

TP EJB

Préparé par
M.G. BELKASMI

Archivage de Photos

Nous allons réaliser une application d'archivage de photos composé de:

- Un ejb permettant d'archiver des photos sur le serveur
- Une application cliente permettant de :
 - choisir les photos en local à archiver sur le serveur
 - Consulter les photos sur serveur
 - Récupérer les photos archivées



Réseau local
de l'entreprise



Application
cliente

```

<<interface>>
ArchiverRemote

+ stocker(nom : String, octets : byte[]) : void
+ restituer(nom : String) : byte[]
+ getListe() : List<Photo>
+ getPhoto(nom : String) : Photo
+ supprimer(nom : String) : void
    
```

Disque dur
client.

RMI

Serveur
d'applications
GLASSFISH

```

<<EJB session>>
Archiver

- répertoire : String = E:/Archivage/

+ stocker(nom : String, octets : byte[]) : void
- <<asynchrone>> enregistrer(nom : String) : void
+ restituer(nom : String) : byte[]
+ getListe() : List<Photo>
+ getPhoto(nom : String) : Photo
+ supprimer(nom : String) : void
    
```

- persistance

```

<<interface>>
EntityManager

+ persist(entité : Object) : void
+ find(classe : Class<T>, cléprimaire : Object) : void
+ remove(entité : Object) : void
+ createNamedQuery(requête : String) : Query
    
```

Création des entités
en mémoire

Gestion des entités
pour la persistance

Enregistrement d'informations
supplémentaires relatives aux photos

Archivage
des photos

Disque dur
du serveur

SQL / JDBC

Base de données
Photos.

```

«entity»
+ Photo

- id : String
- instant : Date
- largeur : int
- hauteur : int
- poids : double

+ Photo()
+ Photo(String, double)
+ getId() : String
+ getHauteur() : int
+ getLargeur() : int
+ getPoids() : double
+ setDimensions(BufferedImage) : void
+ toString() : String
    
```

Associations

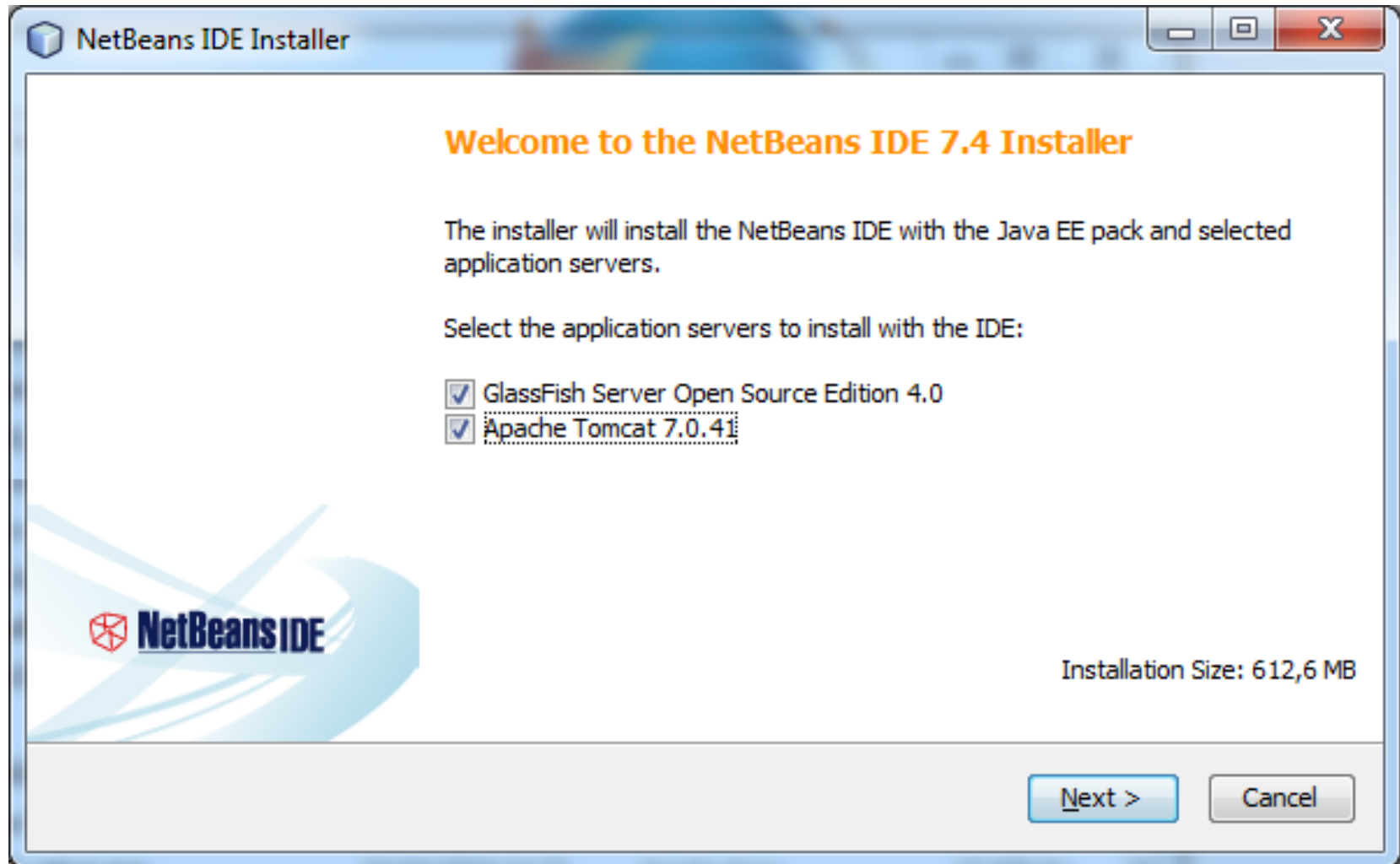
PHOTO		
ID	VARIABLE CHARACTER (255)	NOT NULL
INSTANT	DATE TIME	NULL
LARGEUR	INTEGER	NOT NULL
HAUTEUR	INTEGER	NOT NULL
POIDS	DOUBLE PRECISION	NULL
Clé primaire / unique (ID)		

Archivage de Photos

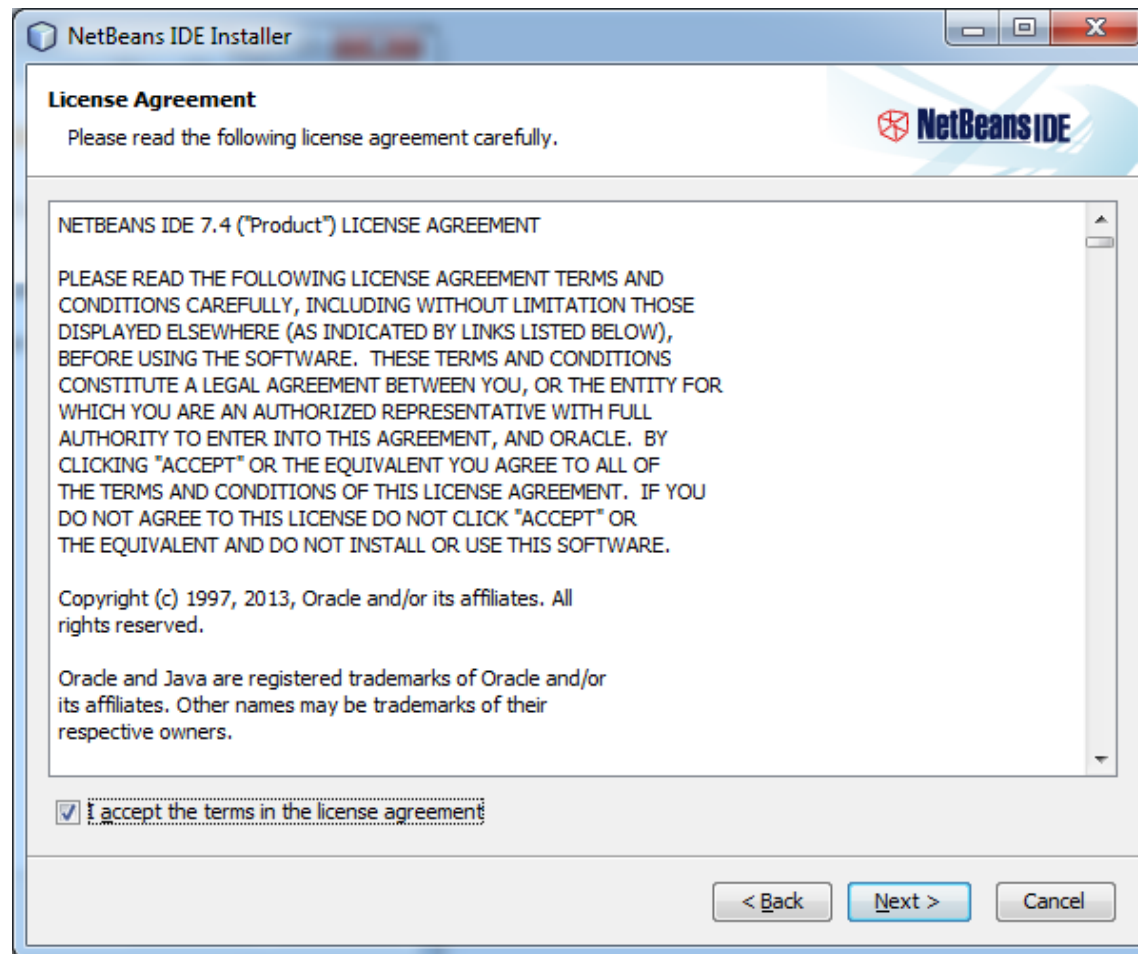
- Nous réaliserons cette application sous NetBeans (avec TomCat et GlassFish incorporés).
- Téléchargeable sur

