Anexo A
Instalación de las plataformas JAIN SLEE

A. Instalación de las plataformas JAIN SLEE

A.1. Instalación de Rhino

Una vez que se ha desempaquetado el fichero *RhinoSDK-1.4.1-ga.tar* utilizando el comando *tar*, ya se puede proceder a la instalación del SLEE:

```
[vhros@rhinoslee PFC]$  tar xvf  RhinoSDK-1.4.1-ga.tar
[vhros@rhinoslee PFC]$  cd rhino-sdk-install
```

Sin embargo, previo a la instalación del SLEE, es conveniente iniciar el servidor PostgreSQL [27]. Para ello, lo primero es fijar la variable de entorno que almacenará la ruta hasta el directorio donde se guardará la información de la base de datos, para lo cual se modificará el fichero de configuración de la shell del usuario:

```
[vhros@rhinoslee ~]$  gedit .bash_profile
```

A continuación se incorpora la variable de entorno *PGDATA*, indicando la ruta donde se quiere almacenar la información de las bases de datos:

```
export PGDATA=/home/vhros/PFC/db/data
```

Recordar que se está haciendo uso de la versión de Java 1.5.0, y no la 1.4.2 que trae el Linux Fedora Core 4 por defecto, así que también se tendrá que añadir la siguiente variable de entorno al fichero de configuración:

```
export JAVA_HOME=/home/vhros/PFC/jdk1.5.0_05
```

Se guarda el archivo y se sale del editor de texto. Para aplicar los cambios se puede reiniciar el sistema, o bien escribir las mismas líneas en la shell que se esté usando, pero teniendo en cuenta que los cambios se perderán en cuanto se cierre dicha shell.

El siguiente paso es crear las estructuras necesarias para iniciar el servidor de base de datos. Esto se consigue con la orden:

```
[vhros@rhinoslee ~]$ initdb
```

Una vez hecho esto ya se puede iniciar el servidor de base de datos, para lo que tienen dos alternativas:

```
[vhros@rhinoslee ~]$ /usr/bin/postmaster -D $PGDATA
```

o

```
[vhros@rhinoslee ~]$ /usr/bin/pg_ctl -D $PGDATA -l logfile start
```

Ya se puede regresar al directorio *rhino-sdk-install* y ejecutar el *script rhino-install.sh* para iniciar el proceso de instalación:

```
[vhros@rhinoslee rhino-sdk-install]$ ./rhino-install.sh
```

Lo que queda es seguir los pasos que vaya indicando el proceso de instalación y se tendrá instalado el SLEE.

A.1.1. Iniciar la base de datos

Al instalar el SLEE, teniendo iniciado el servidor PostgreSQL, se crea por defecto la base de datos *rhino_sdk* para almacenar el estado del SLEE en cada momento. Pues bien, esta base de datos tiene que ser inicializada antes de que el Rhino SLEE pueda ser utilizado:

```
[vhros@rhinoslee rhino]$ ./init-management.sh
```

A.1.2. Iniciar el Rhino SLEE

Ya se puede iniciar el SLEE ejecutando el *script*:

```
[vhros@rhinoslee rhino]$ ./start-rhino.sh
```

Esto inicia el proceso autosuficiente del Rhino SLEE:

1. El *host* lanza un proceso de Máquina Virtual Java.
2. El nodo genera y lee su configuración.
3. El nodo se conecta a la base de datos y sincroniza la memoria de trabajo principal.
4. El SLEE pondrá el estado de arrancado (RUNNING).
5. El Rhino SDK SLEE está listo para recibir comandos de gestión.

Para parar el Rhino SDK SLEE simplemente se utiliza el *script stop-rhino.sh*.

```
[vhros@rhinoslee rhino]$ ./stop-rhino.sh --nice
```

A.2. Instalación de Mobicents

En cuanto a la instalación, no se va a explicar como se instala cada uno de los requisitos 9.2. Sin embargo, resaltar la conveniencia de no instalar nada en un directorio que contenga espacios en blanco en su ruta, ya que puede ocasionar problemas, y no olvidar la creación de las correspondientes variables de entorno:

- JAVA_HOME=C:\jdk1.5.0_05
- ANT_HOME=C:\apache-ant-1.6.5

- PATH=%PATH%;%ANT_HOME%\bin
- JBOSS_HOME=C:\PFC\jboss-3.2.6

Lo que sí se verá es la instalación y el arranque del servidor Mobicents, ya que no se dispone de una guía detallada como la que pueden tener los otros componentes instalados.

