Using the Back-to-Back User Agent service
Running provided proof-of-concept services
Installation and deployment guide

by Luis Teixeira
26 October 2005

Introduction

The Back-to-Back User Agent (B2BUA) is a SIP entity that offers a means of actively controlling a call from the beginning to the end. To achieve this behaviour it acts both as a User Agent Server (UAS) and User Agent Client (UAC), maintaining dialog state and participating in all requests that are sent[i].

The B2BUA service that this document refers to is an implementation of this entity that is deployed in the JSLEE container as a component intended to be reused by services requiring its facilities.

Architecture

From the network and component point of view the B2BUA architecture is made up by the UAS and UAC elements and a generic interface for 3rd Party Call Control (3PCC), allowing both network initiated calls and service initiated calls to be controlled. For each of these two cases there are concrete interfaces exposing the methods particular to each one.
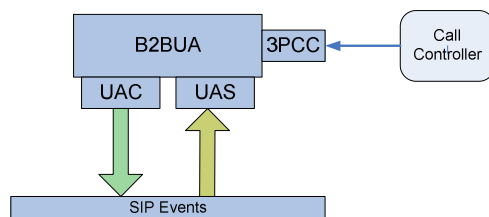
Figure 1 – B2BUA external architecture

SIP events coming from the network enter the service through the SIP Resource Adaptor, reaching the Service Building Blocks (SBB's) that implement the UAC and UAS components. The 3PCC component is accessed through SBB Local Interfaces that expose the methods that can be accessed by other services that require the B2BUA facility.

Using the Back-to-Back User Agent service

To run the B2BUA service the following conditions must be verified:
- The Mobicents AS is running;
- The SIP RA is deployed;

The steps to install this component are:
- Modify the `com.ptin.xmas.control.common.CallControllerListenerImpl.java` replacing the line where the line with the address `abc@10.112.128.233` with the URI corresponding to the destination SIP Phone;
- `ant service-deploy`.

To test if the service is working correctly configure the SIP phones not to register to a domain and place a call to the address where the B2BUA is running. It can be something like `xpto@<MOBICENTS_IP>`.

Running provided proof-of-concept services

Two sample services are available: the CallDuration example, which limits the duration of a call to a given time, and the Click-to-Dial (C2D) service, allowing a user to perform operations on top of active calls, like hanging-up and rejecting calls. The interaction with this service is achieved by means of a XMPP chat window, in which a user can get a listing of the active

calls and perform any of the 3PCC operations.

C2D Service

To install and run the service make sure the B2BUA service is running and the XMPP RA is also installed and active, with a connection to the corresponding server. Prior to deploying the service, make sure you:

- Select a XMPP client like PSI and launch it creating an account on the server for you.
- Modify the `com.ptin.xmas.app.BaseC2 DSbb.java` file, replacing the address data with the one corresponding to your case, in the `onMessage` event handler.

Once those steps have been achieved do:

```
ant service-deploy
```

and the service should now be running. In order to test it do the following:
- with the SIP phone originate a call to the B2BUA. Answer the call on the other phone;
- Open a chat window under the available XMPP service (must match the XMPP service to which messages are sent on the `com.ptin.xmas.app.BaseC2 DSbb.java`) and type the following commands:

```
>SHOW CALLS
1950b172b2389775
>HANGUP 1950b172b2389775
```

the call should hang-up correctly.

CallDuration Service

To install this service simply make sure the B2BUA is running. Run the command:

```
ant service-deploy
```

To test the service simply attempts a call to the B2BUA address and answer the call on the other phone. The default duration of the call is 10 seconds. It can be modified by changing the `duration` profile attribute stored in the default profile that is in the `DurationProfiles` profile table. This operation can be done through the JMX console. Changing this value will affect subsequent calls.

---

[i] SIP: Session Initiation Protocol: http://www.faqs.org/rfcs/rfc3261.html