



AgentService Mobile

A light architecture for mobile devices

The AgentService Team

Copyright @ 2008 – l.i.d.o. – DIST University of Genoa

Abstract

The aim of this guide is to show to use the AgentService Mobile infrastructure in order to execute agents on mobile devices with limited resources.

Index of Contents

AgentService Mobile	1
A light architecture for mobile devices	1
The AgentService Team.....	1
Copyright @ 2008 - <i>l.i.d.o.</i> - DIST University of Genoa	1
Abstract	2
Index of Contents.....	3
1. Introduction	4
2. Deployment of the infrastructure	5
2.1. Configuring the desktop platform.....	5
2.2. Allowed users	6
2.3. Light platform installation	6
2.4. Configuration files.....	6
2.5. The GUI	8

1. Introduction

This brief manual is focused on the deployment and execution of a multi-agent application which will partially run on a mobile device.

AgentService Mobile has been developed with the main goal of executing usual AgentService agents, developed for a standard AgentService installation, on a mobile device as a PDA or a Smart Phone. The agent developer has only to recompile its agents by using the *Compact Framework 3.5*.

In order to execute the distributed multi-agent application the platform administrator must perform the following easy steps:

- *To enable* the Windows Communication Foundation on the desktop platform.
- *To create* an xml file with the allowed user credentials.
- *To install* the mobile infrastructure on the device.
- *To configure* the connection parameters.

These points are going to be discussed in the following paragraphs.

From a general point of view the system is organized as shown in Figure 1. The *desktop platform* exposes its internal basic services like the *Directory Facilitator*, the *White Pages Service* and the *Messaging Module* as web services, through a *SOA server* implemented by using the *Windows Communication Foundation (WCF)*. The mobile platform runs a *SOA client* which is able to consume the aforementioned web services. The mobile platform is essentially able to manage three aspects:

- *Agent execution*: it creates the agent instances.
- *Proxy runtime*: it hides the complex interaction between the *SOA client* and *server* through a proxy which replicates the runtime interface which is usually at the agent disposal.
- *Cyclical download of the event notifications* for each agent hosted in the mobile platform. Events as incoming conversations and incoming messages are signaled to the recipient agents by a thread which cyclically polls the server.

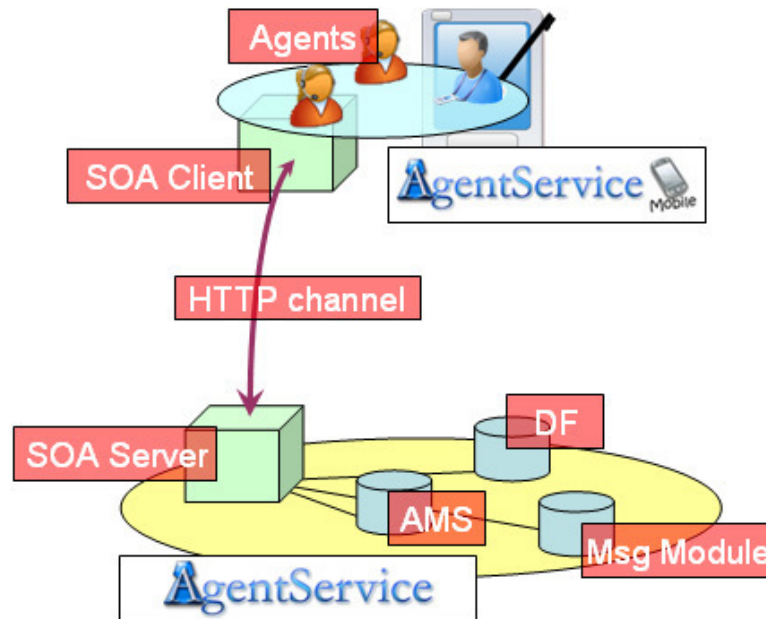


Figure 1: the whole architecture.

2. Deployment of the infrastructure

2.1. Configuring the desktop platform

Install a standard AgentService platform and then open the file `conf/platform.conf`. You must essentially configure three parameters:

- `system.wcfserver.enable` (true or false) in order to run the WCF Server.
- `system.wcfserver.ip` for the local IP Address.
- `system.wcfserver.tcport` is the port opened for the WCF Server.

