

HowToLenyaInNetbeans4

And [StepByStep](#) tutorial explaining how to setup a netbeans 4.0 project with lenya.

First of all excuse me for my English....

Last month I have been working in lenya with netbeans 4.0 and I managed to create a project to work with lenya in this IDE. Here I put the steps I follow to configure the IDE:

1. Download the latest stable source distribution (1.2.3 when I wrote this page)
2. Follow the steps provided by lenya to compile. You obtain a build folder. [BR]
3. Netbeans doesn't allow you to create a project without a web.xml file in WEB-INF folder. Lenya doesn't provide you a web.xml file in src/webapp/WEB-INF folder. You can use the build/lenya/webapp/WEB-INF/build.xml file to cheat lenya.
4. Create a web project with existing sources using new project wizard.
5. Choose folder {LENYA_HOME}/src/java in src files and {LENYA_HOME} in project folder.
6. When you accept, netbeans tells us that this project has an existing build.xml. Netbeans creates a new nbbuild.xml file...
7. Choose {LENYA_HOME}/src/webapp folder in web pages location.
8. Choose {LENYA_HOME}/lib folder in libs location.
9. Creating the project sometimes netbeans hangs up. We can cancel and fortunately the project is created

/***** Changes in NBBUILD.XML *****/

Note: Be careful with ant project name (affects variable ant.project.name). To avoid problems (like not being able to deploy the project), use context path name (for instance lenya) like ant project name.

10. Now we copy all ant import from build.xml file to nbbuild.xml.

11. Now we add our custom targets. For instance:

```
<!-- OUR TARGETS -->
```

```
<import file="src/targets/our_targets_file.xml" />
```

12. Now we "patch" some lenya targets to adapt it to our requirements:

- webapp.xml: Comment the steps in charge of loading Tomcat or Jetty

```
<!-- netbeans takes care of this...
```

```
<condition property="servlet.engine.wls">
```

```
<equals arg1="{web.app.server}" arg2="WLS"/>
```

```
</condition>
```

```
<antcall target="install-wls"/>
```

```
<antcall target="successfully-built-message-wls"/>
```

```
<condition property="servlet.engine.tomcat">
```

```
<equals arg1="{web.app.server}" arg2="Tomcat"/>
```

```
</condition>
```

```
<antcall target="install-tomcat"/>
```

```
<antcall target="successfully-built-message-tomcat"/>
```

```
<condition property="servlet.engine.jetty">
```

```
<equals arg1="{web.app.server}" arg2="Jetty"/>
```

```
</condition>
```

```
<antcall target="successfully-built-message-jetty"/>
```

```
-->
```

- Change the name of clean target to clean-lenya to avoid interferences with netbeans default clean target.
- Change the name of init target to init-lenya to avoid interferences with netbeans default init target.
- compile-build-xml: Comment lenya compile target. We now want to use netbeans compile target.

```
<!-- netbeans takes care of this...
```

```
<target name="compile" depends="prepare-src"
```

```
description="Compile the sources">
```

```
<mkdir dir="{build.dest}"/>
```

```
<javac srcdir="{build.src}"
```

```
destdir="{build.dest}"
```

```
debug="{debug}"
```

```
optimize="{optimize}"
```

```
deprecation="{deprecation}"
```

```
target="{target.vm}"
```

```
nowarn="{nowarn}"
```

```
source="1.4">
```

```
<classpath refid="classpath"/>
```

```
</javac>
```

```
</target>
```

```
-->
```

13. Now we add source libraries needed to compile in properties project. We can obtain them from Cocoon folder or lenya itself:

- avalon-framework-api-xxxx.jar
- avalon-framework-impl-xxxx.jar
- cocoon-xxx.jar
- log4j.jar
- excalibur-source-resolver.jar
- excalibur-io.jar

- excalibur-xmlutils.jar
- jakarta-reg-exp.jar
- xalan-xxx.jar
- commons-lang.jar
- commons-httpclient.jar
- lucene.jar
- jtydy-xxx.jar
- xml-commons-resolver.jar
- rhino-xxx.jar
- quartz-xxx.jar : You can find it at coocon/lib/optional.
- All *.jar from tools/lib folder.