<https://netbeans.org/>

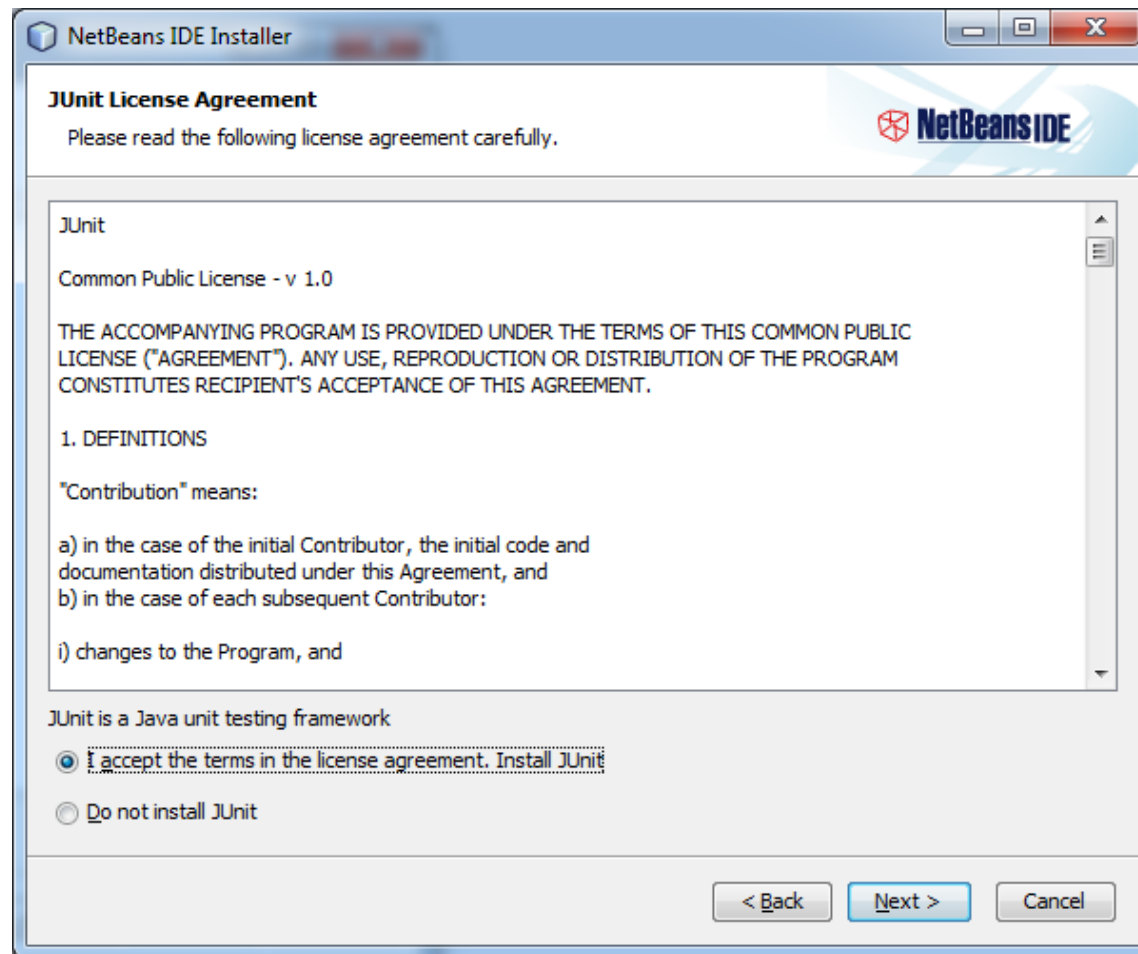
Installation de NetBeans



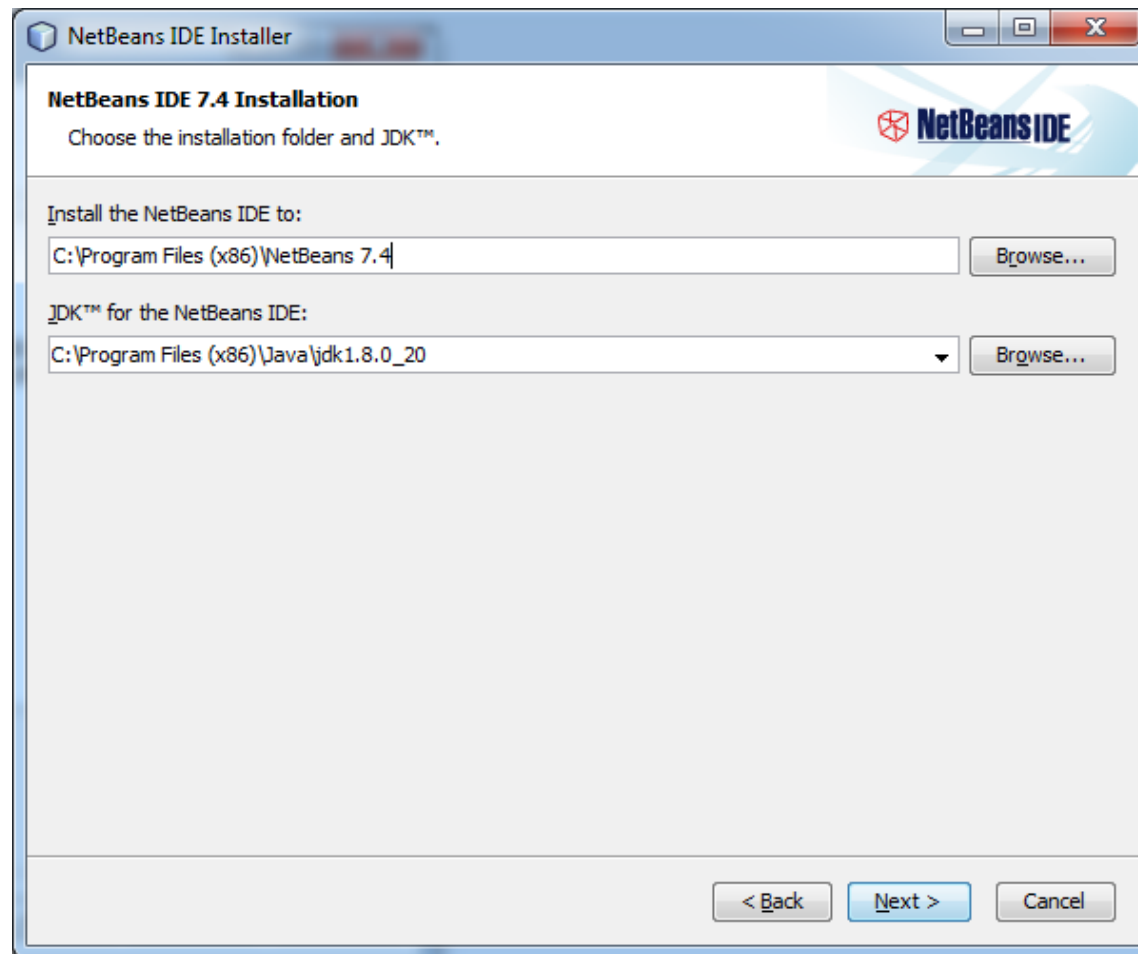
Installation de NetBeans



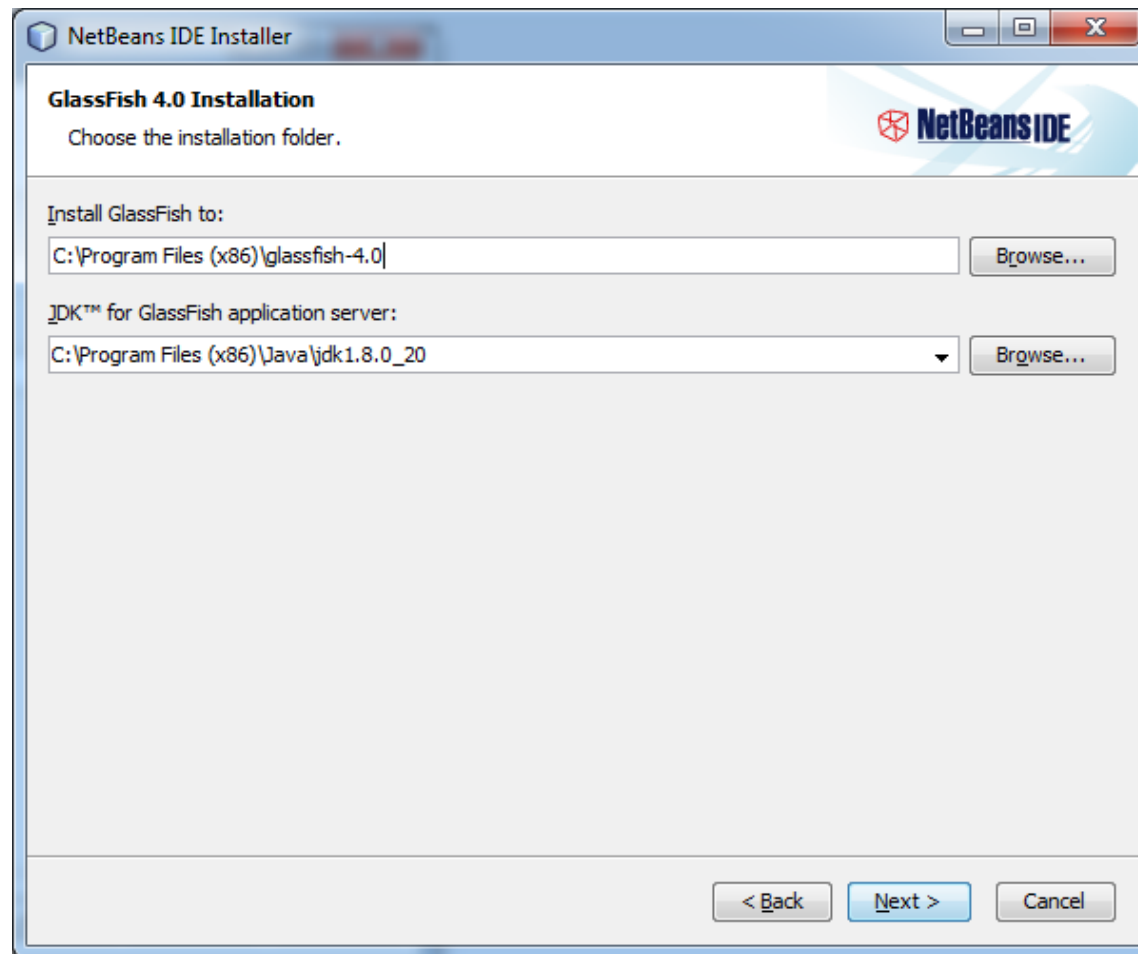
Installation de NetBeans



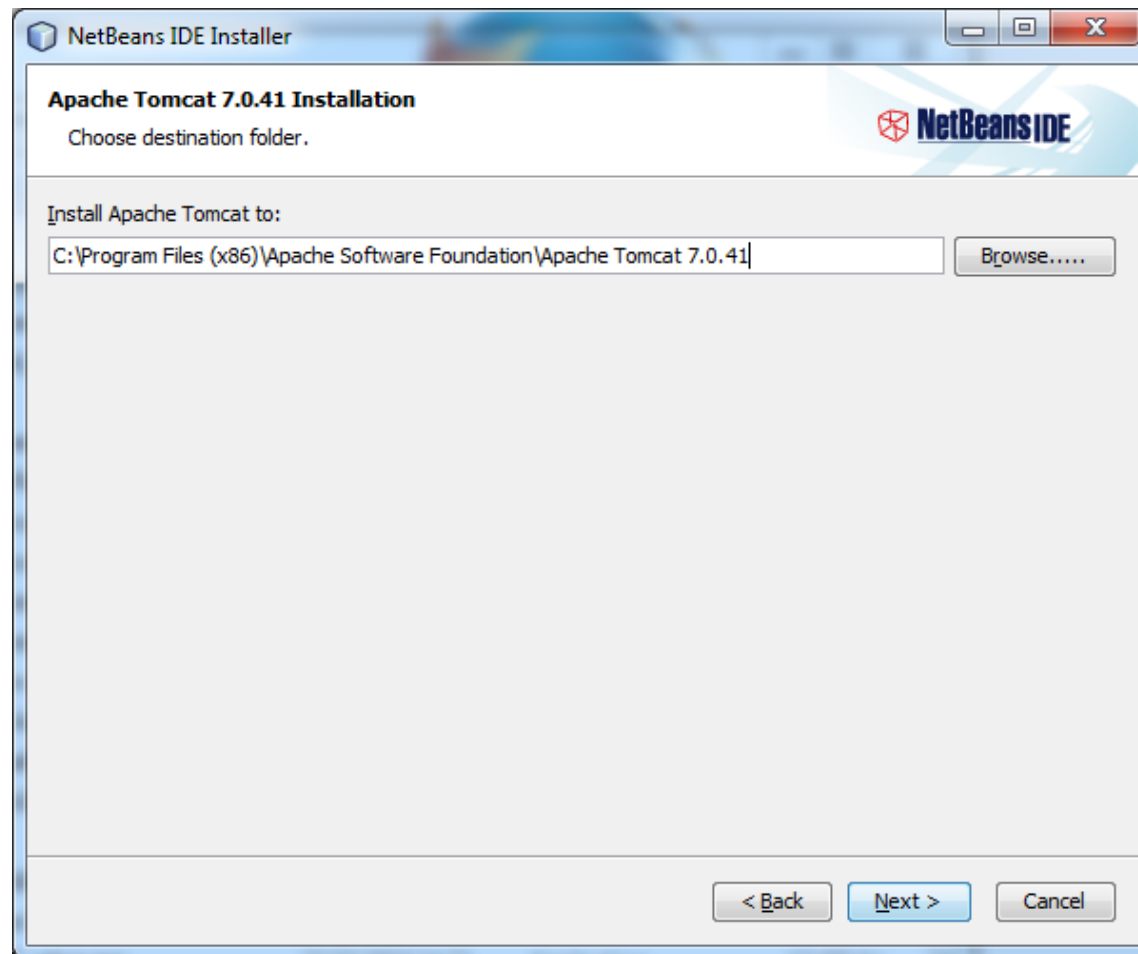
Installation de NetBeans



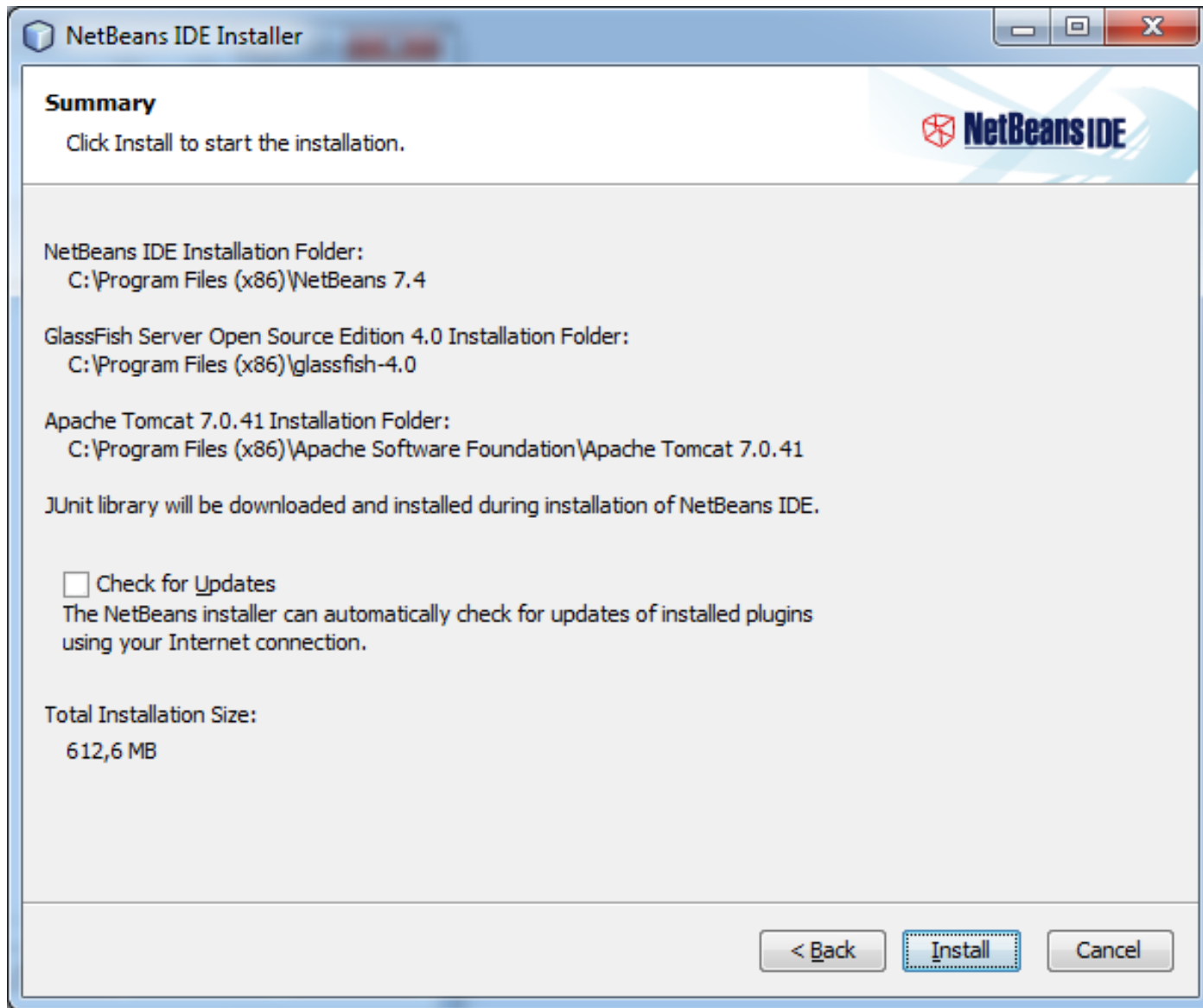
Installation de NetBeans



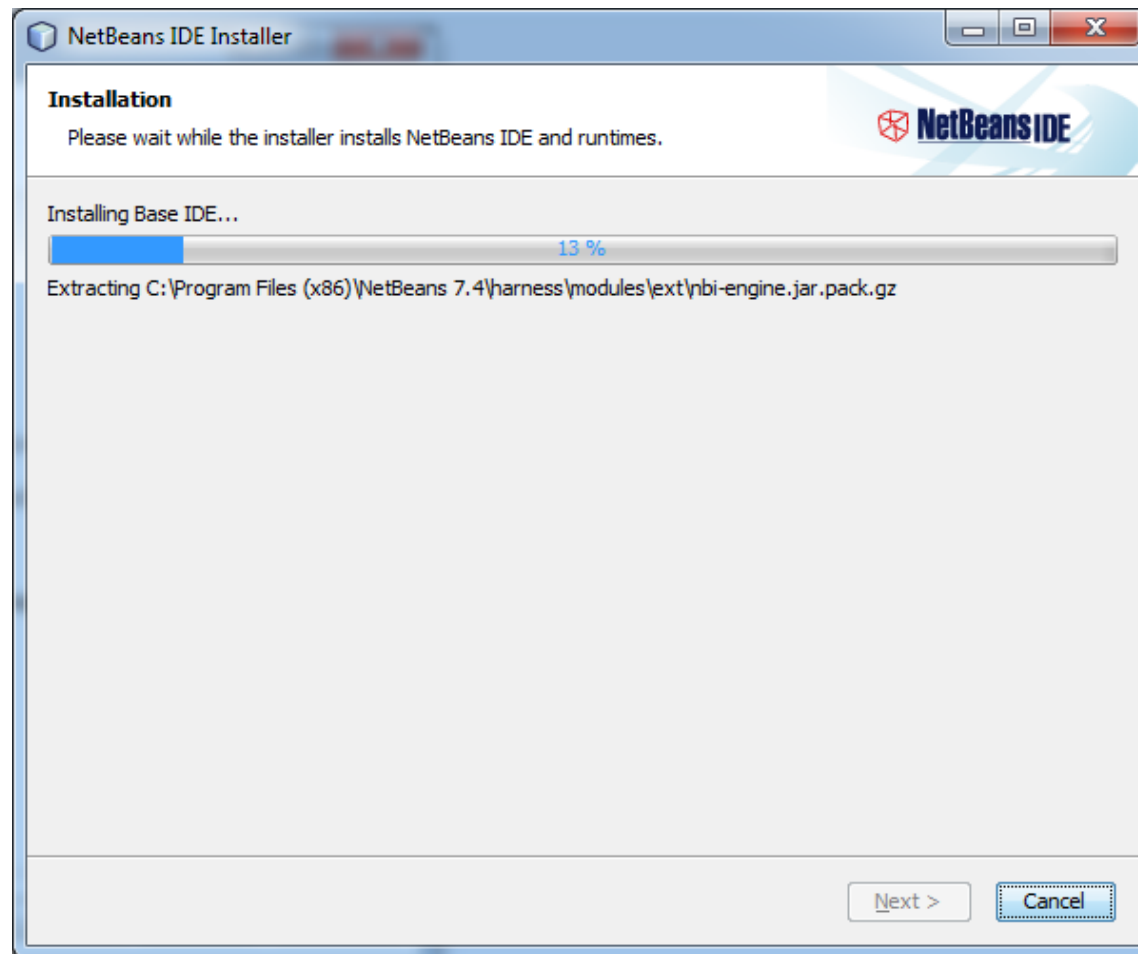
Installation de NetBeans



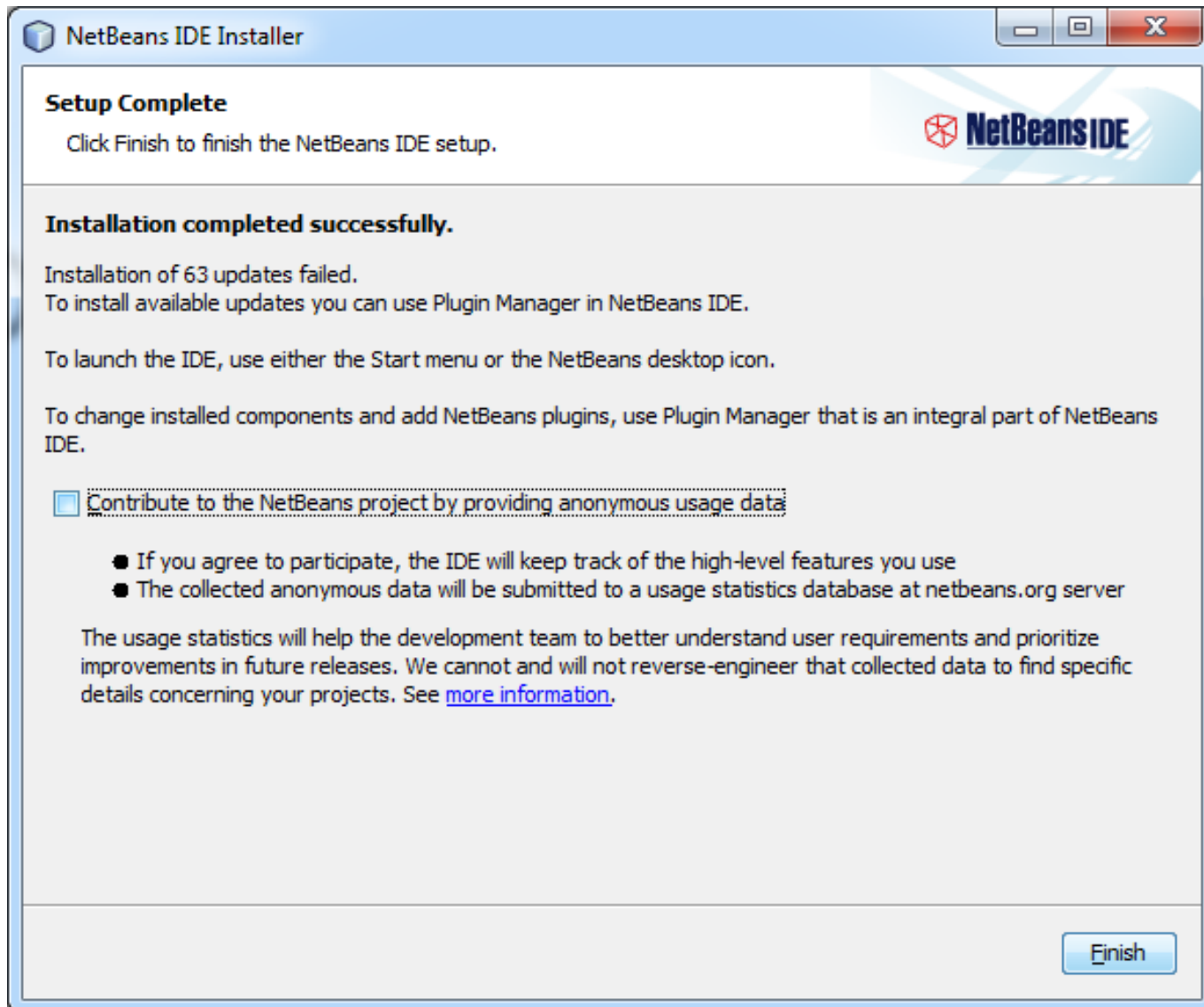
Installation de NetBeans



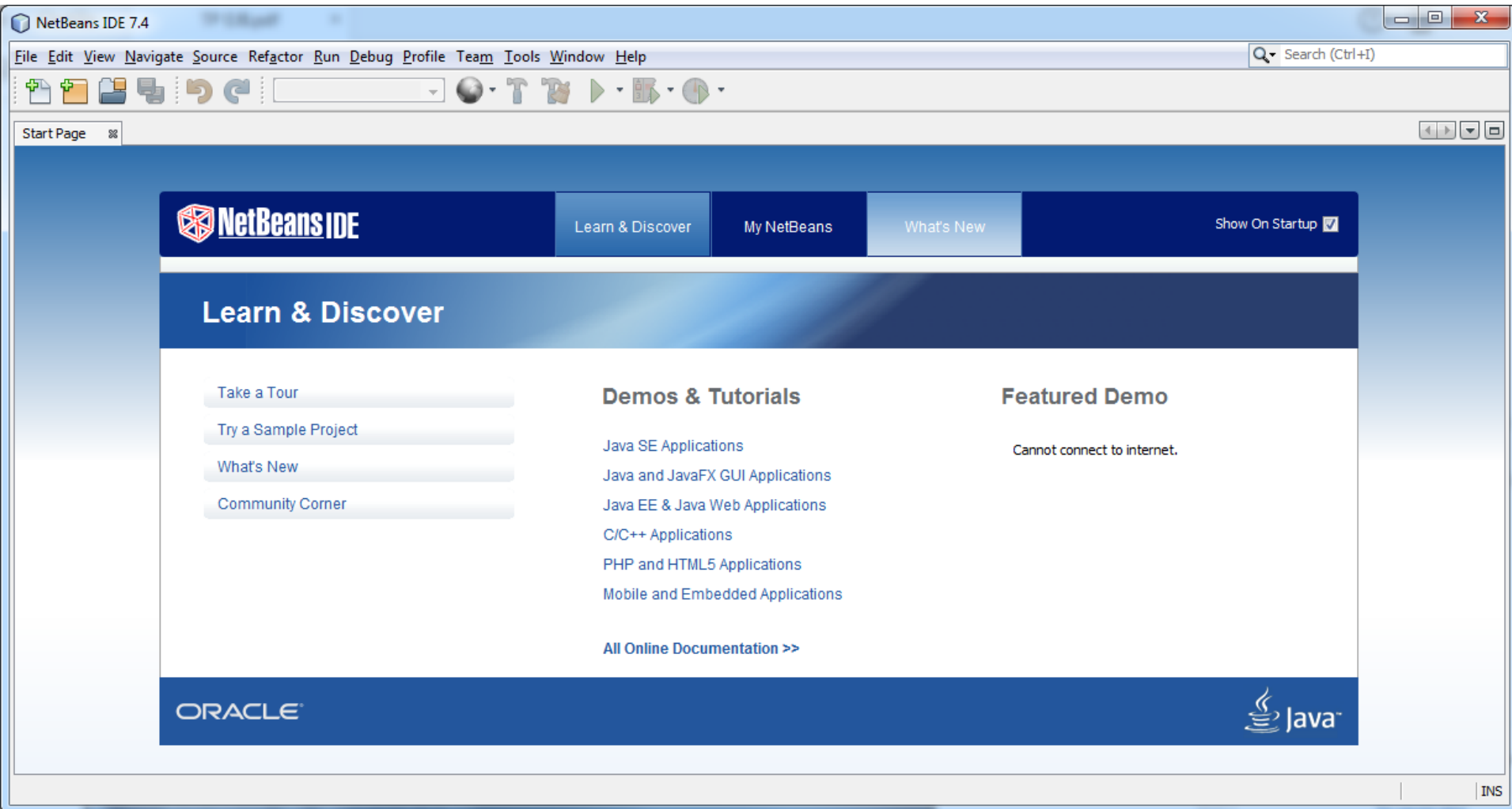
Installation de NetBeans



Installation de NetBeans



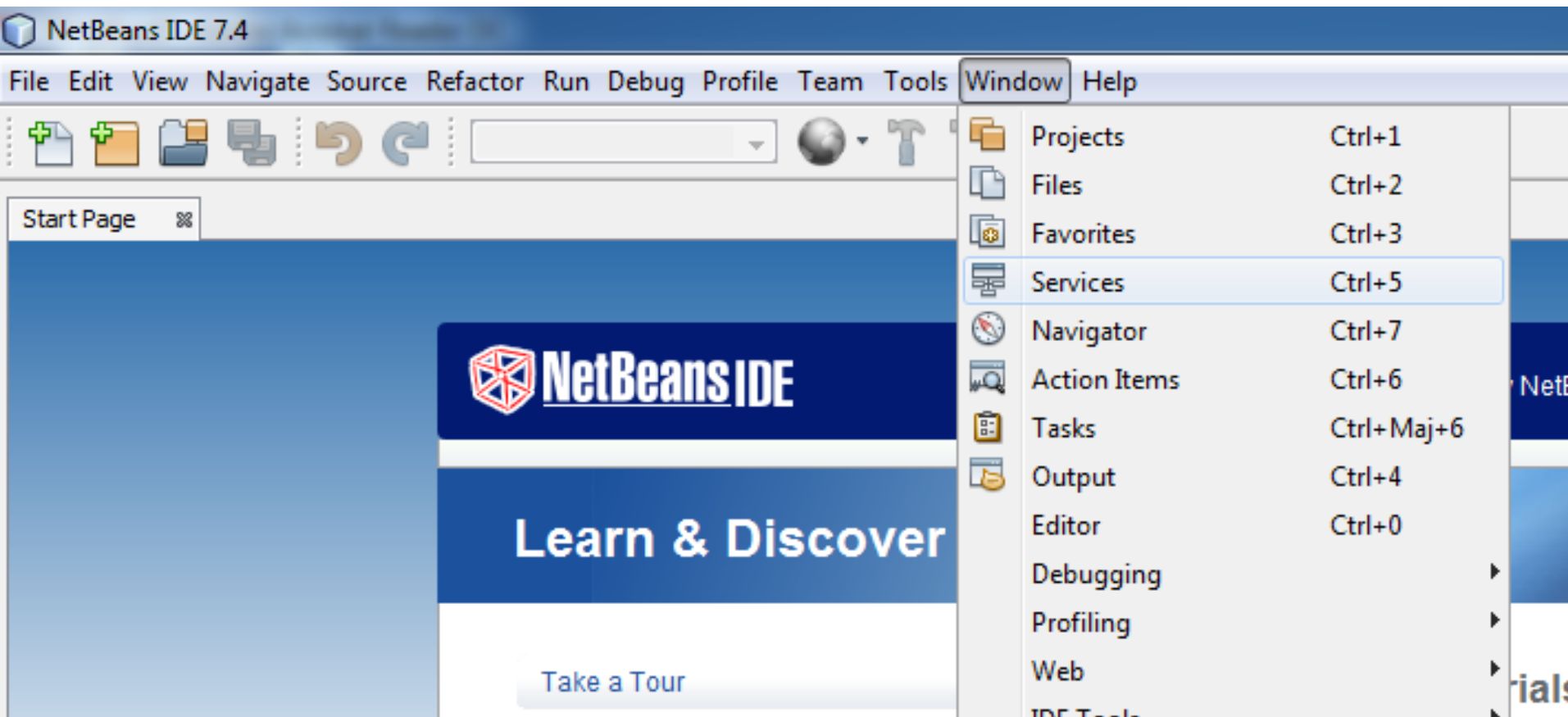
Installation de NetBeans



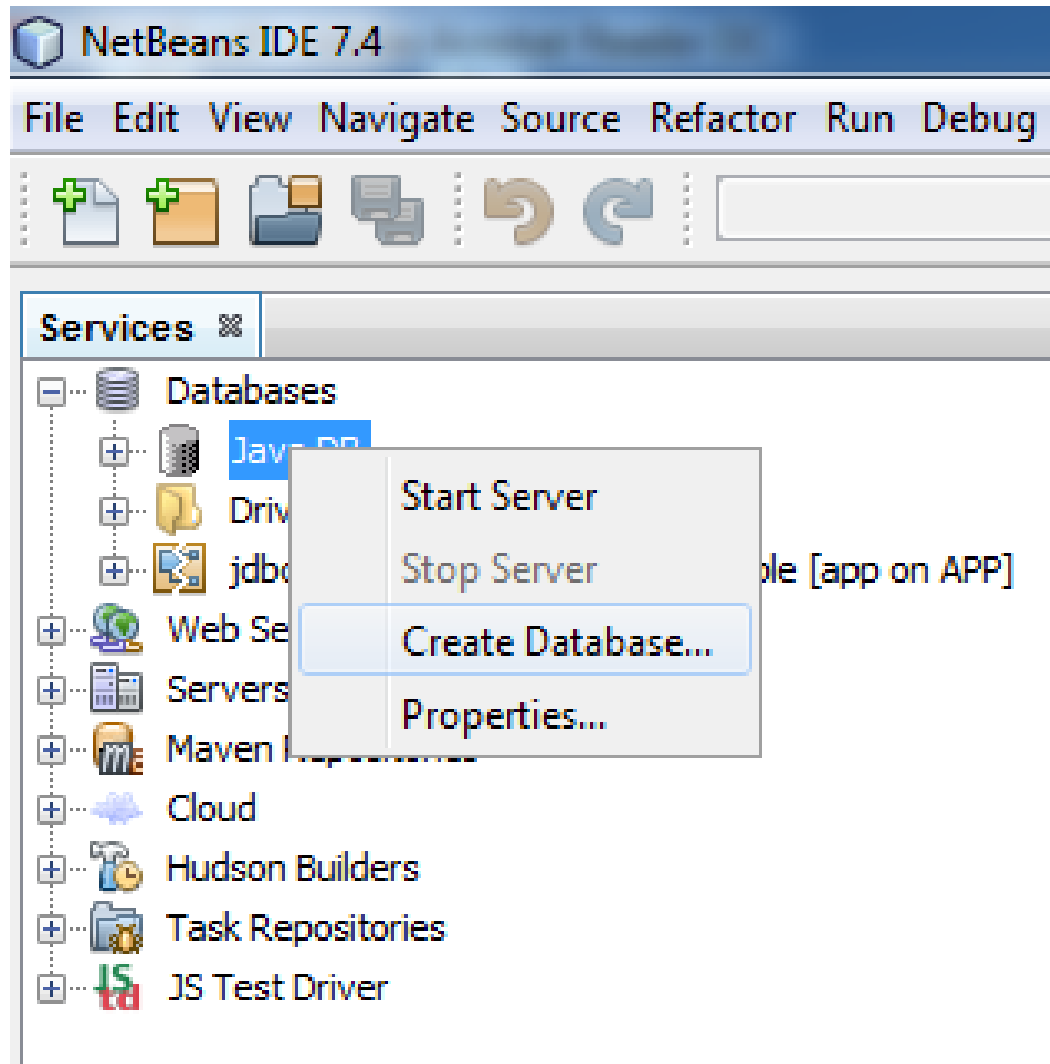
Base de données

- Procédons à créer la base de données.
- Nous optons pour une base de données embarquée JavaDB.

