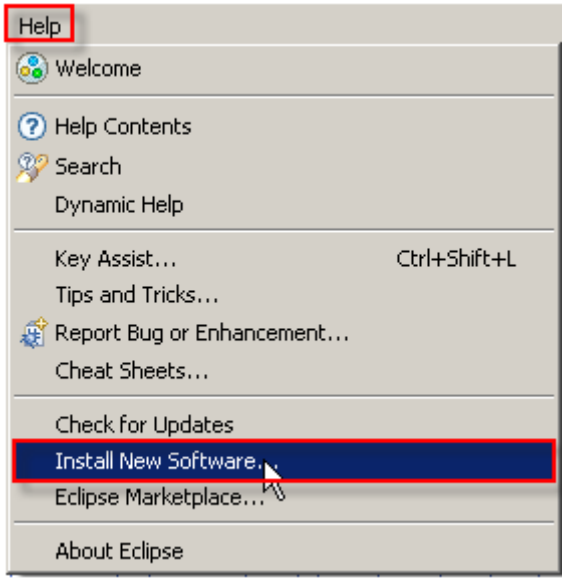
Atelier Hibernate avec Eclipse

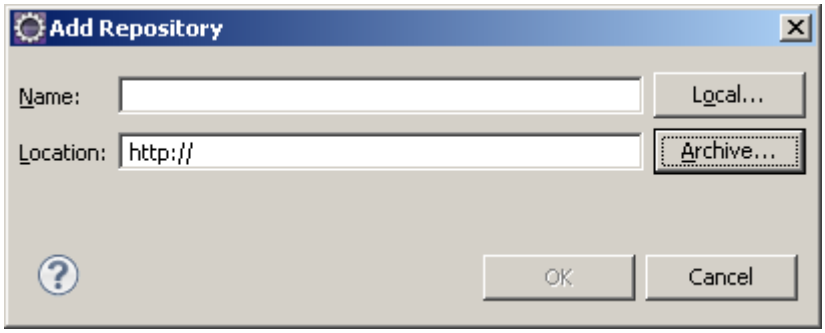
Installation Plug-in

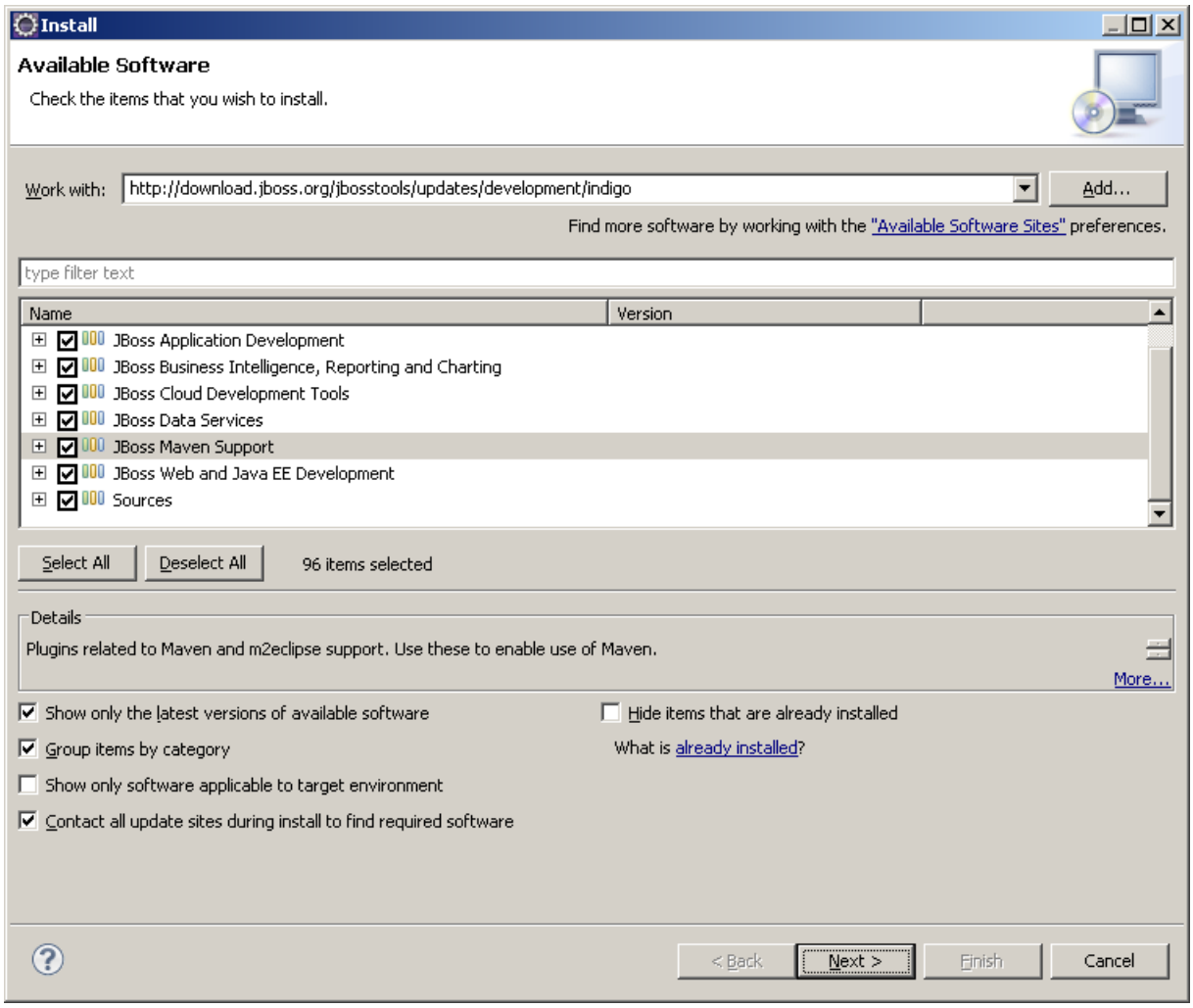
Hibernate est un outil ORM largement utilisé dans la communauté Java pour conserver l'objet Java en utilisant le concept de Relational Mapping objet (ORM). ORM réduit le nombre de lignes d'interagir avec les bases de données avec le langage de requête optimisé qui est **Hibernate langage de requêtes (HQL)** .

Dans cet exemple, nous allons créer une application de connexion simple en utilisant l'outil Hibernate et l'IDE Eclipse. Nous allons utiliser **Eclipse Indigo** (Web Tools Platform), pour installer "Hibernate Tools". Suivre les étapes suivantes:

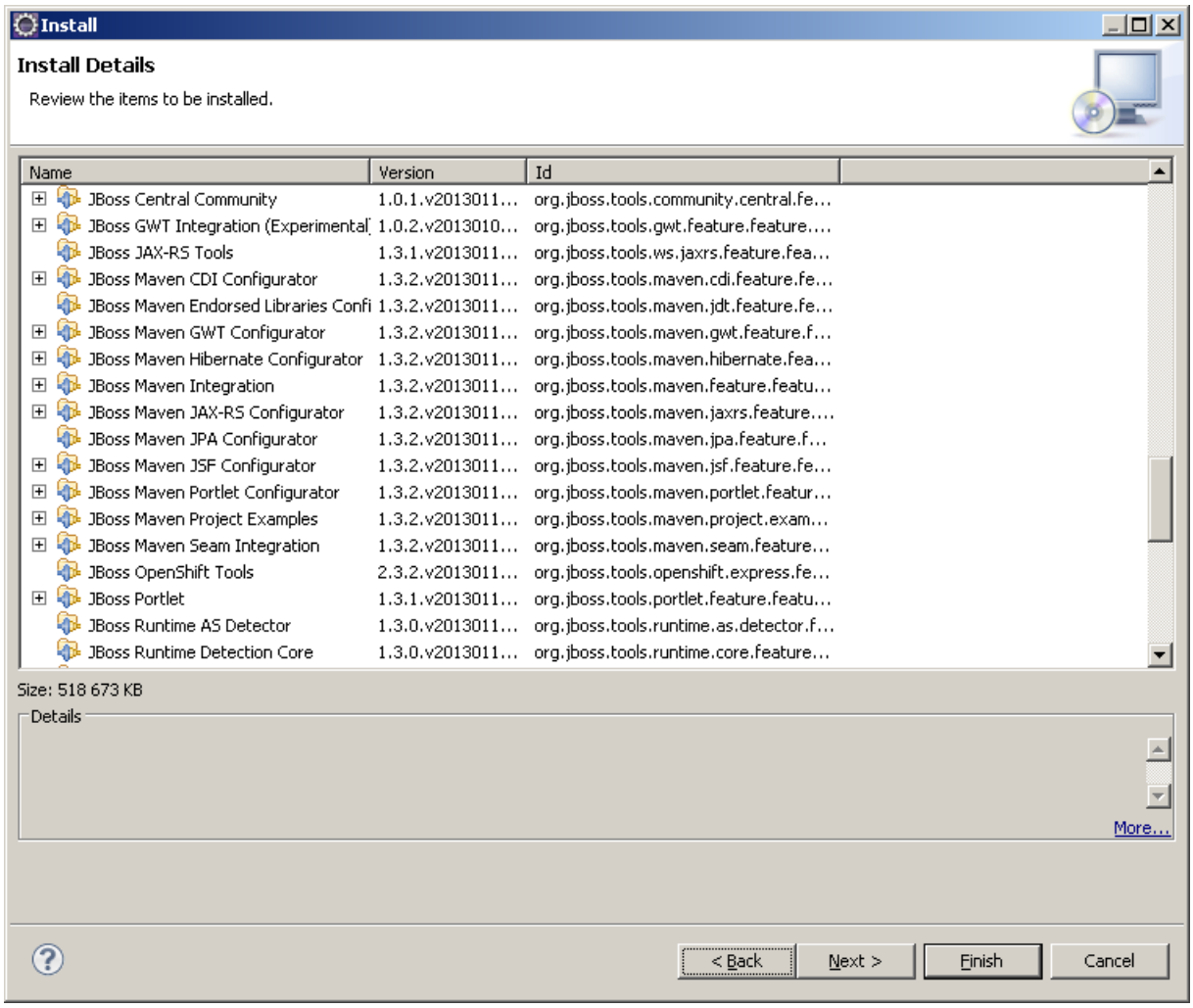
Dans Eclipse IDE, barre de menu, sélectionnez **"help" --> "Install new software ..."** mettre l' URL *"http://download.jboss.org/jbosstools/updates/stable/indigo "*

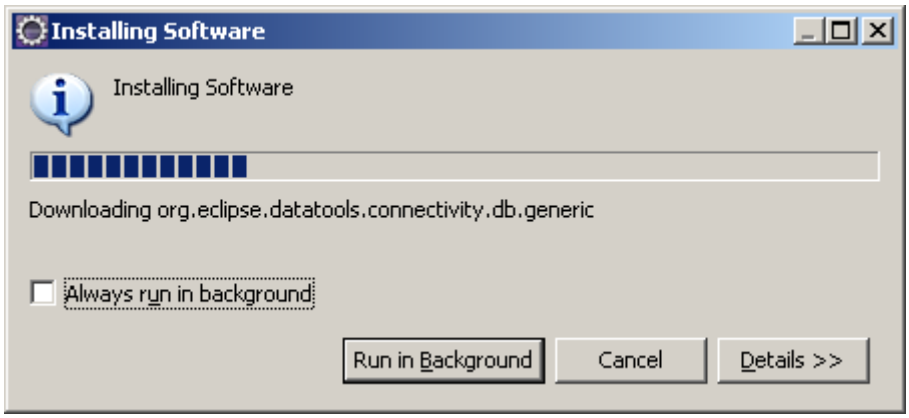


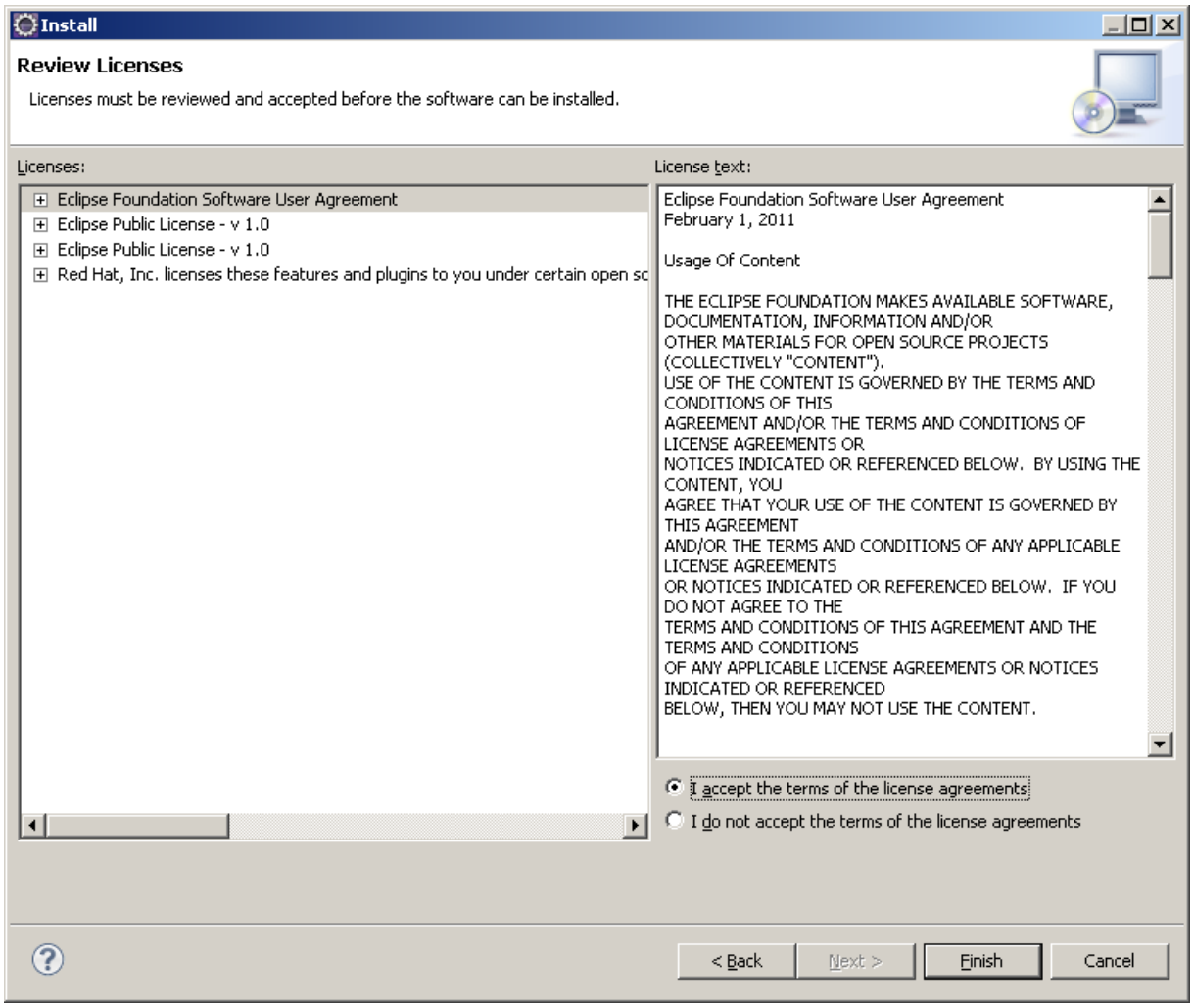




Vous pouvez sélectionner tout ou bien sélectionner Hibernate Tools Uniquement.


Install
Install Details
Review the items to be installed.
Name
Version
Id
JBoss Central Community 1.0.1.v2013011... org.jboss.tools.community.central.fe...
JBoss GWT Integration (Experimental 1.0.2.v2013010... org.jboss.tools.gwt.feature.feature....
JBoss JAX-RS Tools 1.3.1.v2013011... org.jboss.tools.ws.jaxrs.feature.fea...
JBoss Maven CDI Configurator 1.3.2.v2013011... org.jboss.tools.maven.cdi.feature.fe...
JBoss Maven Endorsed Libraries Confi 1.3.2.v2013011... org.jboss.tools.maven.jdt.feature.fe...
JBoss Maven GWT Configurator 1.3.2.v2013011... org.jboss.tools.maven.gwt.feature.f...
JBoss Maven Hibernate Configurator 1.3.2.v2013011... org.jboss.tools.maven.hibernate.fea...
JBoss Maven Integration 1.3.2.v2013011... org.jboss.tools.maven.feature.featu...
JBoss Maven JAX-RS Configurator 1.3.2.v2013011... org.jboss.tools.maven.jaxrs.feature....
JBoss Maven JPA Configurator 1.3.2.v2013011... org.jboss.tools.maven.jpa.feature.f...
JBoss Maven JSF Configurator 1.3.2.v2013011... org.jboss.tools.maven.jsf.feature.fe...
JBoss Maven Portlet Configurator 1.3.2.v2013011... org.jboss.tools.maven.portlet.featur...
JBoss Maven Project Examples 1.3.2.v2013011... org.jboss.tools.maven.project.exam...
JBoss Maven Seam Integration 1.3.2.v2013011... org.jboss.tools.maven.seam.feature...
JBoss OpenShift Tools 2.3.2.v2013011... org.jboss.tools.openshift.express.fe...
JBoss Portlet 1.3.1.v2013011... org.jboss.tools.portlet.feature.featu...
JBoss Runtime AS Detector 1.3.0.v2013011... org.jboss.tools.runtime.as.detector.f...
JBoss Runtime Detection Core 1.3.0.v2013011... org.jboss.tools.runtime.core.feature...
Size: 518 673 KB
Details
More...
< Back
Next >
Finish
Cancel

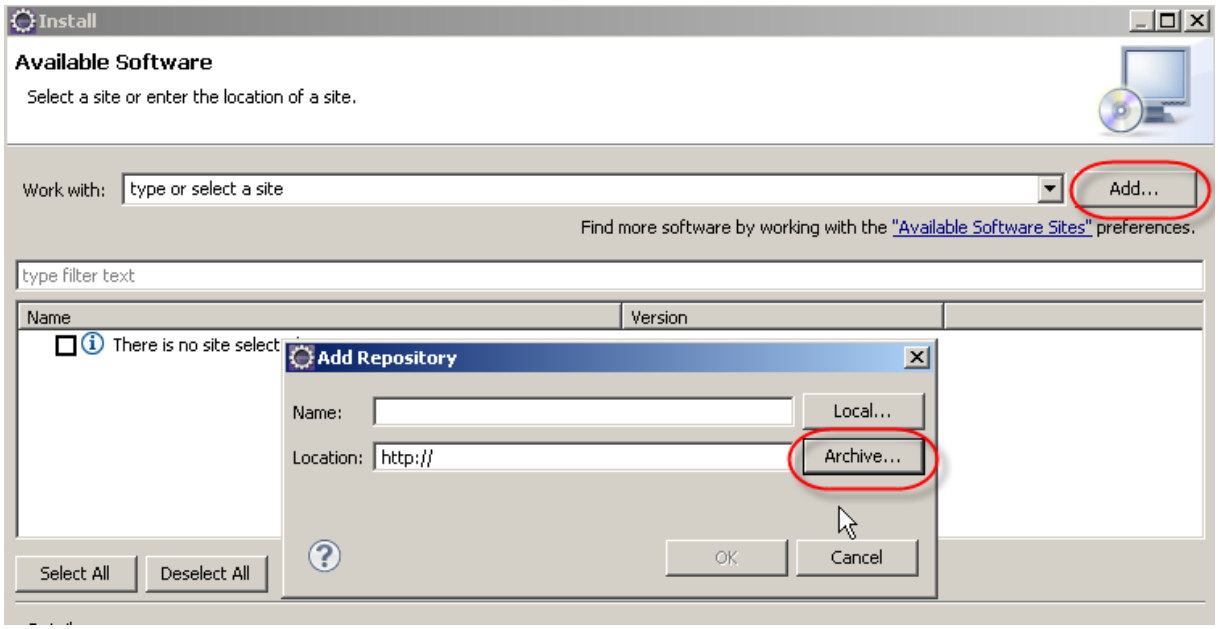

Installing Software
Installing Software
Downloading org.eclipse.datatools.connectivity.db.generic
Always run in background
Run in Background
Cancel
Details >>