14. Now we overwrite the following netbeans ant targets in nbbuild.xml file:

```
<target name="-pre-compile" depends="prepare-src"/>

<target name="-pre-init" depends="init-lenya"/>

<target name="-post-init" depends="sincronizar_build_con_src"/>

<target name="-post-clean" depends="clean-lenya"/>
```

15. Add the following ant targets to nbbuild.xml our our custom targets file (you choose!) You can find Kupu at kupu.build file in src/targets folder)

```
<target name="kupu" description="Build Kupu">

<echo>
*****
• Build Kupu. * ***** ${build.webapp}/lenya/resources/kupu
</echo>
<exec dir="${build.webapp}/lenya/resources/kupu" executable="xsltproc" failonerror="false" failifexecutionfails="false" outputproperty="xsltproc.
output"
/>
<fail unless="xsltproc.output">: *****
• Please install "xsltproc" to build Kupu. *
  ○ *
  ○ xsltproc can be downloaded from *
  ○ ftp://ftp.zlatkovic.com/pub/libxml/ *
  ○ (you also need zlib and iconv) *
  *****
  </fail>
  <exec dir="${build.webapp}/lenya/resources/kupu"
  executable="make"
  failonerror="true"
  failifexecutionfails="false"
  />
</target>

<target name="clean_tomcat_work_folder">
<delete dir="/home/sergi.tur/.netbeans/4.0/jakarta-tomcat-5.0.28_base/work/Catalina/localhost/lenya"/>
</target>
```

16. Customize filke {LENYA_HOME}/build.properties to our specific paths requirements (cococon.src.dir,etc).

17. Modife the following ant properties in file {LENYA_HOME}/nbproject/project.properties:

```
-build.dir=build/ a build.dir=build/lenya
-build.web.dir=${build.dir}/web a build.web.dir=${build.dir}/webapp
```

Here you can find the same steps in my native language (spanish):

1. Bajarse los fuentes de la última versión estable (1.2.3 al escribir este doc)
2. Compilar las fuentes según los pasos indicados en lenya. Se obtiene una carpeta build.
3. Netbeans no permite crear un proyecto web sin fichero web.xml en la carpeta WEB-INF. Como las fuentes no tiene este fichero copiaremos el fichero web.xml de la carpeta build a src/webapp/WEB-INF.
4. Crear un nuevo proyecto de netbeans tipo: Web project with existing sources.
5. Escoger los ficheros fuente (carpeta {LENYA_HOME}/src/java) y la carpeta del proyecto ({LENYA_HOME}).
6. Nombrar el proyecto.
7. Al aceptar no indicará que ya existe un fichero build para este proyecto y que creara uno con el nombre nbbuild.xml.
8. Aceptamos. Elegimos como localización de los ficheros web (web pages location) la carpeta {LENYA_HOME}/src/webapp.
9. Elegimos como localización de las librerías la carpeta {LENYA_HOME}/lib.
9. A veces da un error y se queda colgado al crear el proyecto. Sin embargo si cerramos la ventana de dialogo podremos abrí el proyecto recién creado desde el punto de menu nuevo proyecto.

/***** RETOCAR EL FICHERO NBBUILD.XML *****/

Nota: Ojo con el nombre del proyecto ant que afecta a la variable ant.project.name. Este debe ser igual al nombre del contexto que que estemos utilizando en las propiedades del proyecto.

