# Main.buildinstructions

## How to build

## Build all

- For simplicity, we suggest you to have the following directory structure:

```
base.directory/ eclipse exo-tomcat exo-jonas exo-jboss maven projects/
exoplatform yourproject
```

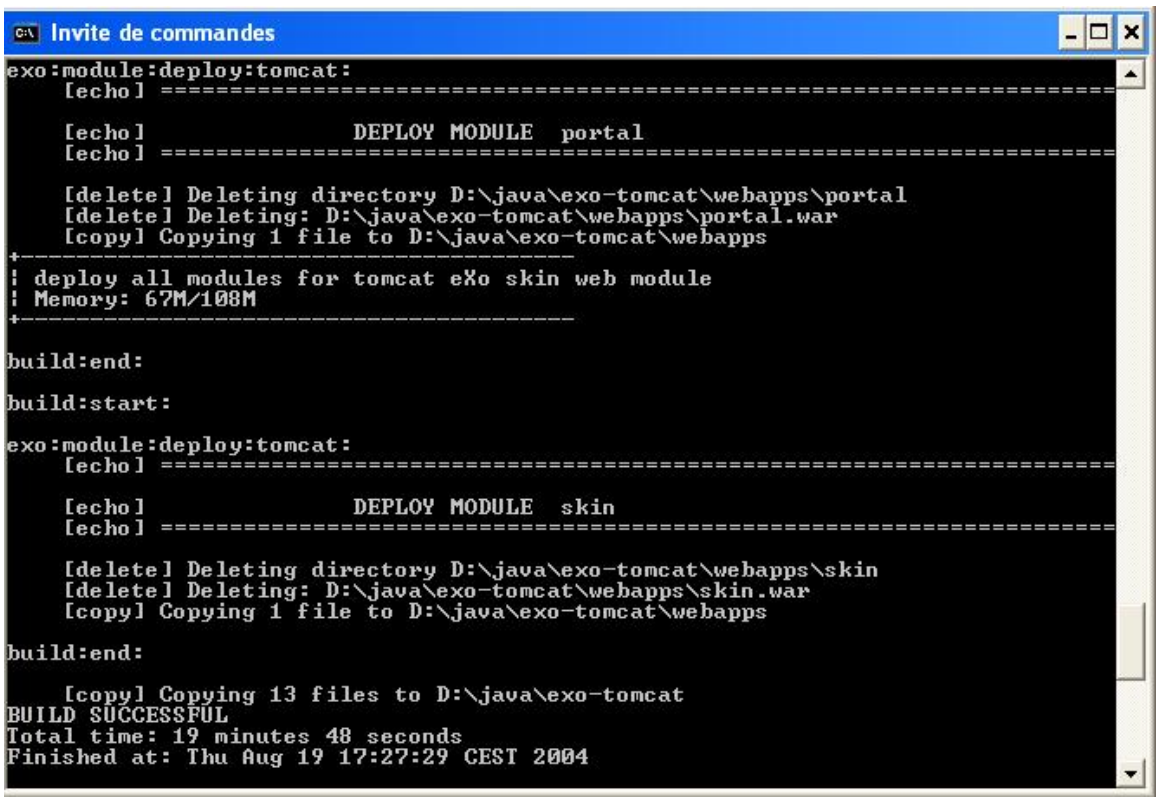
- Go to maven-plugins/exo, run maven plugin:install (If you reinstall, you may need to delete $MAVEN_HOME/plugins/maven-exo-plugins-1.0.jar before install
- Go to build/, edit the build.properties file according to your env, copy that file to your $HOME directory. If you use the directory structure we suggest, you should only need to update the base.directory property. If you are using unix like environement, you need to change maven.repo.remote=file:/${base.directory}/exoplatform/lib to maven.repo.remote=file:${base.directory}/exoplatform/lib
- Run maven multiproject:install
- Run "maven exo" for latest help instructions from the maven exo plugin (Optional)
- Run maven exo:platform:deploy
- For jonas deployment, You need to run maven exo:patch:jonas in build to modify the jonas configuration. Actually the command copy all the files in buid/src/exo-jonas to your exo-jonas.
- For jboss deployment, You need to run maven exo:patch:jboss in build/ to modify the jboss configuration. Actually the command copy all the files in buid/src/exo-jboss to your jboss.