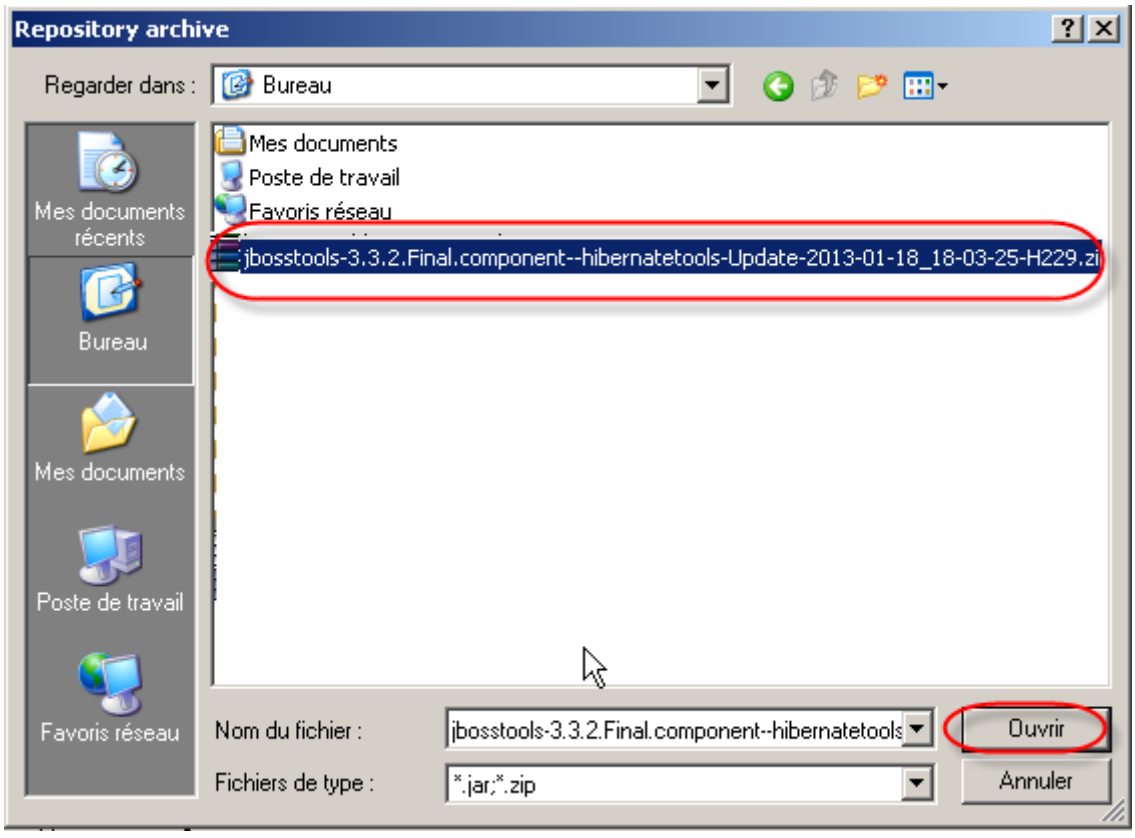


Après l'installation, redémarrez l'éclipse.

Installation Plug-in off-line (hors ligne)

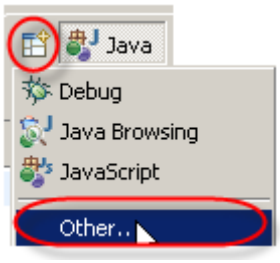
Si vous n'avez pas la connexion Internet et que vous voulez le mode hors ligne pour ajouter des outils Hibernate dans Eclipse. Pour installer les outils Hibernate, télécharger, à partir de l'URL suivante : http://freefr.dl.sourceforge.net/project/jboss/JBossTools/JBossTools3.3.x/, le fichier **jbosstools-3.3.2.Final.component--hibernatetools-Update-2013-01-18_18-03-25-H229.zip**.

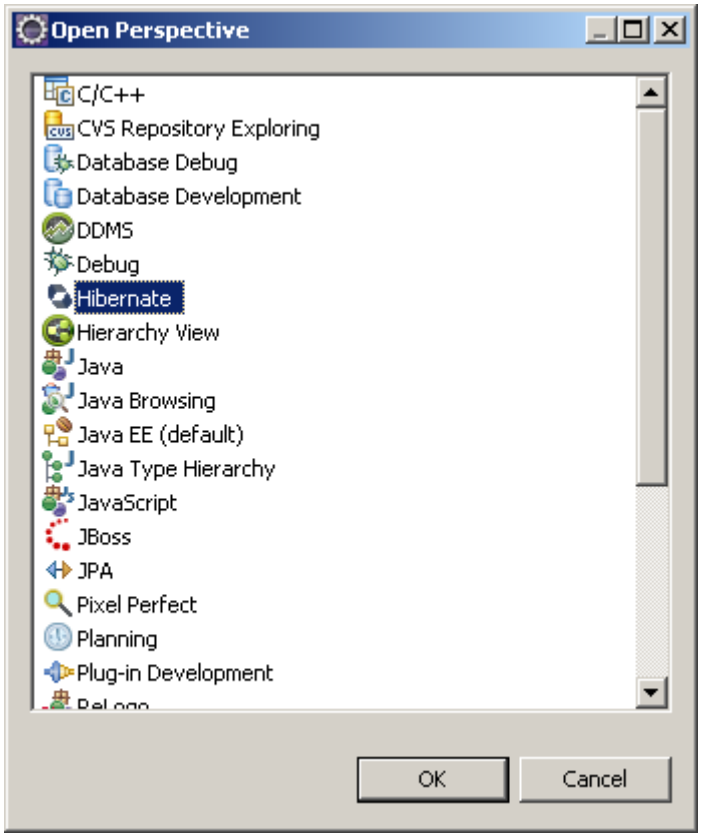




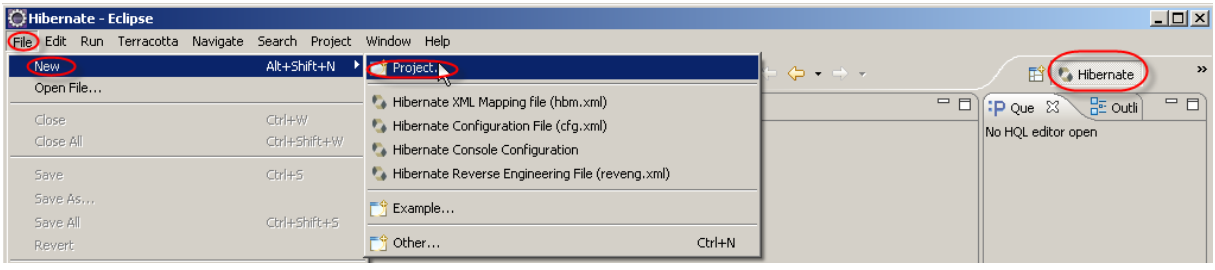
Après le redémarrage, Pour travailler avec la perspective Hibernate :

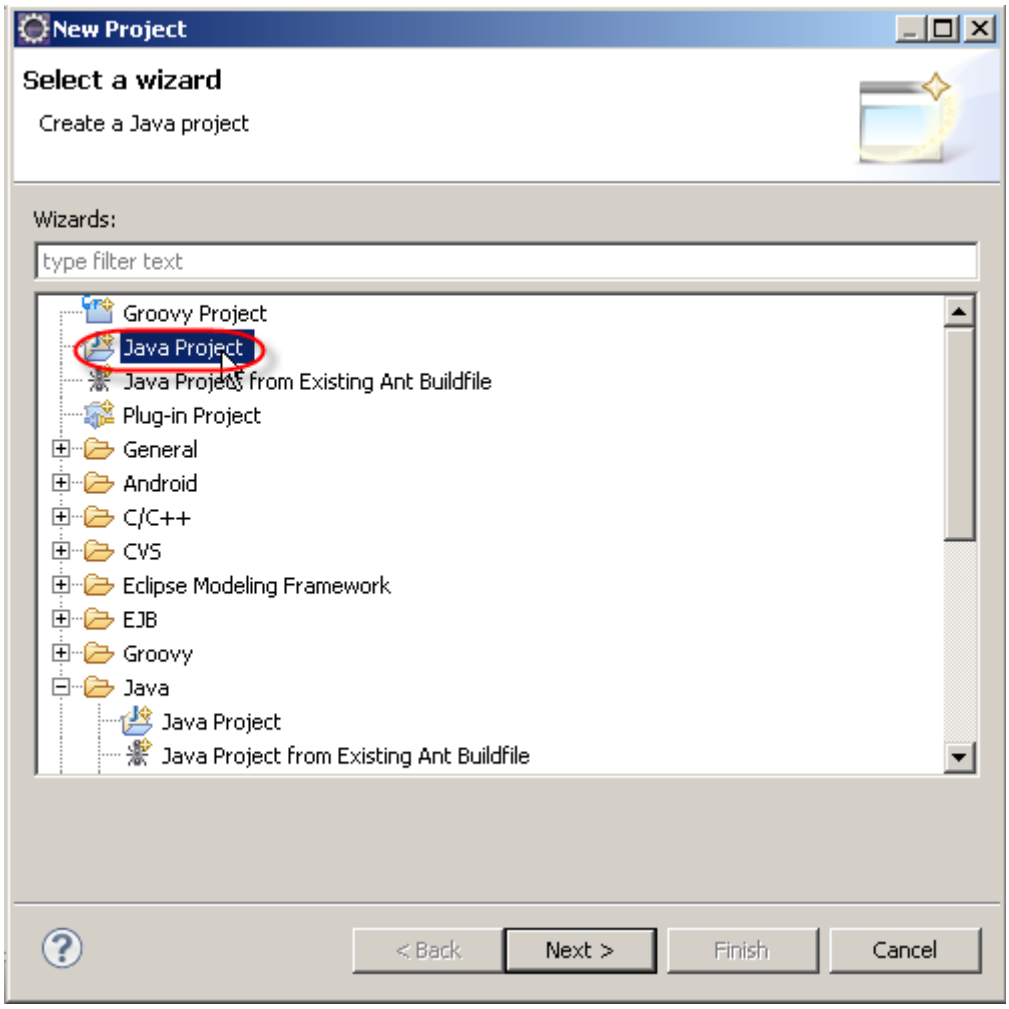
Allez à **Window | Open Perspective | Other**, la boîte de dialogue suivante apparaît, sélectionnez Hibernate et cliquez sur le bouton OK.