Lo primero será acudir al sitio web *http://mobicents.org* y registrarse. Una vez hecho el registro, y tras unirse al proyecto Mobicents, se procede a la descarga del servidor Mobicents vía CVS (*Concurrent Versions System - Sistema de Control de Versiones*). El CVS implementa un sistema de control de versiones, que mantiene el registro de todo el trabajo y los cambios en la implementación de un proyecto, y permite que distintos desarrolladores (potencialmente situados a gran distancia) colaboren.

Finalmente, para realizar la descarga se seguirán los siguientes pasos:

1. Arrancar el WinCvs.
2. Seleccionar *Admin → Login...* e introducir el CVSroot correspondiente al registro *(:pserver:torosvi@cvs.dev.java.net:/cvs*) (Ilustración 1). Después será pedido el *password* (Ilustración 2), y una vez introducido y aceptado ya se podrá iniciar la descarga del proyecto.
3. Seleccionar *Remote → Checkout module...* e introducir el nombre del proyecto a descargar (*mobicents*) y la carpeta donde se quiere que se descargue (Ilustración 3).

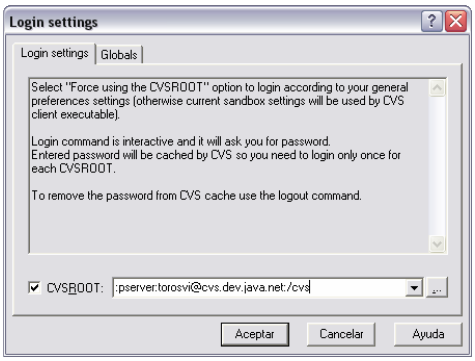

Ilustración 1: WinCvs - Login.

Ilustración 2: WinCvs - Password.

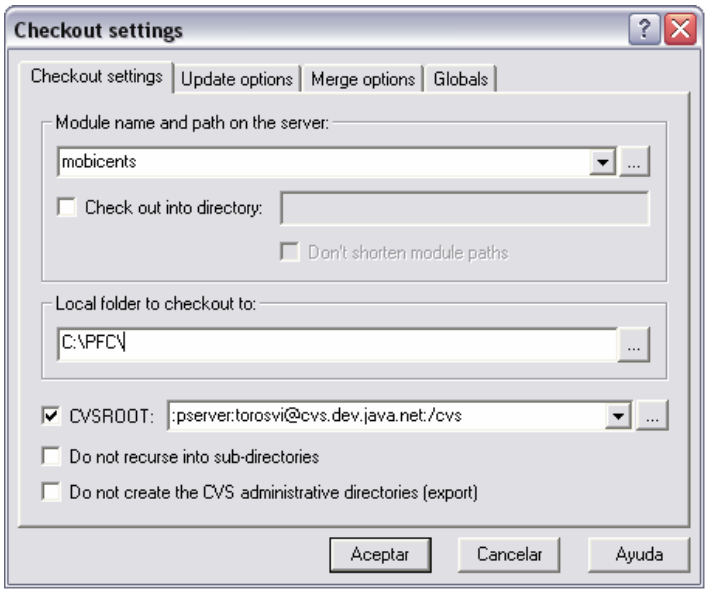

Ilustración 3: WinCvs - Checkout.

Finalizada la descarga, se procede a hacer el despliegue mediante el uso de la herramienta Ant. Para ello, se abrirá una ventana de MS-DOS, se acudirá al directorio correspondiente al proyecto Mobicents, donde se encuentra el fichero constructor *build.xml*, y se ejecutará la siguiente línea de comandos:

```
C:\PFC\mobicents>ant auto-deploy-sip
```

De esta forma, no sólo se llevará a cabo la compilación, sino que también se prepararán los servicios *proxy* y *registrar* para iniciarse en cuanto se arranque el SLEE.

Ya sólo queda arrancar el SLEE, para lo cual se acudirá al directorio *%JBOSS_HOME%* y se ejecutará lo siguiente:

```
C:\PFC\jboss-3.2.6>bin\run.bat -c all -b p4
```

Decir que *-c* se corresponde con el nombre de configuración del servidor (*C:\PFC\jboss-3.2.6\server\all*), que es *all*, el *-b* es el *host* o la dirección IP correspondiente al equipo donde arranca el SLEE y *p4* es precisamente el nombre de dicho equipo.