10. En el fichero nbbuild.xml copiamos todos los imports del fichero build.xml original (se important todos los ficheros de proyecto de la carpeta src/targets)

11. Añadimos un linea como esta:

```
<!-- NUESTROS TARGETS -->
<import file="src/targets/tu_fichero_de_targets.xml" />
```

12. Hay que hacer unos pequeños cambios en los ficheros target del lenya para adaptarlo a netbeans:

- webapp.xml: Comentar la parte encargada de ejecutar Tomcat o Jetty

```
<!-- ESTO SE LO DEJAMOS AL NETBEANS
<condition property="servlet.engine.wls">
<equals arg1="{web.app.server}" arg2="WLS"/>
</condition>
<antcall target="install-wls"/>
<antcall target="successfully-built-message-wls"/>

<condition property="servlet.engine.tomcat">
<equals arg1="{web.app.server}" arg2="Tomcat"/>
</condition>
<antcall target="install-tomcat"/>
<antcall target="successfully-built-message-tomcat"/>

<condition property="servlet.engine.jetty">
<equals arg1="{web.app.server}" arg2="Jetty"/>
</condition>
<antcall target="successfully-built-message-jetty"/>
-->
```

Cambiar el target clean por clean-lenya.

- init-build.xml: Renombramos el target init a init-lenya para evitar conflictos con el init de netbeans.
- compile-build.xml: Hay que comentar el compile de lenya para utilizar el de netbeans:

```
<!-- SUSTITUIDO POR EL DE NETBEANS
<target name="compile" depends="prepare-src"
description="Compile the sources">
<mkdir dir="{build.dest}"/>
<javac srcdir="{build.src}"
destdir="{build.dest}"
debug="{debug}"
optimize="{optimize}"
deprecation="{deprecation}"
target="{target.vm}"
nowarn="{nowarn}"
source="1.4">
<classpath refid="classpath"/>
</javac>
</target>
-->
```

13. Para compilar necesitaremos añadir algunas librerías extras que no están en la carpeta lib (son de cocoon):

- avalon-framework-api-xxx.jar
- avalon-framework-impl-xxx.jar
- cocoon-xxx.jar
- log4j.jar
- excalibur-source-resolver.jar
- excalibur-io.jar
- excalibur-xmlutils.jar
- jakarta-reg-exp.jar
- xalan-xxx.jar
- commons-lang.jar
- commons-httpclient.jar
- lucene.jar
- jtydy-xxx.jar
- xml-commons-resolver.jar
- rhino-xxx.jar
- quartz-xxx.jar : Esta en en coocon/lib/optional.
- Todos los .jar de la carpeta tools/lib.

14. Hay que sobrescribir los targets de netbeans siguientes:

```
<target name="-pre-compile" depends="prepare-src"/>

<target name="-pre-init" depends="init-lenya"/>

<target name="-post-init" depends="sincronizar_build_con_src"/>

<target name="-post-clean" depends="clean-lenya"/>
```

15. Añadir los siguientes targets (el de kupu se copia del fichero kupu.build de src/targets)

```
<target name="kupu" description="Build Kupu">

<echo>
*****
```

- Build Kupu. * ***** \${build.webapp}/lenya/resources/kupu
 <echo>
 <exec dir="\${build.webapp}/lenya/resources/kupu" executable="xsltproc" failonerror="false" failifexecutionfails="false" outputproperty="xsltproc.output"
 />
 <fail unless="xsltproc.output">: *****
 • Please install "xsltproc" to build Kupu. *
 o *
 o xsltproc can be downloaded from *
 o <ftp://ftp.zlatkovic.com/pub/libxml/> *
 o (you also need zlib and iconv) *

 </fail>
 <exec dir="\${build.webapp}/lenya/resources/kupu"
 executable="make"
 failonerror="true"
 failifexecutionfails="false"
 />
 </target>

<target name="clean_tomcat_work_folder">

<delete dir="/home/sergi.tur/.netbeans/4.0/jakarta-tomcat-5.0.28_base/work/Catalina/localhost/lenya"/>

</target>

16. Repasar el fichero {LENYA_HOME}/build.properties para ver si los paths son correctos (cocoa.src.dir,etc).

17. Modificar en el fichero {LENYA_HOME}/nbproject/project.properties:

-build.dir=build/ a build.dir=build/lenya

-build.web.dir=\${build.dir}/web a build.web.dir=\${build.dir}/webapp