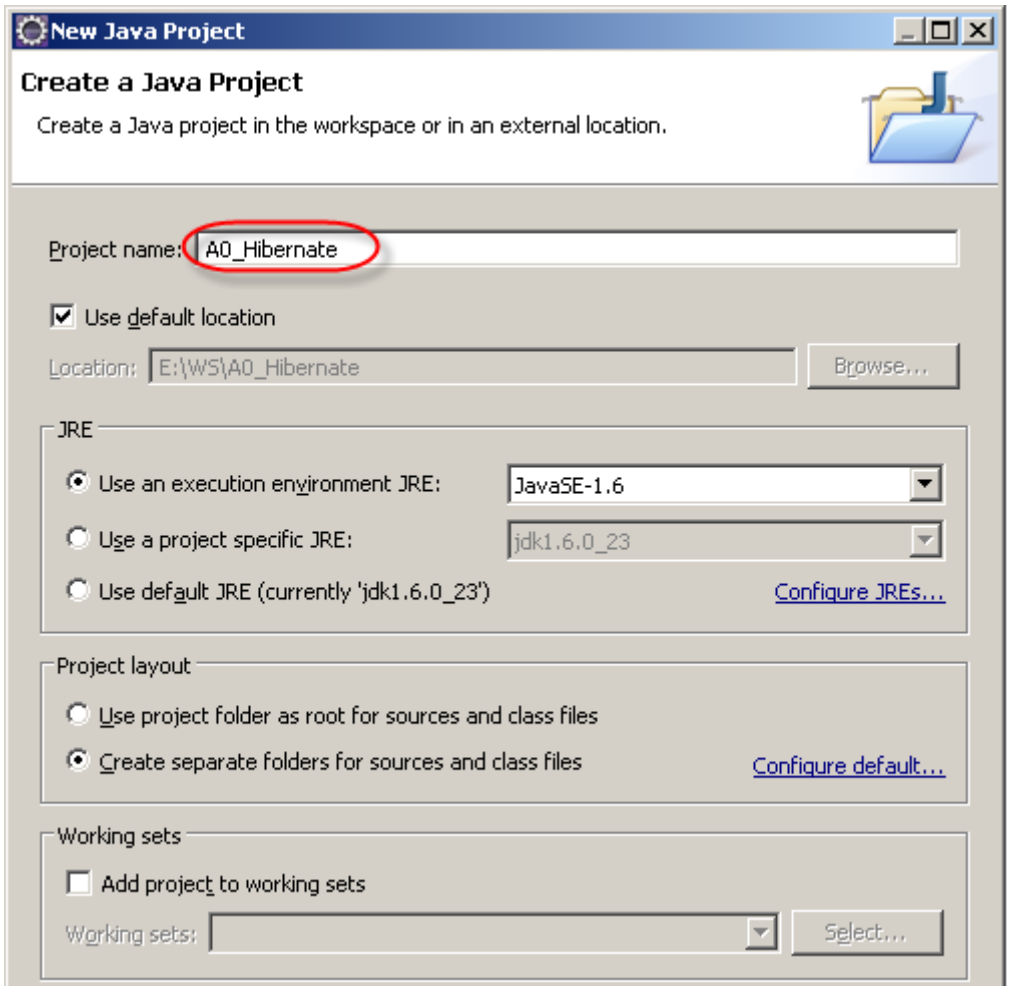


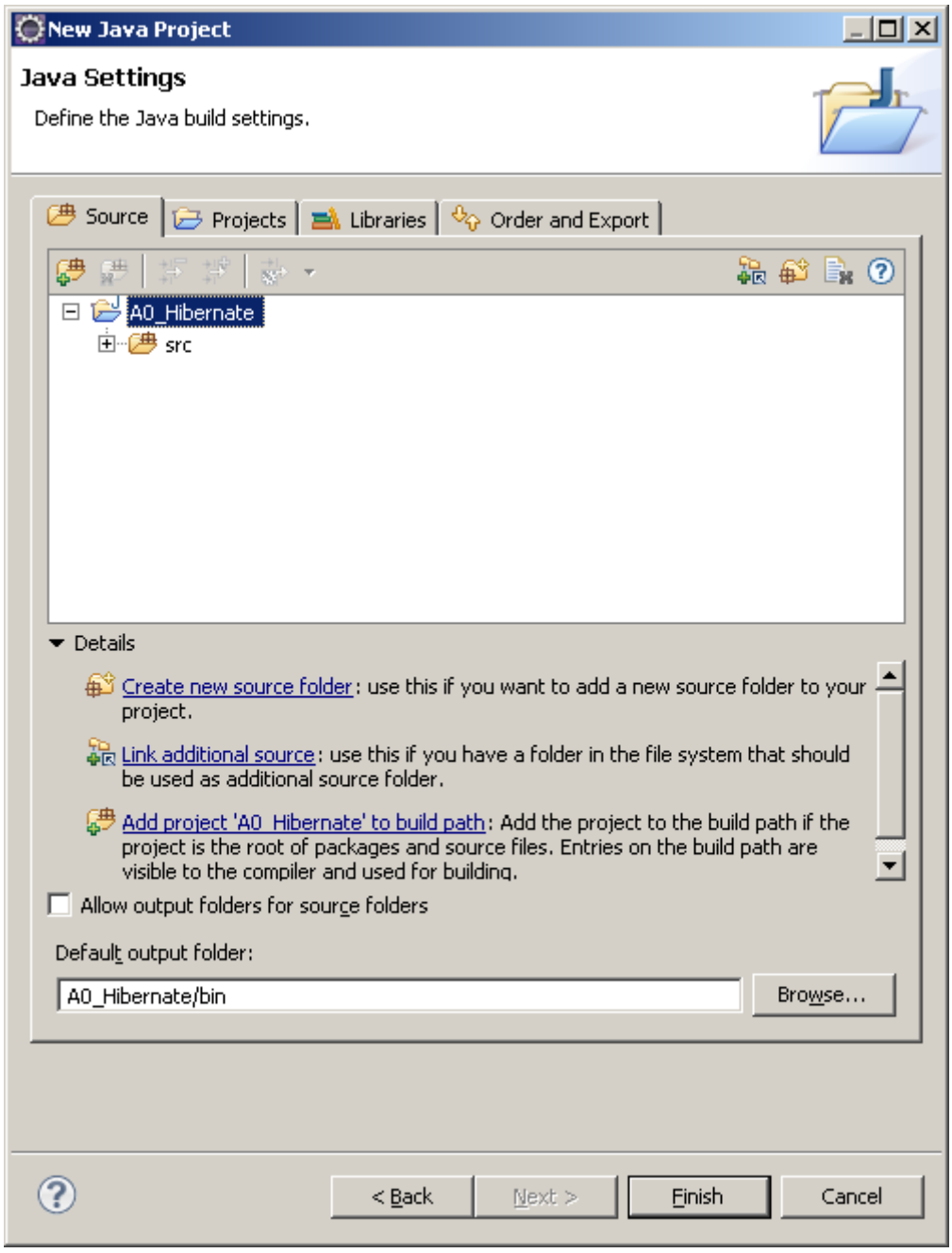


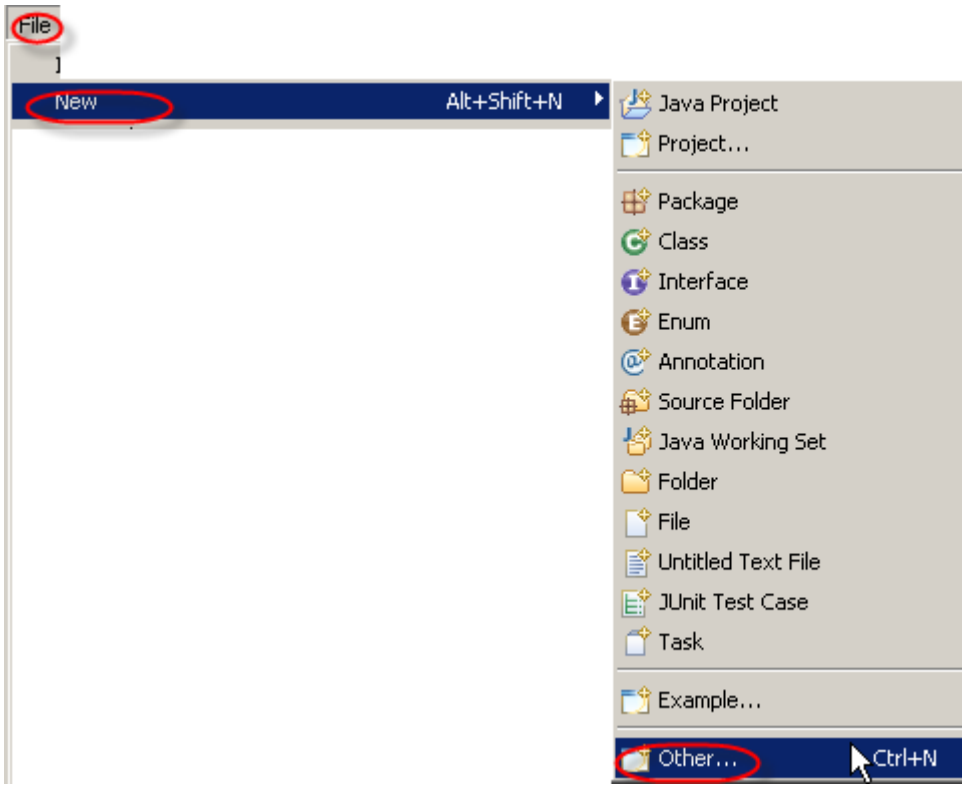
Création Projet Java



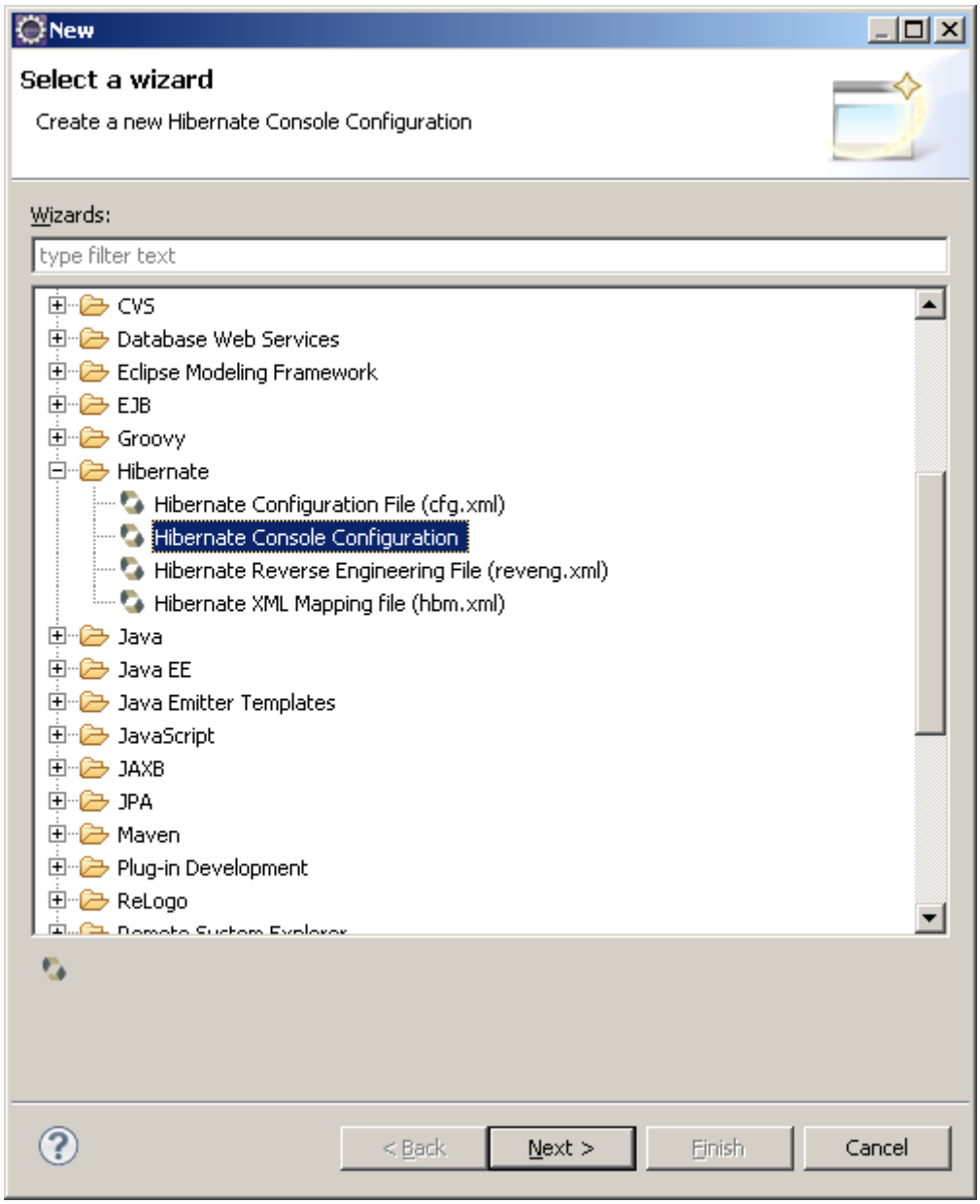

New Project
Select a wizard
Create a Java project
Wizards:
type filter text
Groovy Project
Java Project
Java Project from Existing Ant Buildfile
Plug-in Project
General
Android
C/C++
CVS
Eclipse Modeling Framework
EJB
Groovy
Java
Java Project
Java Project from Existing Ant Buildfile
< Back
Next >
Finish
Cancel

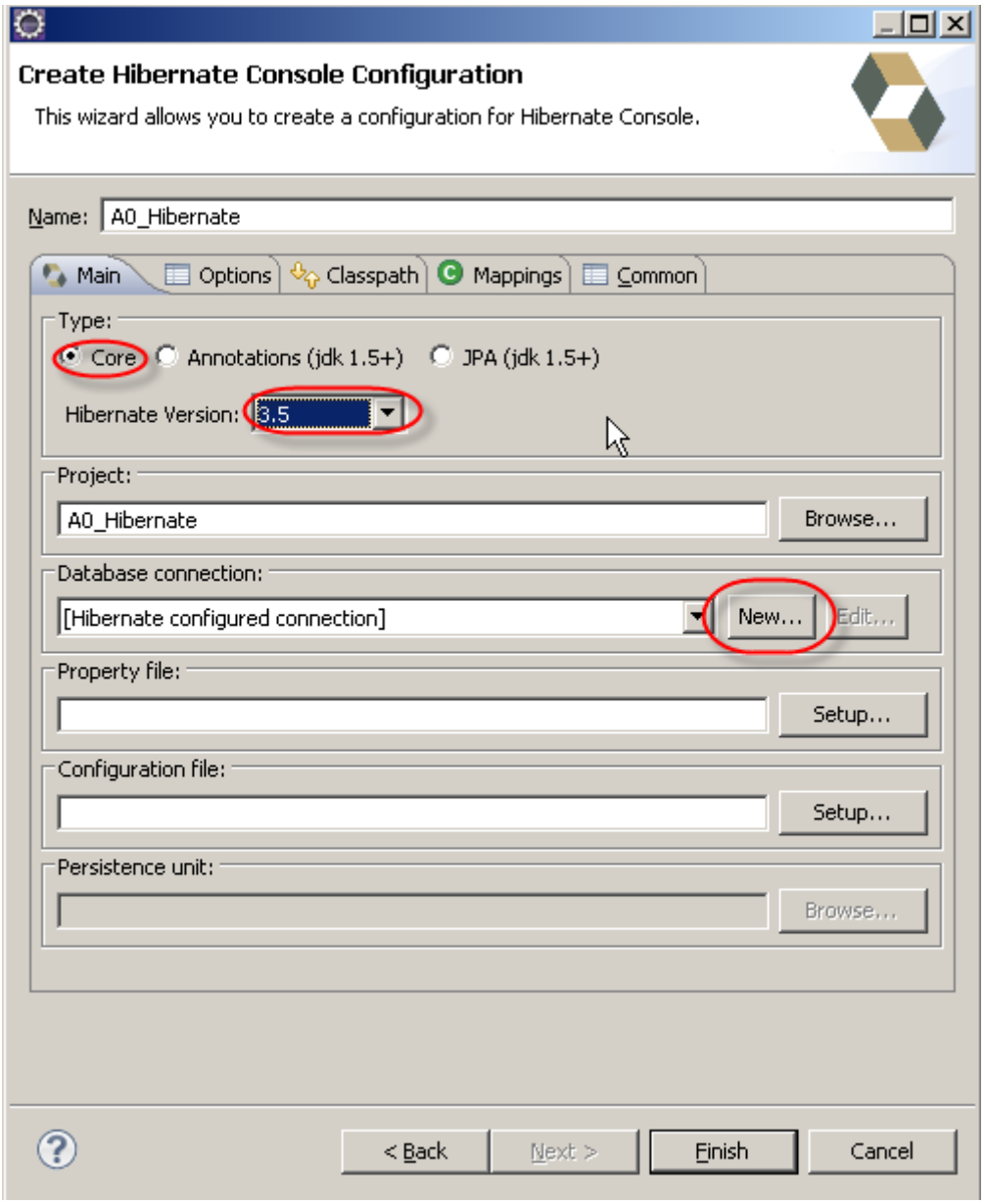

New Java Project
Create a Java Project
Create a Java project in the workspace or in an external location.
Project name: A0_Hibernate
Use default location
Location: E:\WS\A0_Hibernate
Browse...
JRE
Use an execution environment JRE: JavaSE-1.6
Use a project specific JRE: jdk1.6.0_23
Use default JRE (currently 'jdk1.6.0_23')
Configure JREs...
Project layout
Use project folder as root for sources and class files
Create separate folders for sources and class files
Configure default...
Working sets
Add project to working sets
Working sets:
Select...


New Java Project
Java Settings
Define the Java build settings.
Source
Projects
Libraries
Order and Export
A0_Hibernate
src
Details
Create new source folder: use this if you want to add a new source folder to your project.
Link additional source: use this if you have a folder in the file system that should be used as additional source folder.
Add project 'A0_Hibernate' to build path: Add the project to the build path if the project is the root of packages and source files. Entries on the build path are visible to the compiler and used for building.
Allow output folders for source folders
Default output folder:
A0_Hibernate/bin
Browse...
< Back
Next >
Finish
Cancel

