

David ANATON



# CORBA Messenger

Manuel développeur





## 1. Table des matières

1.	Table des matières .....	2
2.	Historique des modifications.....	2
3.	Introduction .....	3
4.	Travail effectué .....	3
4.1.	Server Side.....	3
4.1.1.	Messenger.idl .....	3
4.1.2.	Dossier MessengerApp.....	3
4.1.3.	MessengerImpl.java.....	3
4.1.4.	StartServer.java .....	3
4.2.	Client Side .....	3
4.2.1.	Dossier MessengerApp.....	3
4.2.2.	StartClient.....	3
5.	Améliorations envisagées .....	4
5.1.	Interface graphique .....	4
5.2.	Possibilité de créer des chambres de discussion différentes .....	4
5.3.	Récupération des messages en temps réel.....	4
5.3.1.	Récupération à intervalles réguliers.....	4
5.3.2.	Récupération en temps réel.....	4
6.	Manuel Utilisateur .....	4
6.1.	ORBD.....	4
6.2.	Serveur .....	4
6.1.	Client.....	4

## 2. Historique des modifications

<i>Name</i>	<i>Date (dd/mm/yy)</i>	<i>Reason For Changes</i>	<i>Version</i>



## 3. Introduction

Ce manuel est là pour expliquer l'approche utilisée pour développer l'application Messenger permettant de mettre en réseau plusieurs clients pour que ces derniers puissent échanger des messages textuels.

## 4. Travail effectué

### 4.1. Server Side

Ayant choisi de travailler avec Corba, le développement a commencé avec la rédaction d'un fichier .idl.

Ce dernier spécifie l'ensemble des fonctions accessibles à travers le réseau. La compilation de ce dernier générera un ensemble de classes qui permettront aux clients d'appeler ces fonctions sur le serveur.

#### 4.1.1. Messenger.idl

Contient les prototypes des méthodes accessibles sur le serveur depuis le client.

Compilé en utilisant `idlj -fall Messenger.idl`

#### 4.1.2. Dossier MessengerApp

Celui-ci est généré automatiquement par compilation du fichier IDL.

#### 4.1.3. MessengerImpl.java

Implémentation des méthodes spécifiées dans le fichier IDL.

#### 4.1.4. StartServer.java

Se connecte à l'ORB et met en place un serveur qui attendra les connexions de clients sur un objet java de type MessengerImpl.

### 4.2. Client Side

#### 4.2.1. Dossier MessengerApp

Celui-ci est généré automatiquement par compilation du fichier IDL dans le serveur. Il a été importé pour avoir accès aux différents objets de l'ORB.

#### 4.2.2. StartClient

Connexion à l'ORB. Mise en place d'une boucle qui demande les commandes à effectuer aux clients.



Messenger  
Keep in touch

## 5. Améliorations envisagées

### 5.1. Interface graphique

### 5.2. Possibilité de créer des chambres de discussion différentes

### 5.3. Récupération des messages en temps réel

#### 5.3.1. Récupération à intervalles réguliers

Mise en place d'un thread chez chaque client qui lance la commande `#messages` toutes les 5 secondes.

#### 5.3.2. Récupération en temps réel

Chaque client devient un mini-serveur qui peut recevoir les messages au fur et à mesure que le serveur les lui envoie.

## 6. Manuel Utilisateur

### 6.1. ORBD

Choisir un port.

Aller en ligne de commande (Windows + R, puis taper « `cmd` »). Naviguer jusqu'au dossier du projet et lancer la commande `start orbd - ORBInitialPort <numéro de port>`

### 6.2. Serveur

Naviguer ensuite dans le sous-projet du serveur et lancer le `StartServer` en lui fournissant les paramètres suivants `-ORBInitialPort <numéro de port> -ORBInitialHost localhost`

### 6.1. Client

Naviguer dans le sous-projet du client et lancer le `StartClient` en lui passant les paramètres suivants `-ORBInitialPort <numéro de port>-ORBInitialHost <adresse du serveur>`

Le client est alors invité à entrer son nom, puis le nom de son correspondant. Il a ensuite accès à l'interface de communication.

Il peut alors entrer un texte à envoyer à son correspondant ou plusieurs commandes :

- `#clients` : Retourne la liste de tous les clients connectés
- `#exit` : Déconnecte son client de l'application
- `#help` : Affiche la liste des commandes disponibles
- `#pseudo` : Changer de pseudo
- `#to` : Changer de destinataire
- Chaîne de caractères vide : télécharge la liste des messages adressés au client

# KEEP IN TOUCH !