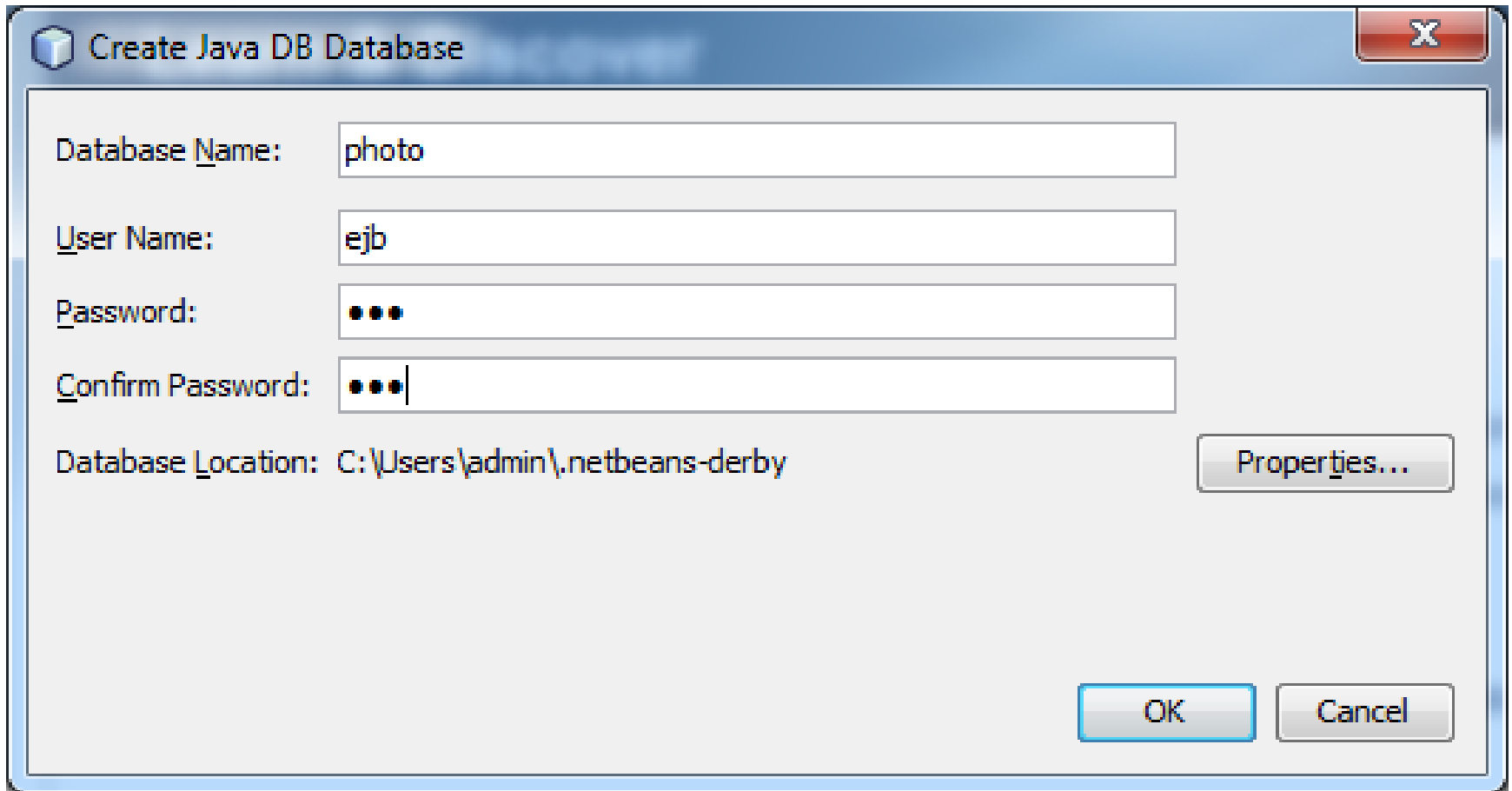
Base de données



Base de données



Base de données



A screenshot of a 'Create Java DB Database' dialog box. The dialog has a title bar with a cube icon and a close button. It contains five input fields: 'Database Name' with 'photo', 'User Name' with 'ejb', 'Password' with three dots, 'Confirm Password' with three dots and a cursor, and 'Database Location' with 'C:\Users\admin\.netbeans-derby'. There is a 'Properties...' button next to the location field, and 'OK' and 'Cancel' buttons at the bottom right.

Create Java DB Database

Database Name: photo

User Name: ejb

Password: ●●●

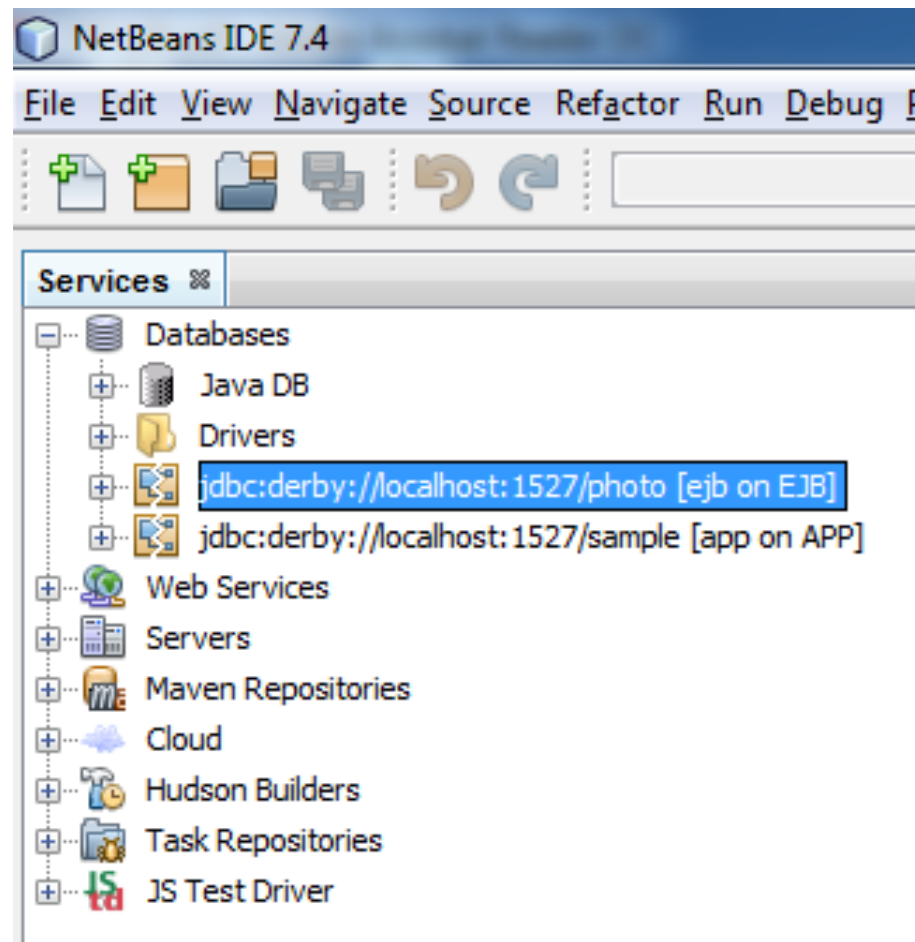
Confirm Password: ●●●|

Database Location: C:\Users\admin\.netbeans-derby

Properties...

OK Cancel

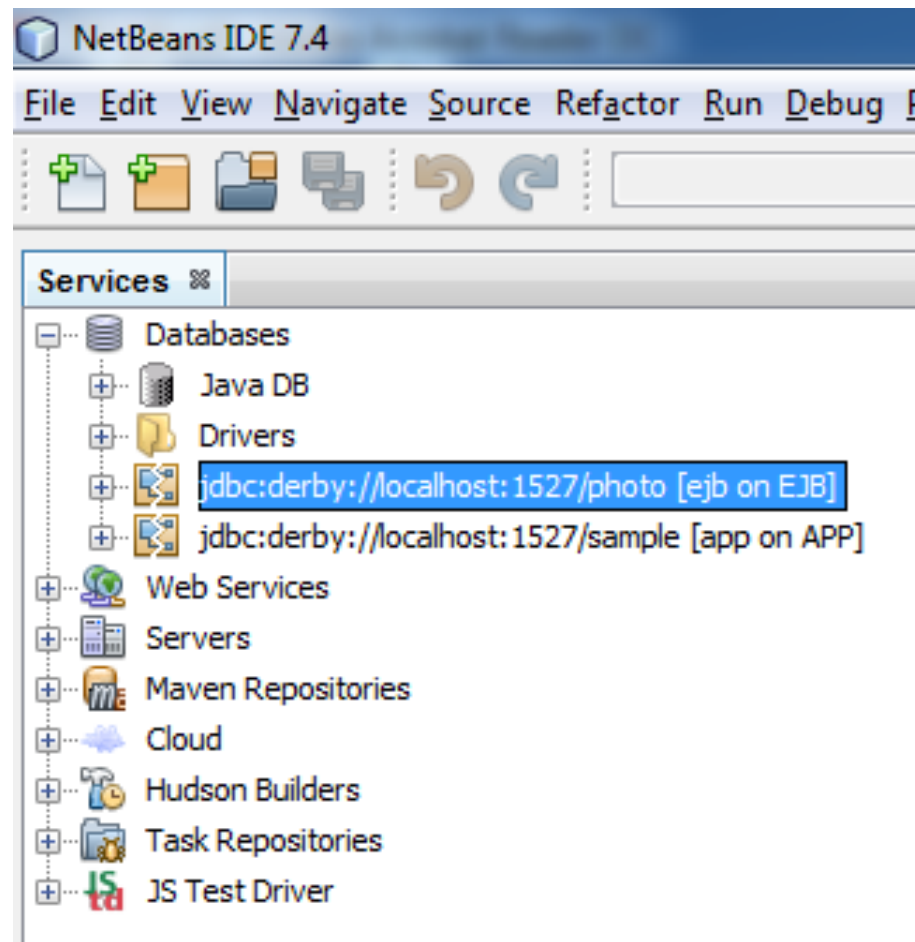
Base de données



Output - Java DB Database Process

```
: Le gestionnaire de sécurité a été installé à l'aide de la stratégie de sécurité de serveur de base.  
: Apache Derby Network Server - 10.10.1.3 - (1557168) démarré et prêt à accepter les connexions sur 1
```

Base de données

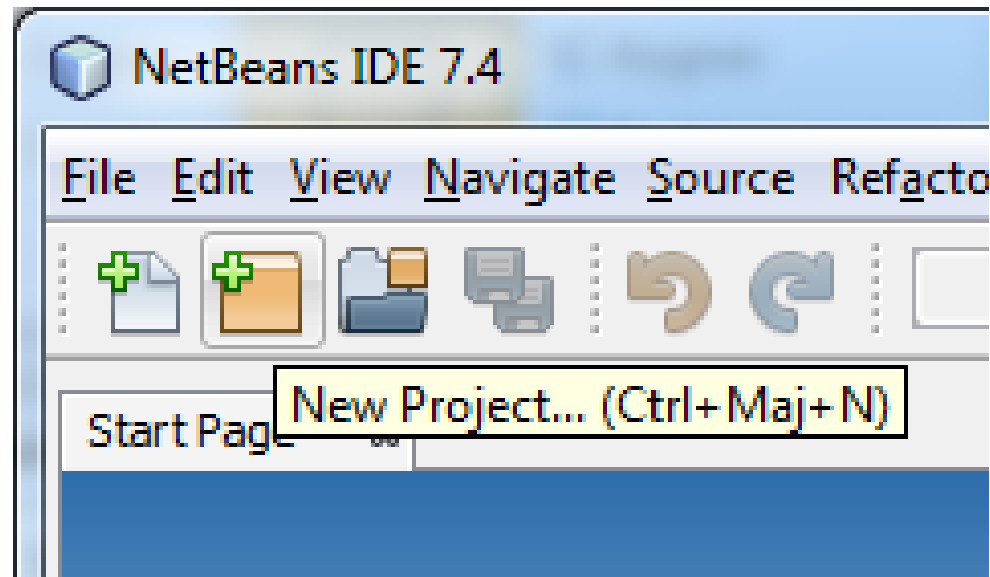


Output - Java DB Database Process

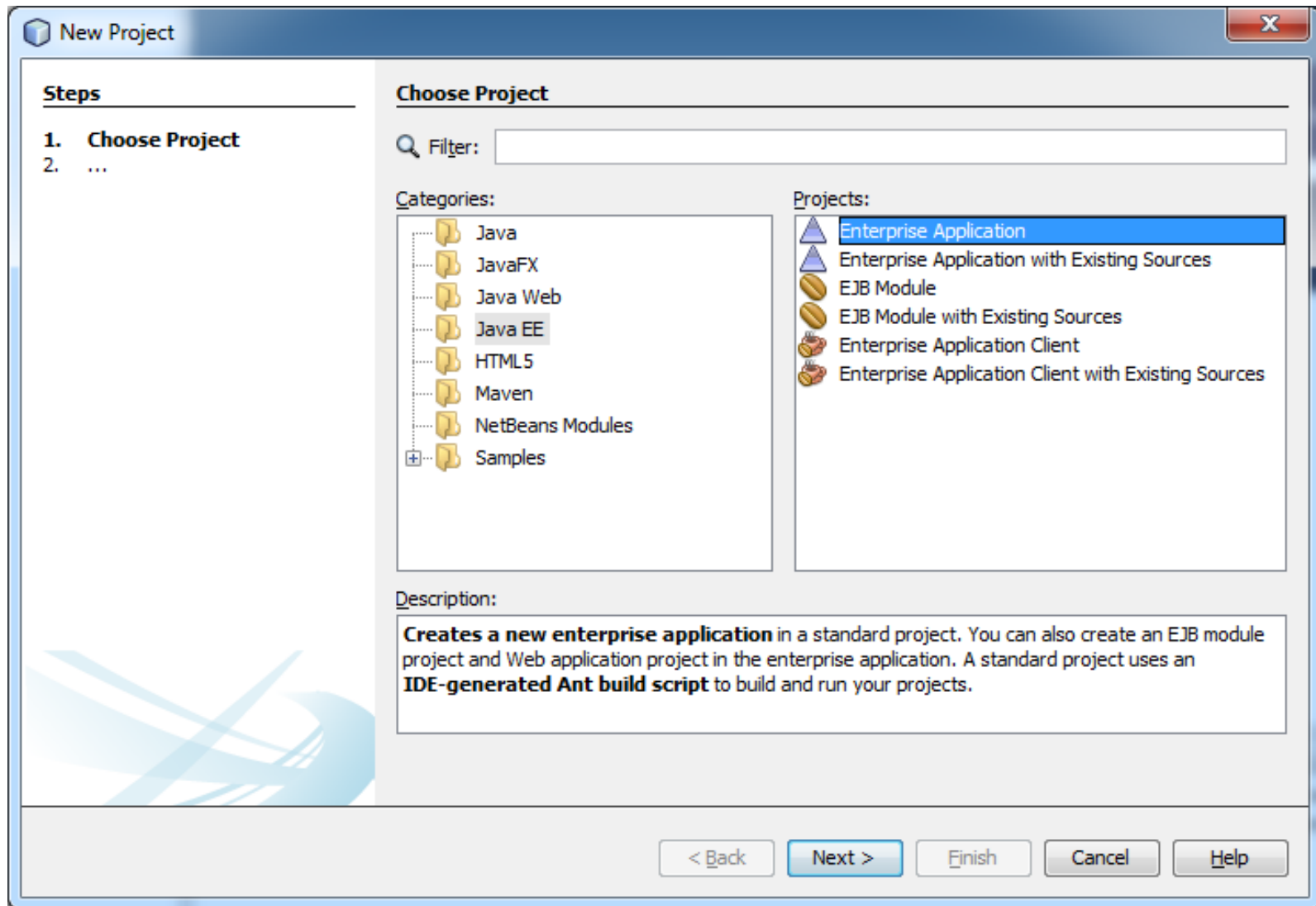
```
: Le gestionnaire de sécurité a été installé à l'aide de la stratégie de sécurité de serveur de base.  
: Apache Derby Network Server - 10.10.1.3 - (1557168) démarré et prêt à accepter les connexions sur 1
```

Projet Archivage-Photos

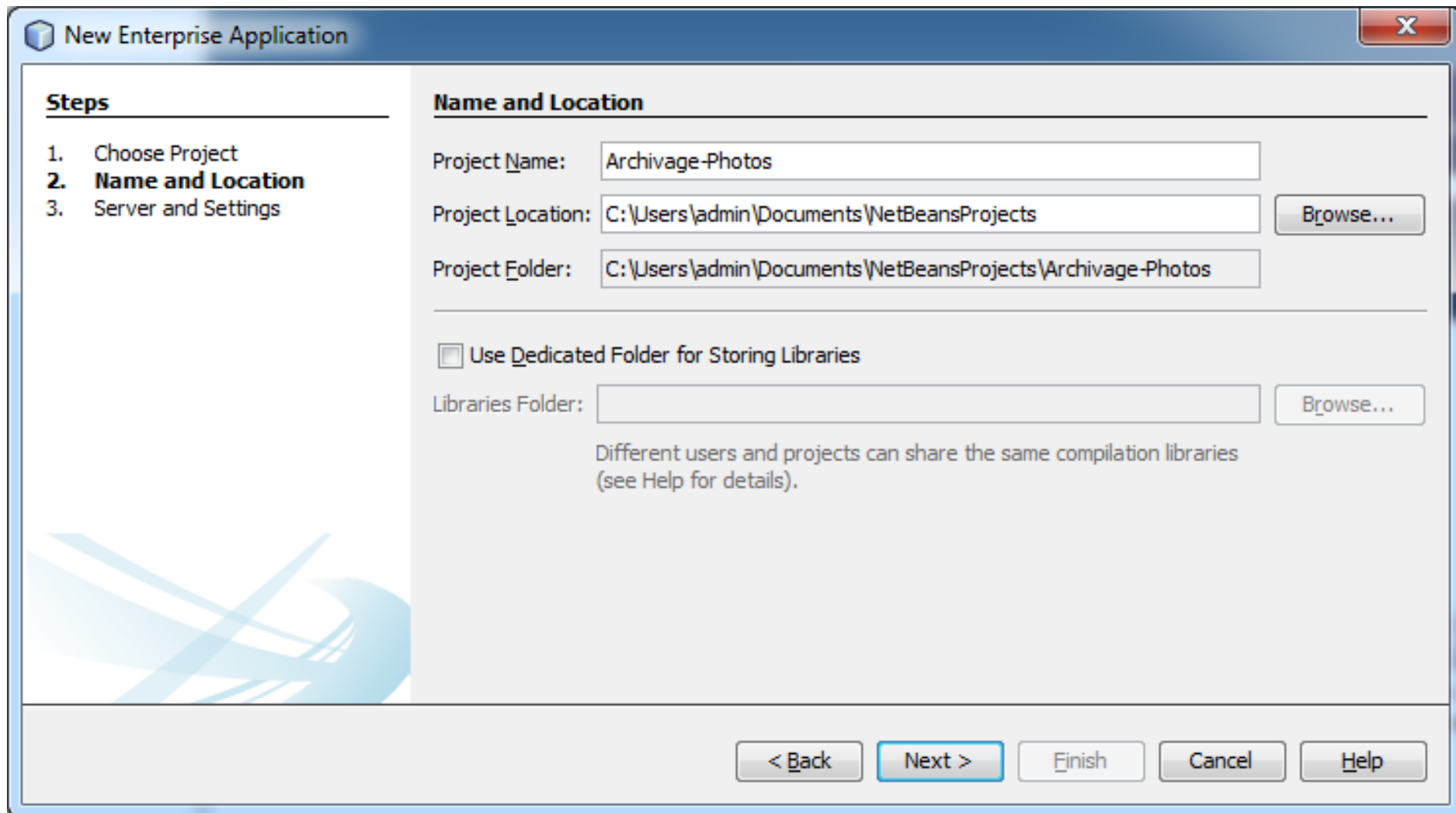
- Créons maintenant notre projet (Enterprise Application)



Project Archivage-Photos



Projet Archivage-Photos



The screenshot shows the 'New Enterprise Application' dialog box. The title bar reads 'New Enterprise Application' with a close button. On the left, a 'Steps' panel lists: 1. Choose Project, 2. **Name and Location**, and 3. Server and Settings. The main area is titled 'Name and Location'. It contains three text fields: 'Project Name' with the value 'Archivage-Photos', 'Project Location' with the value 'C:\Users\admin\Documents\NetBeansProjects', and 'Project Folder' with the value 'C:\Users\admin\Documents\NetBeansProjects\Archivage-Photos'. Each field has a 'Browse...' button to its right. Below these fields is a checkbox labeled 'Use Dedicated Folder for Storing Libraries', which is currently unchecked. To its right is another 'Browse...' button. Below the checkbox, a note states: 'Different users and projects can share the same compilation libraries (see Help for details)'. At the bottom of the dialog are five buttons: '< Back', 'Next >' (highlighted with a blue border), 'Finish', 'Cancel', and 'Help'.