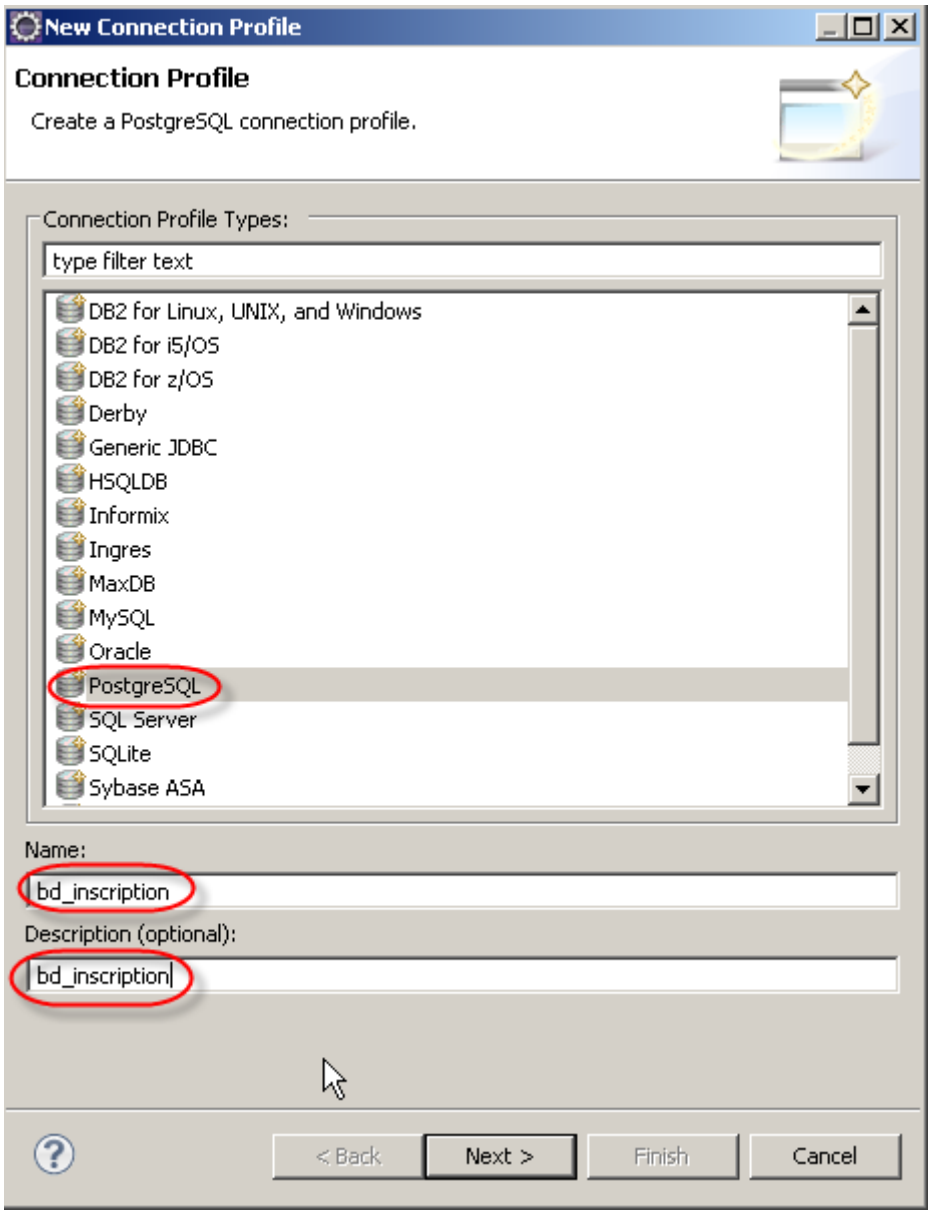


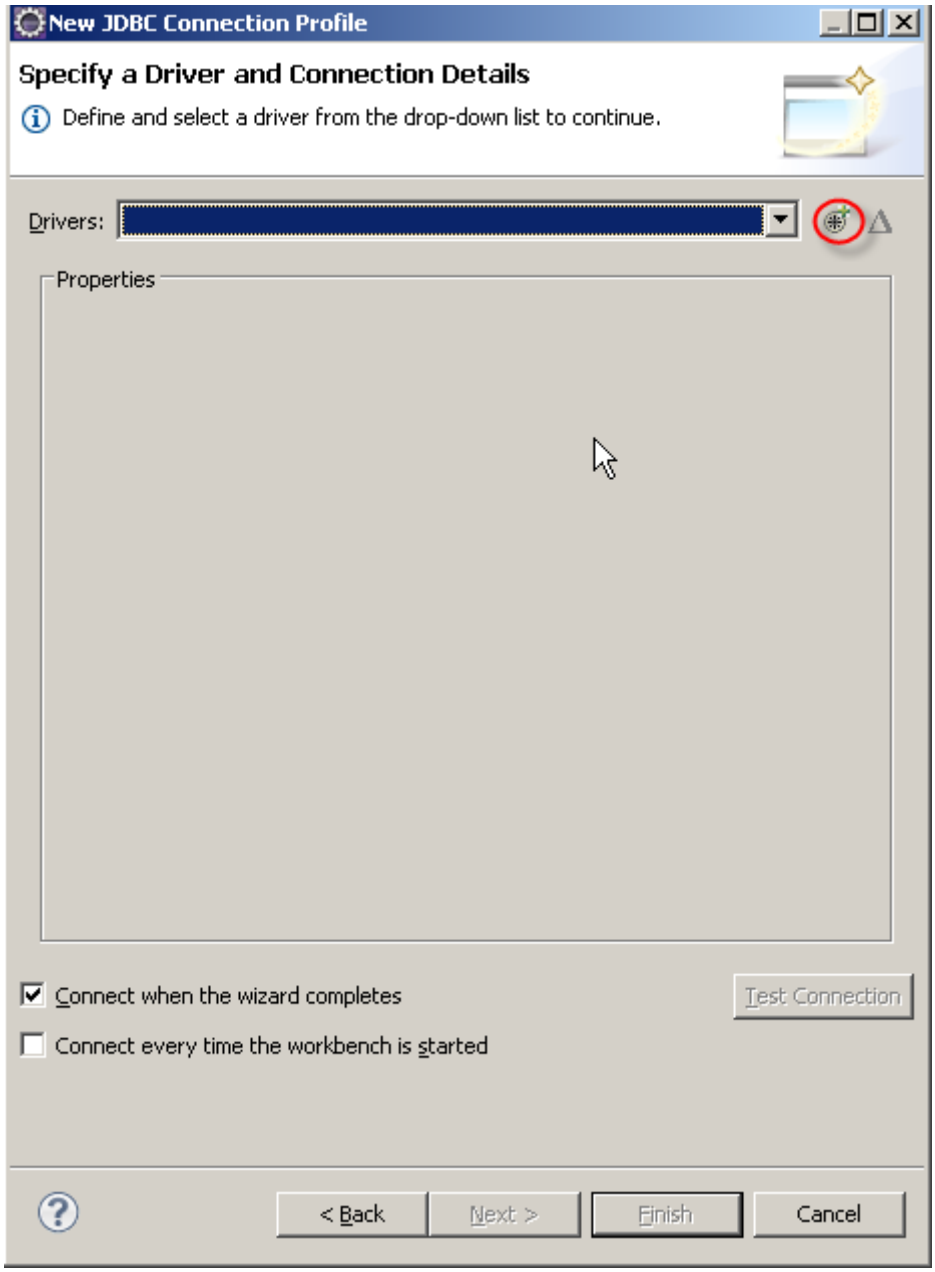
Création Console configuration

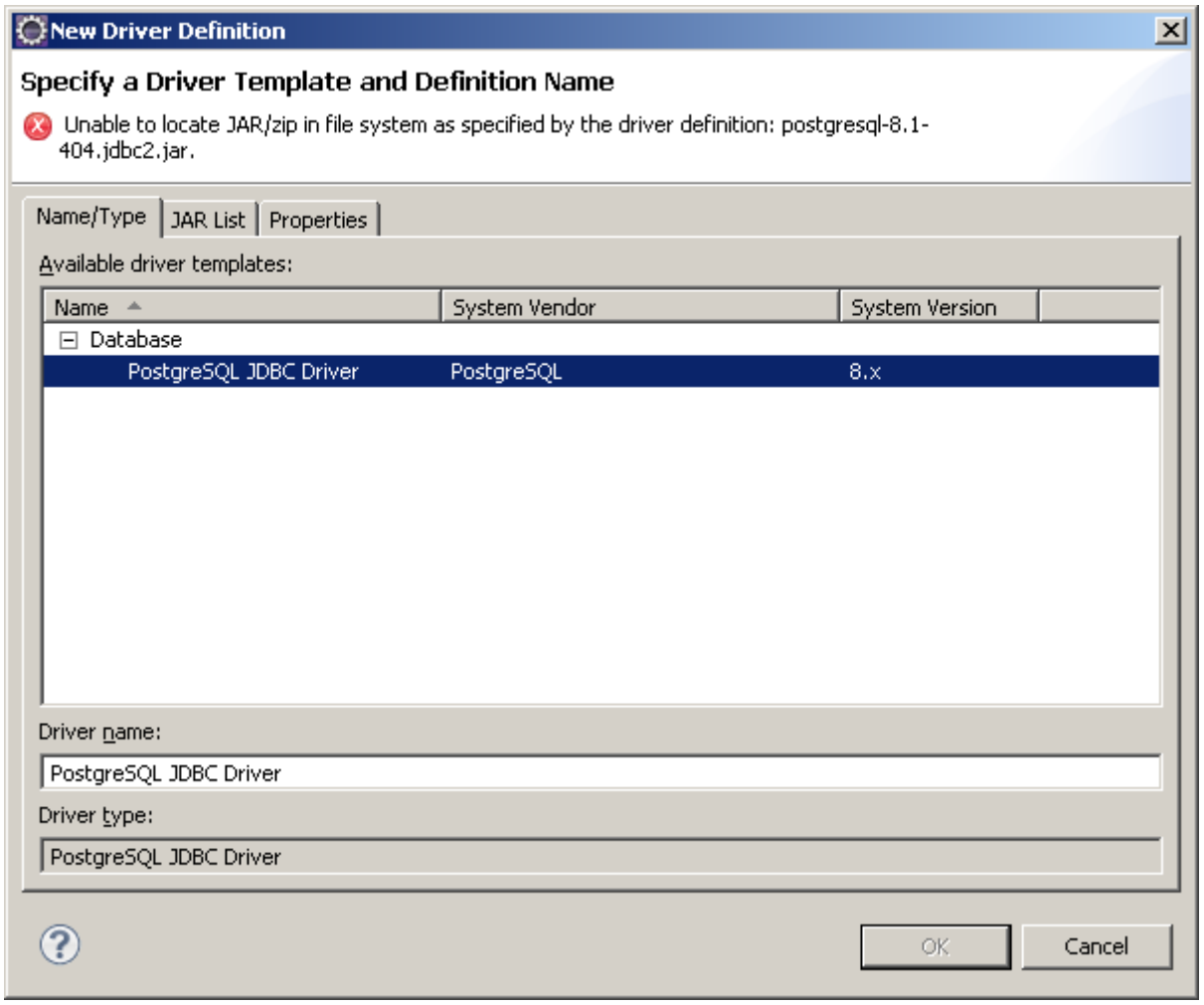

New
Select a wizard
Create a new Hibernate Console Configuration
Wizards:
type filter text
CVS
Database Web Services
Eclipse Modeling Framework
EJB
Groovy
Hibernate
Hibernate Configuration File (cfg.xml)
Hibernate Console Configuration
Hibernate Reverse Engineering File (reveng.xml)
Hibernate XML Mapping file (hbm.xml)
Java
Java EE
Java Emitter Templates
JavaScript
JAXB
JPA
Maven
Plug-in Development
ReLogo
< Back
Next >
Finish
Cancel

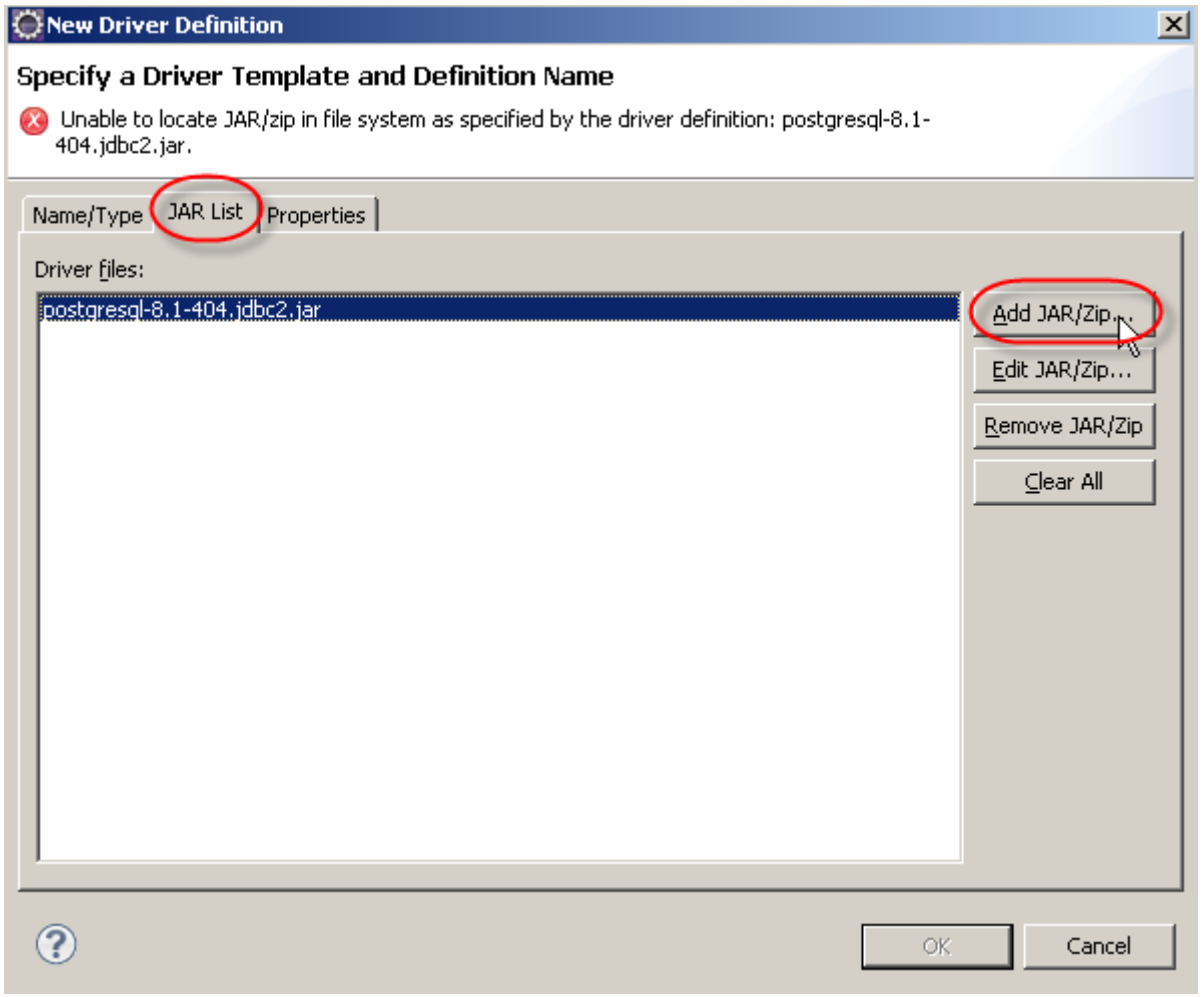


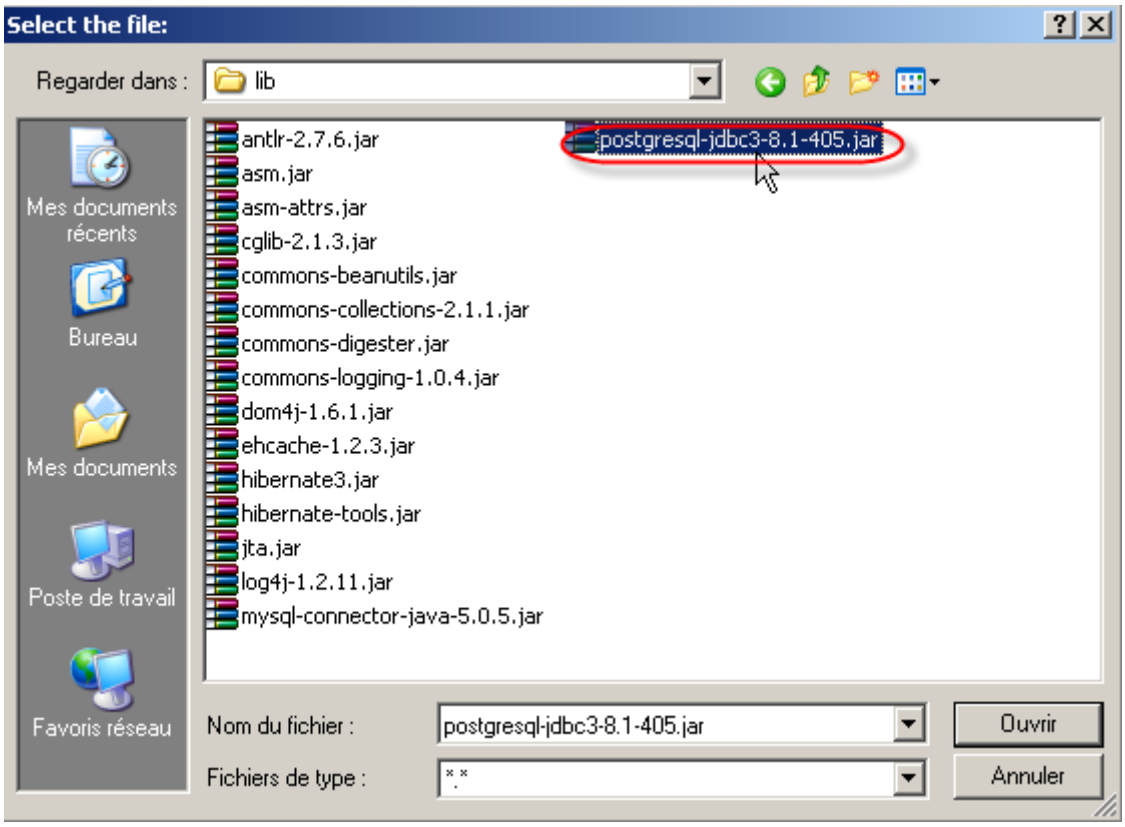
On doit cliquer sur le bouton « New »

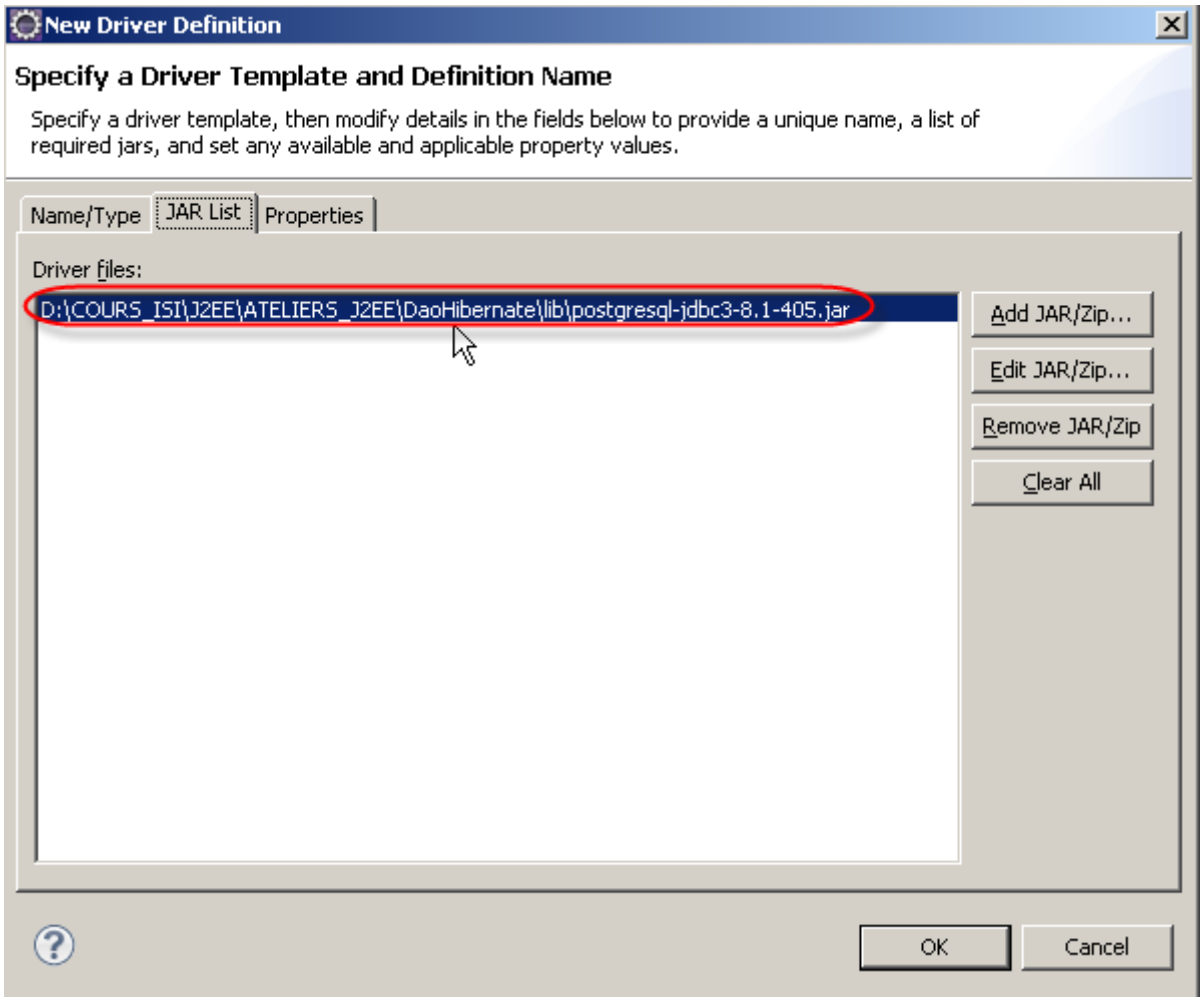

New Connection Profile
Connection Profile
Create a PostgreSQL connection profile.
Connection Profile Types:
type filter text
DB2 for Linux, UNIX, and Windows
DB2 for i5/OS
DB2 for z/OS
Derby
Generic JDBC
HSQLDB
Informix
Ingres
MaxDB
MySQL
Oracle
PostgreSQL
SQL Server
SQLite
Sybase ASA
Name:
bd_inscription
Description (optional):
bd_inscription
< Back
Next >
Finish
Cancel

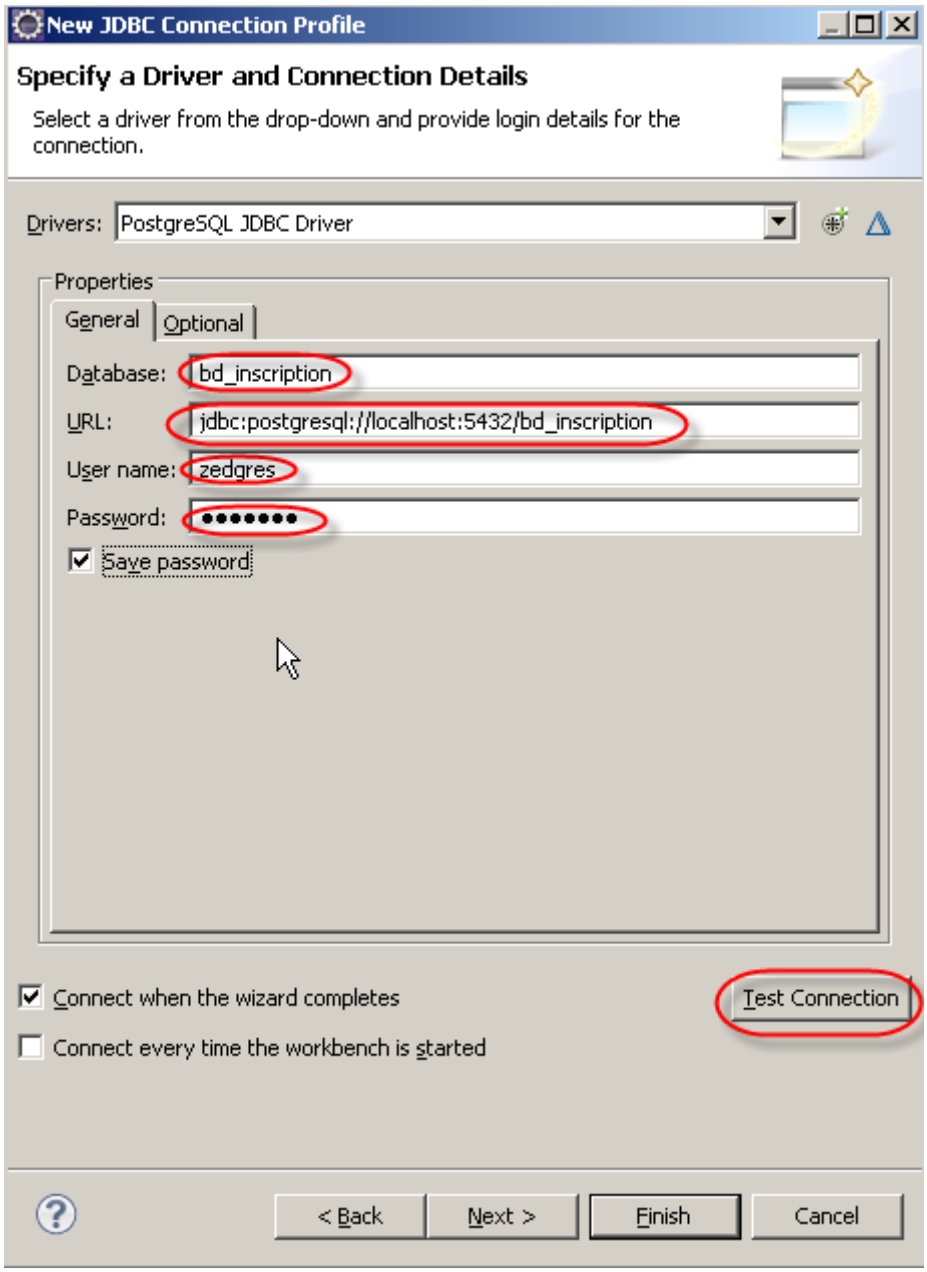

New JDBC Connection Profile
Specify a Driver and Connection Details
Define and select a driver from the drop-down list to continue.
Drivers:
Properties
Connect when the wizard completes
Test Connection
Connect every time the workbench is started
< Back
Next >
Finish
Cancel


