Exemple Ch07_01

I. La compilation

Le fichier build.xml est pris en compte par défaut par la commande ant.
Compiler chacun des programmes consiste donc à exécuter la commande: ***ant***

Les fichiers .class sont créés dans le sous répertoire local ***classes***.
Les fichiers .java générés par la compilation de l'IDL sont créés dans le sous répertoire local ***generated***.

II. Le fichier jacorb.properties :

- le fichier se trouve dans le répertoire *etc* car on exécute les programmes Java en utilisant la commande:
  - cd classes
  - jaco -**Djacorb.config.dir="."** devise.Server

  qui par défaut va le chercher dans le répertoire etc du répertoire référencé par la propriété jacorb.config.dir

- `ORBInitRef.NameService=file:/c:/NS_Ref`
  - le fichier utilisé par défaut pour stocker les IOR

- `jacorb.log.default.verbosity=2`
  - `pour ne pas avoir trop de trace dans l'exécution.`
  - `pour avoir plus d'information 3 sinon 4 (niveau debug)`

III. Cas1_hello

Ce cas est largement commenté en cours dans le support de cours sur le chapitre consacré à CORBA.
Ce cas contient les cas fonctionnels suivants :

- création d'un servant dont l'IOR est écrit dans un fichier
- le client invoque les méthodes du servant en accédant au fichier contenant l'IOR

Pour exécuter ce cas :
runServeur.bat
runClient.bat

IV. Cas2_hello_witn_ns

Ce cas est largement commenté en cours dans le support de cours sur le chapitre consacré à CORBA.
Ce cas contient les cas fonctionnels suivants :

- création d'un servant qui s'enregistre dans un Naming Service

- le client invoque les méthodes du servant en accédant au servant via le Naming Service
- création d'un sous-contexte dans le Naming Service dans lequel on crée un deuxième servant de même nom
- le client invoque ce deuxième servent en utilisant le sous-contexte

Pour exécuter ce cas :

runNS.bat
runServeur.bat
runClient.bat

V. Cas3_devise

Ce cas est largement commenté en cours dans le support de cours sur le chapitre consacré à CORBA.
Ce cas contient les cas fonctionnels suivants :

- un objet Devise est encapsulé dans un servant DeviseOD
- ce servant implémente deux méthodes distantes qui retournent une devise. Une par retour de méthode, l'autre par référence d'un paramètre de la méthode
- "mappage" de l'objet devise en une donnée de l'IDL
- le serveur démontre que l'on peut faire passer la référence d'un objet CORBA dans la communication et utiliser cette référence pour invoquer les méthodes distantes
- le IDL contient à titre d'exemple de nombreux cas afin de montrer les fichiers générés lors de la compilation IDL
- le serveur affiche le contenu du Naming Service
- le client utilise les servants créés par le serveur afin de mettre en évidence les différents cas

Pour exécuter ce cas :

runNS.bat
runServeur.bat
runClient.bat

VI. Cas4_devise avec Tie

Le cas précédent mais dans le cas où la classe d'implémentation du servant ne pouvant pas hériter doit utiliser un "Tie" :

DeviseOD implements InterfaceDeviseODOperations

Pour exécuter ce cas :

runNS.bat
runServeur.bat
runClient.bat

VII. Cas5_event_push

Ce cas consiste à montrer comment il est possible de gérer un canal d'évènement dans lequel des consommateurs (des servants) s'abonnent et attendent que des producteurs leurs poussent des évènements.

Pour exécuter ce cas :

runNS.bat
runChannel.bat
runConsommer.bat (on peut en lancer plusieurs)
runProducter.bat

VIII. Cas6_evant_push_ref_corba

Ce cas est identique au précédent sauf que l'on montre que la donnée échangée dans l'envoi d'un évènement peut être une référence CORBA : le producteur envoi aux consommaterus l'objet CORBA du serveur de devise afin que les consommateurs utilisent de manière distante les services du serveur de devise.

Pour exécuter ce cas :

runNS.bat
runChannel.bat
runServerDevise.bat
runConsommer.bat (on peut en lancer plusieurs)
runProducter.bat