Steps

1. Choose Project
- 2. Name and Location**
3. Server and Settings

Name and Location

Project Name: Archivage-Photos

Project Location: C:\Users\admin\Documents\NetBeansProjects Browse...

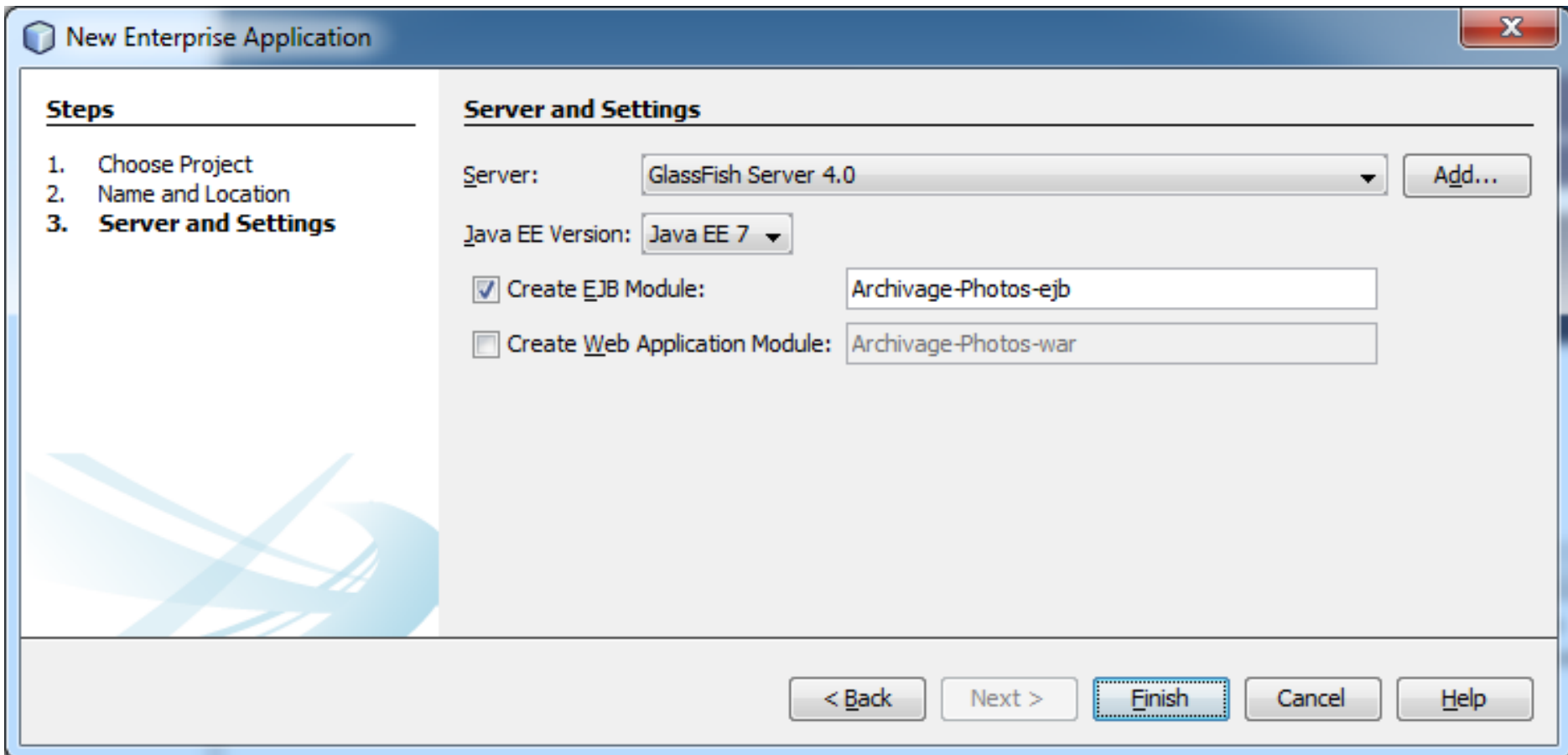
Project Folder: C:\Users\admin\Documents\NetBeansProjects\Archivage-Photos

☐ Use Dedicated Folder for Storing Libraries Browse...

Different users and projects can share the same compilation libraries
(see Help for details).

< Back Next > Finish Cancel Help

Projet Archivage-Photos

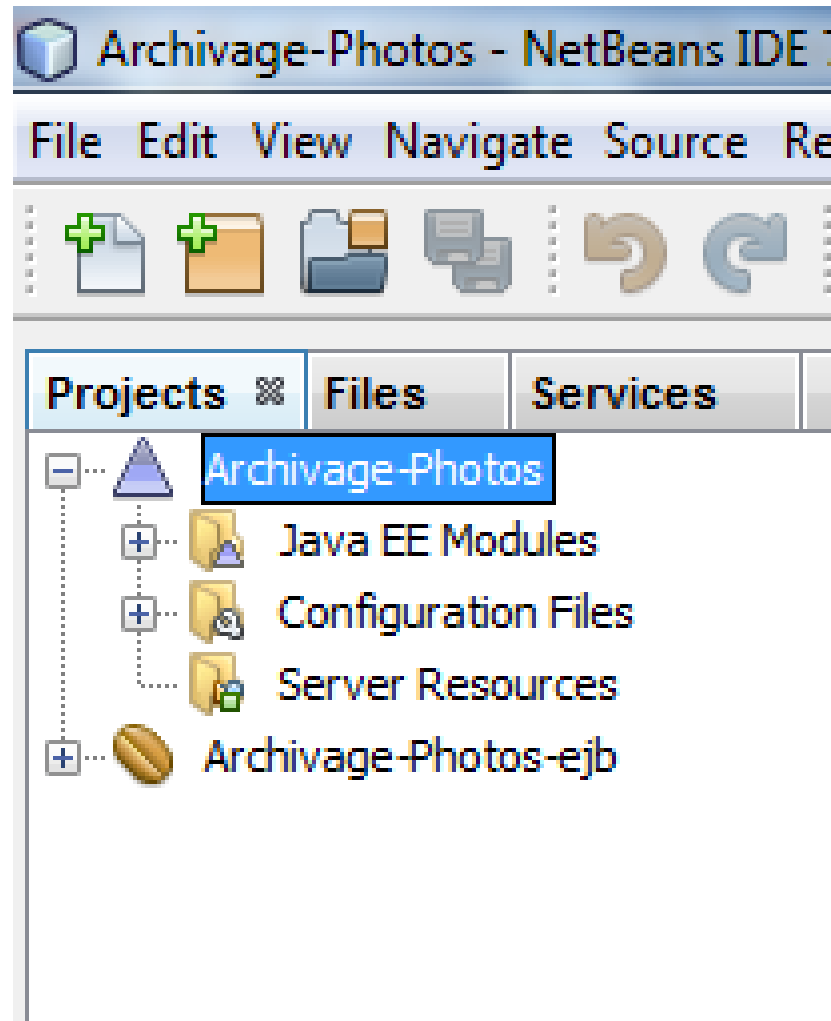


The screenshot shows the 'New Enterprise Application' wizard in the Eclipse IDE. The window title is 'New Enterprise Application'. On the left, a 'Steps' pane lists three steps: '1. Choose Project', '2. Name and Location', and '3. Server and Settings', with the third step being the current active step. The main area is titled 'Server and Settings'. It contains the following fields and controls:

- Server:** A dropdown menu showing 'GlassFish Server 4.0' and an 'Add...' button to its right.
- Java EE Version:** A dropdown menu showing 'Java EE 7'.
- Create EJB Module:** A checked checkbox followed by a text field containing 'Archivage-Photos-ejb'.
- Create Web Application Module:** An unchecked checkbox followed by a text field containing 'Archivage-Photos-war'.

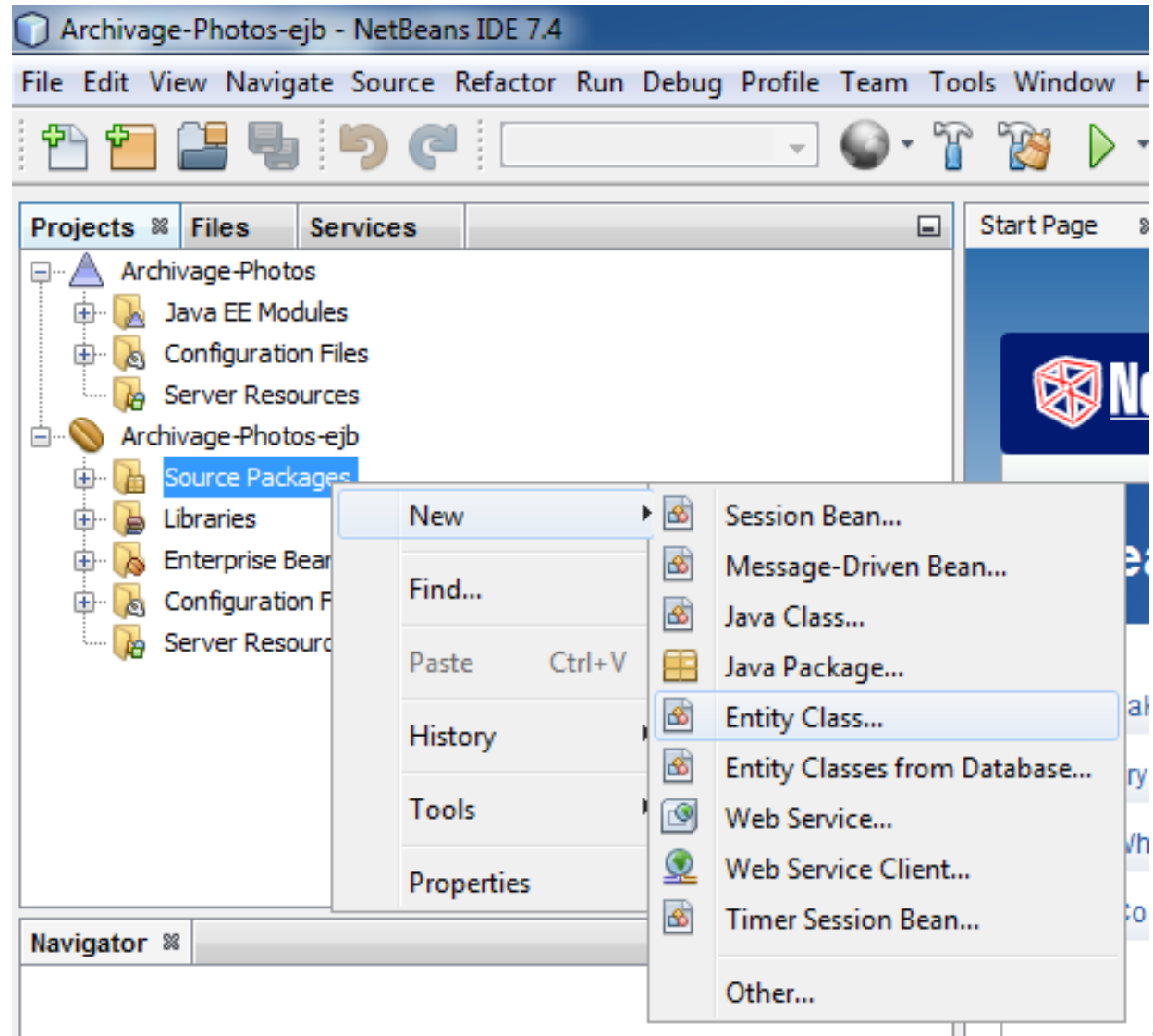
At the bottom of the wizard, there are five buttons: '< Back', 'Next >', 'Finish' (which is highlighted with a dotted border), 'Cancel', and 'Help'.

Projet Archivage-Photos

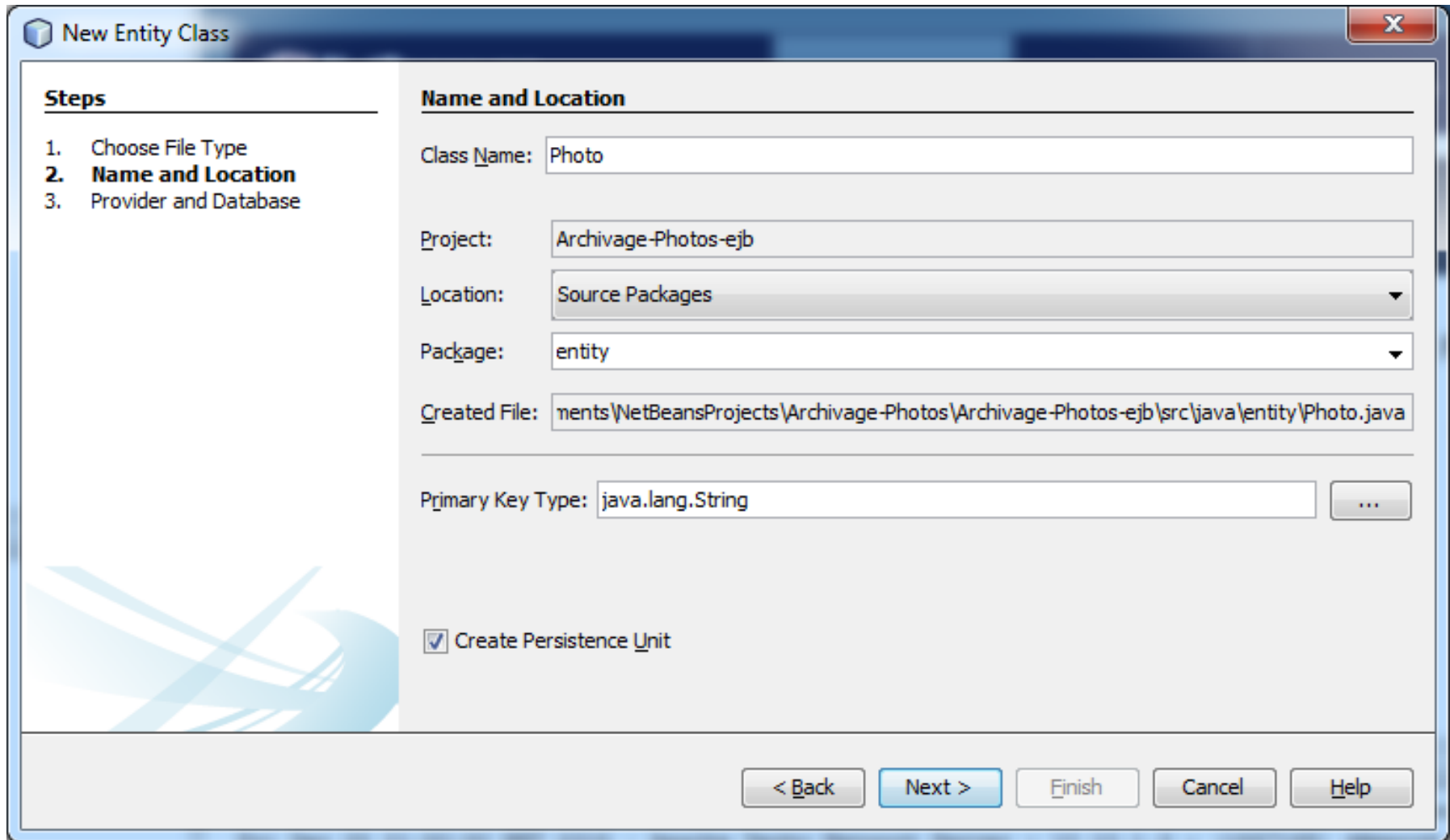


Entity Bean

- Création du bean entité et de l'unité de persistance



Entity Bean



The image shows a 'New Entity Class' dialog box from NetBeans. It has a title bar with a cube icon and a close button. The dialog is divided into two main sections: 'Steps' on the left and 'Name and Location' on the right. The 'Steps' section lists three steps: '1. Choose File Type', '2. Name and Location' (which is currently selected), and '3. Provider and Database'. The 'Name and Location' section contains several input fields: 'Class Name' with the value 'Photo', 'Project' with 'Archivage-Photos-ejb', 'Location' with a dropdown menu showing 'Source Packages', 'Package' with a dropdown menu showing 'entity', and 'Created File' with the path 'ments\NetBeansProjects\Archivage-Photos\Archivage-Photos-ejb\src\java\entity\Photo.java'. Below these fields is a 'Primary Key Type' field with 'java.lang.String' and a button with three dots. At the bottom of the 'Name and Location' section is a checkbox labeled 'Create Persistence Unit' which is checked. At the very bottom of the dialog are five buttons: '< Back', 'Next >', 'Finish', 'Cancel', and 'Help'.

New Entity Class

Steps

1. Choose File Type
- 2. Name and Location**
3. Provider and Database

Name and Location

Class Name: Photo

Project: Archivage-Photos-ejb

Location: Source Packages

Package: entity

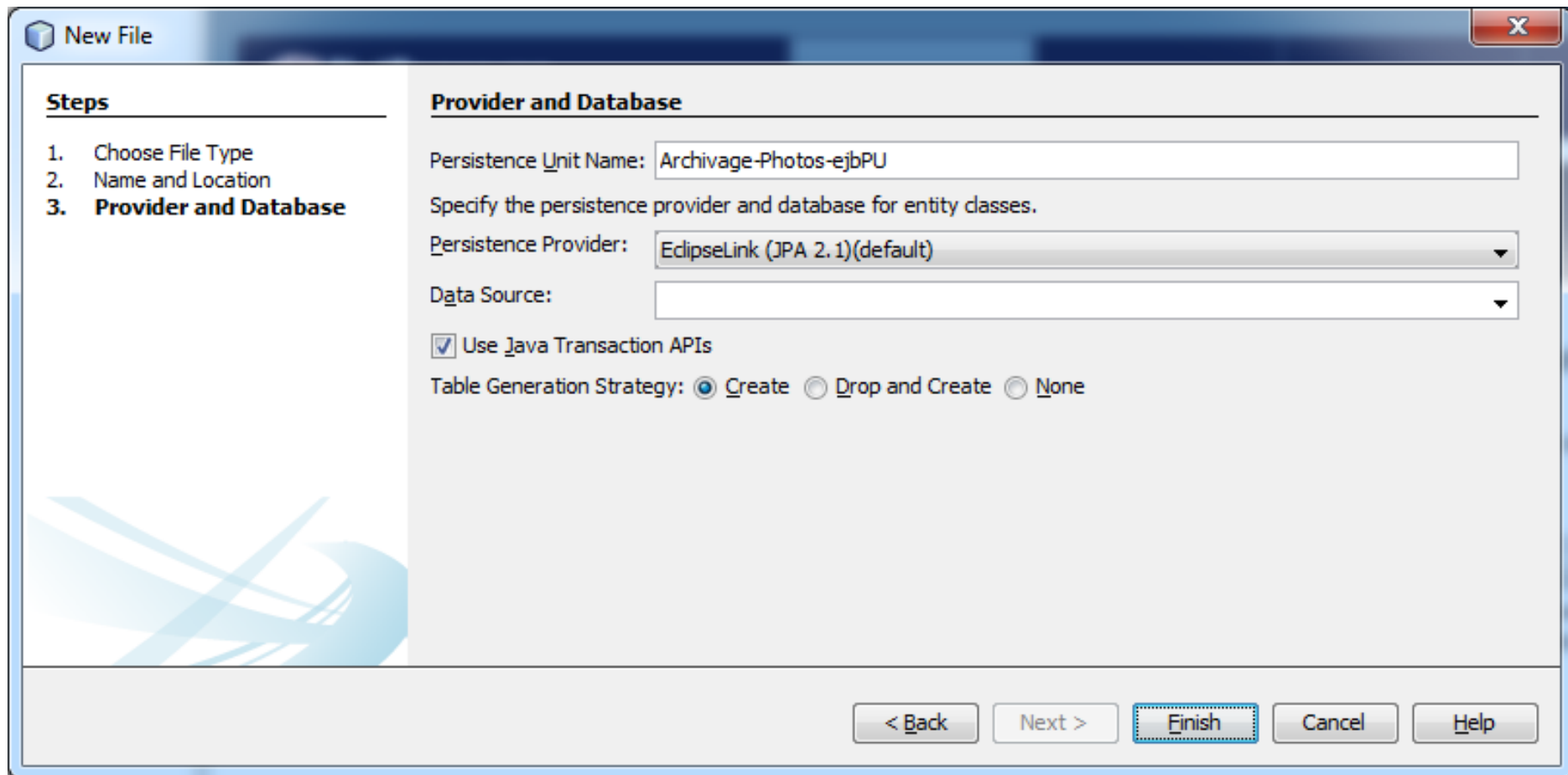
Created File: ments\NetBeansProjects\Archivage-Photos\Archivage-Photos-ejb\src\java\entity\Photo.java

Primary Key Type: java.lang.String

☒ Create Persistence Unit

< Back Next > Finish Cancel Help

Entity Bean



The screenshot shows the 'New File' dialog in the Eclipse IDE. The 'Steps' pane on the left lists three steps: '1. Choose File Type', '2. Name and Location', and '3. Provider and Database', with the third step being the active one. The main area is titled 'Provider and Database' and contains the following fields and options:

- Persistence Unit Name:** A text field containing 'Archivage-Photos-ejbPU'.
- Specify the persistence provider and database for entity classes.**
- Persistence Provider:** A dropdown menu showing 'EclipseLink (JPA 2.1)(default)'.
- Data Source:** A dropdown menu.
- ☒ **Use Java Transaction APIs**
- Table Generation Strategy:** Three radio buttons: 'Create' (selected), 'Drop and Create', and 'None'.