New Driver Definition
Specify a Driver Template and Definition Name
Unable to locate JAR/zip in file system as specified by the driver definition: postgresql-8.1-404.jdbc2.jar.
Name/Type
JAR List
Properties
Available driver templates:
Name
System Vendor
System Version
Database
PostgreSQL JDBC Driver
PostgreSQL
8.x
Driver name:
PostgreSQL JDBC Driver
Driver type:
PostgreSQL JDBC Driver
OK
Cancel

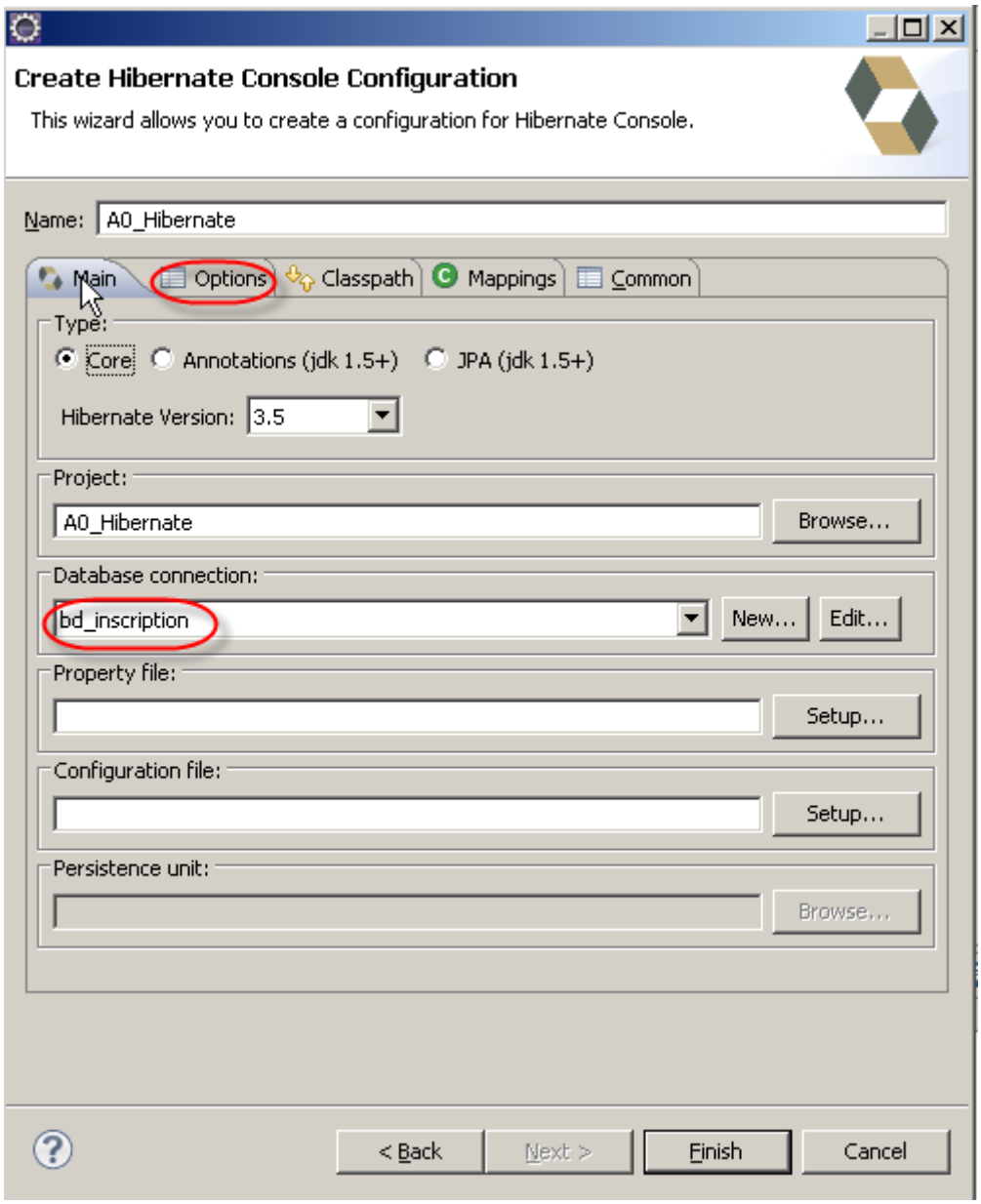

New Driver Definition
Specify a Driver Template and Definition Name
Unable to locate JAR/zip in file system as specified by the driver definition: postgresql-8.1-404.jdbc2.jar.
Name/Type
JAR List
Properties
Driver files:
postgresql-8.1-404.jdbc2.jar
Add JAR/Zip...
Edit JAR/Zip...
Remove JAR/Zip
Clear All
OK
Cancel

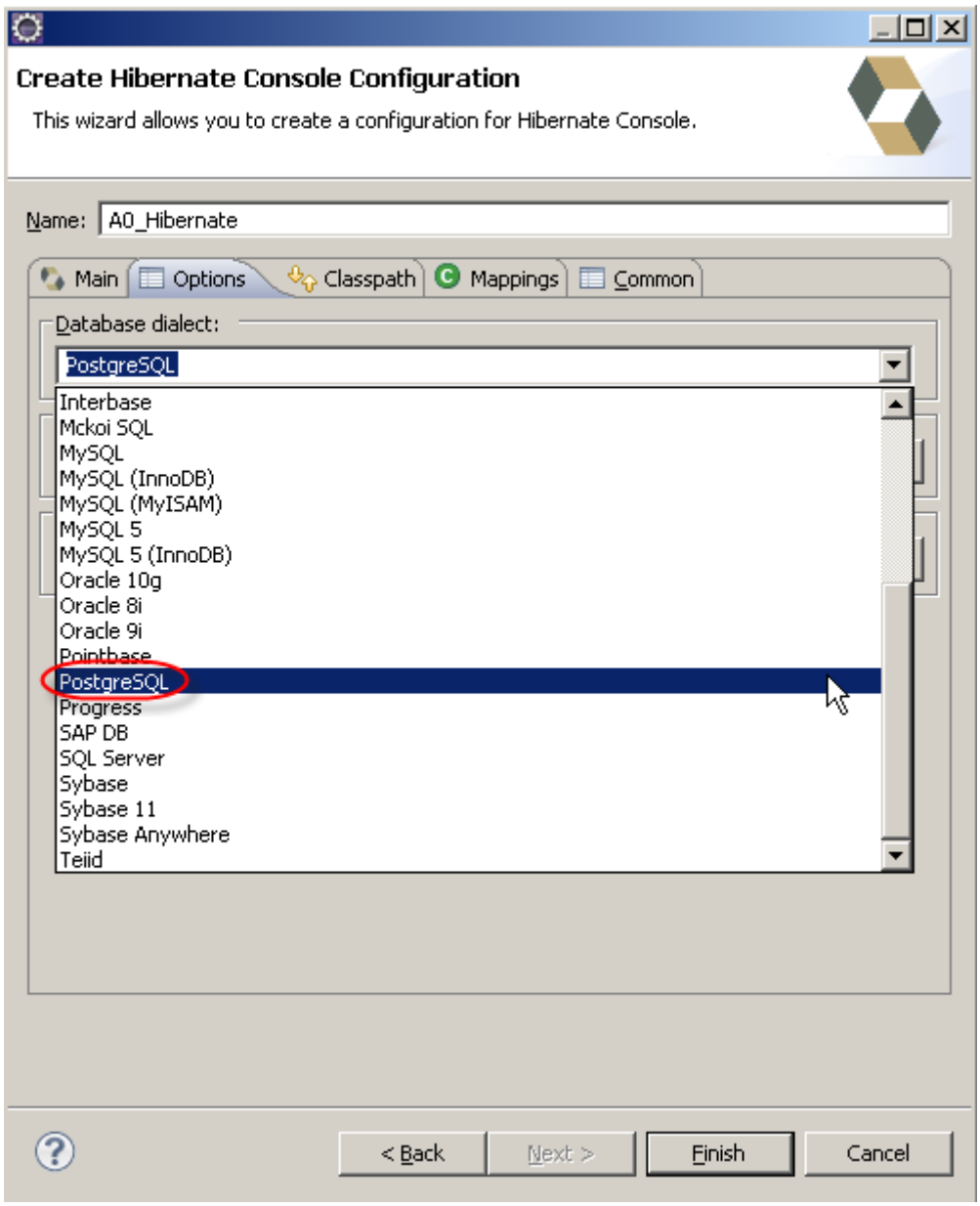

Select the file:
Regarder dans :
lib
Mes documents récents
Bureau
Mes documents
Poste de travail
Favoris réseau
antlr-2.7.6.jar
asm.jar
asm-attrs.jar
cglib-2.1.3.jar
commons-beanutils.jar
commons-collections-2.1.1.jar
commons-digester.jar
commons-logging-1.0.4.jar
dom4j-1.6.1.jar
ehcache-1.2.3.jar
hibernate3.jar
hibernate-tools.jar
jta.jar
log4j-1.2.11.jar
mysql-connector-java-5.0.5.jar
postgresql-jdbc3-8.1-405.jar
Nom du fichier :
postgresql-jdbc3-8.1-405.jar
Fichiers de type :
.
Ouvrir
Annuler


New Driver Definition
Specify a Driver Template and Definition Name
Specify a driver template, then modify details in the fields below to provide a unique name, a list of required jars, and set any available and applicable property values.
Name/Type
JAR List
Properties
Driver files:
D:\COURS_ISI\J2EE\ATELIERS_J2EE\DaoHibernate\lib\postgresql-jdbc3-8.1-405.jar
Add JAR/Zip...
Edit JAR/Zip...
Remove JAR/Zip
Clear All
OK
Cancel

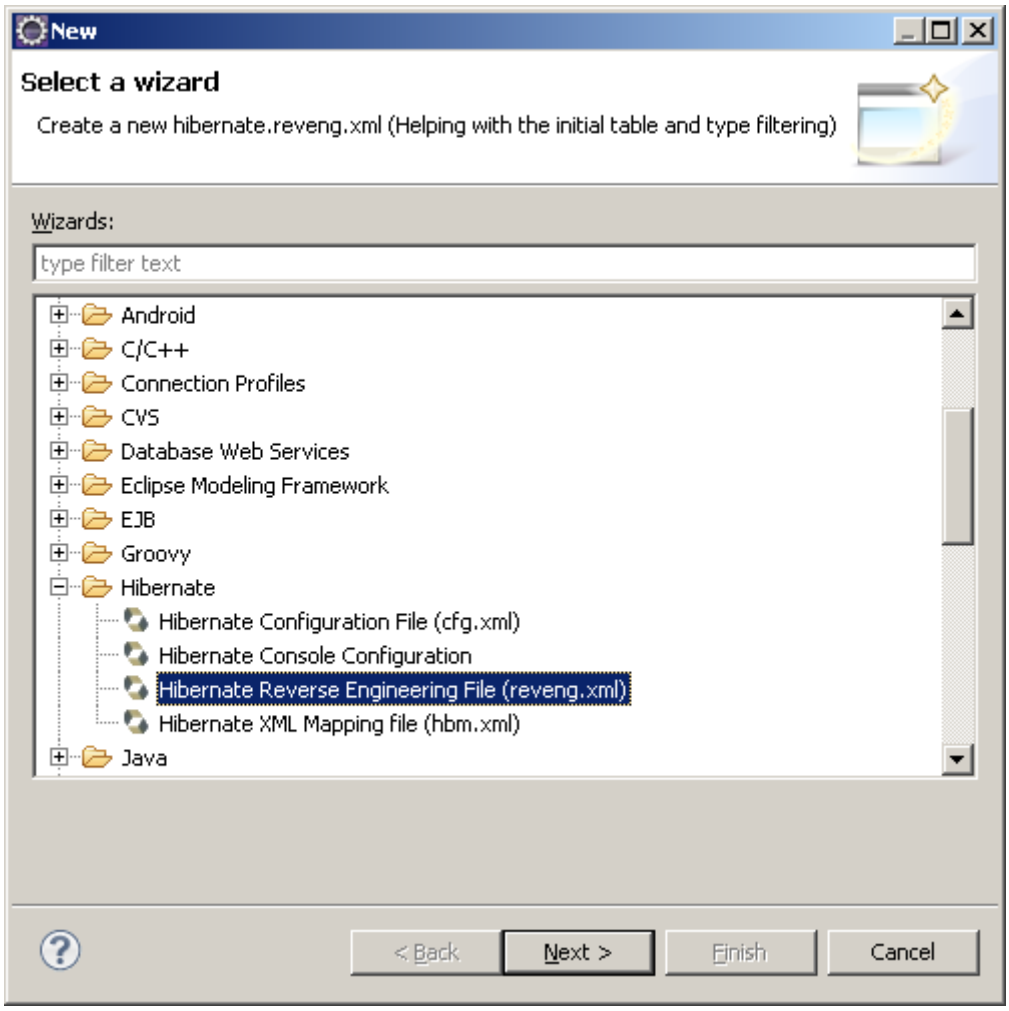

New JDBC Connection Profile
Specify a Driver and Connection Details
Select a driver from the drop-down and provide login details for the connection.
Drivers:
PostgreSQL JDBC Driver
Properties
General
Optional
Database:
bd_inscription
URL:
jdbc:postgresql://localhost:5432/bd_inscription
User name:
zedgres
Password:
Save password
Connect when the wizard completes
Connect every time the workbench is started
Test Connection
< Back
Next >
Finish
Cancel

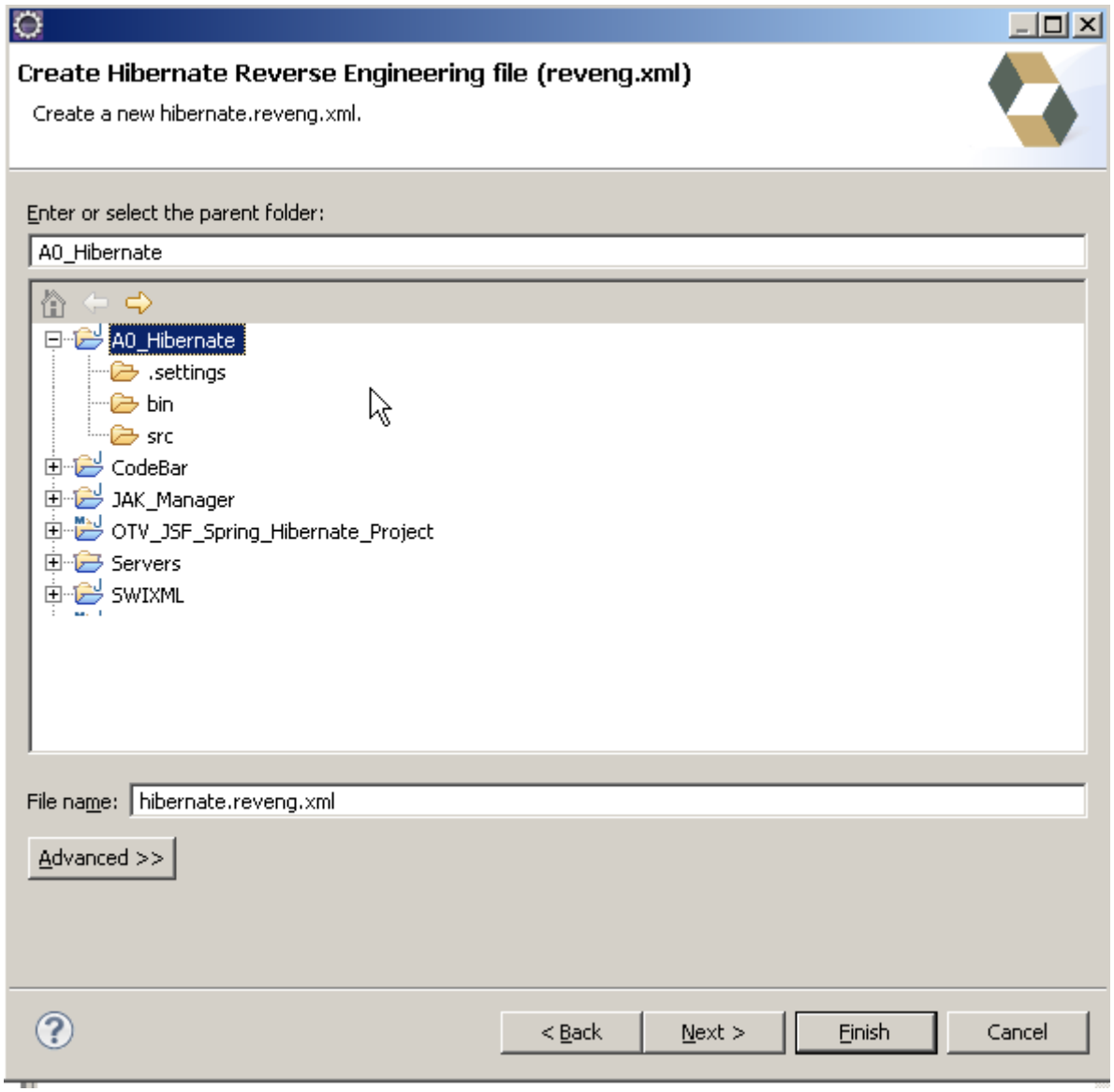

Success
Ping succeeded!
OK

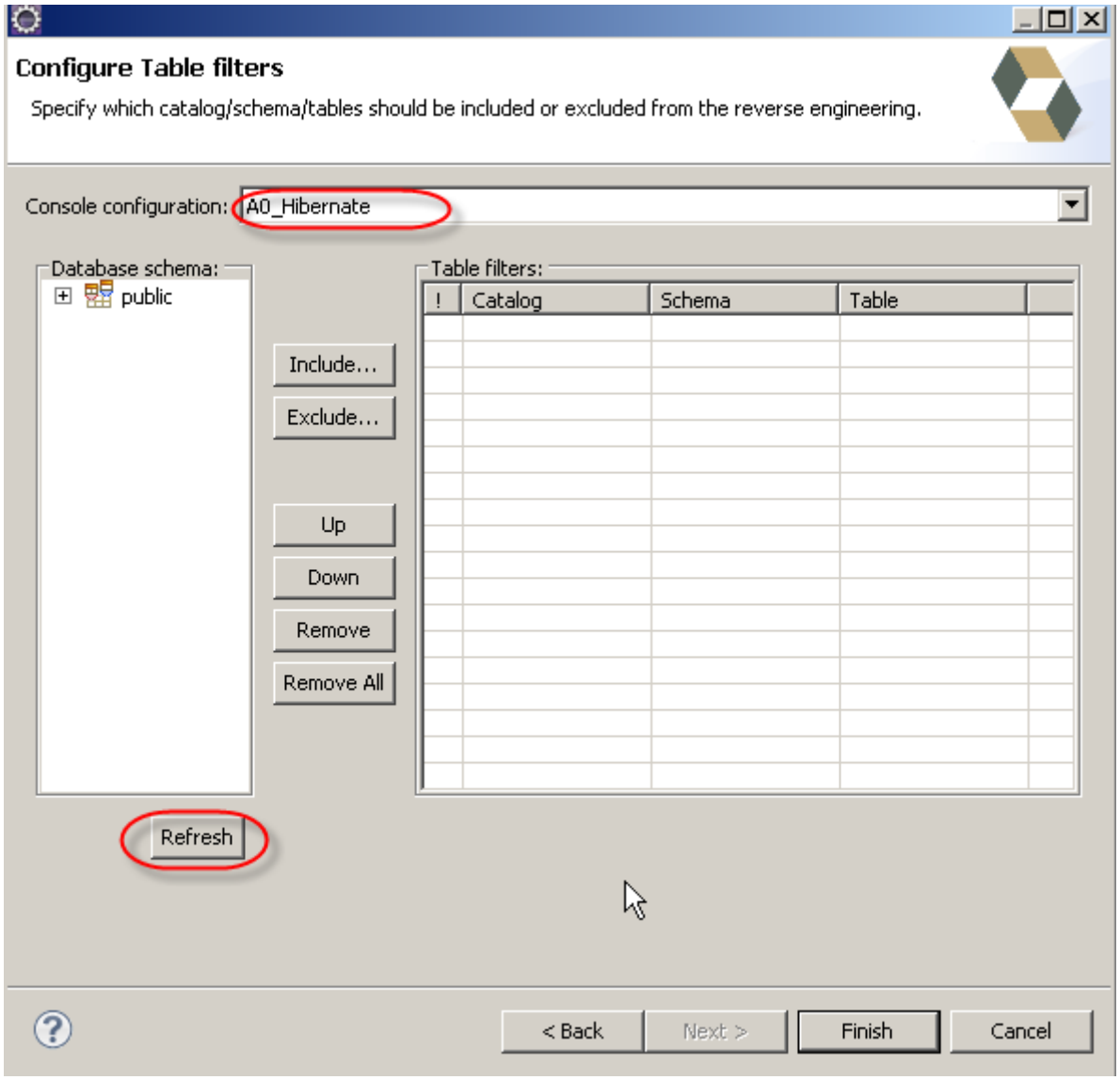

Create Hibernate Console Configuration
This wizard allows you to create a configuration for Hibernate Console.
Name: A0_Hibernate
Main
Options
Classpath
Mappings
Common
Type:
Core
Annotations (jdk 1.5+)
JPA (jdk 1.5+)
Hibernate Version: 3.5
Project:
A0_Hibernate
Browse...
Database connection:
bd_inscription
New...
Edit...
Property file:
Setup...
Configuration file:
Setup...
Persistence unit:
Browse...
< Back
Next >
Finish
Cancel