Para asegurarse de que todo ha ido bien, se abrirá la consola JMX y se comprobará que se pueden ver los MBeans bajo el dominio del SLEE (Ilustración 4).

Para terminar con este apartado, se dirá que la información correspondiente al Adaptador de Recurso SIP se encuentra en el fichero *sipra.properties* (C.2) en la ruta *C:\PFC\mobicents\ra\sipra\ra\src\org\mobicents\slee\resource\sip*. En dicho fichero se hace un comentario al respecto de la dirección IP utilizada por el adaptador de recurso,

y cómo se puede indicar esa dirección por línea de comandos, como se ha hecho al escribir *-b p4*, si la variable *javax.sip.IP_ADDRESS* no está especificada.

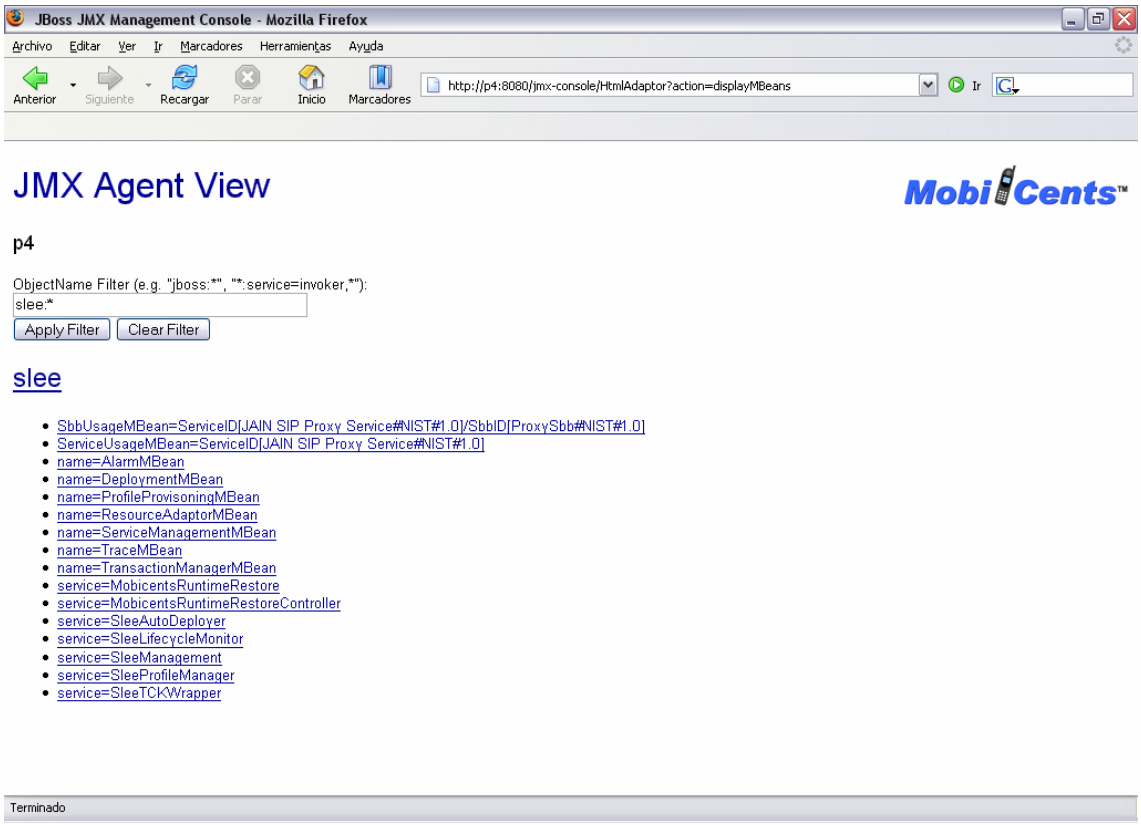

Ilustración 4: Consola JMX.

Anexo B
Código Rhino

B. Código Rhino

En este anexo se incluirá el código correspondiente a la plataforma Rhino de Open Cloud que pueda resultar de utilidad para el seguimiento de la memoria.