At the bottom of the dialog are five buttons: '< Back', 'Next >', 'Finish' (highlighted with a dashed border), 'Cancel', and 'Help'.

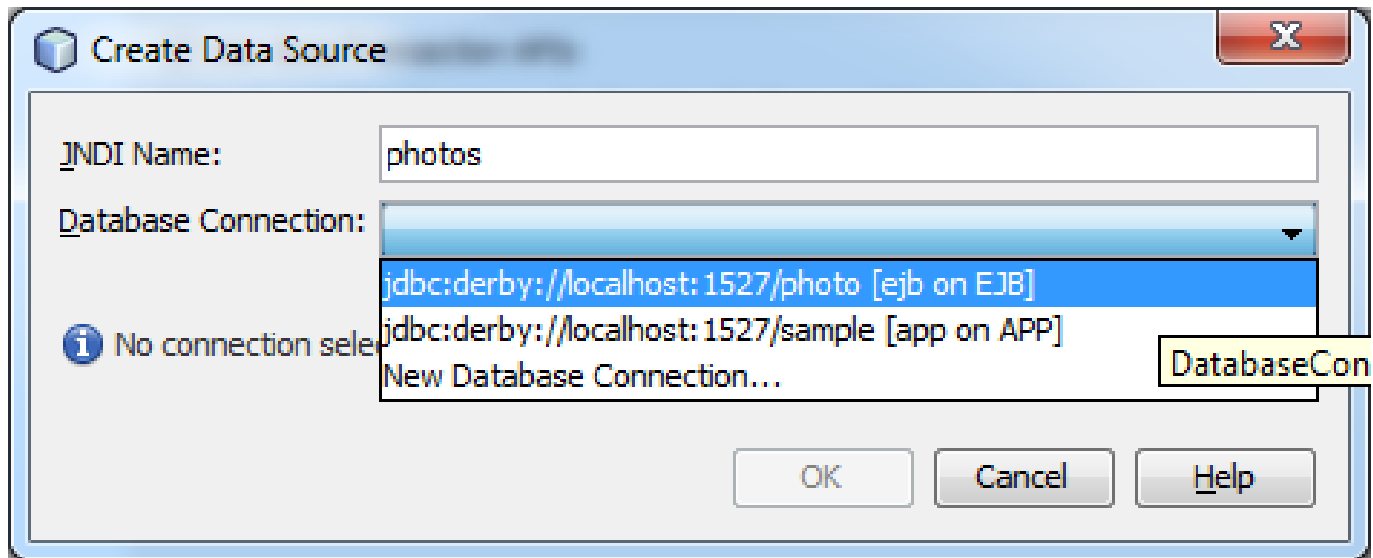
Entity Bean

The screenshot shows the 'New File' dialog in the Eclipse IDE. The 'Steps' panel on the left indicates the current step is '3. Provider and Database'. The main panel is titled 'Provider and Database' and contains the following fields and options:

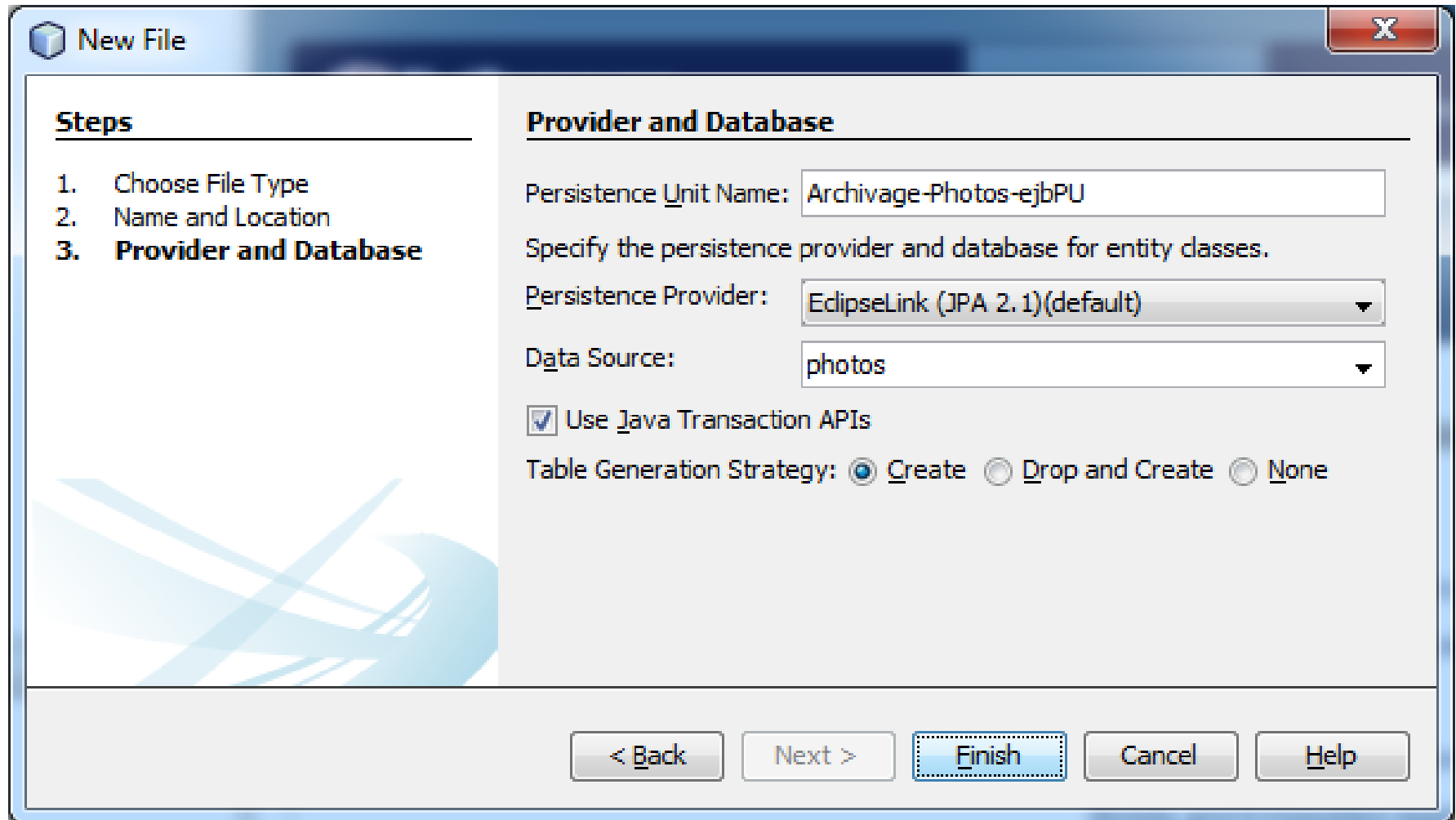
- Persistence Unit Name:** A text field containing 'Archivage-Photos-ejbPU'.
- Specify the persistence provider and database for entity classes.**
- Persistence Provider:** A dropdown menu showing 'EclipseLink (JPA 2.1)(default)'.
- Data Source:** A dropdown menu showing a list of data sources: 'java:comp/DefaultDataSource', 'jdbc/__default', 'jdbc/__TimerPool', 'jdbc/sample', and 'New Data Source...'. The 'New Data Source...' option is currently selected and highlighted in blue.
- ☒ **Use Java Transaction**
- Table Generation Strategy:** (This label is partially obscured by the data source dropdown)

At the bottom of the dialog, there are five buttons: '< Back', 'Next >', 'Finish' (highlighted in blue), 'Cancel', and 'Help'.

Entity Bean



Entity Bean



New File

Steps

1. Choose File Type
2. Name and Location
3. **Provider and Database**

Provider and Database

Persistence Unit Name: Archivage-Photos-ejbPU

Specify the persistence provider and database for entity classes.

Persistence Provider: EclipseLink (JPA 2.1)(default)

Data Source: photos

☒ Use Java Transaction APIs

Table Generation Strategy: ☒ Create ☐ Drop and Create ☐ None

< Back Next > **Finish** Cancel Help

Entity Bean

The screenshot displays an IDE interface with the following components:

- Project Explorer (Left):** Shows a project structure with folders like 'Archivage-Photos', 'Java EE Modules', 'Configuration Files', 'Server Resources', and 'Archivage-Photos-ejb'. Under 'Source Packages', there is an 'entity' package containing 'Photo.java'.
- Photo - Navigator (Bottom Left):** Shows the 'Members' of the 'Photo' class, including 'equals(Object object) : boolean', 'getId() : String', 'hashCode() : int', and 'setId(String id)'.
- Source Editor (Right):** Displays the source code of 'Photo.java'.

```
7 package entity;
8
9 import java.io.Serializable;
10 import javax.persistence.Entity;
11 import javax.persistence.GeneratedValue;
12 import javax.persistence.GenerationType;
13 import javax.persistence.Id;
14
15 /**
16  *
17  * @author admin
18  */
19 @Entity
20 public class Photo implements Serializable {
21     private static final long serialVersionUID = 1L;
22     @Id
23     @GeneratedValue(strategy = GenerationType.AUTO)
```

Voir le code source photo.java

Entity Bean

Photo.java

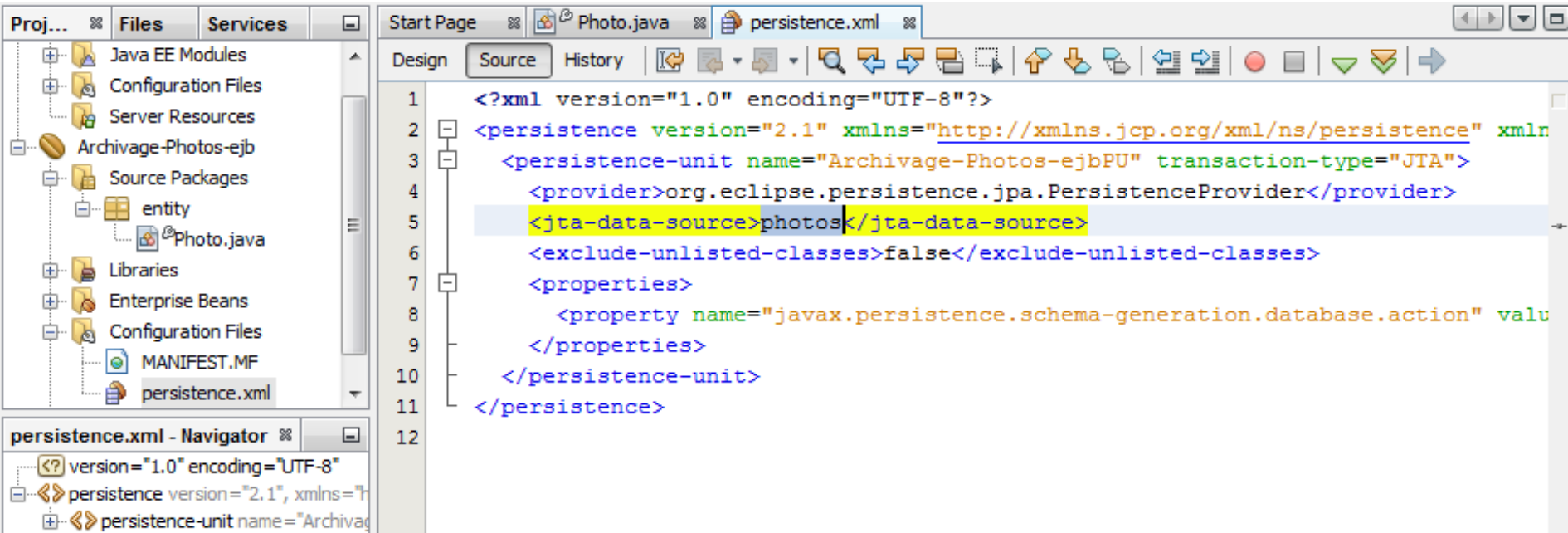
- Remarquer que Netbeans propose systématiquement que chaque entité implémente l'interface Serializable.
- Choix est parfaitement judicieux si on désire que cette information se propage sur le réseau afin que le client puisse en bénéficier.

Entity Bean

The screenshot displays the Eclipse IDE interface for configuring an Entity Bean. The left sidebar shows the project structure with 'Archivage-Photos-ejb' and its 'entity' package containing 'Photo.java'. The 'persistence.xml' file is selected in the 'persistence.xml - Navigator' view. The main editor shows the 'General' tab of the configuration for 'persistence.xml'. The configuration includes:

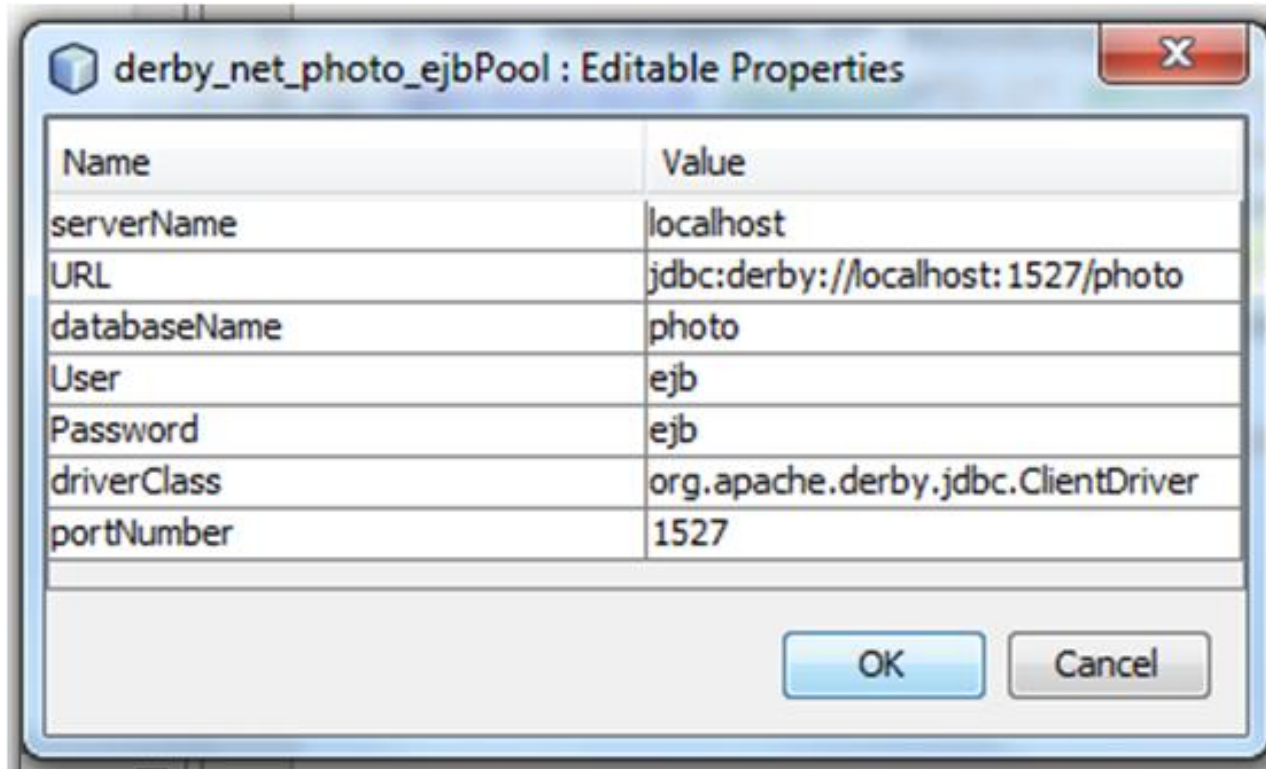
- Persistence Unit Name:** Archivage-Photos-ejbPU
- Persistence Provider:** EclipseLink (JPA 2.1)(default)
- Data Source:** photos
- ☒ **Use Java Transaction APIs**
- Table Generation Strategy:** ☒ Create ☐ Drop and Create ☐ None
- Validation Strategy:** ☒ Auto ☐ Callback ☐ None
- Shared Cache Mode:** ☐ All ☐ None ☐ Enable Selective ☐ Disable Selective ☒ Unspecified
- ☒ **Include All Entity Classes in "Archivage-Photos-ejb" Module**
- Include Entity Classes:** (Empty text box)

Entity Bean



- L'unité de persistance Archivage-Photos-ejbPU fait référence au nom JNDI photos qui lui même fait référence au pool de connexion derby_net_photos_ejbPool (nom par défaut proposé par Netbeans).

Entity Bean



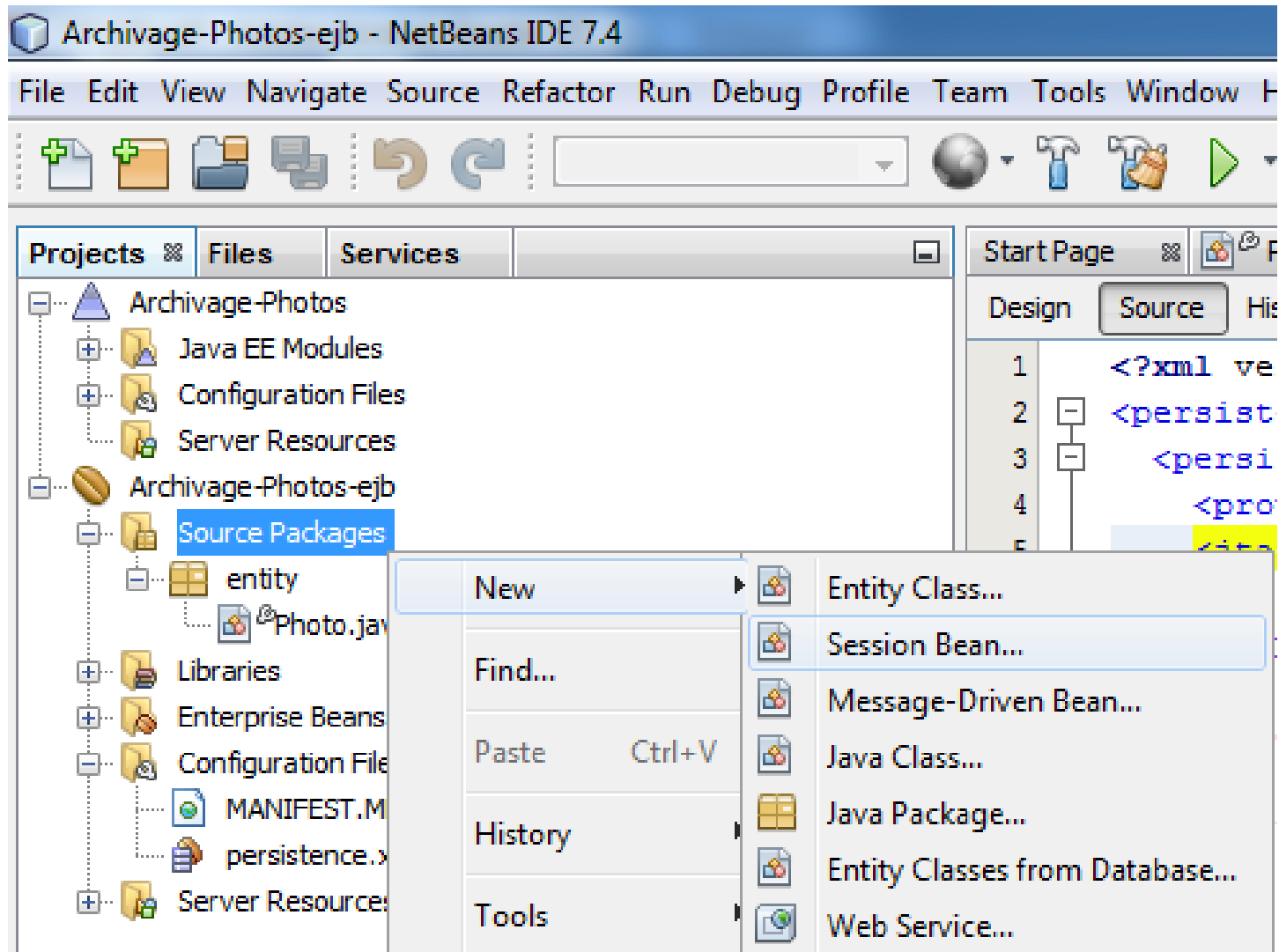
derby_net_photos_ejbPool possède toutes les caractéristiques essentielles de la base de données : serveur utilisé, driver associé, numéro de service, nom de la base, nom d'utilisateur,