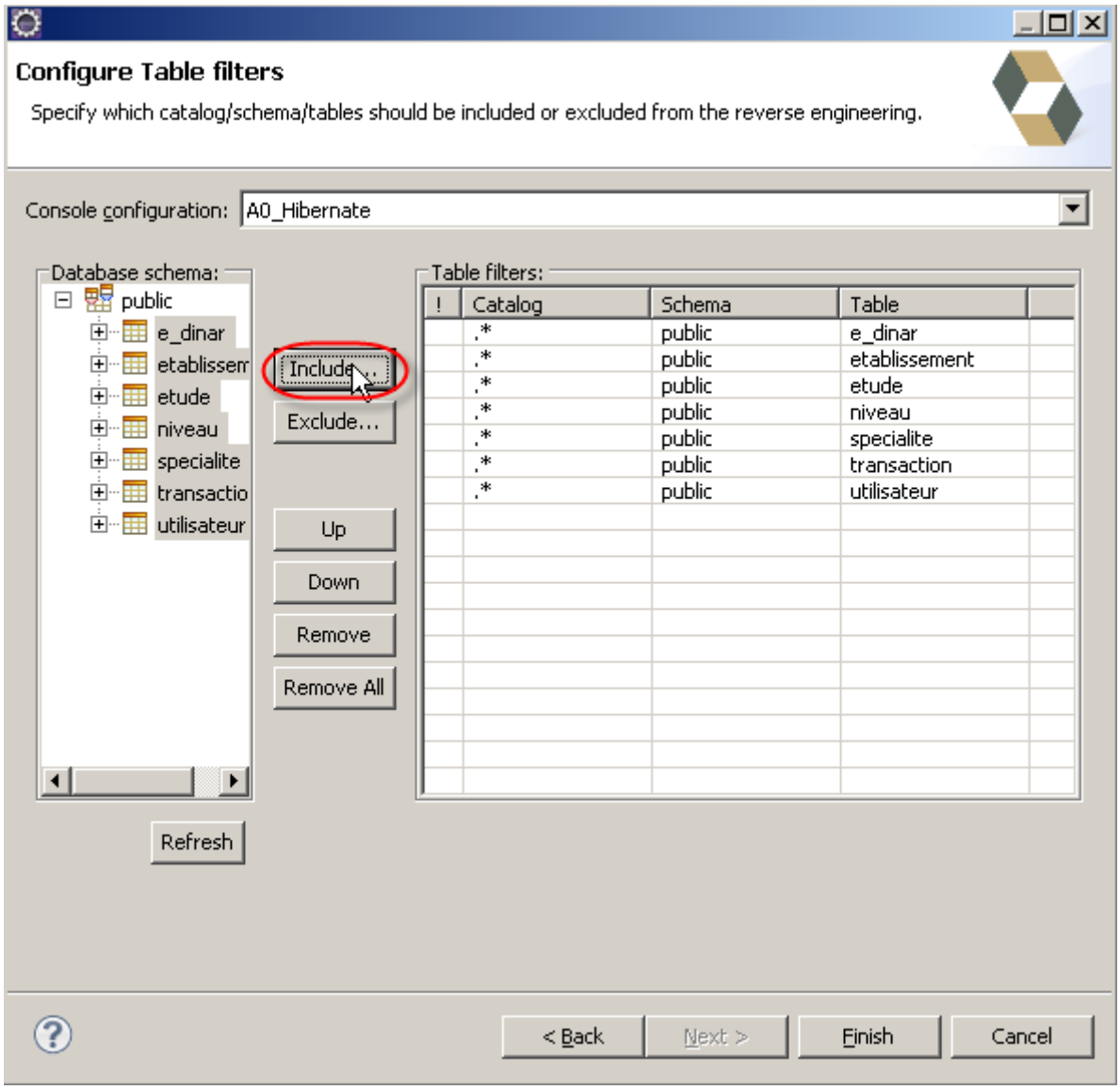


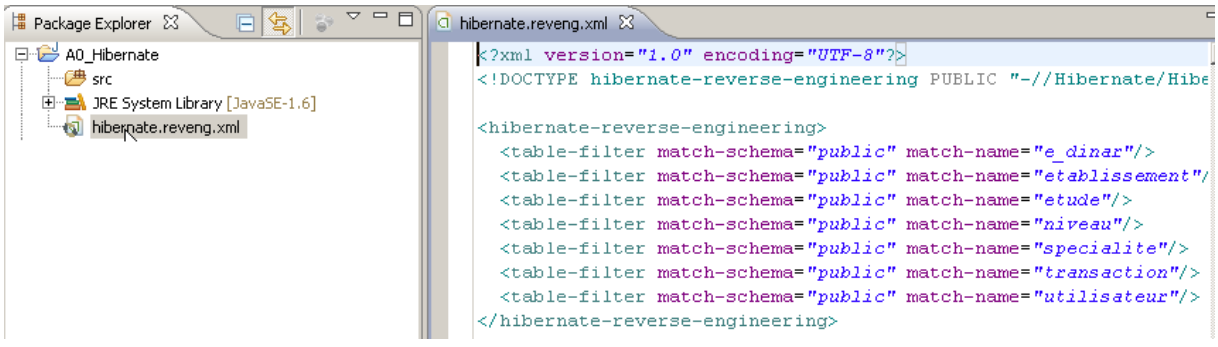
Création Fichier reveng.xml


New
Select a wizard
Create a new hibernate.reveng.xml (Helping with the initial table and type filtering)
Wizards:
type filter text
Android
C/C++
Connection Profiles
CVS
Database Web Services
Eclipse Modeling Framework
EJB
Groovy
Hibernate
Hibernate Configuration File (cfg.xml)
Hibernate Console Configuration
Hibernate Reverse Engineering File (reveng.xml)
Hibernate XML Mapping file (hbm.xml)
Java
< Back
Next >
Finish
Cancel

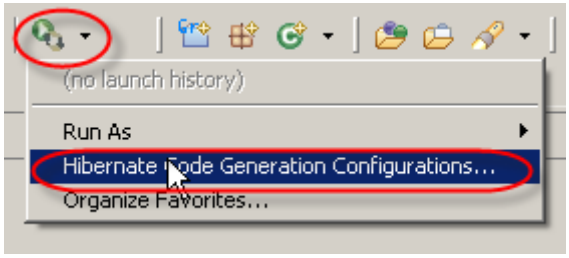

Create Hibernate Reverse Engineering file (reveng.xml)
Create a new hibernate.reveng.xml.
Enter or select the parent folder:
A0_Hibernate
A0_Hibernate
.settings
bin
src
CodeBar
JAK_Manager
OTV_JSF_Spring_Hibernate_Project
Servers
SWIXML
File name:
hibernate.reveng.xml
Advanced >>
< Back
Next >
Finish
Cancel

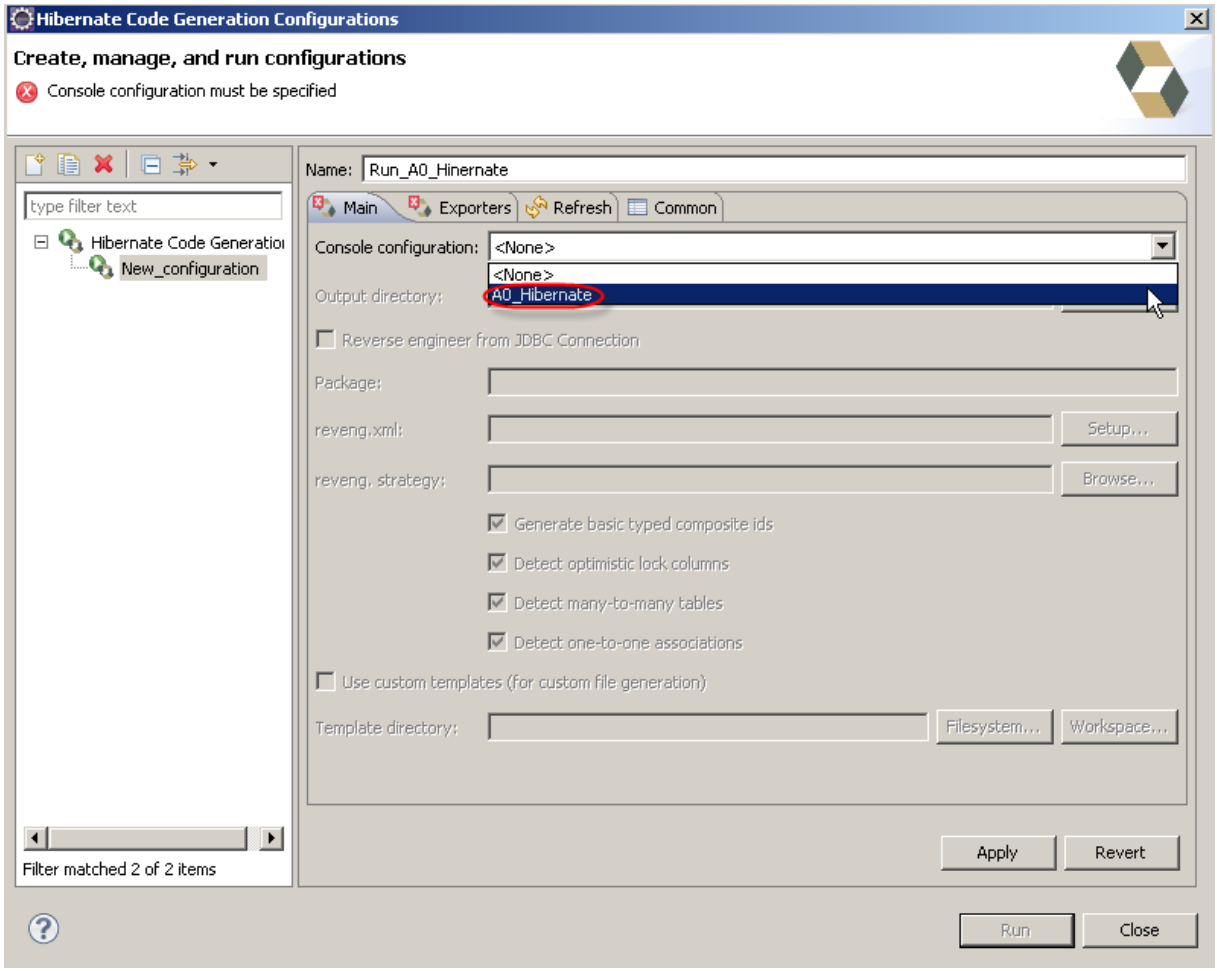

Configure Table filters
Specify which catalog/schema/tables should be included or excluded from the reverse engineering.
Console configuration:
A0_Hibernate
Database schema:
public
Include...
Exclude...
Up
Down
Remove
Remove All
Table filters:
!
Catalog
Schema
Table
Refresh
< Back
Next >
Finish
Cancel

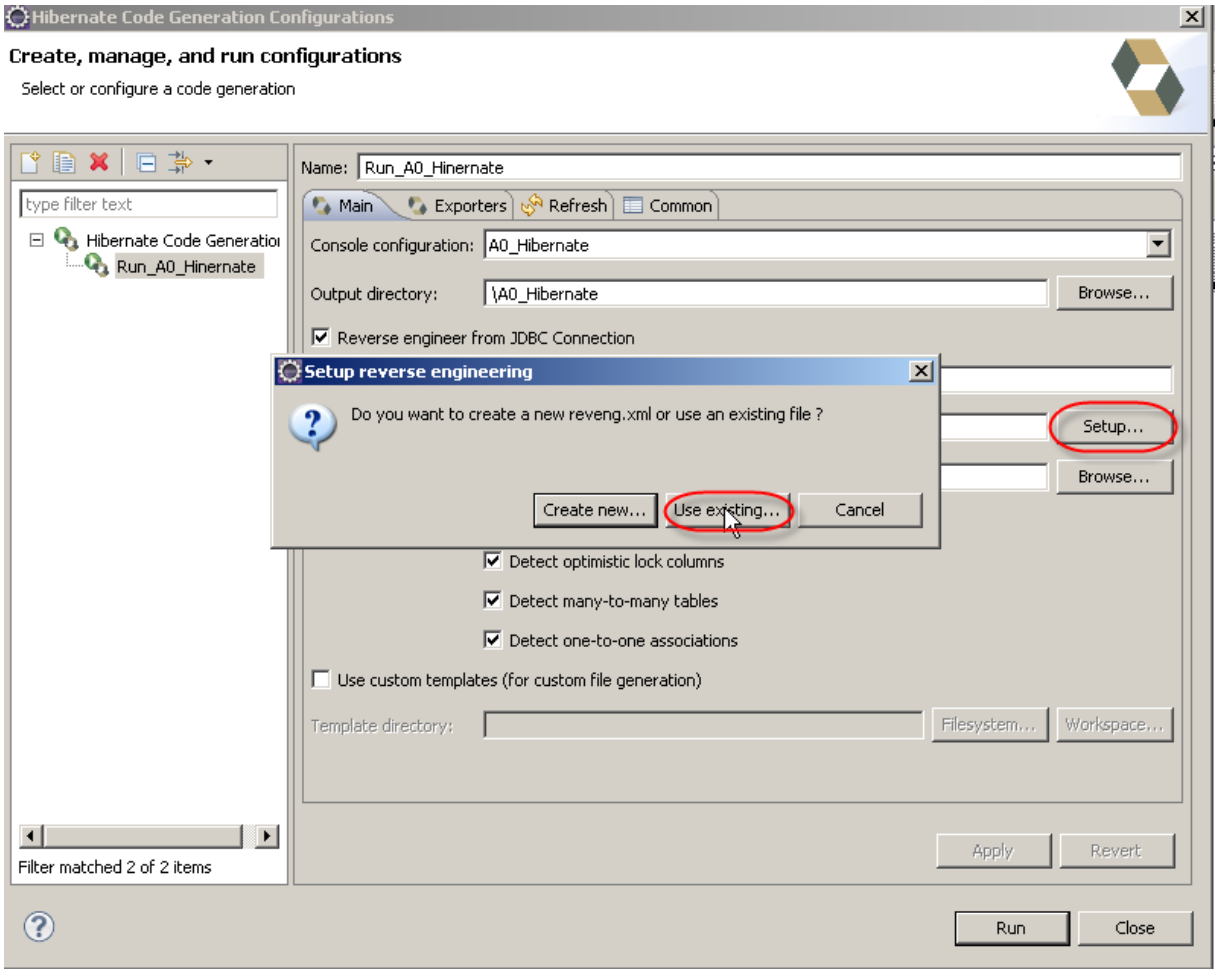

Configure Table filters
Specify which catalog/schema/tables should be included or excluded from the reverse engineering.
Console configuration: A0_Hibernate
Database schema:
public
e_dinar
etablisser
etude
niveau
specialite
transactio
utilisateur
Include...
Exclude...
Up
Down
Remove
Remove All
Refresh
Table filters:
! Catalog Schema Table
.* public e_dinar
.* public etablissement
.* public etude
.* public niveau
.* public specialite
.* public transaction
.* public utilisateur
< Back
Next >
Finish
Cancel

