

# TPs CORBA

---

## Plan du TP :

1. [Prérequis](#)
  2. [L'application "Convertisseur Euro"](#)
  3. [Configurations particulières](#)
  4. [L'application Talk](#)
- 

## OpenORB: avant de commencer...

OpenORB est une solution gratuite et libre qui fournit une implémentation Java des spécifications OMG 2.4.2 de CORBA (cf. <http://openorb.sourceforge.net/>). En complément des bibliothèques de développement, OpenORB propose un compilateur IDL vers Java et une implémentation des services CORBA standards (service de noms, d'événements,...).

### Configuration préalable de la machine virtuelle Java afin de pouvoir utiliser OpenORB:

Copiez le fichier "orb.properties" du dossier c:\openorb-1.3.1\config vers le dossier lib de votre java home (pour connaître le java home de votre environnement, exécutez la class java suivante [ShowMeJavaHome.class](#))

---

## L'application "Convertisseur Euro"

Convertisseur Euro est une application client/serveur CORBA qui permet de convertir une somme en Euros vers la devise de son choix et vice-versa.

1/ [Code source du convertisseur Euro](#) (à décompresser dans un dossier Euro de votre espace de travail)

Organisation des fichiers:

- Convertisseur.idl : description IDL de l'interface Euro.
- EuroImpl.java : implémentation Java du servant Euro.
- ServeurConvertisseurEuro.java : application convertisseurEuro côté serveur.
- ClientConvertisseurEuro.java : application convertisseurEuro côté client