Session Bean

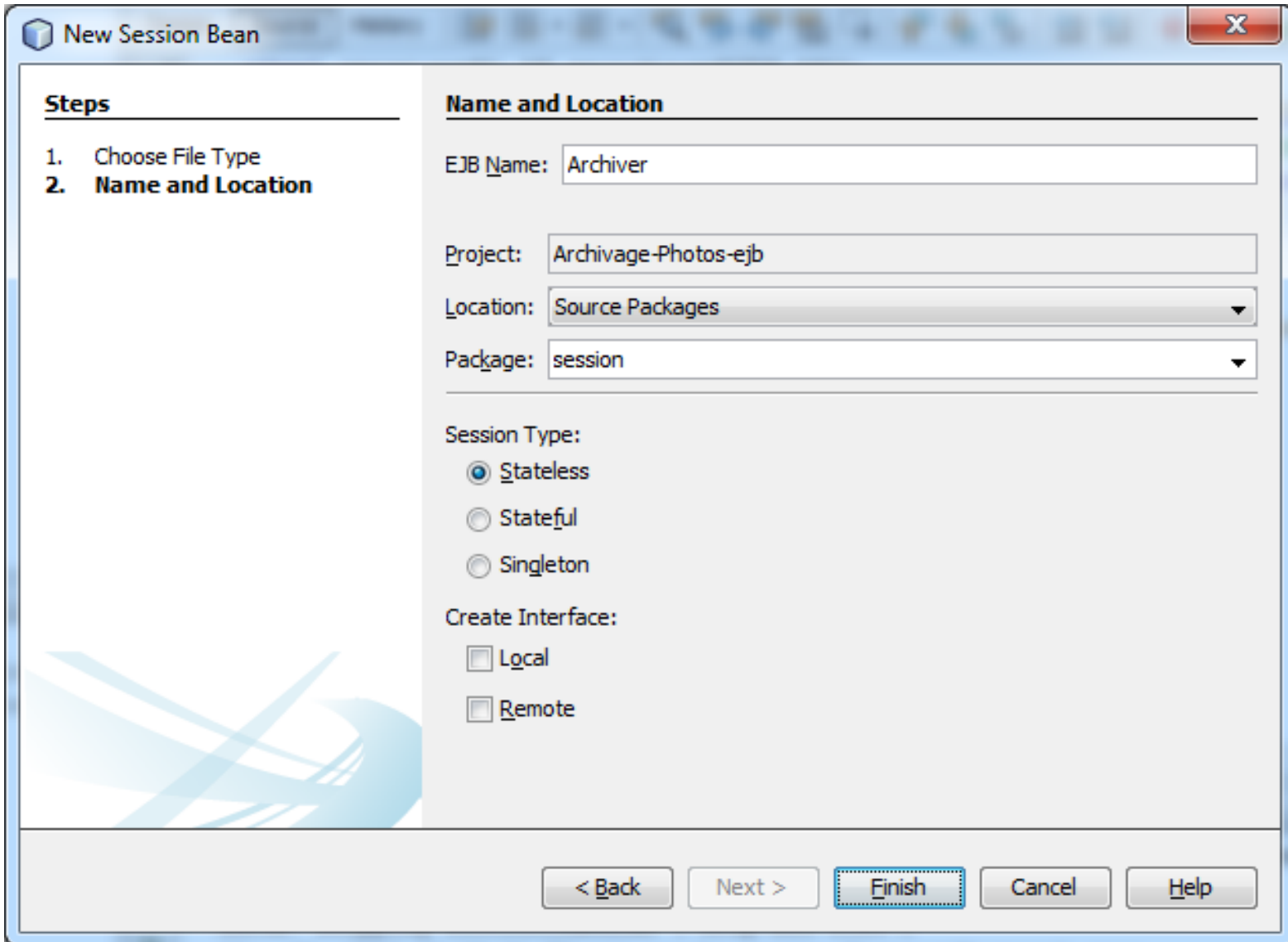
Création du bean session Archiver

- Il va s'occuper de tout le traitement nécessaire côté serveur :
 - enregistrement physique des différentes photos sur le disque dur du serveur
 - gestion complète de la persistance des entités relatives aux informations supplémentaires des photos archivées.

Session Bean



Session Bean



The image shows a 'New Session Bean' wizard dialog box. It has a title bar with a cube icon and the text 'New Session Bean'. The dialog is divided into two main sections. The left section, titled 'Steps', contains a list of two steps: '1. Choose File Type' and '2. Name and Location', with the second step being the current one. The right section, titled 'Name and Location', contains several input fields and options. The 'EJB Name' field is set to 'Archiver'. The 'Project' field is set to 'Archivage-Photos-ejb'. The 'Location' dropdown menu is set to 'Source Packages'. The 'Package' dropdown menu is set to 'session'. Below these fields, there are three radio buttons for 'Session Type': 'Stateless' (selected), 'Stateful', and 'Singleton'. At the bottom of this section, there are two checkboxes for 'Create Interface': 'Local' and 'Remote'. At the very bottom of the dialog, there are five buttons: '< Back', 'Next >', 'Finish' (highlighted with a dashed border), 'Cancel', and 'Help'.

New Session Bean

Steps

1. Choose File Type
- 2. Name and Location**

Name and Location

EJB Name:

Project:

Location:

Package:

Session Type:

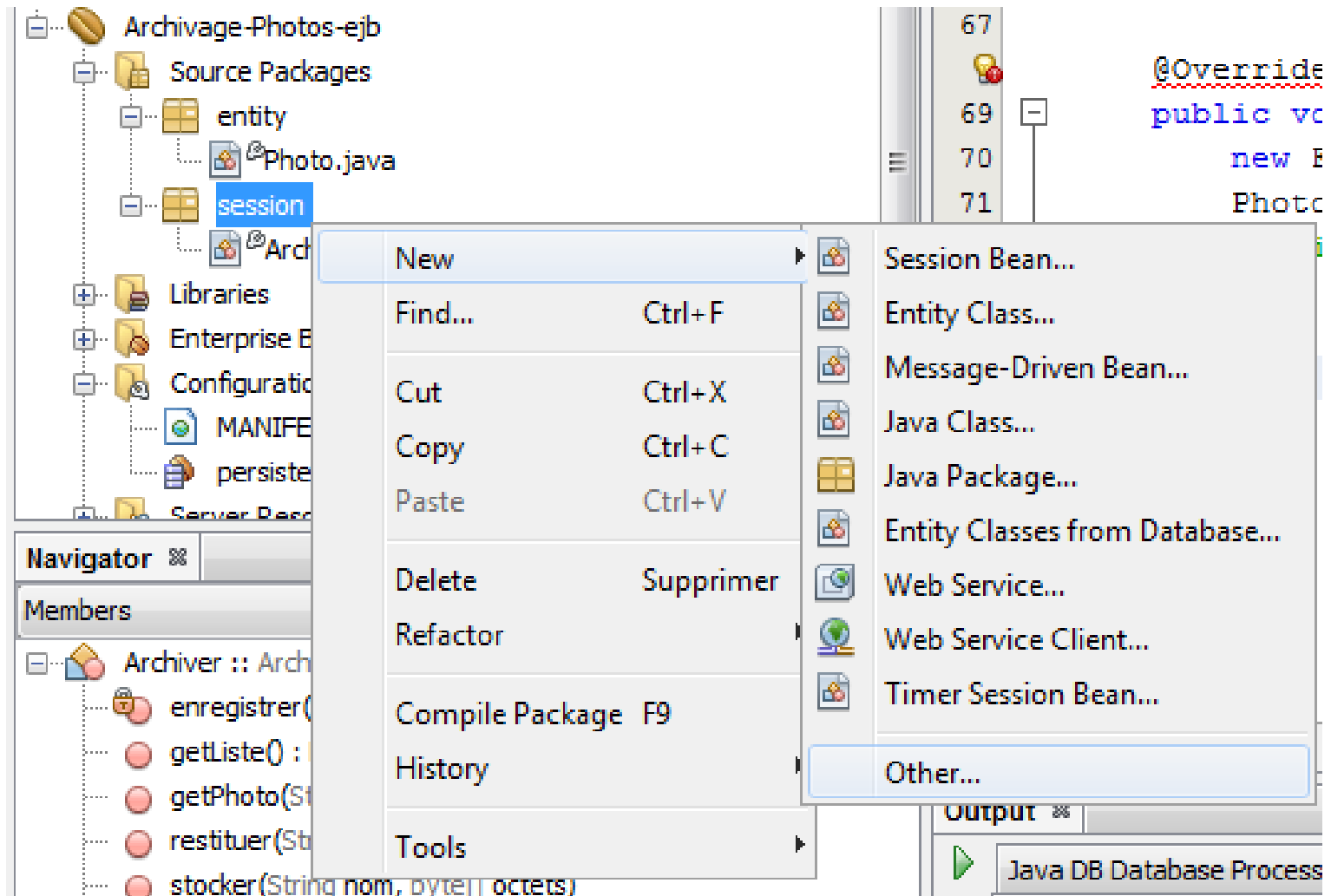
- ☒ Stateless
- ☐ Stateful
- ☐ Singleton

Create Interface:

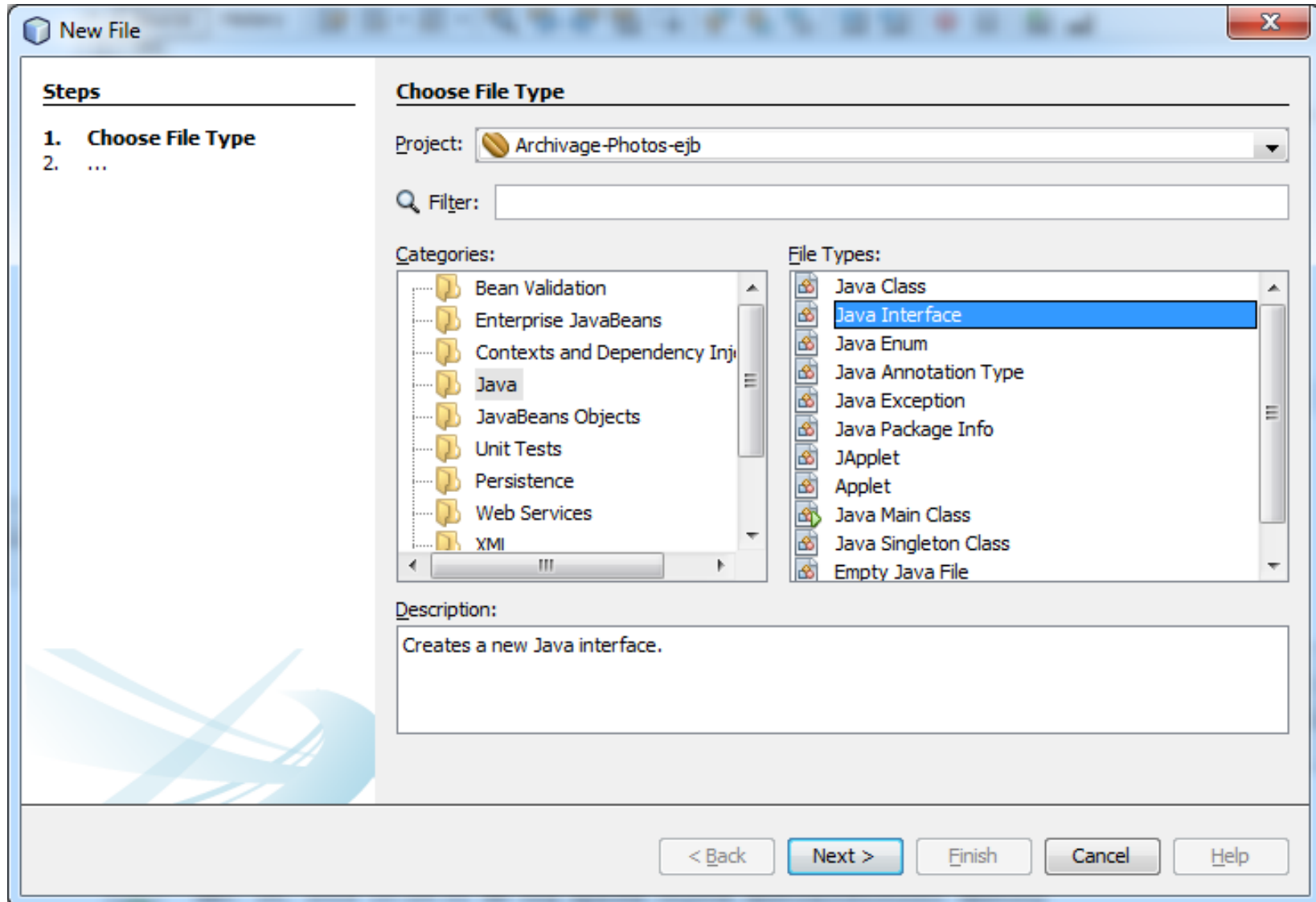
- ☐ Local
- ☐ Remote

< Back Next > **Finish** Cancel Help

Session Bean



Session Bean



Session Bean

New Java Interface

Steps

1. Choose File Type
2. **Name and Location**

Name and Location

Class Name:

Project:

Location:

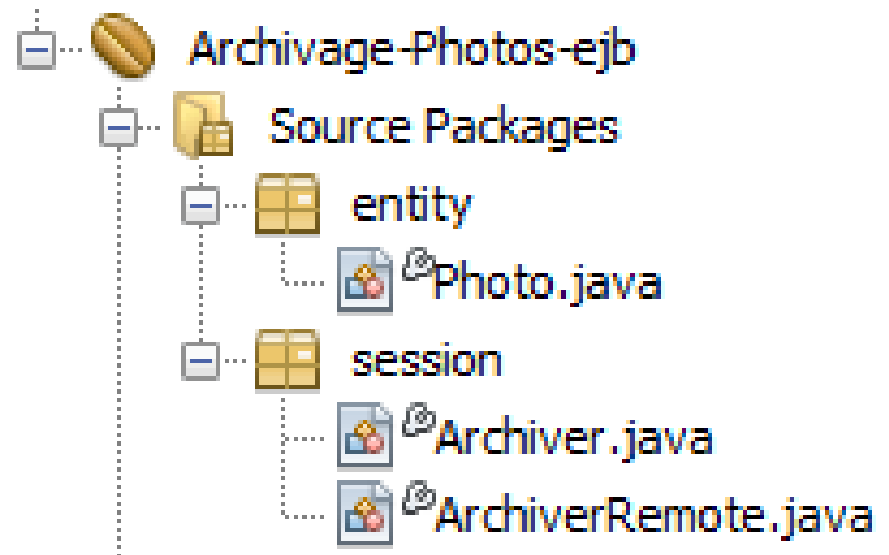
Package:

Created File:

< Back Next > **Finish** Cancel Help

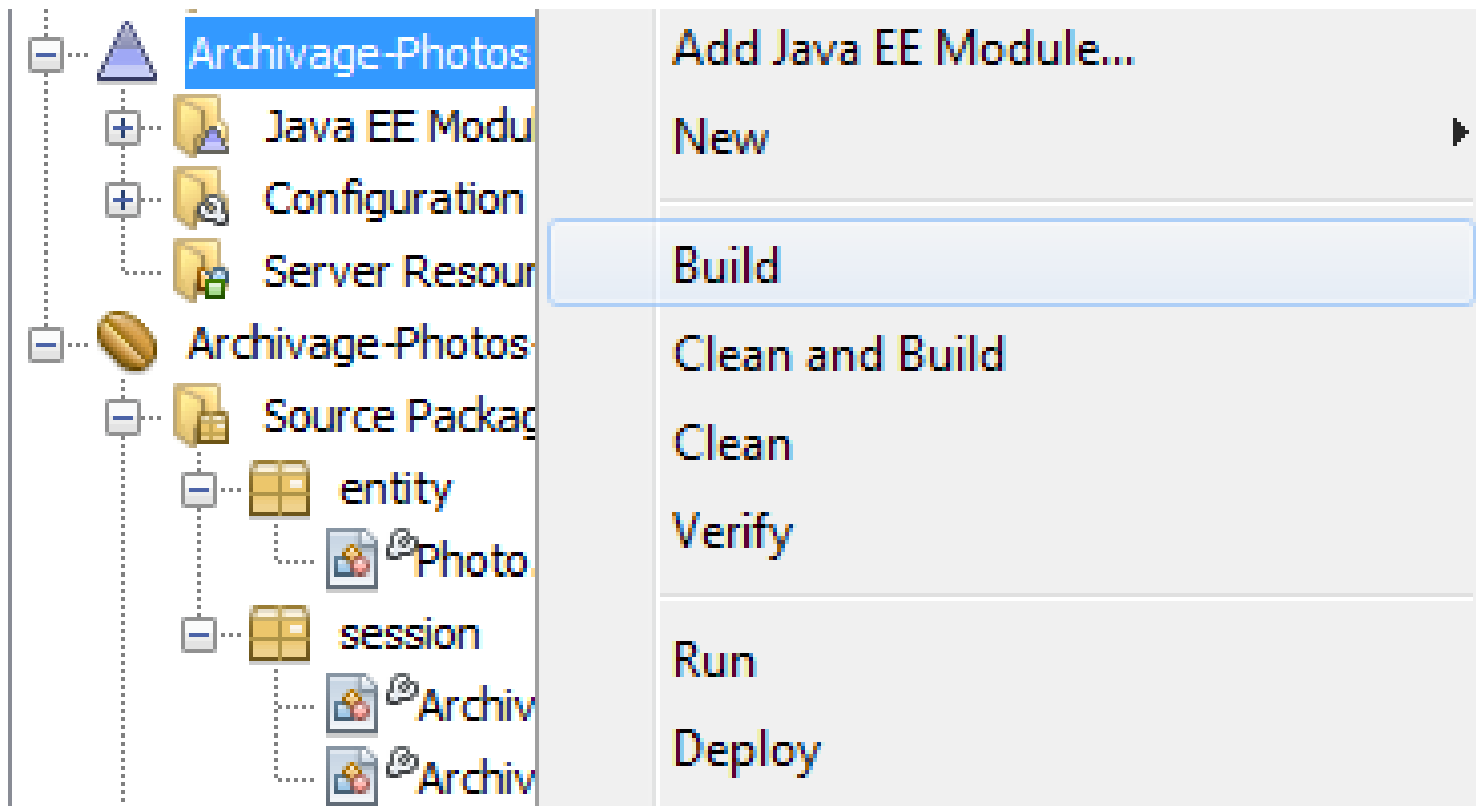
Session Bean

- Voir les sources :
 - Archiver.java
 - ArchiverRemote.java
- créer le dossier
D:\Archivage

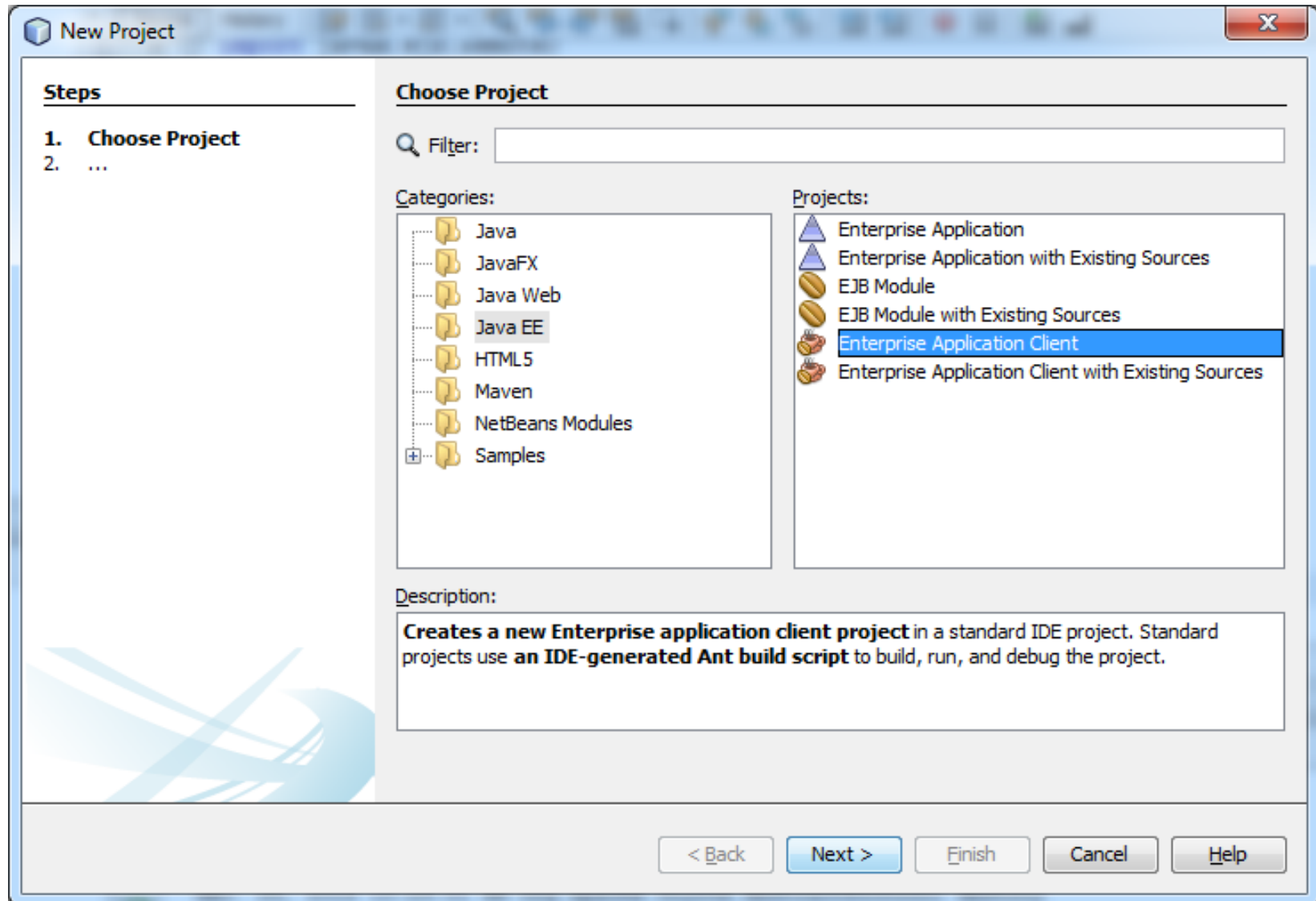


Build

- Faire un premier build



L'application cliente



L'application cliente

New Enterprise Application Client

Steps

1. Choose Project
- 2. Name and Location**
3. Server and Settings

Name and Location

Project Name: AppClient

Project Location: C:\Users\admin\Documents\NetBeansProjects [Browse...](#)

