y,  
suspend=n,address=5005  
-Xdebug-Xnoagent-Djava.compiler=NONE  
-Xrunjdwp:transport=dt_socket,server=y,suspe  
nd=n,address=5005
```

Run Jetty in console mode., You only need conf file. You don't have to install this as Windows Service.

```
wrapper.exe -c jetty.conf
```

Now you can use IntelliJ Remote debugging facility. See Remote tab under Run-> Edit Configurations menu option.