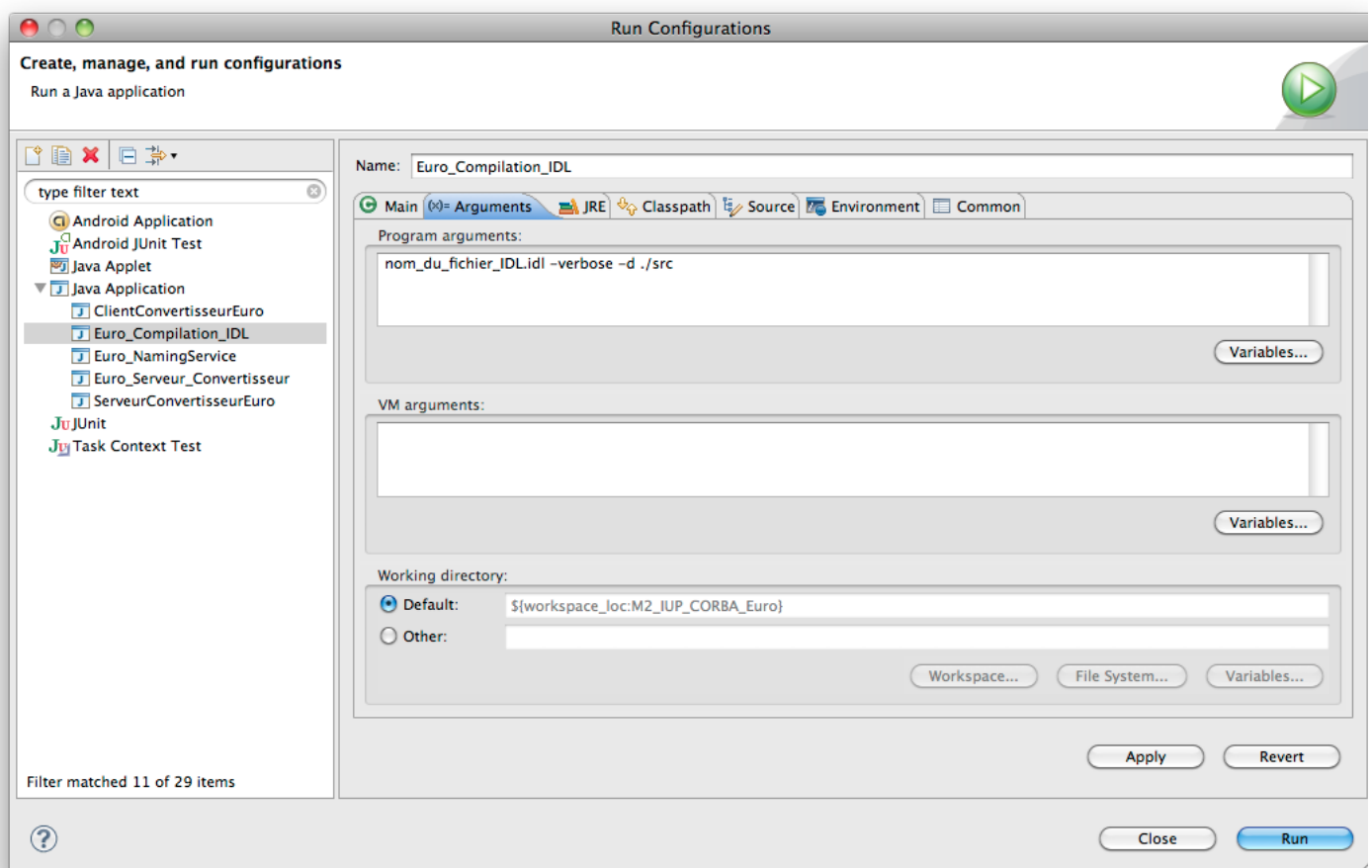
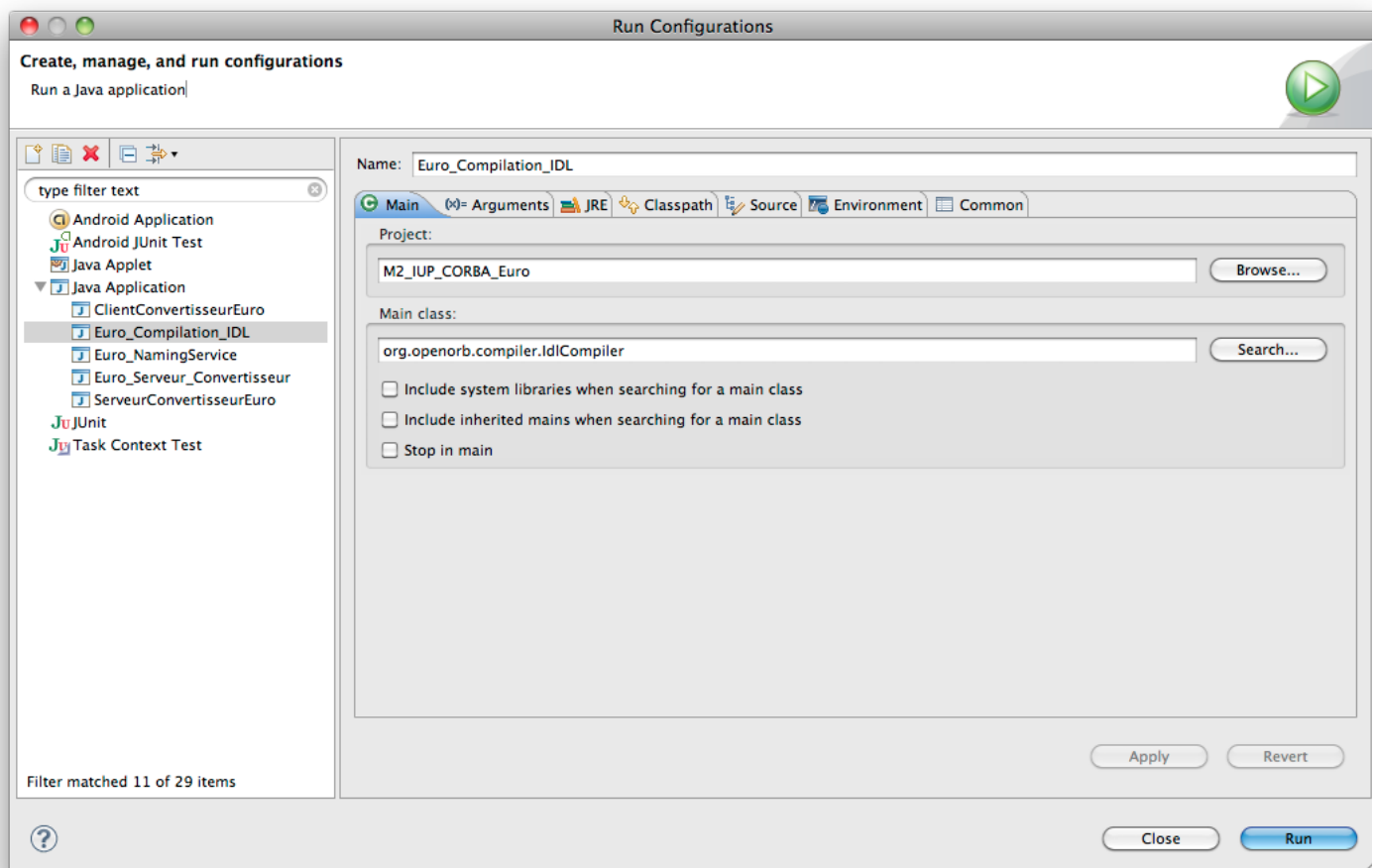
2/ Criez un nouveau projet dans Eclipse