B.1. init-management-db.sh

```
#!/bin/sh

RHINO_HOME=`dirname $0`
# Read the values of our configuration
. $RHINO_HOME/read-config-variables
cd $RHINO_HOME

if [ -z "$PSQL_CLIENT" -o "$PSQL_CLIENT" = "-" ]; then
    die "\$PSQL_CLIENT has not been configured. Check your
config/config_variables file."
fi

$PSQL_CLIENT -h $MANAGEMENT_DATABASE_HOST -U $MANAGEMENT_DATABASE_USER
-p $MANAGEMENT_DATABASE_PORT template1 <<EOF
drop database $MANAGEMENT_DATABASE_NAME;
create database $MANAGEMENT_DATABASE_NAME;
\c $MANAGEMENT_DATABASE_NAME;

create table versioning (
  name          text    not null primary key,
  application   text    not null,
  ejb           text    not null,
  description   text    not null,
  version       integer not null
);
COMMENT ON TABLE versioning IS 'EJB CMP metadata';

CREATE TABLE keyspaces (
  dbid       text    not null,
  key_id     text    not null,
  mode       int4    not null,
  timeout    int4    not null,
  table_name text    not null,
  primary key (dbid, key_id, mode)
);
COMMENT ON TABLE keyspaces IS 'Persistent-memdb metadata';

CREATE TABLE timestamps (
  dbid        text    not null   primary key,
  era         int8    not null,
  last_update int8    not null
);
COMMENT ON TABLE timestamps IS 'Persistent-memdb metadata';

create table registrations ( -- for SIP location service
  sipaddress varchar(80) not null,
  contactaddress varchar(80) not null,
  expiry bigint,
```

```
  qvalue integer,
  cseq integer,
  callid varchar(80),
  primary key (sipaddress, contactaddress)
);
COMMENT ON TABLE registrations IS 'SIP Location Service
registrations';

create table findme ( -- for SIP Find Me Follow Me Service
  sipaddress varchar(80) not null,
  backup_sipaddress varchar(80) not null,
  seq_no integer,
  primary key (sipaddress, seq_no)
);
COMMENT ON TABLE findme IS 'FMFM Subscribers';

EOF
```

B.2. config_variables

```
RHINO_HOME=/home/vhros/PFC/rhino
RHINO_BASE=/home/vhros/PFC/rhino
RHINO_WORK_DIR=/home/vhros/PFC/rhino/work
JAVA_HOME=/home/vhros/PFC/jdk1.5.0_05
FILE_URL=file:
MANAGEMENT_DATABASE_HOST=localhost
MANAGEMENT_DATABASE_USER=vhros
MANAGEMENT_DATABASE_PORT=5432
MANAGEMENT_DATABASE_NAME=rhino_sdk
MANAGEMENT_DATABASE_PASSWORD=*******
PSQL_CLIENT=/usr/bin/psql
RMI_MBEAN_REGISTRY_PORT=1199
RMI_MBEAN_OBJECT_PORT=1200
JMX_SERVICE_PORT=1202
HEAP_SIZE=256m
RHINO_PUBLIC_STORE_PASS=changeit
RHINO_PRIVATE_STORE_PASS=changeit
RHINO_PUBLIC_KEY_PASS=changeit
RHINO_PRIVATE_KEY_PASS=changeit
WEB_CONSOLE_HTTP_PORT=
WEB_CONSOLE_HTTPS_PORT=8443
WEB_CONSOLE_KEY_PASS=changeit
WEB_CONSOLE_STORE_PASS=changeit
WEB_CONSOLE_HOSTNAME=rhinoslee.proxy.sip
RHINO_PASSWORD=password
RHINO_USERNAME=admin
LOCALIPS="[0:0:0:0:0:0:0:1%1] 127.0.0.1
[fe80:0:0:0:20f:b0ff:fe6b:4120%2] 172.16.17.179"
RHINO_WATCHDOG_DUMPTHREADS=/home/vhros/PFC/rhino/dumpthreads.sh
```

