

# Créez votre premier agent avec JADE et ECLIPSE

par Youghourta BENALI ([Page perso de Djug](#))

Date de publication : 23.03.2009

Dernière mise à jour :


JADE est une plate-forme qui permet de construire des systèmes multi agents (SMA) créée par le laboratoire TILAB, entièrement implémenté en JAVA. Nous allons découvrir comment installer et créer notre premier agent avec cette Plate-forme

I - Introduction.....	3
II - Installation.....	5
III - Création du premier agent avec JADE et ECLIPSE.....	7
IV - Compilation et exécution de l'agent.....	8
V - Remerciements.....	11

## I - Introduction

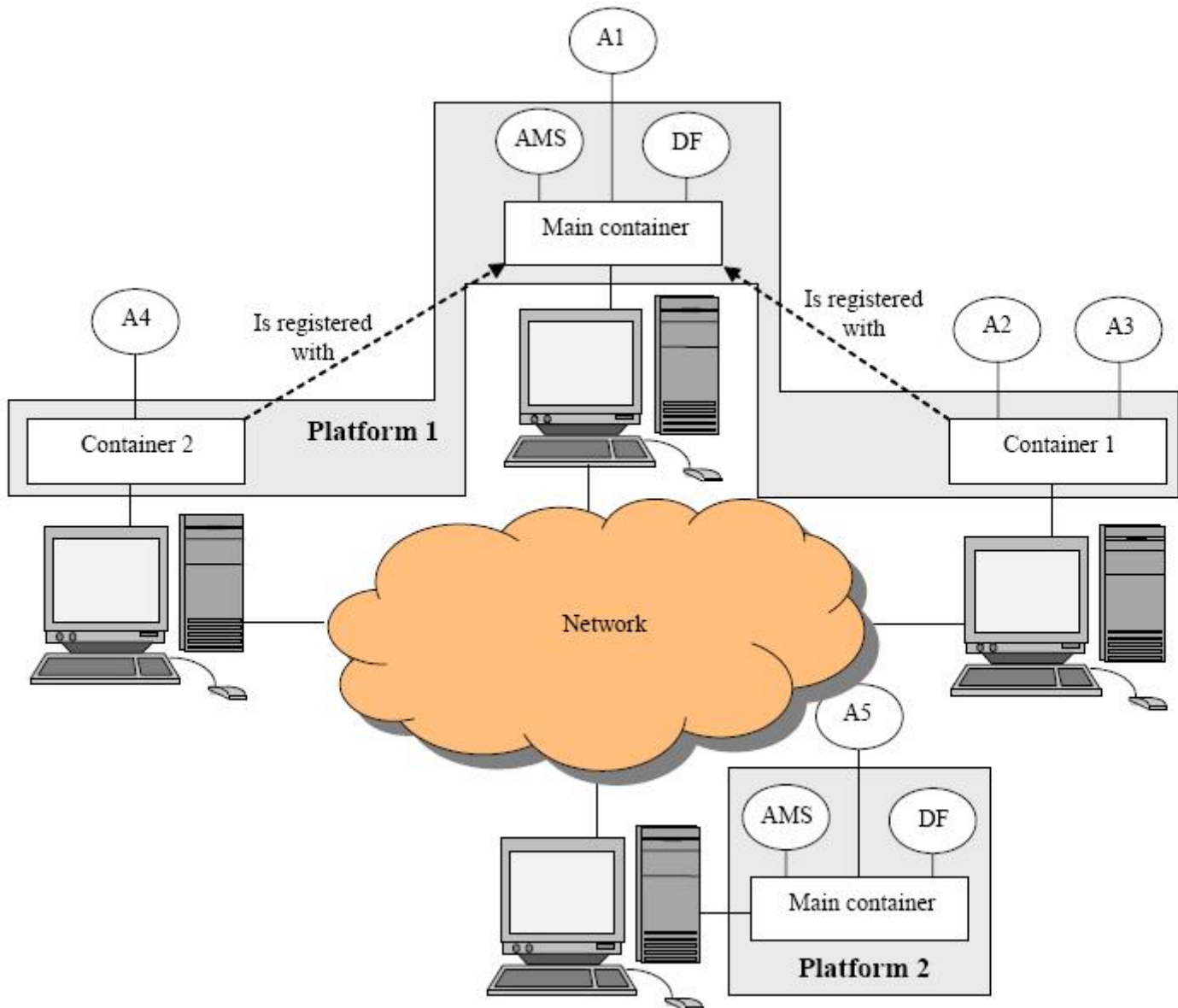


Jade est un middleware qui facilite le développement des  **systèmes multi agents (SMA)**. JADE contient :