- New Project => Java Project
- Donnez un nom au projet (e.g. Euro) et un chemin d'accès au répertoire dans lequel vous avez téléchargés les fichiers sources (*Create project at external location*)
- Dans l'onglet *Libraries*, ajoutez les bibliothèques suivantes (add external JARs):
  - C:\OpenORB-1.3.1\lib\openorb-1.3.1.jar
  - C:\OpenORB-1.3.1\lib\openorb\_tools-1.3.1.jar
  - C:\OpenORB-1.3.1\lib\avalon-framework.jar
  - C:\OpenORB-1.3.1\lib\logkit.jar
- Validez le projet

3/ Configuration de Eclipse pour compiler le fichier IDL

- Clic droit sur le projet, rubrique Run / Run...
- Faire une nouvelle configuration (que vous appelez par exemple Compilation-IDL)
- Dans le champ *Main class*, indiquez **org.openorb.compiler.IdlCompiler**
- Cochez l'option "Include external jars when searching for main class".
- Dans l'onglet *Arguments*, indiquez les arguments du programme: **Nom\_du\_fichier\_IDL.idl -verbose -d ./src**

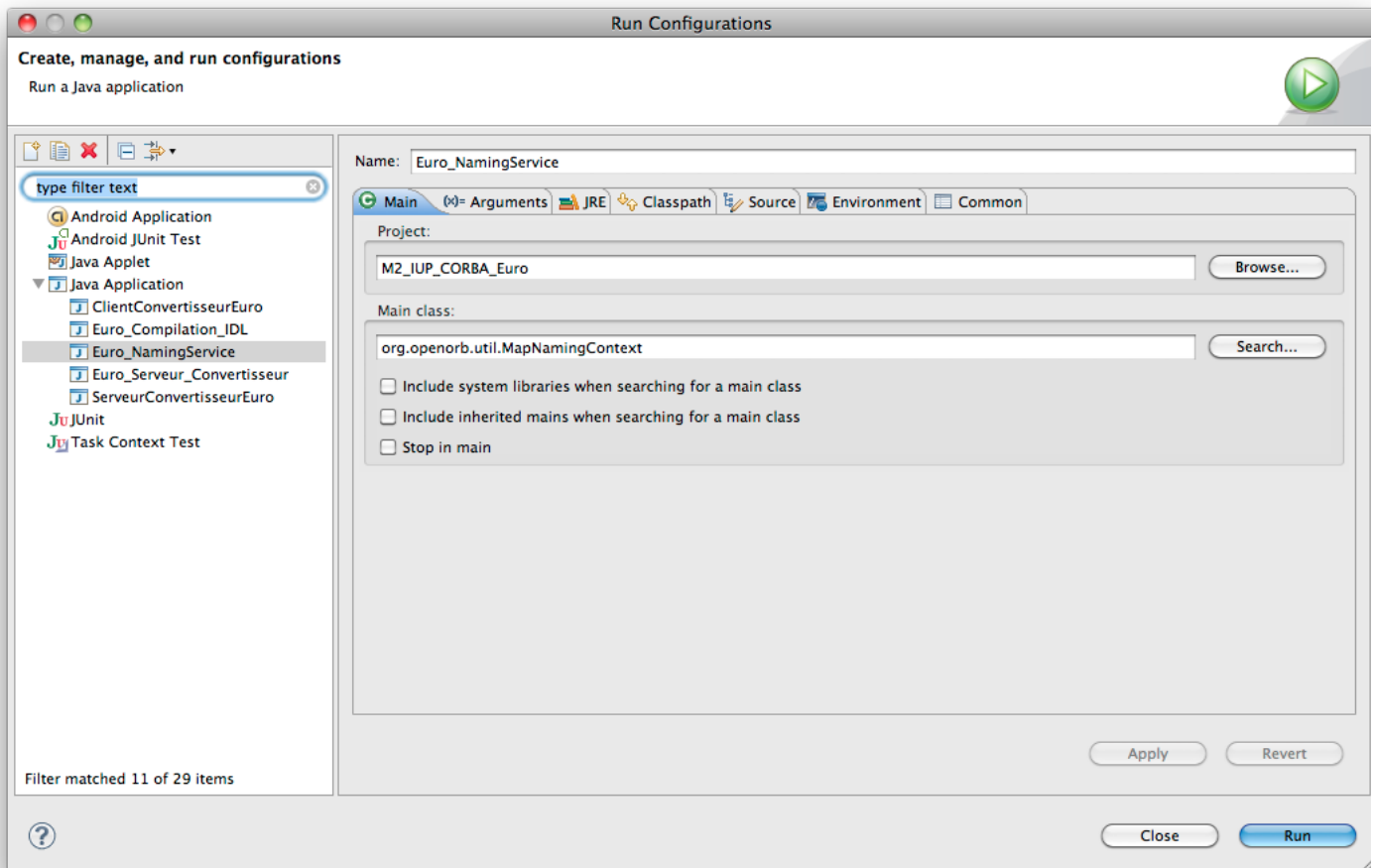
Voici ce que vous devez obtenir:

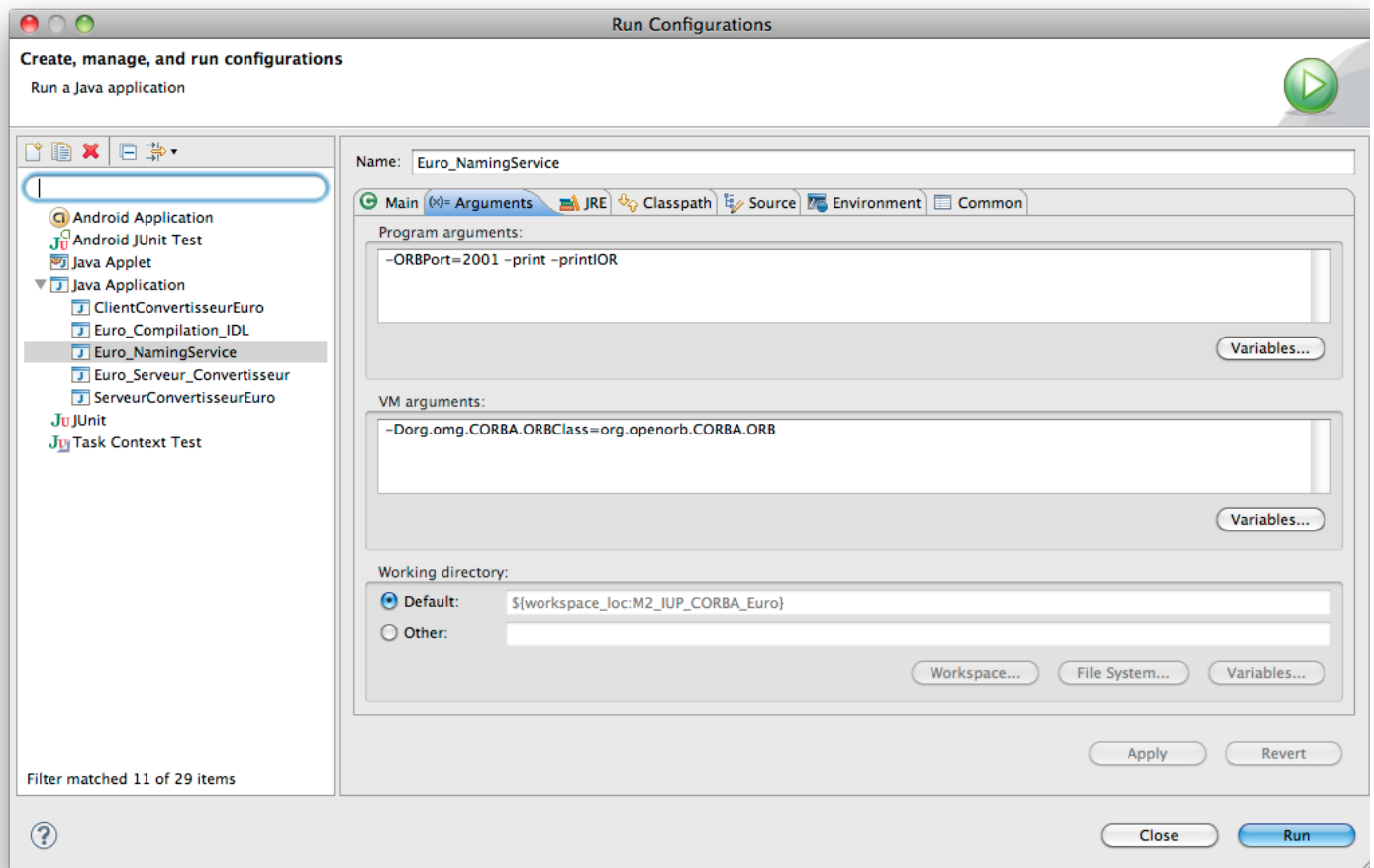


## 5/ Configuration de Eclipse pour utiliser le service de nommage de OpenORB

- Clic droit sur le projet, rubrique Run / Run...
- Faire une nouvelle configuration (que vous appelez par exemple ServiceNoms)
- Dans le champ *Main class*, indiquez **org.openorb.util.MapNamingContext**
- Cochez l'option "Include external jars when searching for main class".
- Dans l'onglet *Arguments*, indiquez les arguments du programme: **-ORBPort=2001 -print** et les arguments de la JVM - **Dorg.omg.CORBA.ORBClass=org.openorb.CORBA.ORB**

Voici ce que vous devez obtenir:





6/ Lancez le service de noms

7/ Exécutez l'application Convertisseur Euro :

- Clic droit sur le projet, rubrique Run / Run...
- Faire une nouvelle configuration (que vous appelez par exemple Lancement)
- Cochez l'option "Include external jars when searching for main class".
- Dans l'onglet *Arguments*, indiquez les arguments de la JVM **-Dorg.omg.CORBA.ORBClass=org.openorb.CORBA.ORB**

8/ Exécutez l'application de façon distribuée (service de noms sur une machine distincte puis client et serveur sur des machines distinctes).

## Configurations particulières :

### Configuration d'OpenORB pour utiliser un service de noms hébergé sur un autre hôte :

- Copiez le fichier "OpenORB.xml" (qui se trouve dans c:\openorb-1.3.1\config) dans la racine de votre répertoire utilisateur (i.e. c:\documents and settings\nomero-de-salle\).
- Editez le fichier. Dans la propriété "NameService", indiquez l'adresse *corbaloc* du service de noms.

Exemple: `<property name="NameService" value="corbaloc:iiop:1.2@193.21.44.14:2001/NameService" />`

### Problème de localisation du *Naming Service* en cas de configuration incorrecte en salle de TP

Le problème se présente par un échec de la méthode `orb.resolve_initial_references("NameService")` alors que le service de nommage a été **correctement** lancé. Une solution **palliative** est de se passer de la configuration de l'ORB et de localiser directement le service de nommage au travers de son IOR (ou de sa corbaloc). Pour cela, il faut remplacer `orb.resolve_initial_references("NameService")` par :

- *Utilisation de la corbaloc* : `orb.string_to_object("corbaloc://corbaloc_affichee_au_lancement_du_Name_service");`
- *Utilisation de l'IOR* : `orb.string_to_object("IOR:IOR_affichee_au_lancement_du_Name_service");`

## L'application Talk

- L'application qu'on vous demande de réaliser est un talk, i.e. un échange de chaînes de caractères entre deux utilisateurs (une fenêtre d'émission, une fenêtre de réception). Dans un premier temps, la communication sera unidirectionnelle, puis dans un second temps, elle devra être bidirectionnelle.  
Rq. Assurez-vous d'avoir le même contrat IDL que votre voisin afin de pouvoir tester l'application de façon répartie.

- Une fois cette application implémentée, effectuez les modifications nécessaires pour transformer ce talk en chat (un émetteur, plusieurs récepteurs).

---

Binares [OpenORB-1.3.1.zip](#)