To customize your own portal components edit maven.exo.module.includes in build.properties and include your portlet choices. The web/.../project.xml and portal/project.xml are mandatory.

```
#minimum requirement for a basic portal
maven.exo.module.includes=web/*/project.xml, portal/project.xml,
portlets/core/admin/project.xml, portlets/core/user/project.xml,
portlets/core/nav/project.xml, portlets/core/content/project.xml #deploy
all the portlets example
maven.exo.module.includes=portlets/**/project.xml, web/*/project.xml,
portal/project.xml
```

IMPORTANT: You need to configure your build.properties according to the instruction from maven exo command.

Here is what you should see at the end :




## Incremental build

It is possible to make incremental builds by going to each maven subproject and run :

```
maven exo:module:deploy
```

the maven command will check your module type (jar or war) and call maven jar:install or maven war:install and copy the output jar/war file to your deploy directory

## Build your project

We include a sample project that has a customized portal , a sample ioc service and sample portlets. You can create your own project base on that sample project. It will allow you to have your code separated from exoplatform src code while you will still be able to reuse our convenient maven plugin (maven exo:platform:deploy, maven exo:module:deploy) and our eclipse setting in case you want to debug both your code and exoplatform code.

The sample project has the directory structure

```
sample/ build/ src/ exo-tomcat exo-jboss web/ ic3/ skin/ services/
sample-service/ api impl portlets/ sample-portlet
```

In the build directory , you will find a minimum requirement build.properties that can be used for both exoplatform and your project. It is up to you to choose the build.properties in exoplatform/build or sample/build as the template for your $HOME/build.properties file. In the sample/build/project.properties, you shoudl find the properties:

```
maven.exo.thirdparty.dir=${base.directory}/projects/ic3-exo-2004
maven.exo.thirdparty.module.includes=portlets/**/project.xml,
web/*/project.xml
```

Those properties define wich war modules will be deployed to tomcat or jboss when you run the command maven exo:platform:deploy. If you want to deploy other jar modules or third party jar files with the war module. You need to define the dependency in the war project.xml file

```
<dependencies> [....] <dependencies> <dependency>
<groupId>exo-demo</groupId>
<artifactId>exo-demo.services.sample.api</artifactId>
<version>1.0</version> <type>jar</type> <properties> <!-- tell maven
exo:module:deploy command to copy the this jar file to the deploy dir
--> <tomcat.dependency>true</tomcat.dependency>
<war.manifest.classpath>true</war.manifest.classpath>
<eclipse.dependency>true</eclipse.dependency> </properties>
</dependency> <dependency> <groupId>exo-demo</groupId>
<artifactId>exo-demo.services.sample.impl</artifactId>
<version>1.0</version> <type>jar</type> <properties>
<tomcat.dependency>true</tomcat.dependency>
```

```
</war.manifest.classpath> <eclipse.dependency>true</eclipse.dependency>
</properties> </dependency> [....] </dependencies>
```

- To build your project , you need to build exoplatform first. Refer to the instruction above to build exoplatform. The reason that you need to build exoplatform is because we do not have a maven repository that store the exoplatform.*.jar file so you can download via internet. Also if you want to use our mock unit testing environment, you need to access certain resources that are stored in the exoplatform/ directory.
- After building the exoplatform, do not run maven exo:platform:deploy. Go to sample/build and run maven multiproject:install or maven multiproject:clean multiproject:install. This step will compile your project code, test , create jar/war file and copy to the maven repository
- Run maven exo:platform:deploy. This step will copy the module that you define in

```
maven.exo.thirdparty.dir=${base.directory}/projects/ic3-exo-2004
maven.exo.thirdparty.module.includes=portlets/**/project.xml,
web/*/project.xmlmaven.exoplatform.dir=${base.directory}/projects/exoplatform
#minimum requirement for a basic portal
maven.exo.module.includes=web/*/project.xml, portal/project.xml,
portlets/core/admin/project.xml, portlets/core/user/project.xml,
portlets/core/navigation/project.xml,
portlets/core/communications/project.xml,
portlets/core/content/project.xml #deploy all the portlets example
#maven.exo.module.includes=portlets/**/project.xml,
```

1. web/*/project.xml,

to the deploy dir.

- If you are using jonas , run maven exo:patch:jonas in both exoplatform/build and sample/build.
- If you are using jboss , run maven exo:patch:jboss in both exoplatform/build and sample/build.
- If you are using tomcat, run maven exo:patch:tomcat in sample/build. This step apply to the first time you build exoplatform and the sample project only.

Main.buildinstructions (en)
CrÃ©ateur: XWiki.exo  Creation Date: 2004/08/01 16:48
Dernier Auteur: XWiki.benjamin.mestrallet  Last Modification Date: 2005/05/12 18:05