- **Un runtime Environment** : l'environnement où les  **agents** peuvent vivre. Ce runtime environnement doit être activé pour pouvoir lancer les agents.
- **Une librairie de classes** : que les développeurs utilisent pour écrire leurs agents
- **Une suite d'outils graphiques** : qui facilitent la gestion et la supervision de la plateforme des agents

Chaque instance du JADE est appelée conteneur " container ", et peut contenir plusieurs agents. Un ensemble de conteneurs constitue une plateforme. Chaque plateforme doit contenir un conteneur spécial appelé main-container et tous les autres conteneurs s'enregistrent auprès de celui-là dès leur lancement

La figure suivante illustre les concepts de base du jade en montrant un petit exemple de deux plateformes jade composées respectivement de trois et un conteneur. Chaque agent est identifié par un identifiant unique et peut communiquer avec n'importe quel autre agent sans avoir besoin de connaître son emplacement :



- Dans le même conteneur (exemple agents A2 et A3)
- Dans la même plateforme mais dans des conteneurs différents (ex A1 et A2)
- Dans deux plateformes différentes (ex. A4 and A5).

Un main-conteneur se distingue des autres "simples" conteneurs par une autre chose; il contient toujours deux agents spéciaux appelés **AMS** et **DF** qui sont lancés automatiquement au lancement du main-conteneur

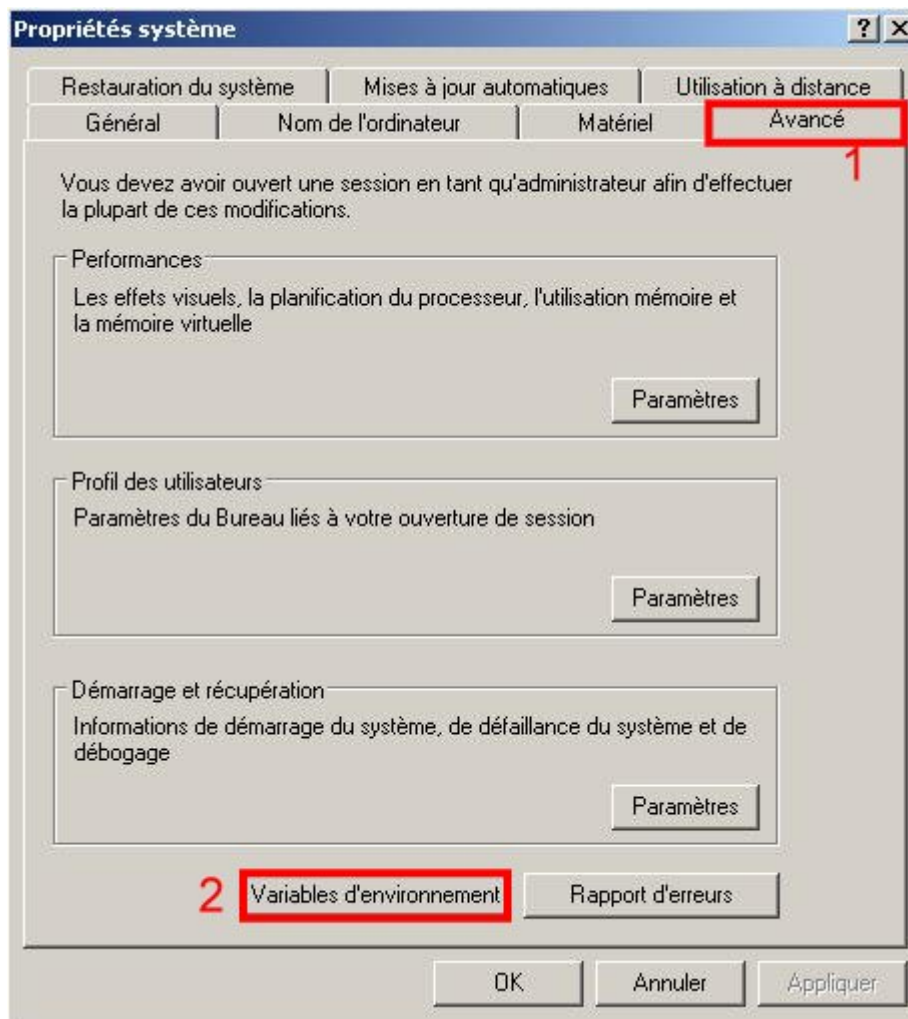
- **AMS** (Agent Management System) qui fournit le service de nommage (pour assurer par exemple que chaque agent possède un identifiant unique dans la plateforme) et qui représente l'autorité de la plateforme (par exemple il est possible de créer/arrêter des agents en envoyant des requêtes à l'AMS)
- **DF** (Directory Facilitator) qui fournit un système de pages jaunes qui permet aux agents de retrouver les agents fournisseurs de services.

