

Fornax-Platform : 4. Archetype Tutorial (CSC)

This page last changed on Nov 26, 2007 by [patrik_nordwall](#).

[3. Advanced Tutorial Sculptor \(CSC\)](#)
[\(CSC\)](#)

[5. CRUD GUI
Tutorial \(CSC\)](#)

Sculptor Archetype Tutorial

In this tutorial we will create a simple Java EE application from scratch using the Maven archetypes provided by Sculptor. It consists of the following projects:

- sculptor-helloworld-parent - Only a maven project for building the other parts.
- sculptor-helloworld - Business tier. EJB project containing the services and domain objects.
- sculptor-helloworld-web - Presentation tier. Web application with CRUD GUI.
- sculptor-helloworld-ear - EAR package of the deployable application.

Table of Contents:

- [Sculptor Archetype Tutorial](#)
 - [Setup maven projects](#)
 - [Import into Eclipse](#)
 - [Complete Business Tier](#)
 - [Build](#)
 - [Setup Database](#)
 - [Deploy](#)
 - [Test](#)
 - [Deployment as WAR without EJBs](#)
 - [Hot Deploy](#)
 - [Debugging](#)
 - [Adding Dependencies](#)
 - [EJB 2.1](#)
 - [Source](#)

Setup maven projects

We start with creating a script that calls the 4 maven archetypes to generate the project structures and maven build files. It also does an initial build and generation of Eclipse project with the maven eclipse plugin. Of course you can execute these commands one by one from the command prompt, but the script is useful when doing this several times.

Copy the following script to `sculptor-archetypes.cmd`, located in the root of your Eclipse workspace. Adjust paths to your environment.

```
set MVN_HOME=C:\devtools\Maven-2.0.6
set JAVA_HOME=C:\devtools\jdk-1.5.0_12
set path=%MVN_HOME%\bin;%JAVA_HOME%\bin
```

```
set PACKAGE=%1
set SYS_NAME=%2
```

```
call mvn archetype:create -DarchetypeGroupId=org.fornax.cartridges
-DarchetypeArtifactId=fornax-cartridges-sculptor-archetype-parent -DarchetypeVersion=1.1.1
-DremoteRepositories=http://www.fornax-platform.org/m2/repository -DgroupId=%PACKAGE%
-DartifactId=%SYS_NAME%-parent
```

```
call mvn archetype:create -DarchetypeGroupId=org.fornax.cartridges
-DarchetypeArtifactId=fornax-cartridges-sculptor-archetype -DarchetypeVersion=1.1.1
-DremoteRepositories=http://www.fornax-platform.org/m2/repository -DgroupId=%PACKAGE%
-DartifactId=%SYS_NAME%
```

```
call mvn archetype:create -DarchetypeGroupId=org.fornax.cartridges
-DarchetypeArtifactId=fornax-cartridges-sculptor-archetype-web -DarchetypeVersion=1.1.1
-DremoteRepositories=http://www.fornax-platform.org/m2/repository -DgroupId=%PACKAGE%
-DartifactId=%SYS_NAME%-web
```

```
call mvn archetype:create -DarchetypeGroupId=org.fornax.cartridges
-DarchetypeArtifactId=fornax-cartridges-sculptor-archetype-ear -DarchetypeVersion=1.1.1
-DremoteRepositories=http://www.fornax-platform.org/m2/repository -DgroupId=%PACKAGE%
-DartifactId=%SYS_NAME%-ear
```

```
pause
```

```
cd %SYS_NAME%-parent
```

```
call mvn install
```

```
pause
```

```
call mvn -DdownloadSources=false eclipse:eclipse
```

```
cd ..
```

Run this script by defining the package name as the first parameter and the application id (artifact id of business tier) as the second parameter.

```
sculptor-archetypes org.helloworld helloworld
```



Maven 2 Plugin for Eclipse

The above instruction uses `eclipse:eclipse` to generate Eclipse `.project` and `.classpath` files. This is a simple and fully functional approach in most cases. An alternative to `eclipse:eclipse` is to use the [Maven 2 Plugin for Eclipse](#). If you prefer that plugin there is no problem with using that approach instead.

Import into Eclipse

Open Eclipse and import the 4 projects.

Complete Business Tier

Open `model.design` located in `src/main/resources` of the helloworld project.
Add something like this to the design file.

```
Application Universe {
    basePackage=org.helloworld

    Module milkyway {
        Entity Planet {
            scaffold
            String name key;
            String message;
            Integer diameter nullable;
            Integer population nullable;
            - Set<@Moon> moons opposite planet;
        }
        Entity Moon {
            not aggregateRoot // belongs to Planet Aggregate
            String name key;
            Integer diameter nullable;
            - @Planet planet opposite moons;
        }
    }
}
```

Note the `scaffold` feature of `Planet`. It will result in automatically generated CRUD operations in the `PlanetRepository` and `PlanetService`.

Build

1. Build the application with `mvn -Dmaven.test.skip=true clean install` from the helloworld-parent project. We skip the JUnit tests in this tutorial, see [Hello World Tutorial](#) for more information about the JUnit tests.
2. Refresh the 4 projects in Eclipse.
3. The resulting EAR file is located in `helloworld-ear/target/helloworld-ear.ear`.



Fornax Maven Launcher

You can checkout/import the Fornax Maven Launcher into the workspace to be able to run maven inside Eclipse. The menu items for this are available in the external tools menu.

Checkout <https://fornax.svn.sourceforge.net/svnroot/fornax/trunk/maven-launcher/>.
You also need to add a runtime variable in Eclipse, as described [here](#).