Project Folder: C:\Users\admin\Documents\NetBeansProjects\AppDataClient

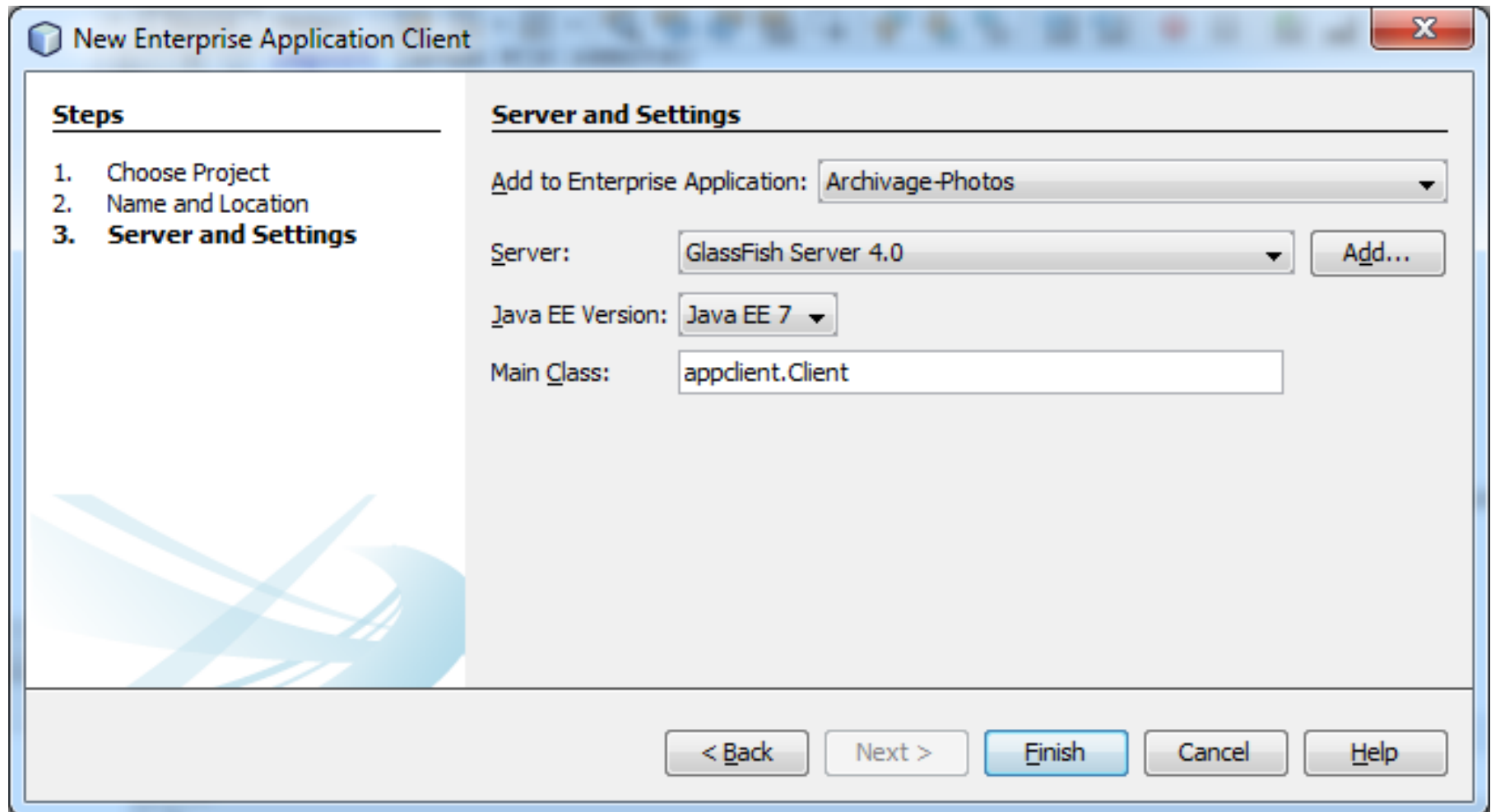
☐ Use Dedicated Folder for Storing Libraries

Libraries Folder: [Browse...](#)

Different users and projects can share the same compilation libraries (see Help for details).

< Back Next > Finish Cancel Help

L'application cliente



The screenshot shows a Java IDE wizard titled "New Enterprise Application Client". The window is divided into two main sections. On the left, a "Steps" pane lists three steps: "1. Choose Project", "2. Name and Location", and "3. Server and Settings", with the third step being the current active step. On the right, the "Server and Settings" section contains several configuration options. At the top, there is a dropdown menu labeled "Add to Enterprise Application:" with "Archivage-Photos" selected. Below this, the "Server:" dropdown is set to "GlassFish Server 4.0", with an "Add..." button to its right. The "Java EE Version:" dropdown is set to "Java EE 7". The "Main Class:" text field contains the value "appclient.Client". At the bottom of the window, there are five buttons: "< Back", "Next >", "Finish" (which is highlighted in blue), "Cancel", and "Help".

New Enterprise Application Client

Steps

1. Choose Project
2. Name and Location
3. **Server and Settings**

Server and Settings

Add to Enterprise Application: Archivage-Photos

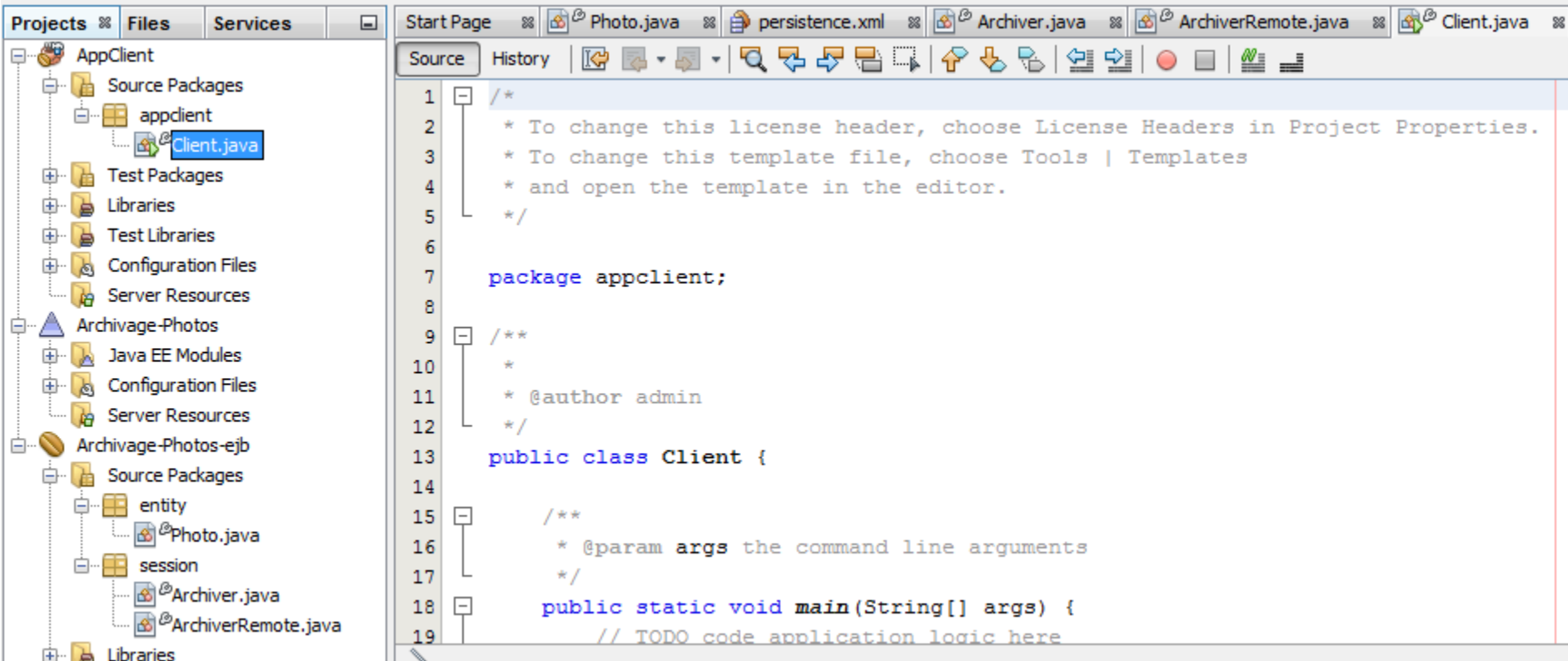
Server: GlassFish Server 4.0 Add...