Other parameters are the following:

- `system.wcfserver.debug` (*true or false*), true to see debug information.
- `system.wcfserver.checktime` (in seconds) is the timeout for the deletion of a mobile device (and then of its mobile agents) which does not interact with the platform for the submitted value seconds. A mobile platform running on a device is considered *alive* if periodically polls the server acquiring the list of events for the mobile platform (see 2.4).
- `system.wcfserver.msgsize`: is the maximum size for a message exchanged between a mobile platform and the desktop platform.

2.2. Allowed users

In order to allow a mobile device to connect to the platform, the user must be authenticated.

In the root of the AgentService installation, create a file named *users.xml* and fill it with *usernames* and *passwords*:

```
<Users>
  <User userid="andrea" password="123"/>
  <User userid="alberto" password="345"/>
  <User userid="davide" password="567"/>
</Users>
```

Figure 2: user credentials.

The desktop platform is now configured. Be sure that the computer on which the platform runs is reachable from the mobile device (check the network configuration, firewalls, etc.).

2.3. Light platform installation

Copy in your mobile device the folder containing libraries and the executable file for running the mobile platform. Compile the AgentServiceMobile Visual Studio Project and copy the following assemblies from the *bin\debug* directory to a folder on the mobile device:

- *ADOHelper.dll*
- *AgentService.dll*
- *AgentServiceInterface.dll*
- *CompactFormatterPlus.dll*
- *ISerial.dll*
- *MobilePractices.OpenFileDialogEx.dll*
- *AgentServiceMobile.exe*

2.4. Configuration files

In order to save configuration data, you must create an xml file containing the basic information. You must create a *config.xml* file in the main folder you created in the previous paragraph, with this tag and these attributes:

```
<Platform
  ip="130.251.22.100"
  port="8383"
  pollingFreq = "2000"
  platformName = "AgentService"
  logFile="log.txt"
  user="user"
  password="pwd"
/>
```

Figure 3: the config.xml file.

ip and *port* point the address of the remote AgentService desktop platform. The *pollingFreq* (milliseconds) attribute states the frequency of the polling thread which downloads from the remote platform all the events addressed to the agents hosted by the mobile platform.

The *platformName* indicates the name of the remote platform. Formally, the mobile platform is a branch of the remote platform, and then its mobile agents really belong to that platform. *LogFile* contains the name of the file showing the log of the platform execution. *User* and *password* contain the credentials which grant the access to the platform.

Another file which helps the user to automate the starting of the platform is an xml file that reports the list of agents which must be created.

```
<Agents>
  <Agent
    assembly = "LidoMobile.dll"
    namespace = "LidoMobileClient.Student"
    name = "Agent007"
    persistent = "false"
    configFile = ""
  />
</Agents>
```

Figure 4: the list of agents to be created.

As shown in Figure 4, we want to instantiate only one type of agent: its class full name is *LidoMobileClient.Student* and the name of the instance we want to create is *Agent007*. The definition of the agent is stored in the assembly *LidoMobile.dll*. Usually the user defined assemblies containing the agent definitions are stored in the main folder of the AgentServiceMobile installation on the mobile device. The *configFile* contains a possible user defined configuration file. The *persistence* is currently under development, so this feature is, at the present moment, disabled.

You can name this file as you want: as shown in the next paragraph, you can select from the graphical interface the batch file you prefer.

2.5. The GUI

Before starting your multi-agent application, be sure that the remote desktop platform is in execution. Click on the *AgentServiceMobile.exe* and the main form will appear.



Figure 5: the main form.

From the main form you can modify the configuration you entered in the *config.xml* file. By clicking on the *Load Agents* button you can select the agent types and the instances you want to create. Finally, click on *Execute* to start the mobile platform.



Figure 6: the platform configuration.

Figure 6 shows the configuration form where you can set all the parameters contained in *config.xml*. The second screenshot illustrates how to select the batch file containing the references to the agent classes and instances.

Finally, in Figure 6 the monitoring form shows the execution of the mobile platform. Click on *Start* when you want to connect to the remote platform and create agent instances and click on *Stop and Close* in order to abort the application (only the mobile platform: the desktop one continues to run).



Figure 7: the mobile platform console.