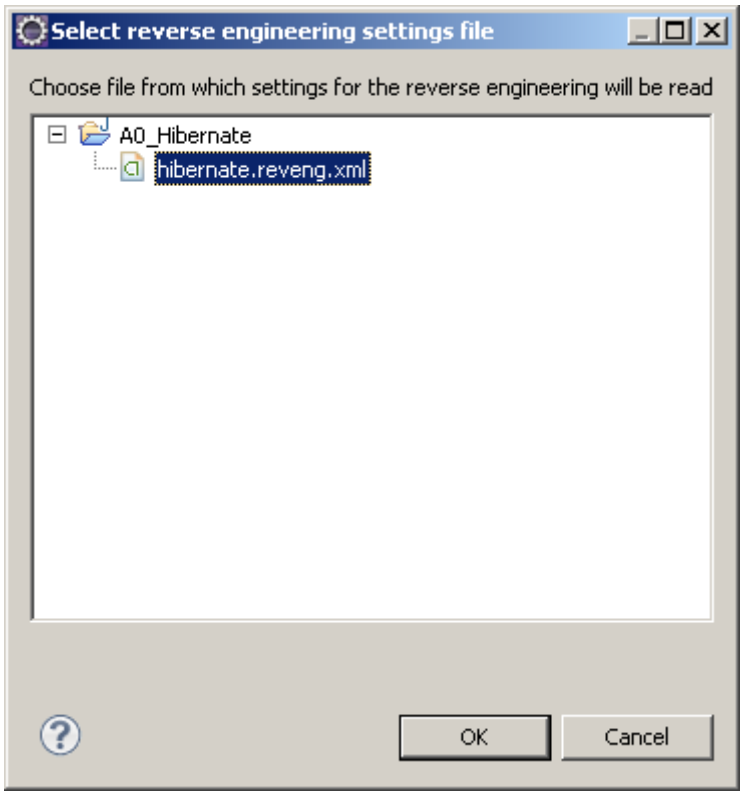

Package Explorer
A0_Hibernate
src
JRE System Library [JavaSE-1.6]
hibernate.reveng.xml
hibernate.reveng.xml
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE hibernate-reverse-engineering PUBLIC "-//Hibernate/Hibe
<hibernate-reverse-engineering>
<table-filter match-schema="public" match-name="e_dinar"/>
<table-filter match-schema="public" match-name="etablissement"/
<table-filter match-schema="public" match-name="etude"/>
<table-filter match-schema="public" match-name="niveau"/>
<table-filter match-schema="public" match-name="specialite"/>
<table-filter match-schema="public" match-name="transaction"/>
<table-filter match-schema="public" match-name="utilisateur"/>
</hibernate-reverse-engineering>

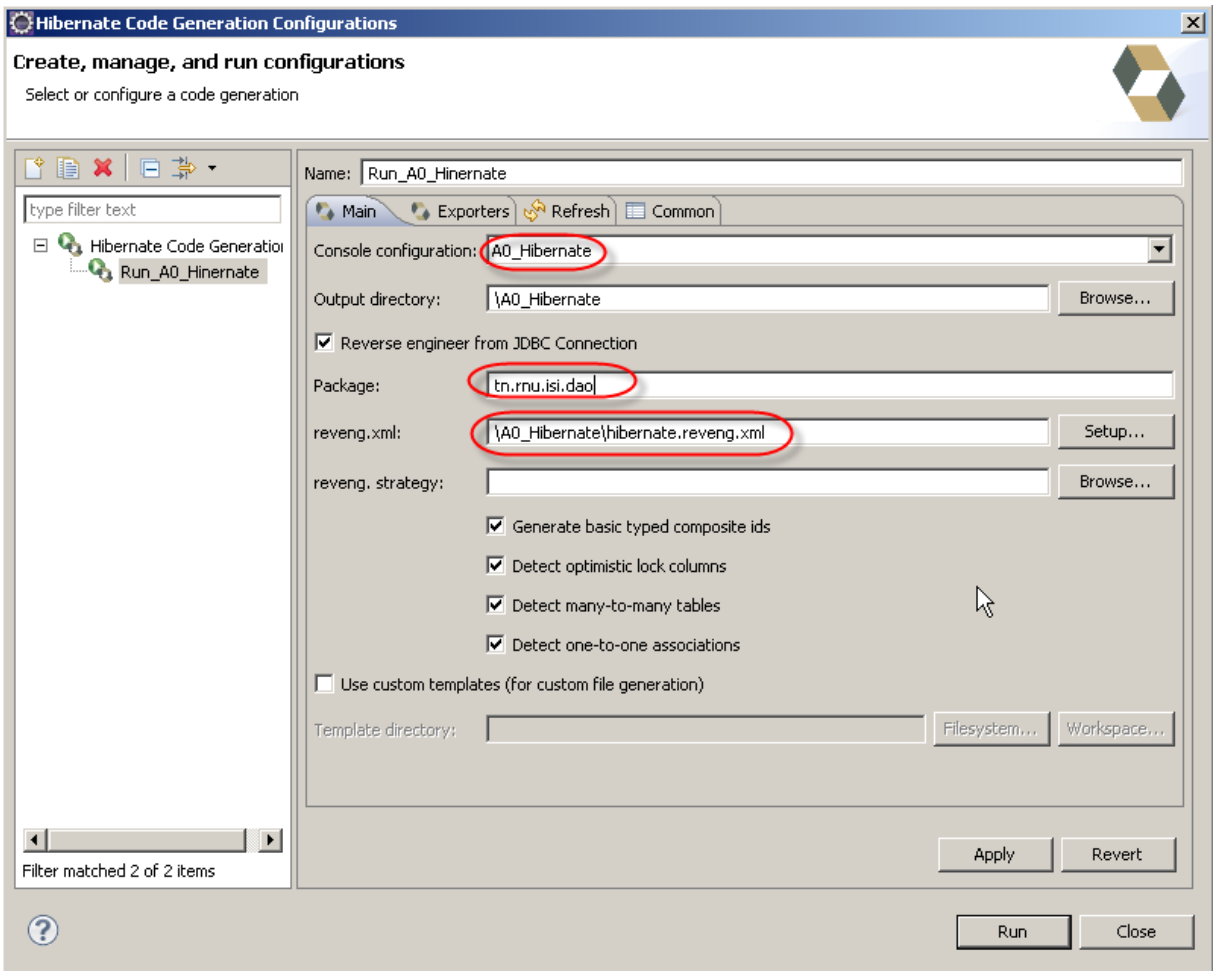
Génération Code Hibernate - Console configuration

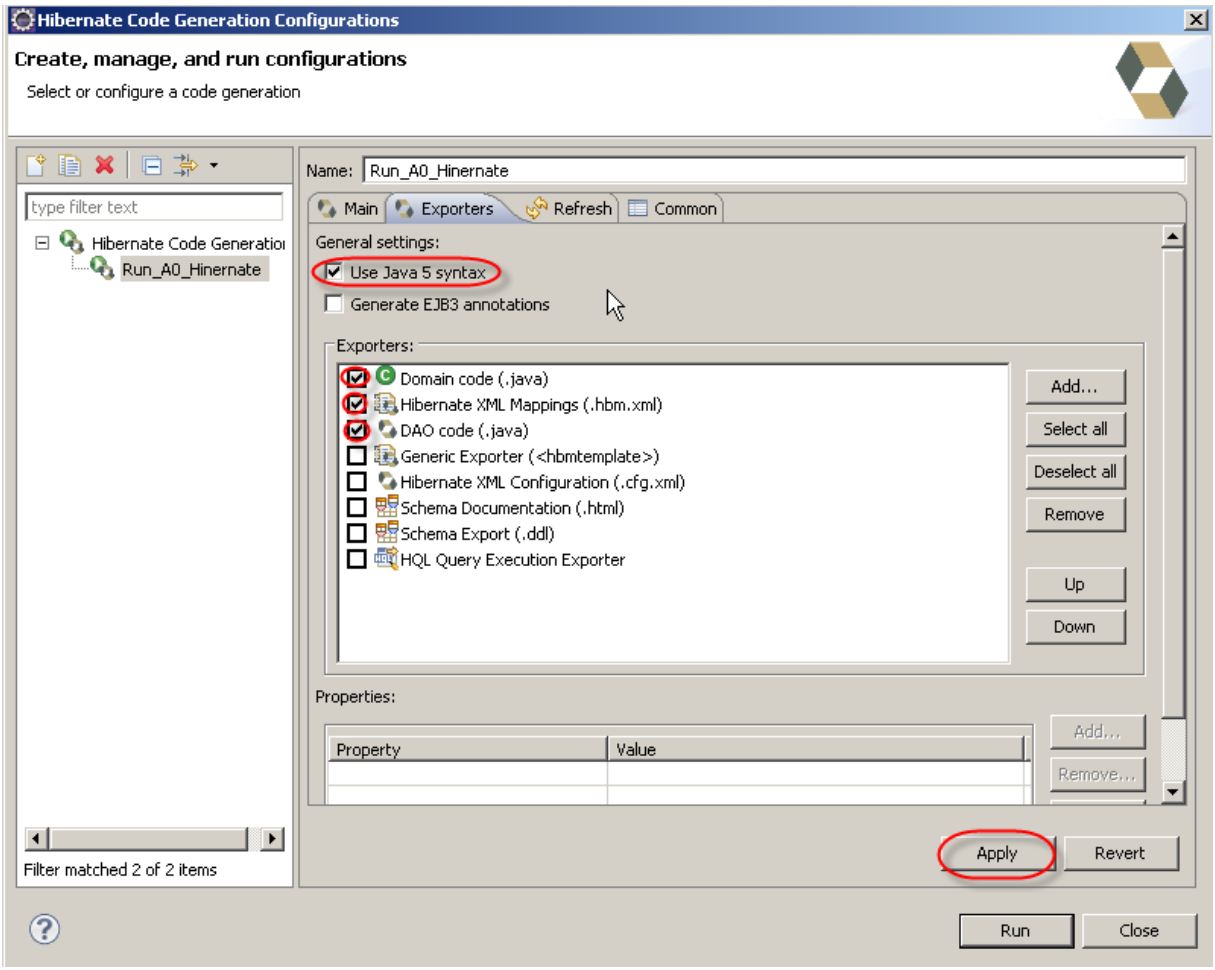




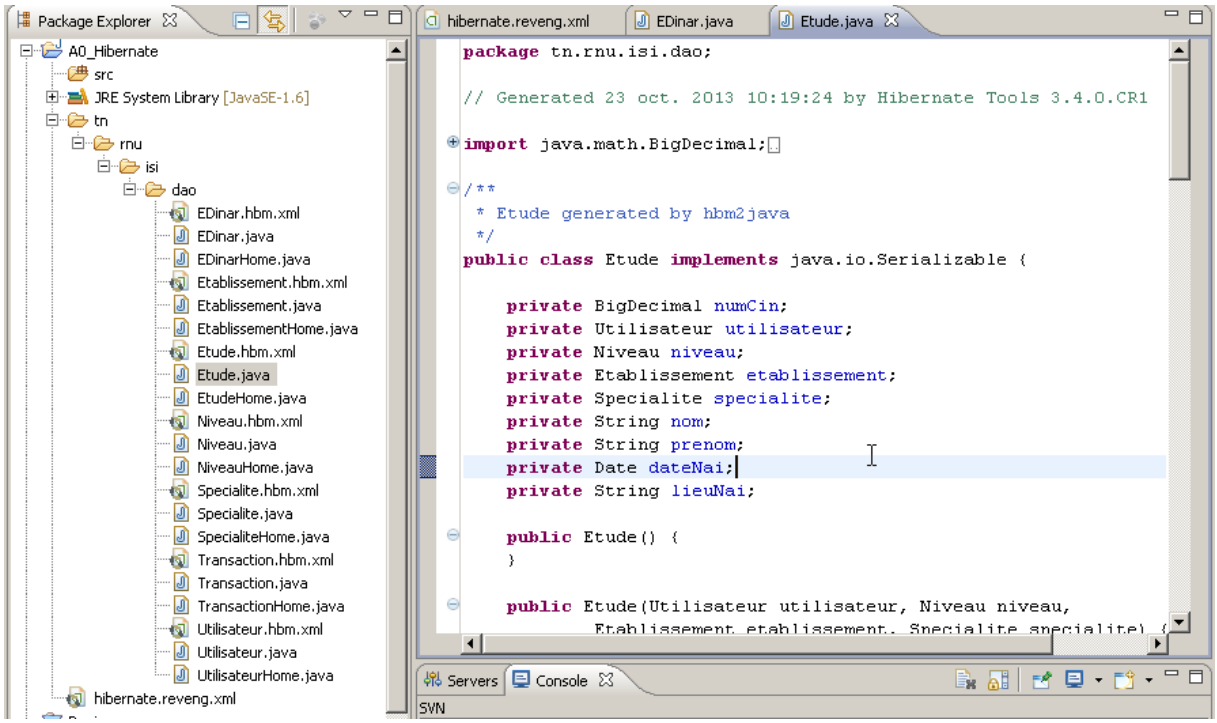

Hibernate Code Generation Configurations
Create, manage, and run configurations
Select or configure a code generation
type filter text
Hibernate Code Generatio
Run_A0_Hinernate
Filter matched 2 of 2 items
Name: Run_A0_Hinernate
Main
Exporters
Refresh
Common
Console configuration: A0_Hibernate
Output directory: \A0_Hibernate
Browse...
Reverse engineer from JDBC Connection
Setup reverse engineering
Do you want to create a new reveng.xml or use an existing file ?
Create new...
Use existing...
Cancel
Setup...
Browse...
Detect optimistic lock columns
Detect many-to-many tables
Detect one-to-one associations
Use custom templates (for custom file generation)
Template directory:
Filesystem...
Workspace...
Apply
Revert
Run
Close


Select reverse engineering settings file
Choose file from which settings for the reverse engineering will be read
A0_Hibernate
hibernate.reveng.xml
OK
Cancel

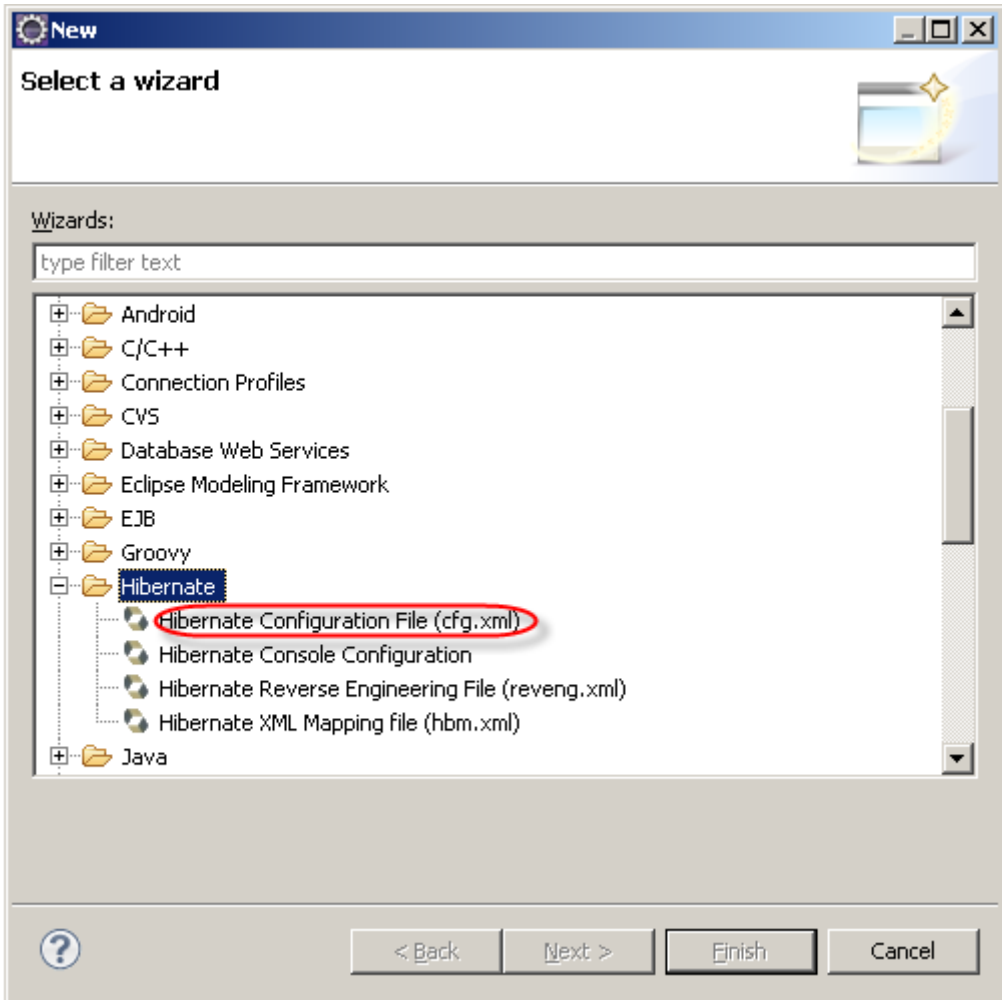

Hibernate Code Generation Configurations
Create, manage, and run configurations
Select or configure a code generation
type filter text
Hibernate Code Generatio
Run_A0_Hinernate
Filter matched 2 of 2 items
Name: Run_A0_Hinernate
Main
Exporters
Refresh
Common
Console configuration: A0_Hibernate
Output directory: \A0_Hibernate
Browse...
Reverse engineer from JDBC Connection
Package: tn.rnu.isi.dao
reveng.xml: \A0_Hibernate\hibernate.reveng.xml
Setup...
reveng. strategy:
Browse...
Generate basic typed composite ids
Detect optimistic lock columns
Detect many-to-many tables
Detect one-to-one associations
Use custom templates (for custom file generation)
Template directory:
Filesystem...
Workspace...
Apply
Revert
Run
Close