Java EE Version: Java EE 7

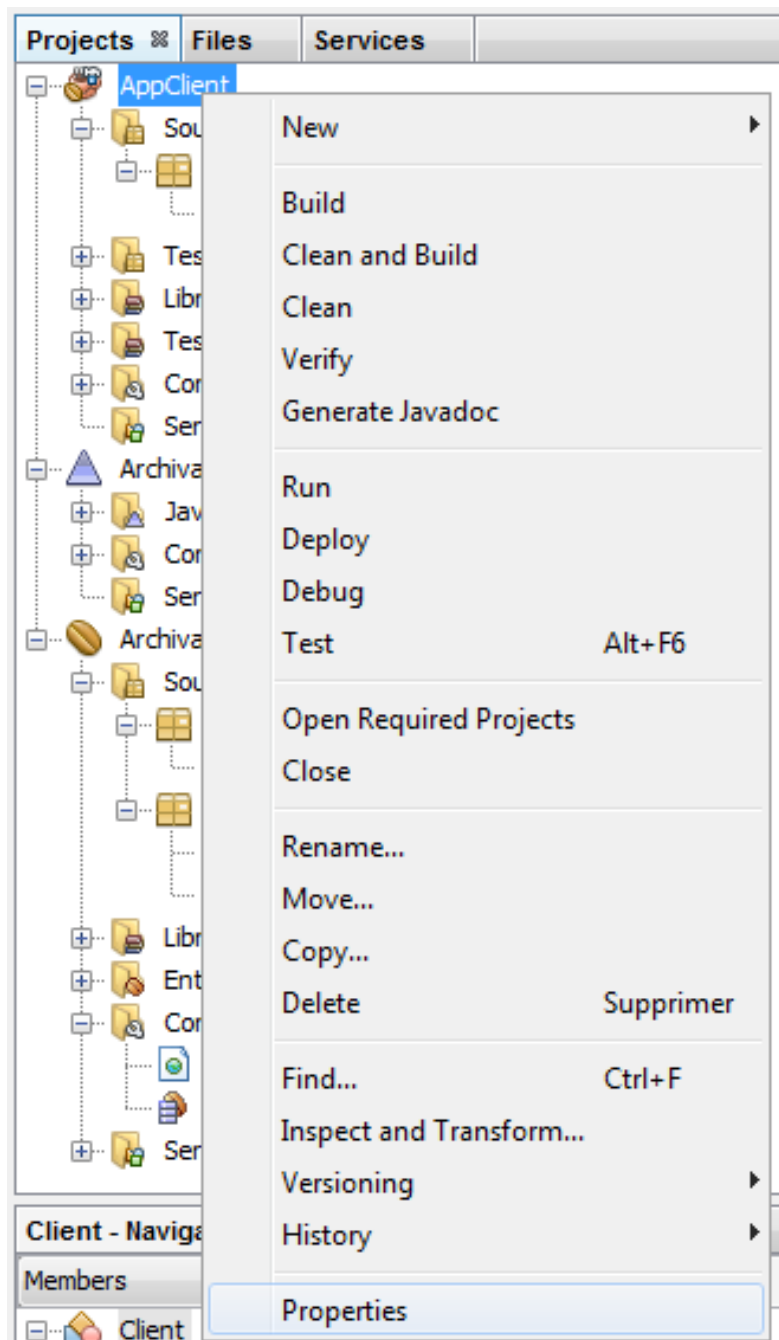
Main Class: appclient.Client

< Back Next > Finish Cancel Help

L'application cliente

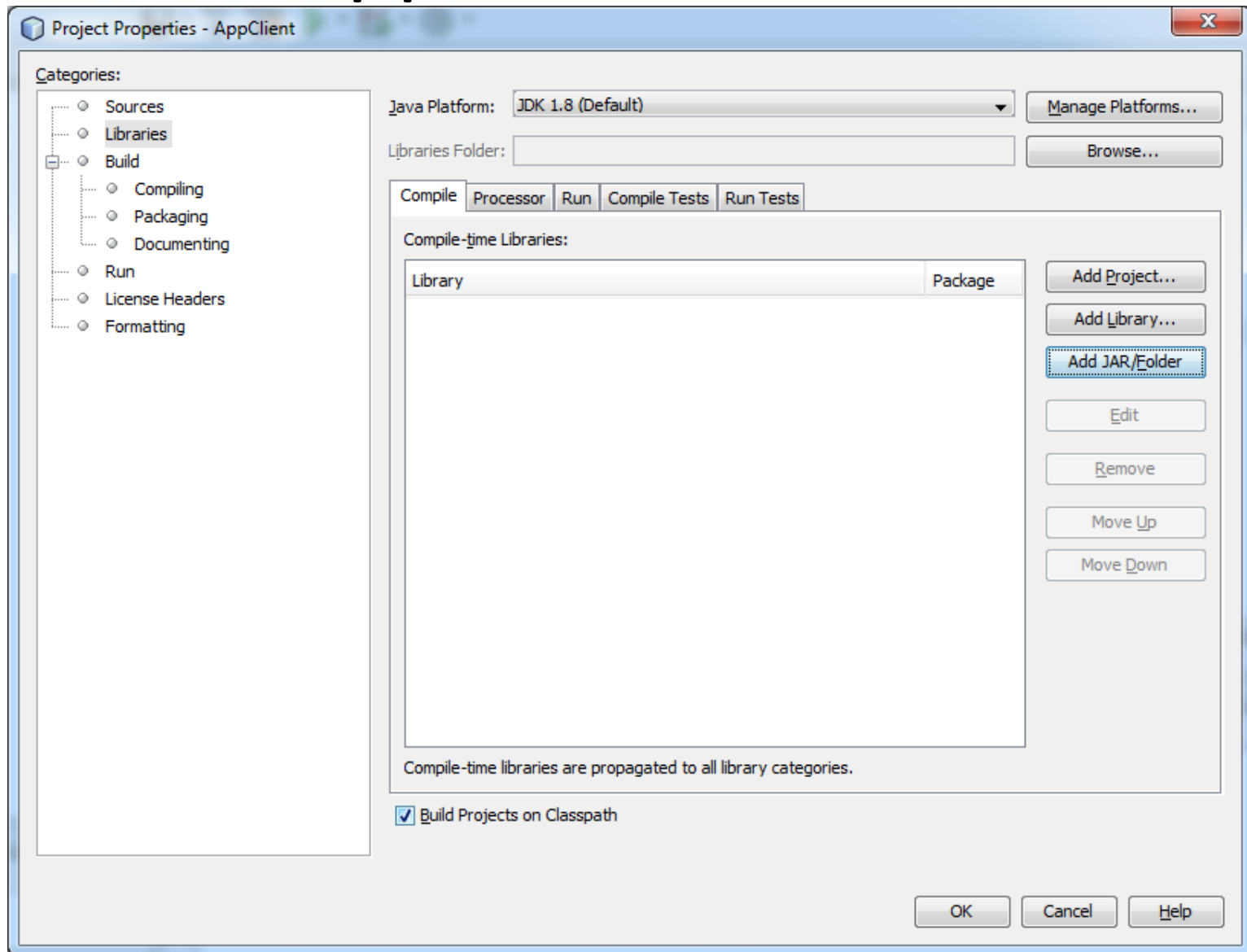


Voir le code source Client.java

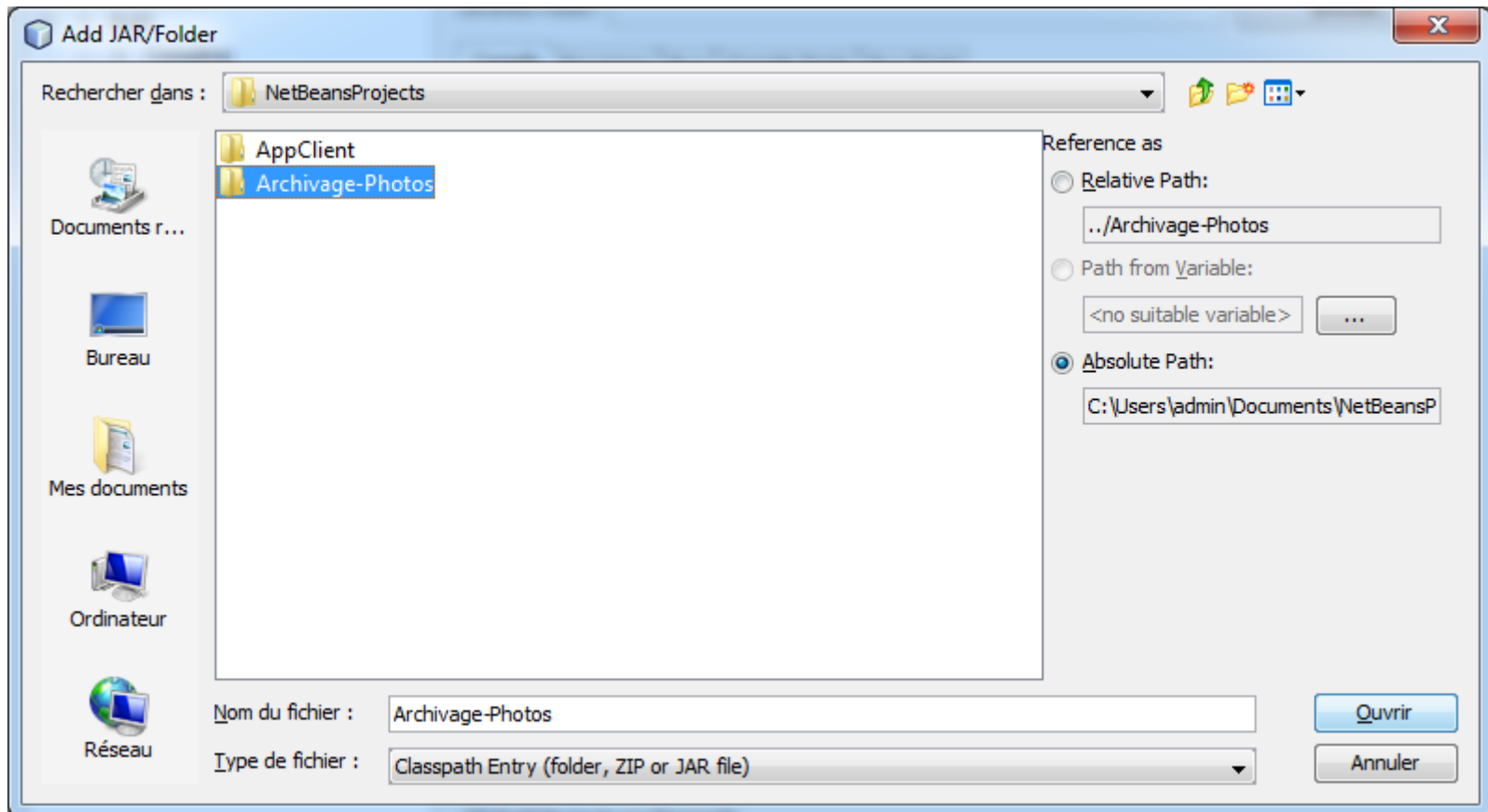


L'application cliente

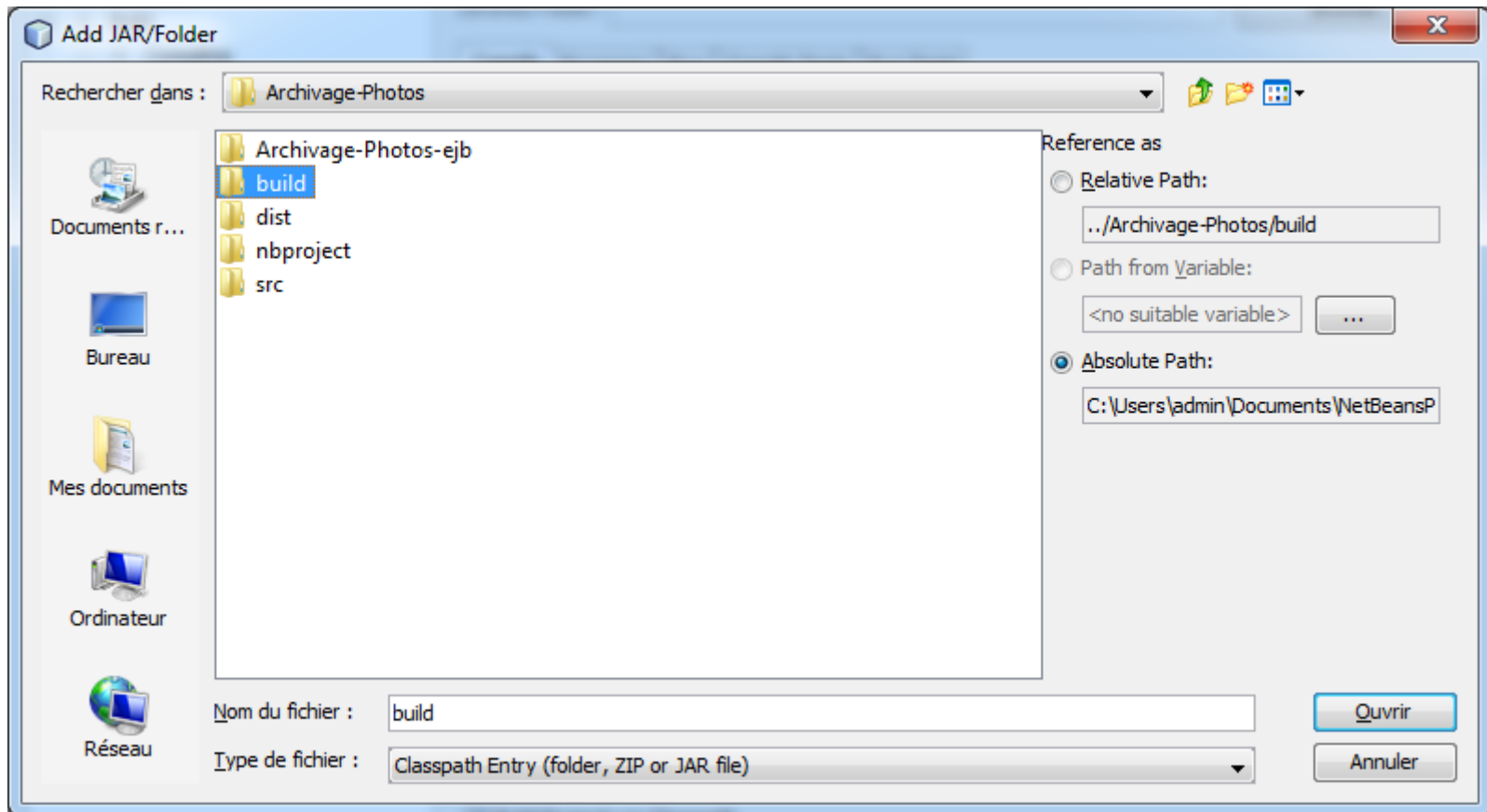
L'application cliente



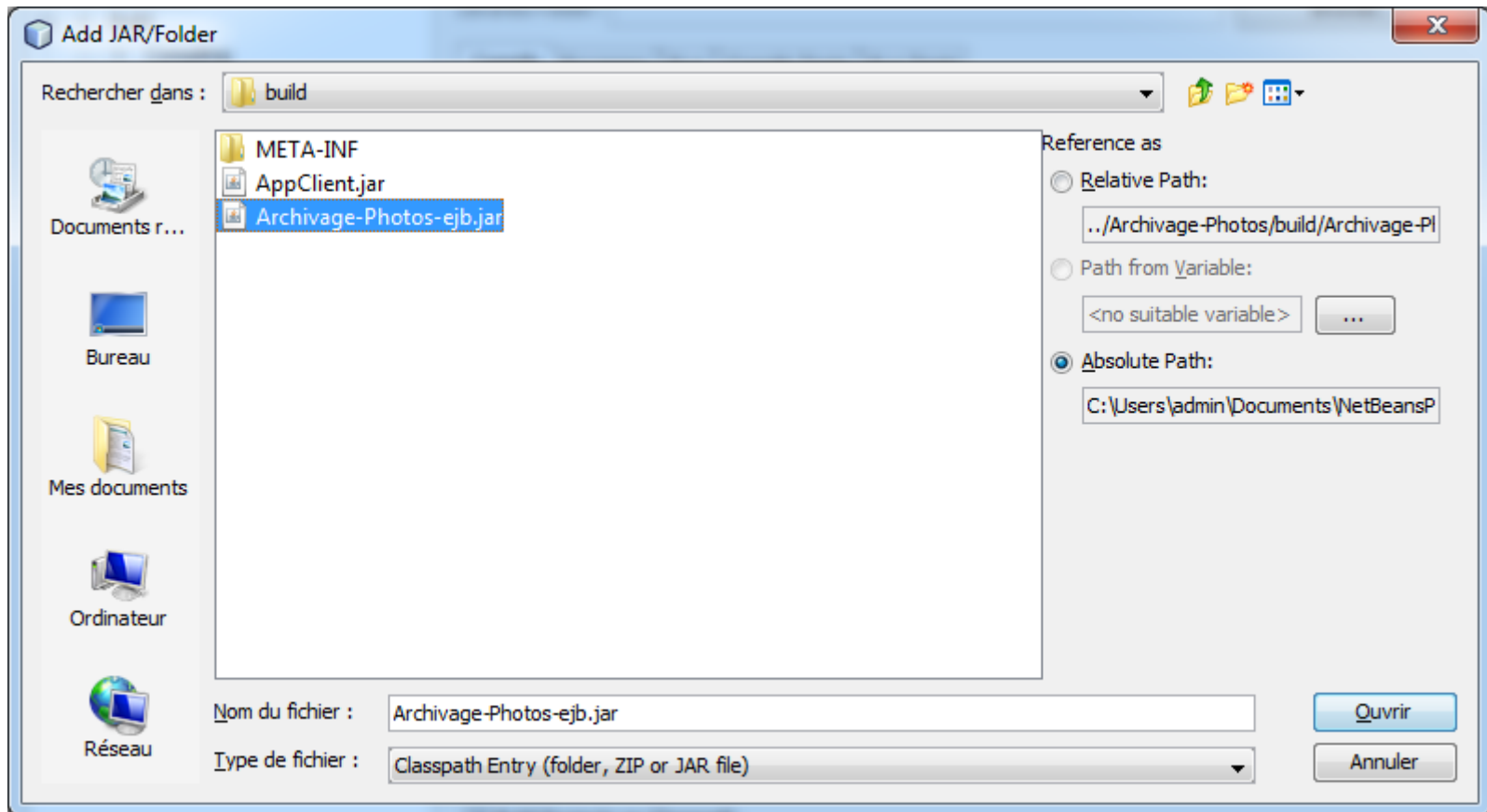
L'application cliente



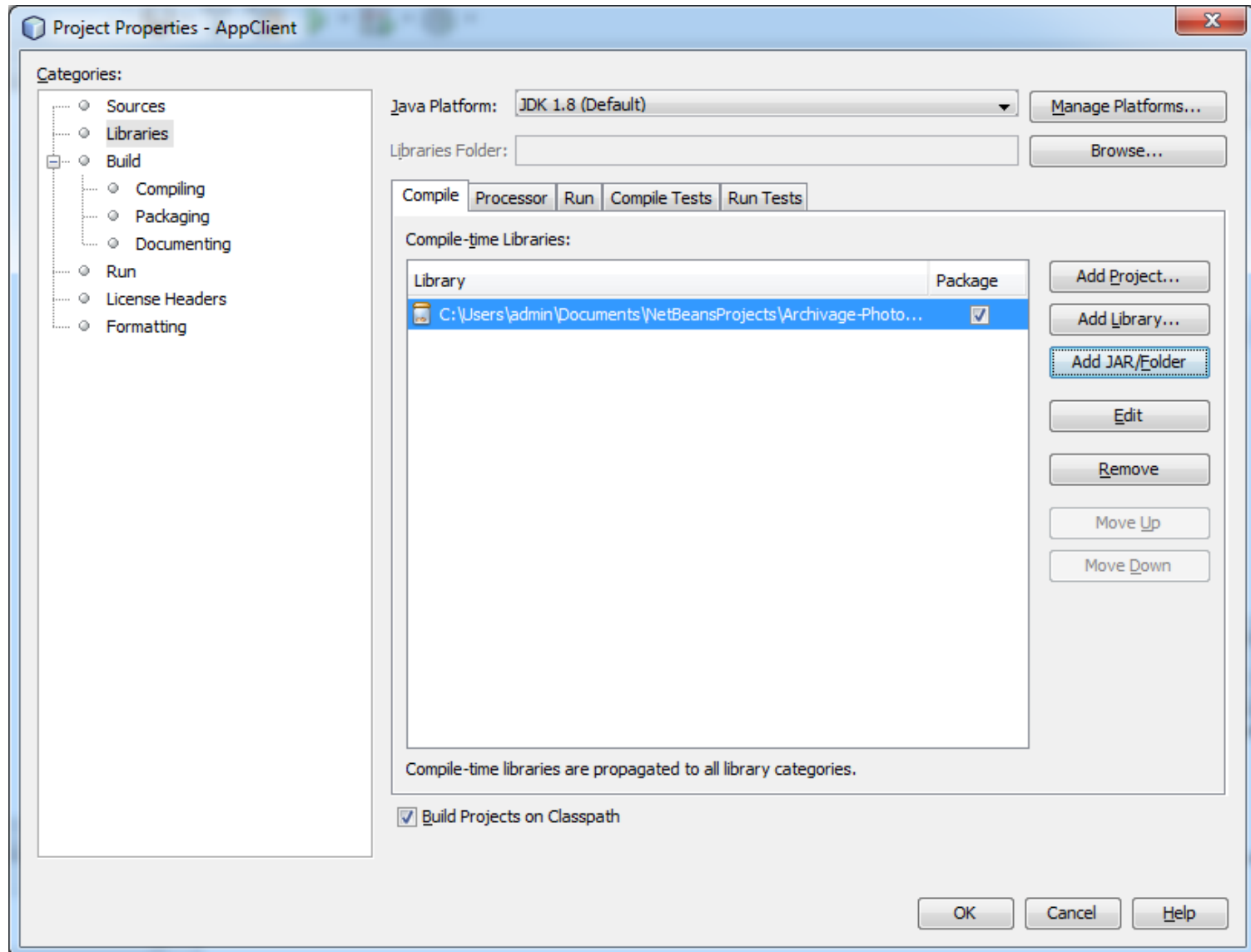
L'application cliente



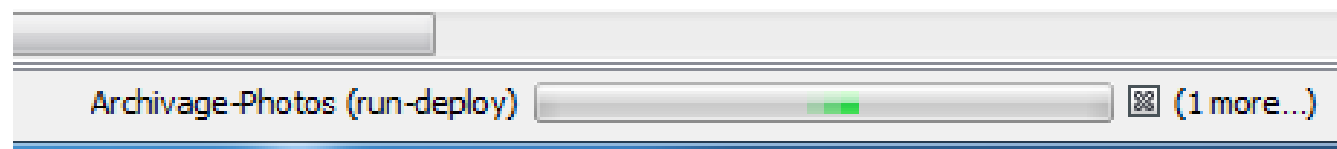
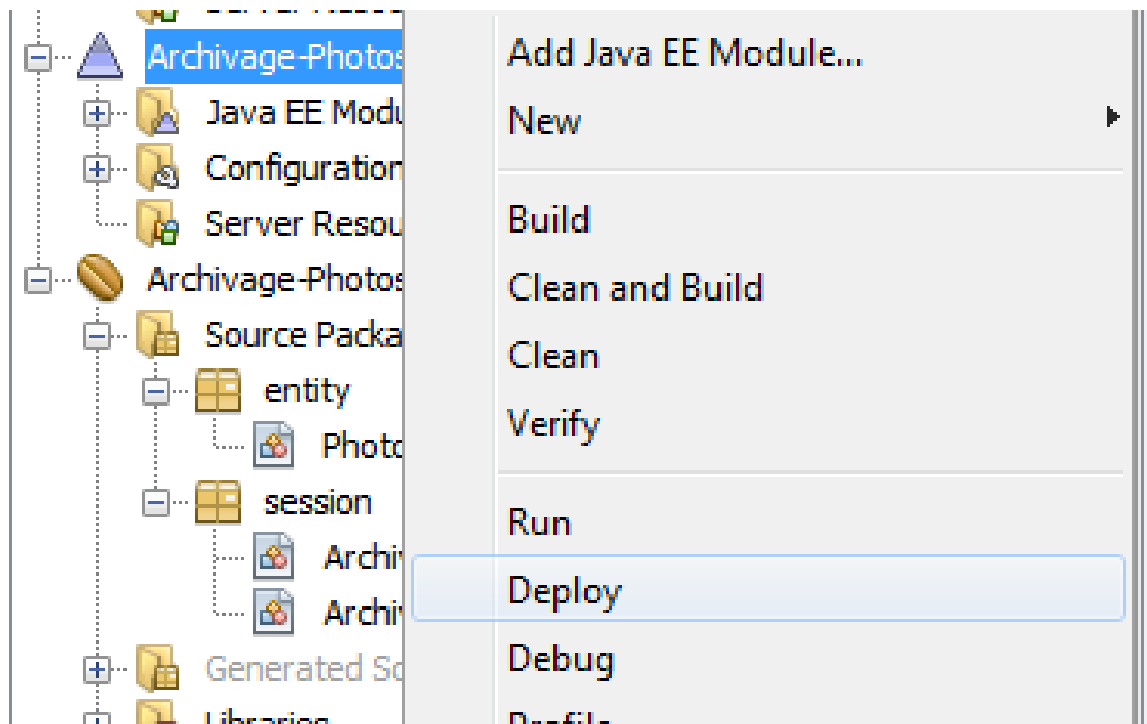
L'application cliente



L'application cliente



Déploiement



Exécution

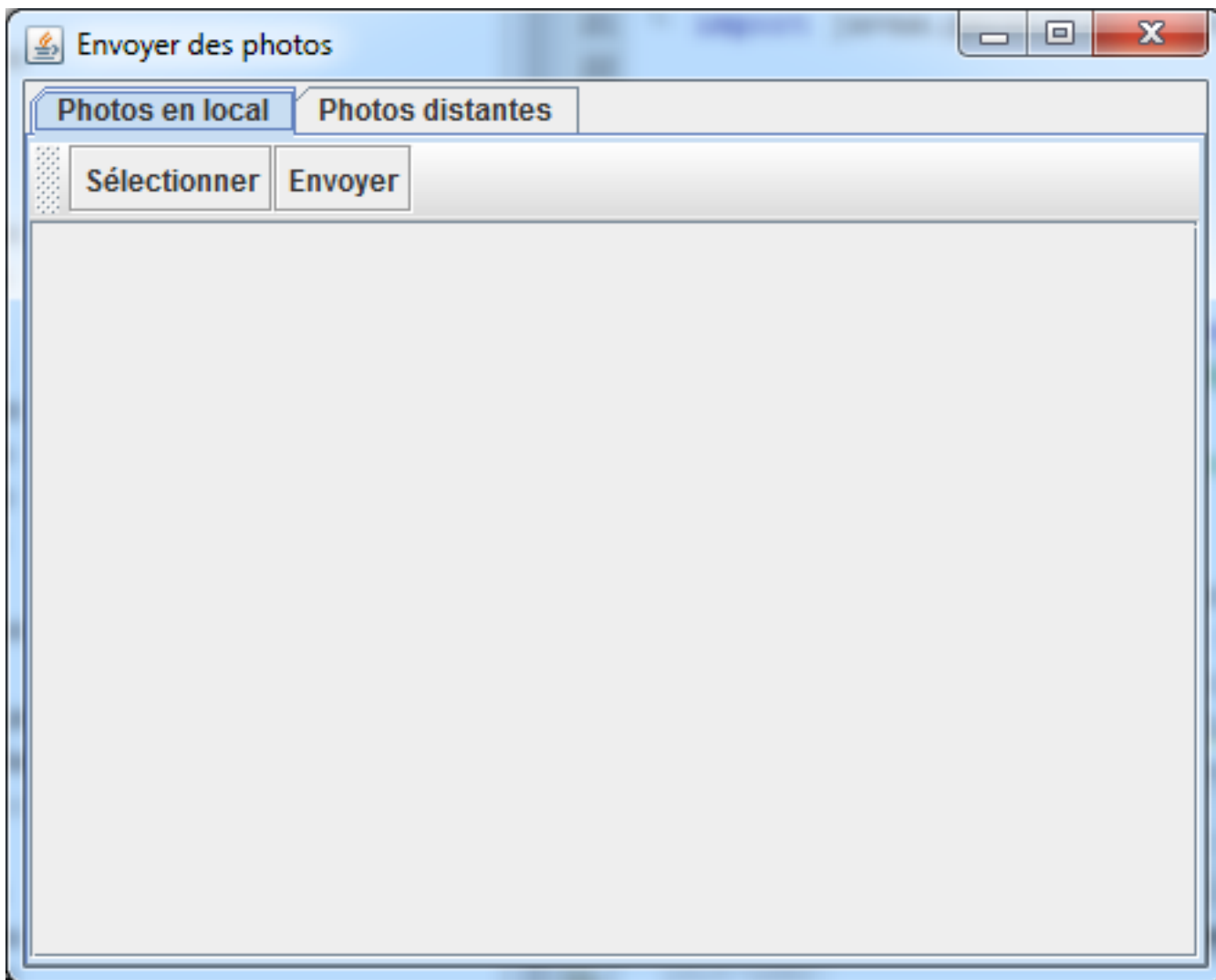


Output ⓘ

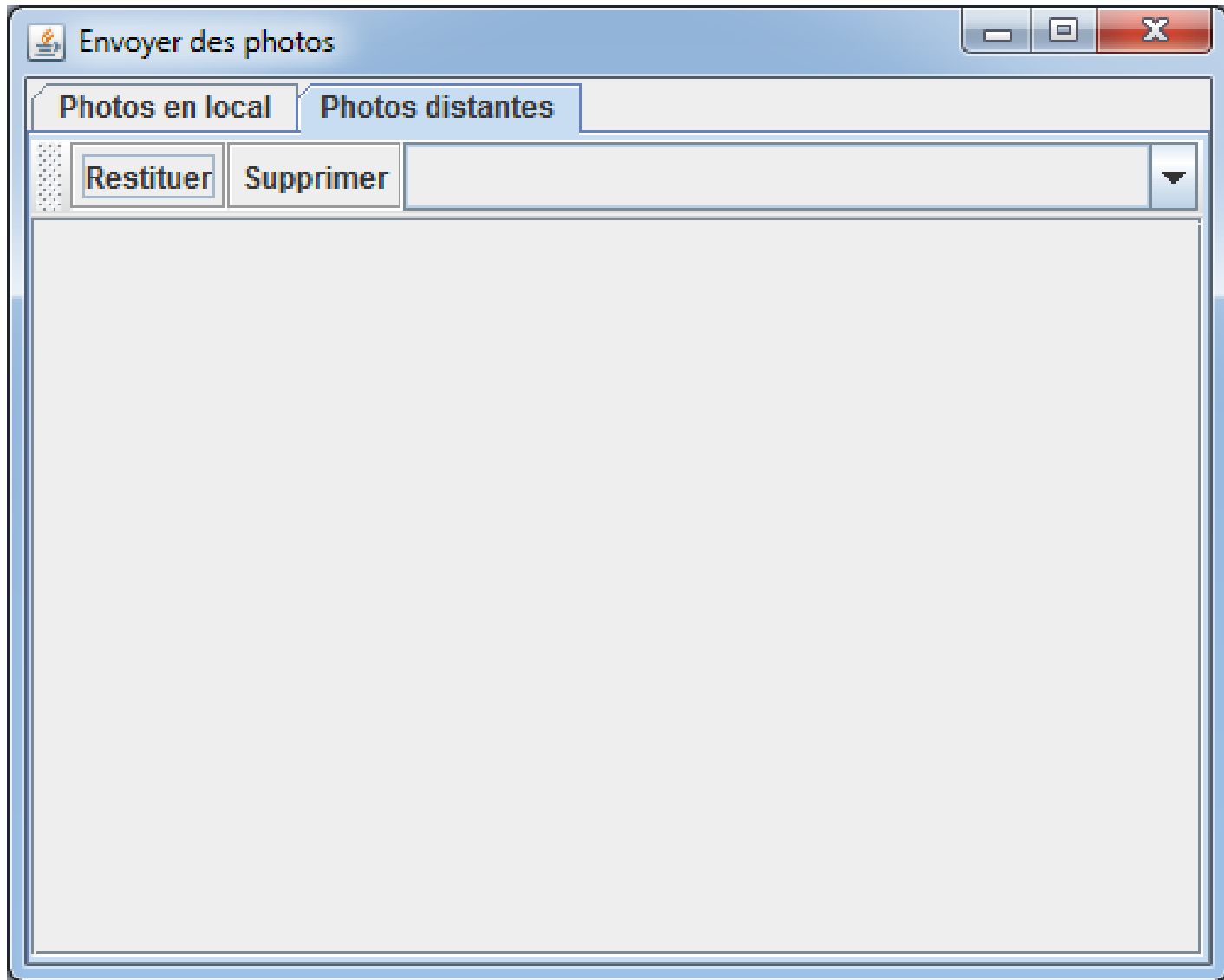
Java DB Database Process ⓘ **GlassFish Server 4.0** ⓘ **Apache Tomcat 7.0.41.0 Log** ⓘ Apache Tomcat 7.0.41.0 ⓘ Archivage-Photos (run) ⓘ

```
do-init:
post-init:
init-check:
init:
Copying 1 file to C:\Users\admin\Documents\NetBeansProjects\Archivage-Photos\dist
Copying 4 files to C:\Users\admin\Documents\NetBeansProjects\Archivage-Photos\dist\Archivage-PhotosClient
run:
BUILD SUCCESSFUL (total time: 1 minute 56 seconds)
```


Exécution



Exécution



Consultation BD

The screenshot displays an IDE interface for database consultation. On the left, the 'Services' tab is active, showing a tree view of the database structure. Under 'Databases', 'Java DB' is listed, containing 'photo' and 'sample'. Under 'Drivers', a connection 'jdbc:derby://localhost:1527/photo [ejb on EJB]' is shown. Under 'EJB', a folder 'Tables' contains a table 'PHOTO'. A context menu is open over the 'PHOTO' table, offering actions like 'View Data...', 'Execute Command...', 'Add Column...', 'Refresh', 'Delete', 'Grab Structure...', and 'Supprimer'. The main editor area shows a SQL query: `select * from EJB.PHOTO;`. Below the query, a result table is displayed with the following data:

#	ID	HAUTEUR	INSTANT	LARGEUR	POIDS
1	axis.png		365 2015-12-25 23:19:56.823	934	