Source: "Developing multi-agent applications with JADE. Tutorial for beginners" de la documentation officielle du JADE

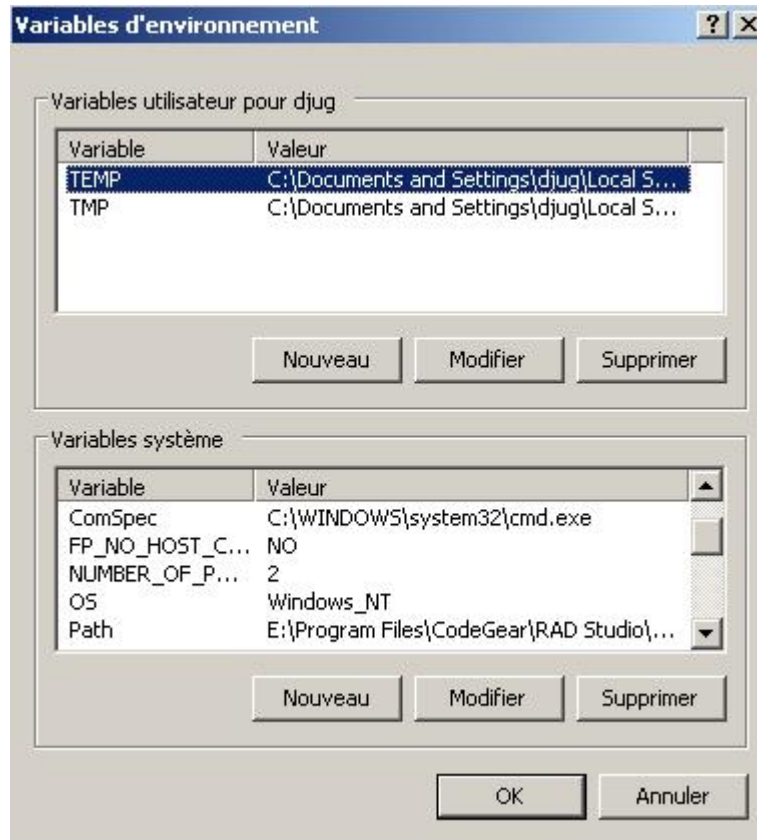
## II - Installation

Voici les étapes à suivre pour installer JADE :