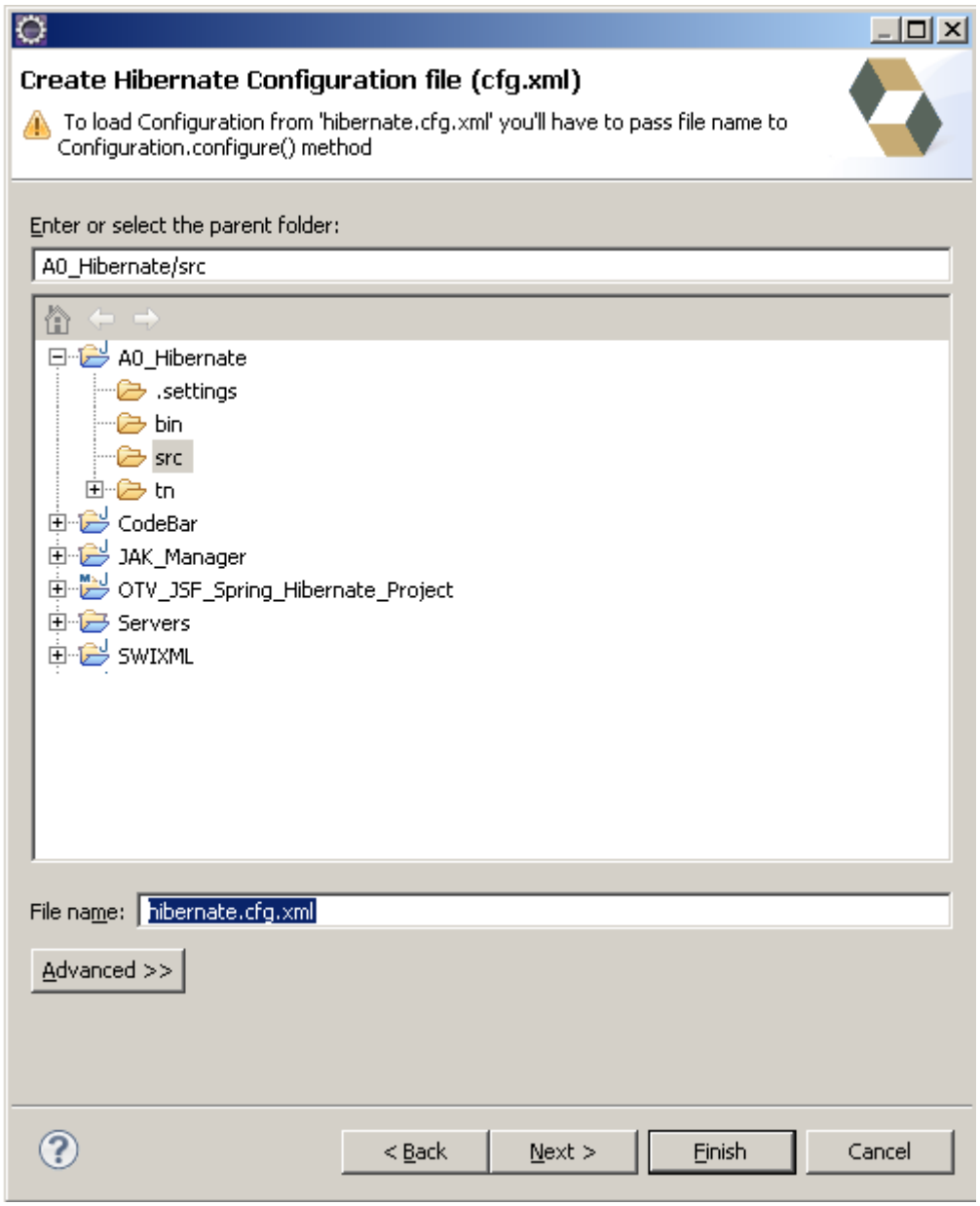


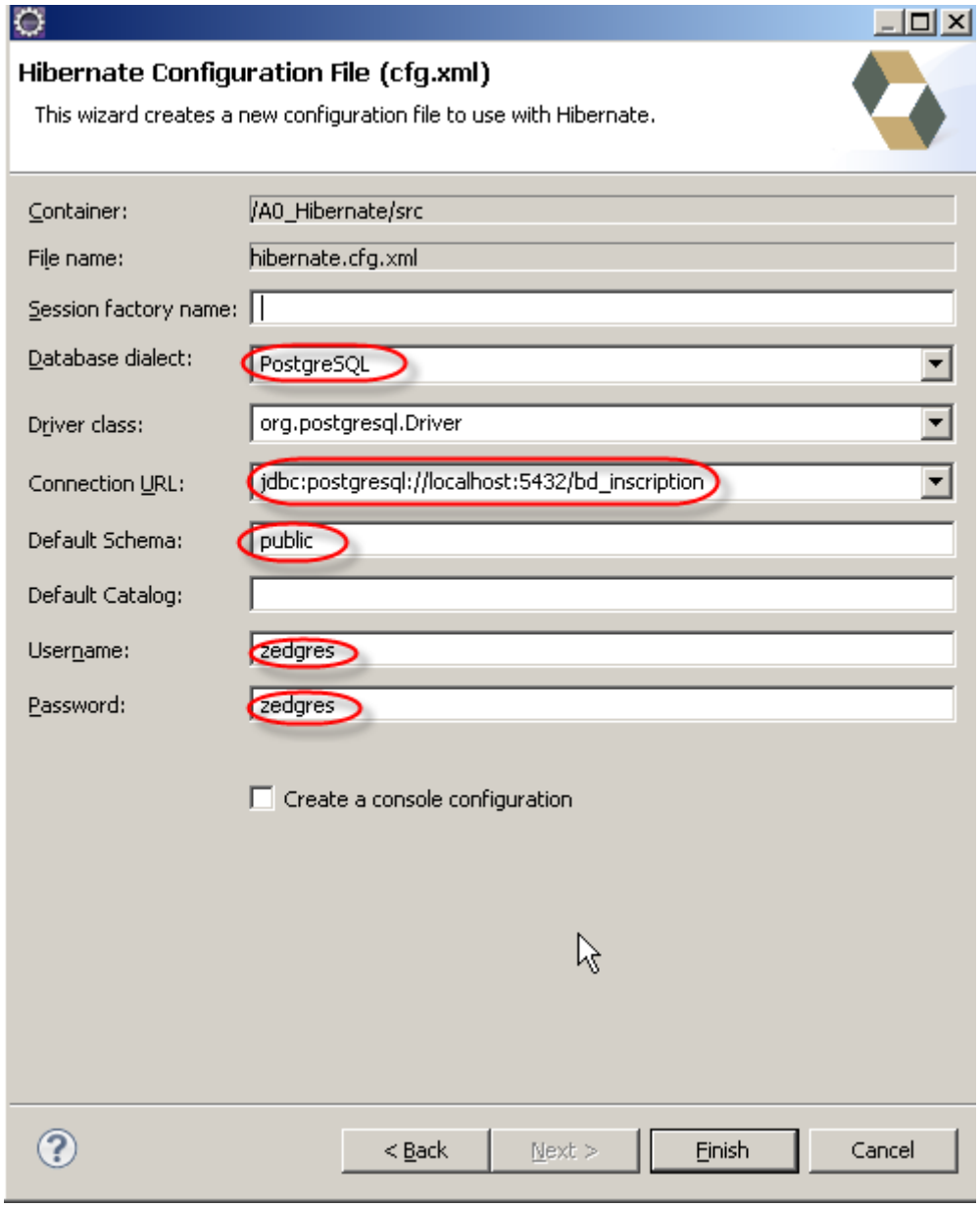
Cliquer sur « Run »

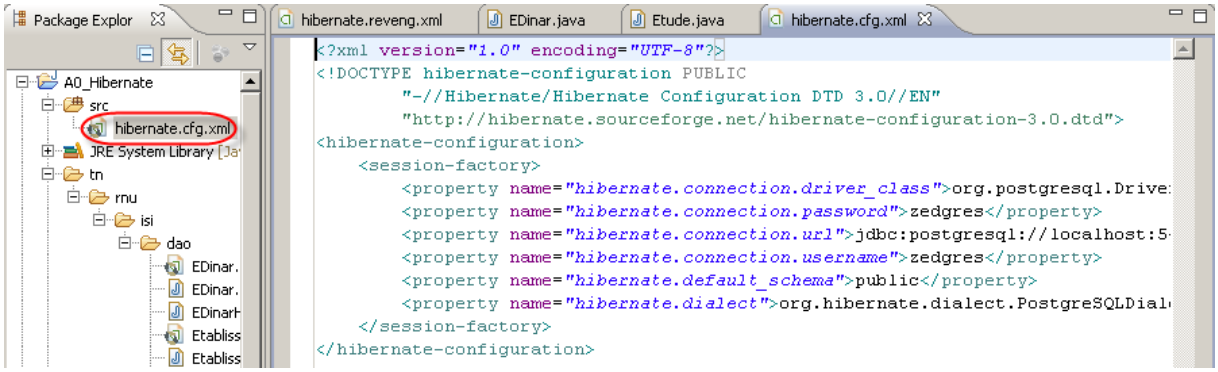


Création Fichier cfg.xml



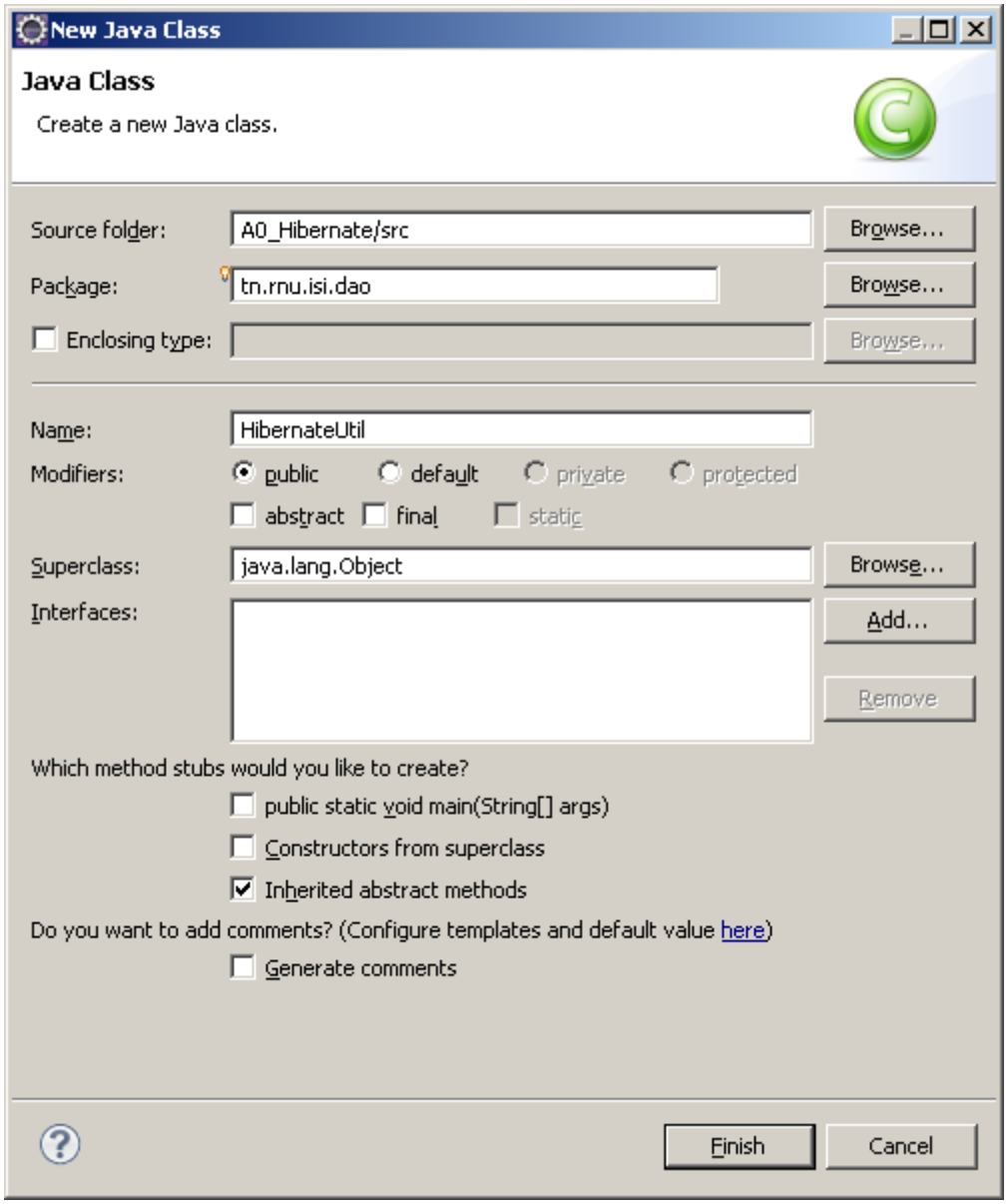

Create Hibernate Configuration file (cfg.xml)
To load Configuration from 'hibernate.cfg.xml' you'll have to pass file name to Configuration.configure() method
Enter or select the parent folder:
A0_Hibernate/src
A0_Hibernate
.settings
bin
src
tn
CodeBar
JAK_Manager
OTV_JSF_Spring_Hibernate_Project
Servers
SWIXML
File name:
hibernate.cfg.xml
Advanced >>
< Back
Next >
Finish
Cancel





Création Classe HiberateUtil.java

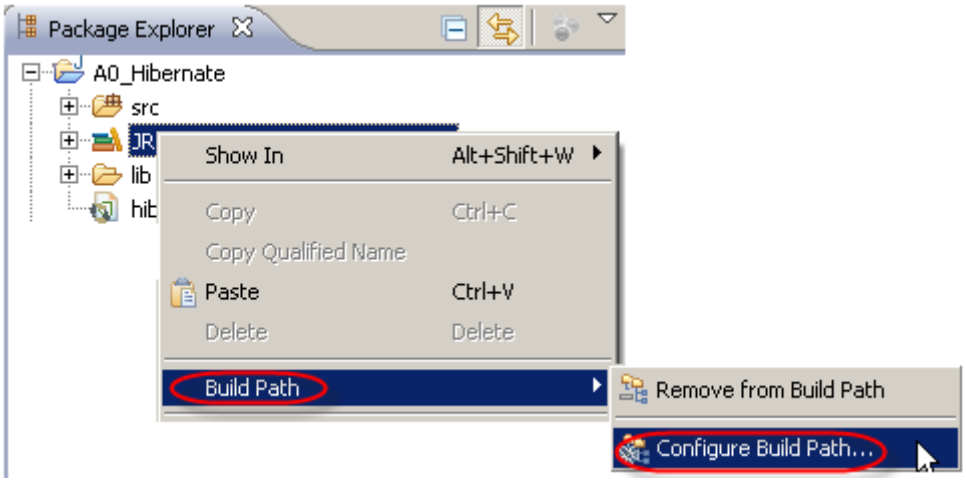
Maintenant, créez la classe HibernateUtil. HibernateUtil contribue à créer la `SessionFactory` à partir du fichier de configuration Hibernate. La SessionFactory est threadsafe, il n'est donc pas nécessaire d'obtenir un pour chaque thread. Voici le modèle singleton est utilisé pour instancier la SessionFactory. La mise en œuvre de la classe HibernateUtil est illustré ci-dessous.

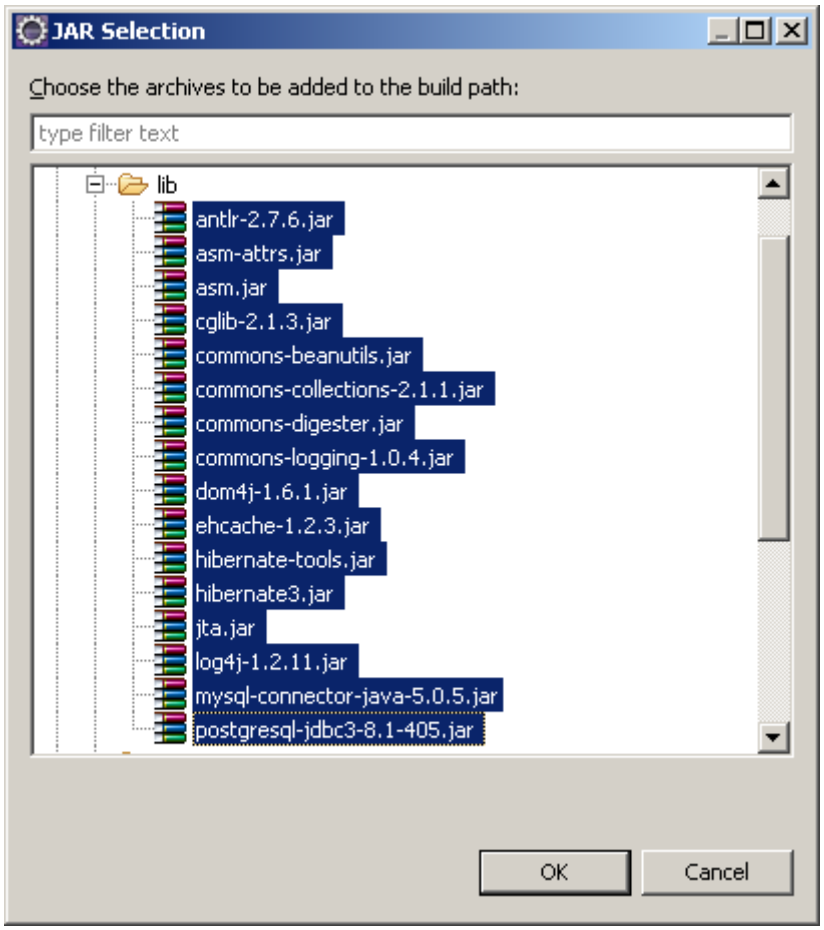


```
package tn.rnu.isi.dao;

import org.hibernate.SessionFactory;

import org.hibernate.cfg.Configuration;

public class HibernateUtil {

    private static final SessionFactory sessionFactory;

    static {

        try {

            sessionFactory = new Configuration().configure()
            .buildSessionFactory();
```

Ajout des jars de Hiberate

Les erreurs ci-dessus sont dues au manque des **jars** de Hibernate nécessaires. Pour ajouter les jars , il faut suivre l'écran suivant :






```
package tn.rnu.isi.dao;

import org.hibernate.SessionFactory;

import org.hibernate.cfg.Configuration;

public class HibernateUtil {

	private static final SessionFactory sessionFactory;

	static {

		try {

			sessionFactory = new Configuration().configure()
			.buildSessionFactory();

		} catch (Throwable ex) {

			System.err.println("Initial SessionFactory creation
failed." + ex);

			throw new ExceptionInInitializerError(ex);

		}

	}

	public static SessionFactory getSessionFactory() {
```

```
            return sessionFactory;

      }
}
```

Création de la classe de test

Maintenant, créez la classe de test ayant méthode principale main:

```
import java.math.BigDecimal;
import java.util.List;

import org.hibernate.Session;
import org.hibernate.Transaction;

import tn.rnu.isi.dao.Etude;
import tn.rnu.isi.dao.HibernateUtil;
import tn.rnu.isi.dao.Utilisateur;

public class TestMain {

      /**
       * @param args
       */
      public static void main(String[] args) {
            TestMain obj = new TestMain();
            //obj.save();
            obj.update(new BigDecimal(1));
//          obj.delete(new BigDecimal(1));
            obj.getList();
      }

//    public void save()
//    {
//          Etude u = new Etude();
//          Session session =
HibernateUtil.getSessionFactory().openSession();
//          Transaction transaction = null;
//          try
//          {
//                transaction = session.beginTransaction();
//                session.save(u);
//                transaction.commit();
//                System.out.println("Data Saved");
//          }catch(Exception e)
//          {
//                e.printStackTrace();
//          }finally{session.close();}
//
//    }
      public void delete(BigDecimal Id)
      {
            Session session =
HibernateUtil.getSessionFactory().openSession();
            Transaction transaction = null;
            try
            {
                  transaction = session.beginTransaction();
```

```
                  Etude u = (Etude)session.get(Etude.class,Id);
                  session.delete(u);
                  transaction.commit();
                  System.out.println("Data Deleted");
            }
            catch(Exception e)
            {
                  e.printStackTrace();
            }
            finally{
                  session.close();
            }
      }
      public void update(BigDecimal Id)
      {
            Session session =
HibernateUtil.getSessionFactory().openSession();
            Transaction transaction = null;
            try
            {
                  transaction = session.beginTransaction();
                  Etude u = (Etude)session.get(Etude.class,Id);
                  u.setPrenom("Mariem");
                  transaction.commit();
                  System.out.println("Data Updated");
            }
            catch(Exception e)
            {
                  e.printStackTrace();
            }
            finally{
                  session.close();
            }
      }

      public void getList()
      {
            Session session =
HibernateUtil.getSessionFactory().openSession();
            Transaction transaction = null;
            try
            {
                  transaction = session.beginTransaction();
                  List<Etude> uList = session.createQuery("from
Etude").list();
                  for(Etude u : uList)
                  {
                        System.out.println("Prenom Name - "+u.getPrenom());
                  }
            }
            catch(Exception e)
            {
                  e.printStackTrace();
            }
            finally{
                  session.close();
            }
      }
}
```