Setup Database

1. Run `helloworld/src/generated/resources/dbschema/Universe_ddl.sql` to create the tables in your [MySQL](#) database.
2. Add a mbean datasource in JBoss (`server/default/deploy/universe-mysql-ds.xml`).

```
<?xml version="1.0" encoding="UTF-8"?>
<datasources>
  <local-tx-datasource>
    <jndi-name>jdbc/UniverseDS</jndi-name>
    <connection-url>jdbc:mysql://localhost/helloworld</connection-url>
    <driver-class>com.mysql.jdbc.Driver</driver-class>
    <user-name>root</user-name>
    <password>root</password>
  </local-tx-datasource>
</datasources>
```

3. Copy `mysql-connector-jar` to JBoss lib directory (`server/default/lib/`)
It is likely that the jar is in your maven repository:

```
repository\mysql\mysql-connector-java\3.1.14\mysql-connector-java-3.1.14.jar
```

Deploy

1. You need to active the EJB3 support in JBoss. If you don't know how you can run the JEMS installer from <http://labs.jboss.com/jemsinstaller/downloads/> and select the EJB3 option.
2. JBoss 4.0.5 has an old version of Hibernate. You must replace `hibernate3.jar` in `server/default/lib` with `hibernate-3.2.5.ga.jar`, which you can find in your maven repository:

```
repository\org\hibernate\hibernate\3.2.5.ga\hibernate-3.2.5.ga.jar
```

3. Deploy the EAR file to JBoss. Copy it to `server/default/deploy/`
4. Start JBoss

Test

Open <http://localhost:8080/helloworld-web> in your browser.

Try the CRUD GUI and create some favourite Planets and Moons.

The features of the generated web application is explained in [CRUD GUI Tutorial](#).

Deployment as WAR without EJBs

Sculptor also support deployment in a web container without support for EJB, such as [Tomcat](#). [Developer's Guide](#) describes what configuration adjustments need to be done.

Hot Deploy

A short roundtrip is essential to achieve an efficient development environment. You need to be able to do quick hot deployment from your IDE. The generated projects contain Ant build files named `antbuild-ear.xml` and `antbuild-war.xml`. You can run these from inside Eclipse, right click and select `Run As > Ant Build`. But first you must define the location of your JBoss installation. Define the following Ant runtime property:

```
jboss.home=C:\JBoss-4.0.5
```

This setting is found in Eclipse: `Window > Preferences > Ant > Runtime > Properties`

Use `antbuild-ear.xml` if you are deploying ear file, and `antbuild-war.xml` if you are deploying war file.

Run the default target `deploy-copy` from the parent project. It will unzip the ear/war file to JBoss deploy directory. It will also copy your current class and configuration files from the target directory. This means that when you have done some changes you don't need to do a full `mvn install`. Eclipse has compiled classes to the target directory and `deploy-copy` will copy them to JBoss and do a hot redeploy, by touching some files.

When you have done changes to `model.design` you can do a quick generation by using `mvn generate-sources` and thereafter run `deploy-copy`.

Debugging

An easy way to be able to use the debugger is to install [JBoss IDE](#) and start JBoss in Eclipse with JBoss IDE.

Adding Dependencies

When you need to add dependencies to other jar files you do that by adding them to the maven pom files and thereafter run `mvn eclipse:eclipse` again. You must always run `mvn eclipse:eclipse` from the parent project.

Note that `eclipse:eclipse` also supports Eclipse project dependencies, as opposed to dependencies via jar files in the repository. In the above application the web project will have a project dependency to the business tier project. You can add other project dependencies by adding modules in the pom of the parent project and thereafter run `mvn eclipse:eclipse` again.

```
<module>../another-project</module>
```

EJB 2.1

Sculptor also supports EJB 2.1. To activate EJB 2.1 instead of EJB 3 you add this line to `sculptor-generator.properties`.

```
ejb.version=2.1
```

Sculptor will generate home and remote interface for each Service. It will also generate deployment descriptor. It will not generate any Service proxy class, since it is better to use Spring:

```
<jee:remote-slsb id="planetServiceProxy"  
  jndi-name="ejb/PlanetService"  
  lookup-home-on-startup="true"  
  refresh-home-on-connect-failure="true"  
  business-interface="org.helloworld.milkyway.serviceapi.PlanetService"/>
```

Source

The complete source code for this tutorial is available in Subversion.

Web Access (read only):

- <http://fisheye3.cenqua.com/browse/fornax/trunk/cartridges/sculptor/sculptor-helloworld-parent>
- <http://fisheye3.cenqua.com/browse/fornax/trunk/cartridges/sculptor/sculptor-helloworld>
- <http://fisheye3.cenqua.com/browse/fornax/trunk/cartridges/sculptor/sculptor-helloworld-web>
- <http://fisheye3.cenqua.com/browse/fornax/trunk/cartridges/sculptor/sculptor-helloworld-ear>

Anonymous Access (read only):

- <https://fornax.svn.sourceforge.net/svnroot/fornax/trunk/cartridges/sculptor/sculptor-helloworld-parent>
- <https://fornax.svn.sourceforge.net/svnroot/fornax/trunk/cartridges/sculptor/sculptor-helloworld>
- <https://fornax.svn.sourceforge.net/svnroot/fornax/trunk/cartridges/sculptor/sculptor-helloworld-web>
- <https://fornax.svn.sourceforge.net/svnroot/fornax/trunk/cartridges/sculptor/sculptor-helloworld-ear>