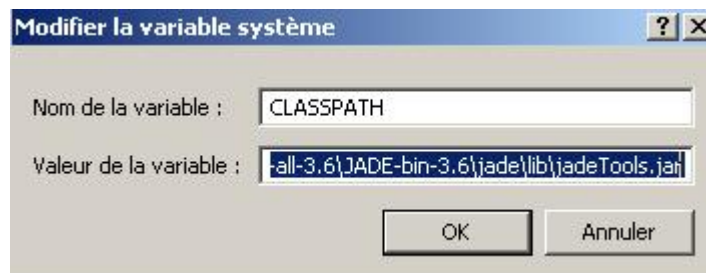
- 1 téléchargez le fichier JADE-all-3.6.zip de l'adresse suivante : <http://jade.tilab.com/download.php>
- 2 décompressez le fichier (on va supposer tout au long de ce tutorial que le chemin du répertoire JADE-all-3.6 est le c:\ JADE-all-3.6 ). Après avoir décompressé le fichier vous retrouvez quatre autres fichiers ZIP (JADE-bin-3.6.zip , JADE-doc-3.6.zip, JADE-examples-3.6.zip, JADE-src-3.6.zip). Décompressez ces 4 fichiers
- 3 on doit maintenant mettre à jour la variable **classpath** (si elle n'existe pas encore il faut la créer) En faisant comme suit :
  - a par Clic droit sur le poste de travail, choisissez propriétés. La fenêtre propriétés système apparaît ,choisissez l'onglet Avancé Puis cliquez sur variables d'environnement



Une petite fenêtre intitulée " variables d'environnement " apparaît



Dans la zone variables système, essayez de trouver la variable d'environnement qui porte le nom CLASSPATH. Si vous ne la trouvez pas, il faut la créer. Maintenant que la variable est trouvée/créée, on doit lui attribuer une valeur. Cette valeur est la concaténation des chemins des quatre fichiers jar http.jar, iiop.jar, jade.jar, jadeTools.jar situés dans le chemin c:\JADE-all-3.6\JADE-bin-3.6\jade\lib

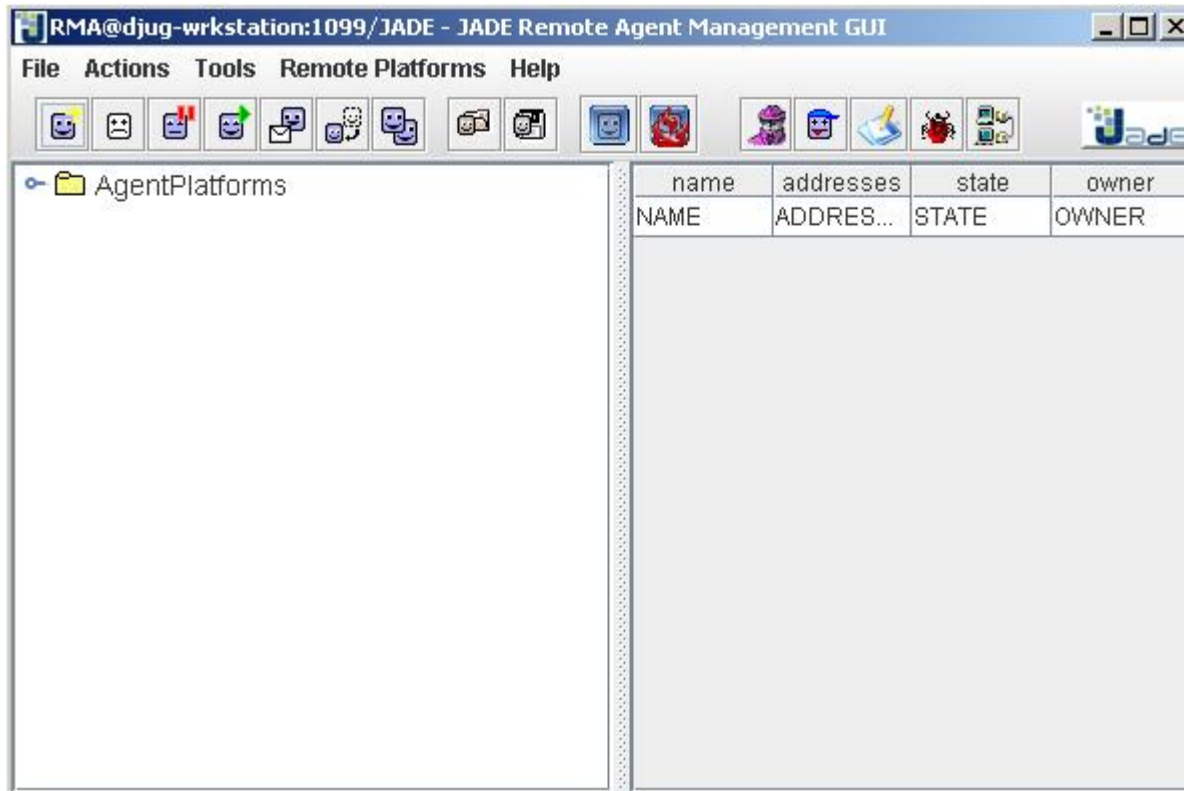


```
C:\JADE-all-3.6\JADE-bin-3.6\jade\lib\http.jar;C:\JADE-all-3.6\JADE-bin-3.6\jade\lib\iiop.jar;
C:\JADE-all-3.6\JADE-bin-3.6\jade\lib\jade.jar;C:\JADE-all-3.6\JADE-bin-3.6\jade\lib\jadeTools.jar
```

