

## TP 4 Introduction au framework OSGi

### Exercice 1. Mini-tutoriel Equinox

- Mettre en place l'environnement de travail :
  - Télécharger Equinox :
    - Sur la page Web du cours :
      - Télécharger le fichier `org.eclipse.osgi_xxx.jar` se trouvant ici (v3.7) :  
[http://www.lirmm.fr/~tibermacin/ens/hmin304/tp/03\\_osgi/lib/org.eclipse.osgi\\_3.7.1.R37x\\_v20110808-1106.jar](http://www.lirmm.fr/~tibermacin/ens/hmin304/tp/03_osgi/lib/org.eclipse.osgi_3.7.1.R37x_v20110808-1106.jar)
- Démarrer le framework Equinox : (depuis la version 3.8 le démarrage du framework en mode console nécessite d'autres jar et une configuration particulière)
  - Aller dans le répertoire où se trouve le JAR téléchargé et taper la commande suivante :  
**java -jar org.eclipse.osgi\_3.7.1.R37x\_v20110808-1106.jar -console**
- Après quelques instants, vous aurez accès au prompt OSGi : **osgi>**
- Saisir la commande *help* pour tester l'environnement, puis lire attentivement ce qui s'affiche (toutes les commandes possibles à partir de ce prompt) :  
**osgi> help**
- Saisir la commande *ss* (qui veut dire « *short status* ») et observer ce qui se passe :  
**osgi> ss**  
Cette commande affiche la liste des bundles OSGi installés dans le conteneur, leur id et leur état (Active, Installed, Resolved, ...). La première fois que vous démarrez cet environnement, il y a déjà un bundle installé et actif ; il s'agit d'Equinox !!!
- Saisir la commande *headers <id-du-bundle>* pour afficher le contenu du Manifest du bundle dont l'identifiant est *<id-du-bundle>* :  
**osgi> headers 0**
- Saisir la commande *bundle <id-du-bundle>* pour afficher l'état détaillé d'un bundle :  
**osgi> bundle 0**
- Quitter l'environnement Equinox : **osgi> close**
- Vous pouvez démarrer ce prompt en lançant Eclipse en mode console. En effet, lorsque vous démarrez Eclipse, il s'agit d'Equinox qui s'exécute, et qui charge les bundles (plugins) correspondants à Eclipse :  
**eclipse -console**

Le prompt **osgi>** s'affiche. Vous pouvez ainsi saisir : **osgi> ss** et voir les bundles qui constituent Eclipse déployés dans le container OSGi Equinox.

Quitter ce prompt et revenez sur celui obtenu en démarrant Equinox (Étape 2)

- Gérer le cycle de vie d'un bundle (l'installer, l'activer, l'arrêter, ...) :
  - Installer un bundle : **osgi> install file:<repertoire/fichier.jar>**
  - Consulter les bundles installés (vérifier l'identifiant du bundle) : **osgi> ss**
  - Activer un bundle : **osgi> start <id-du-bundle>**  
La méthode `start` de la classe « Activator » est invoquée, le cas échéant
  - Mettre à jour un bundle installé/activé (si vous corrigez des erreurs dans votre bundle) : **osgi> update <id-du-bundle>**  
Si le bundle est installé seulement, ce bundle va être dés-installer puis ré-installé (*uninstall* puis *install*). Si le bundle est actif, ce bundle va être ré-installé et ré-activé (*stop*, *uninstall*, *install* puis *start*).
  - Désactiver un bundle : **osgi> stop <id-du-bundle>**  
La méthode `stop` de la classe « Activator » est invoquée, le cas échéant
  - Désinstaller un bundle du container : **osgi> uninstall <id-du-bundle>**
- Consulter les logs dans le sous-répertoire configuration créé par Equinox dans le répertoire à partir duquel vous avez lancé le framework

Vous pouvez utiliser votre éditeur de texte préféré pour écrire vos interfaces et classes Java, et le MANIFEST.MF (N.B. : Il faut toujours ajouter une ligne vide à la fin du Manifest)

Pour compiler vos interfaces et classes, il faut utiliser le compilateur du SDK en ligne de commande. Ne pas oublier d'ajouter au CLASSPATH le JAR d'Equinox : `org.eclipse.osgi_xxx.jar`

Cela vous permet de compiler vos interfaces et classes qui, rappelons le, dépendent parfois de l'API OSGi (BundleActivator, BundleContext, ...).

Pour composer vos bundles, il faut utiliser l'outil `jar` fourni par le SDK de la façon suivante :  
**jar cvmf META-INF/MANIFEST.MF <nom-du-jar> <fichiers-du-jar>**

Pour afficher sur la console le contenu d'un JAR : `jar tvf <fichier-jar>`

Pour extraire le contenu d'un JAR : `jar xvf <fichier-jar>`

Il est conseillé de travailler sur deux consoles : la première pour la compilation et la construction des Jars et la seconde pour saisir les commandes sur le prompt OSGi.

## **Exercice 2.** Application HelloWorld

Écrire le bundle OSGi (vu en cours) qui affiche « Hello World ! » au moment de son activation et « Goodbye World ! » au moment de sa désactivation.

Tester les deux versions vues en cours (en utilisant un « *Service Bundle* » et sans celui-ci. Commencer par cette dernière). Tester un « *update* » des bundles (Hello -> Bonjour).

Tester la version avec les *Declarative Services*.

C. TIBERMACHINE