Séparez les chemins par des ; et Sauvegardez les modifications. Pour vérifier que l'opération est bien réalisée, tapez dans la fenêtre exécuter (démarrer->Exécuter) ou dans l'invite de commande la commande suivante :

```
Java jade.Boot -gui
```

Une fenêtre dos s'ouvre qui lance la plateforme jade. La fenêtre suivante apparaît :



### III - Création du premier agent avec JADE et ECLIPSE

Maintenant, nous allons créer notre premier agent. Nous allons utiliser l'exemple fourni avec la plateforme situé dans : C:\JADE-all-3.6\JADE-examples-3.6\jade\src\examples\hello. Ouvrez eclipse et créez un nouveau projet (MyFirstAgent par exemple), ajoutez un package (firstAgent) puis créez une nouvelle classe appelée HelloWorldAgent. Collez le code suivant dans la classe.

```

package firstAgent;
import jade.core.Agent;

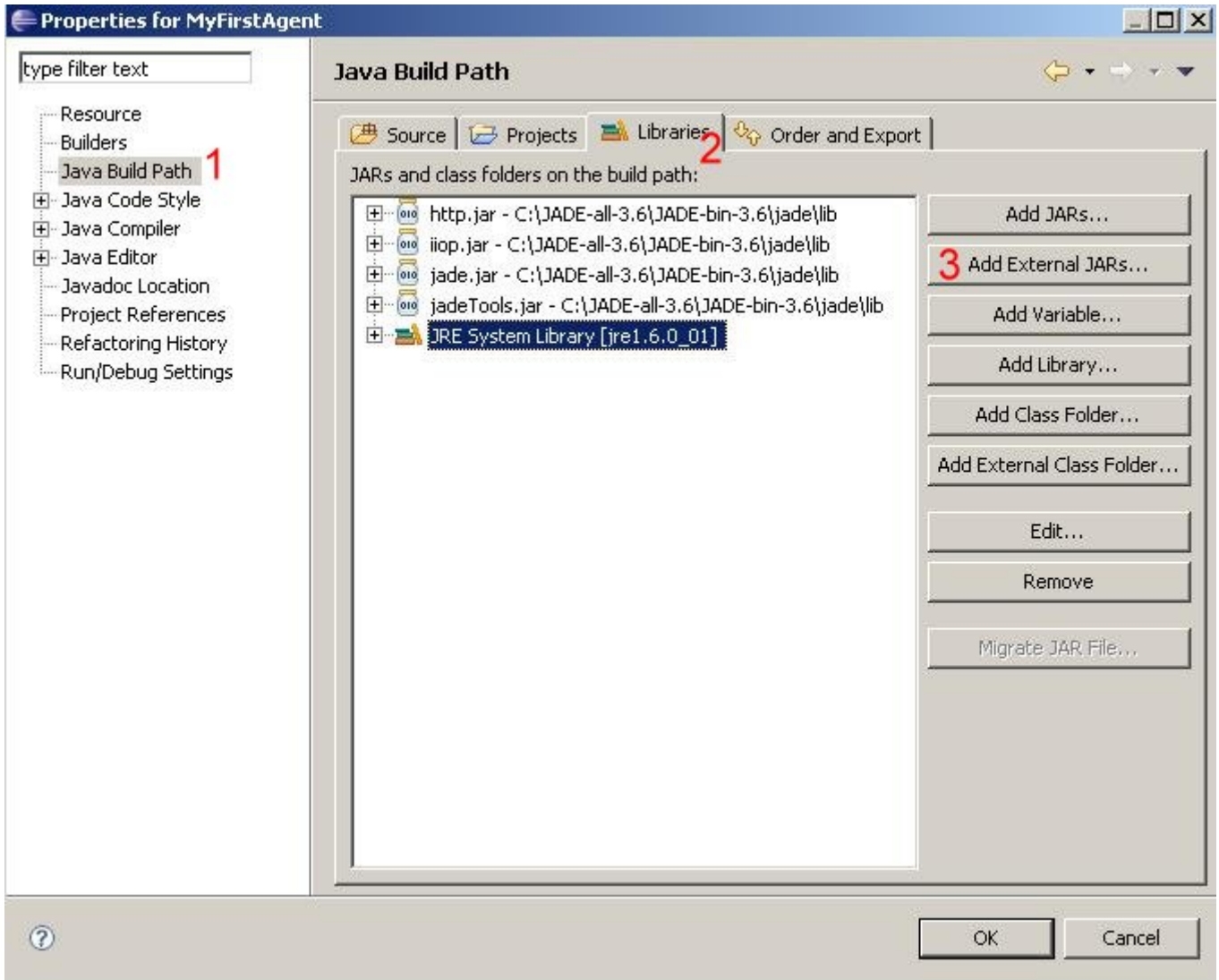
/**
 * This example shows a minimal agent that just prints "Hello World!"
 * and then terminates.
 * @author Giovanni Caire - TILAB
 */
public class HelloWorldAgent extends Agent {

    protected void setup() {
        System.out.println("Hello World! My name is "+getLocalName());
        // Make this agent terminate
        doDelete();
    }
}

// On ne va pas s'intéresser à comprendre chaque ligne du code mais juste comment compiler et lancer l'agent.

```

Vous remarquez l'existence de plusieurs erreurs dans ce petit code. Pour résoudre ce petit problème, effectuez un clic droit sur le nom du projet Puis choisissez propriétés. Cliquez sur java build path >> Libraries>> add external JARs

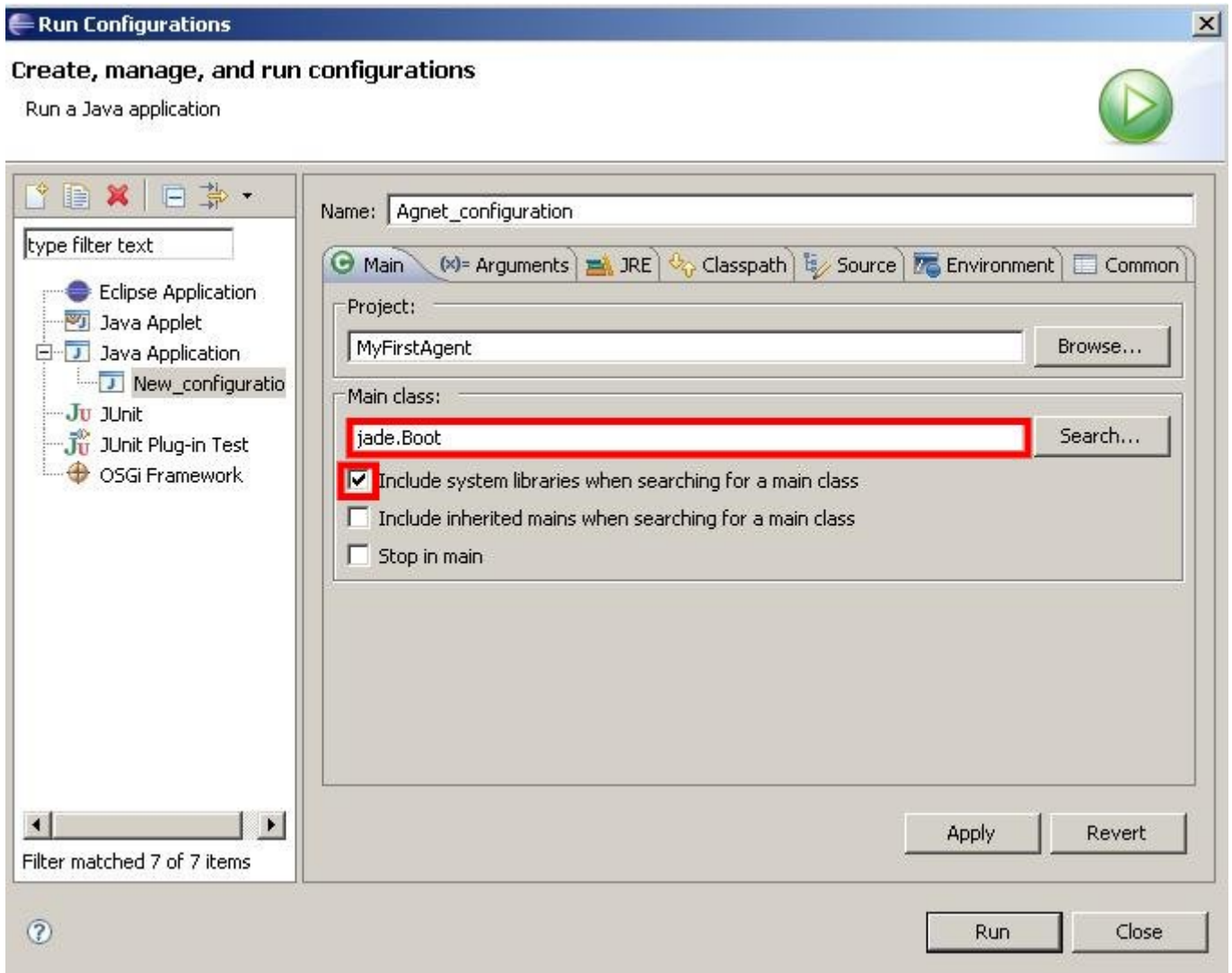


Ajoutez les quatre jar "http.jar" "iiop.jar" "jade.jar" "jadeTools.jar" situés dans C:\JADE-all-3.6\JADE-bin-3.6\jade\lib puis cliquez sur Ok

#### IV - Compilation et exécution de l'agent

Il reste à compiler et lancer l'agent pour cela : Allez dans run>>Run configuration puis double-cliquez sur java application Dans l'onglet " main " et dans la zone de saisie Main class, tapez le code suivant : jade.Boot puis cochez la case : " Include librairies when searching for a main class "





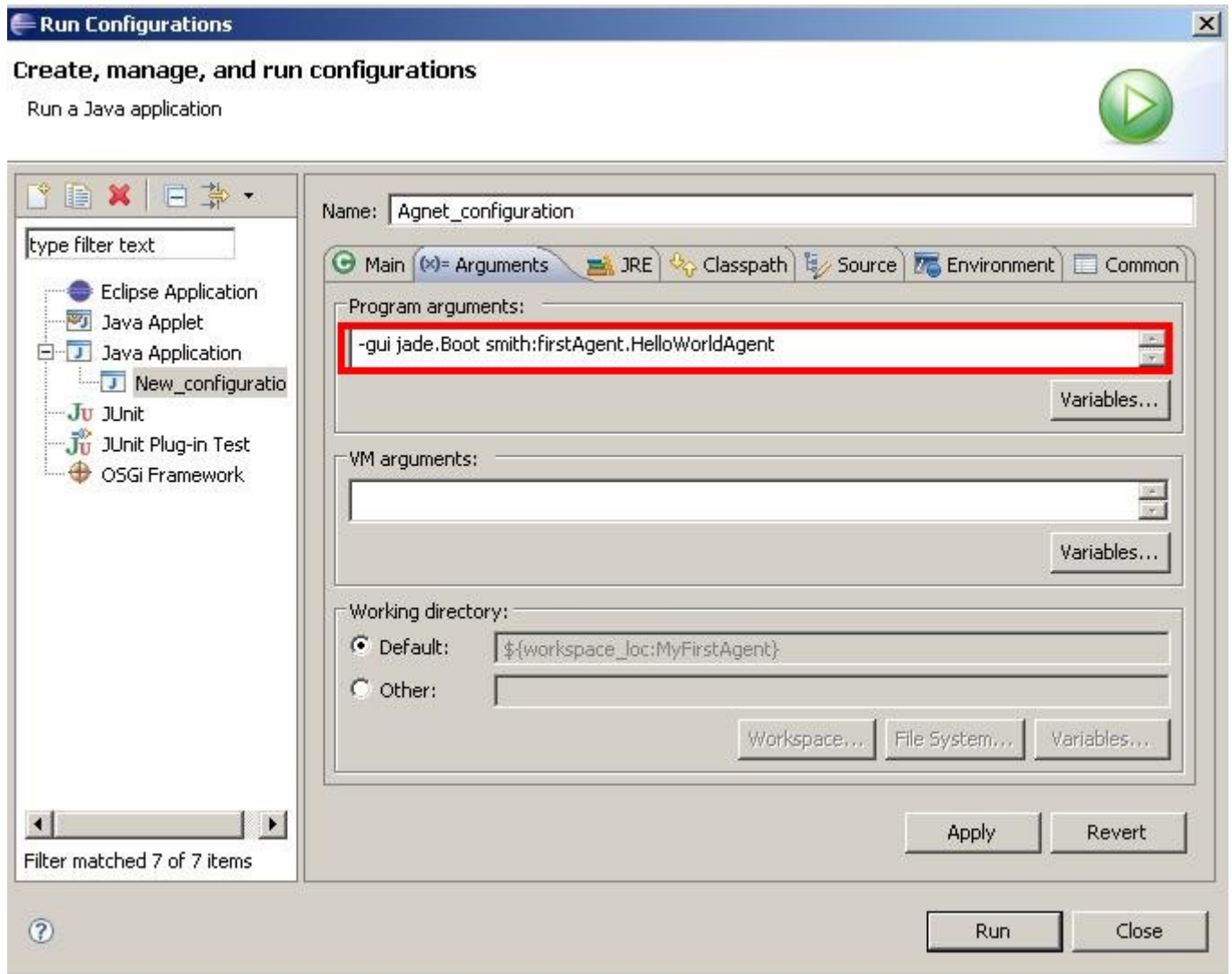
Et dans l'onglet arguments : tapez le code suivant :

```
-gui jade.Boot NomDuL'agent:LeNomDuPackage.LeNomDeLaClasse
```

Dans notre exemple on tape le code suivant :

```
-gui jade.boot smith:firstAgent.HelloWorldAgent
```

puis cliquez sur apply pour ne pas refaire cette configuration plusieurs fois dans le même projet



Cliquez sur run pour voir le résultat (affichage de Hello World! My name is smith Dans la console et ouverture du la plateforme jade)

downloaded in Open Source, under LGPL restrictions,  
at <http://jade.tilab.com/>

```
-----
11 déc. 2008 13:26:39 jade.core.BaseService init
INFO: Service jade.core.management.AgentManagement initialized
11 déc. 2008 13:26:39 jade.core.BaseService init
INFO: Service jade.core.messaging.Messaging initialized
11 déc. 2008 13:26:39 jade.core.BaseService init
INFO: Service jade.core.mobility.AgentMobility initialized
11 déc. 2008 13:26:39 jade.core.BaseService init
INFO: Service jade.core.event.Notification initialized
11 déc. 2008 13:26:39 jade.core.messaging.MessagingService clearCachedSlice
INFO: Clearing cache
11 déc. 2008 13:26:39 jade.mtp.http.HTTPServer <init>
INFO: HTTP-MTP Using XML parser com.sun.org.apache.xerces.internal.jaxp.SAXParserImpl:
11 déc. 2008 13:26:40 jade.core.messaging.MessagingService boot
INFO: MTP addresses:
http://DJUG-WRKSTATION:7778/acc
11 déc. 2008 13:26:40 jade.core.AgentContainerImpl joinPlatform
INFO: -----
Agent container Main-Container@djug-wrkstation is ready.
-----
Hello World! My name is smith
```

## V - Remerciements

Je remercie Ricky81 et Pierre Fauconnier pour la relecture et l'encouragement.