B.3. mlet.conf

```
<?xml version="1.0" encoding="ISO-8859-1"?>

<!DOCTYPE mlets PUBLIC "-//Open Cloud Ltd.//DTD JMX MLet Config
1.1//EN" "http://www.opencloud.com/dtd/mlet_1_1.dtd">
```

```
<mlets>
    <mlet enabled="true">
        <classpath>
           <jar-url>@FILE_URL@@RHINO_BASE@/lib/notification-
recorder.jar</jar-url>
           <security-permission-spec>
            grant codeBase "@FILE_URL@@RHINO_BASE@/lib/notification-
recorder.jar" {
              permission javax.management.MBeanPermission
"javax.management.MBeanServerDelegate#-
[JMImplementation:type=MBeanServerDelegate]","addNotificationListener"
;

              permission javax.management.MBeanPermission
"com.opencloud.rhino.management.notification.Notifier#-
[SLEE:name=NotificationService]", "addNotificationListener";
              permission javax.management.MBeanPermission
"com.opencloud.rhino.management.trace.Trace#-[SLEE:name=Trace]",
"addNotificationListener";
              permission javax.management.MBeanPermission
"com.opencloud.rhino.management.alarm.Alarm#-[SLEE:name=Alarm]",
"addNotificationListener";
            };
           </security-permission-spec>
        </classpath>

<class>com.opencloud.slee.mlet.notificationrecorder.NotificationRecord
er</class>
    </mlet>

    <mlet enabled="true">
      <classpath>
            <jar-url>@FILE_URL@@RHINO_BASE@/lib/jmxr-adaptor.jar</jar-
url>
            <jar-
url>@FILE_URL@@RHINO_BASE@/client/lib/jmxremote.jar</jar-url>
            <jar-url>@FILE_URL@@RHINO_BASE@/client/lib/jmxri-
1.2.1.jar</jar-url>
            <security-permission-spec>
                grant{

                  permission javax.security.auth.AuthPermission
"createLoginContext.other";
                  permission javax.security.auth.AuthPermission
"createLoginContext.jmxr-adaptor";

                  permission java.security.SecurityPermission
"createAccessControlContext";

                  permission java.lang.RuntimePermission
"setContextClassLoader";
                  permission java.lang.RuntimePermission
"getClassLoader";
                  permission java.lang.RuntimePermission
"createClassLoader";
                  permission java.lang.RuntimePermission
"accessDeclaredMembers";
```

```
                permission java.lang.RuntimePermission
"accessClassInPackage.sun.rmi.registry";

                permission java.util.PropertyPermission
"rmissl.jmxr-adaptor-ssl.properties", "read";
                permission java.util.PropertyPermission
"jmx.remote.protocol.provider.pkgs","read";

                permission java.io.FilePermission
"@RHINO_CONFIG_DIR@/rmissl.rmi-adaptor.properties", "read";
                permission java.io.FilePermission "@RHINO_BASE@/rmi-
adaptor-server.keystore", "read";
                permission java.io.FilePermission "@RHINO_BASE@/rmi-
adaptor-client.keystore", "read";
                permission java.io.FilePermission
"@RHINO_BASE@/lib/jmxr-adaptor.jar", "read";
                permission java.io.FilePermission
"@RHINO_BASE@/client/lib/*", "read";
                permission java.io.FilePermission
"@RHINO_BASE@/client/lib/-", "read";
                //rhino libraries needs access for impersonation
                permission java.io.FilePermission
"@RHINO_BASE@/lib/jmxr-adaptor.jar","read";
                permission java.io.FilePermission
"@RHINO_BASE@/lib/*","read";
                permission java.io.FilePermission
"@RHINO_BASE@/lib/-","read";

                permission java.net.SocketPermission
"@LOCALIPS@","accept,connect,resolve";
                permission java.net.SocketPermission
"@LOCALIPS@:@RMI_MBEAN_REGISTRY_PORT@", "listen,resolve";
                permission java.net.SocketPermission
"@LOCALIPS@:@JMX_SERVICE_PORT@", "listen,resolve";

                //minimal standard jmx permissions required to
support web console
                //give jmx-remote the permissions to impersonate
anyone and do anything with mbeans
                permission
javax.management.remote.SubjectDelegationPermission "*";
                permission javax.management.MBeanServerPermission
"*";
                permission javax.management.MBeanPermission "*","*";

                //
                //Additional permission should be granted here if
connections from other
                //hosts are needed.
                //
                //permission java.net.SocketPermission "remotehost",
"accept,resolve";
                //permission java.io.FilePermission
"@RHINO_CONFIG_DIR@/mlet.conf","read";

            };
```

```
               //the jmx-remote role can impersonate the following
subjects credentials
              grant principal
com.opencloud.rhino.security.principal.RhinoRole "jmx-remote" {
                  permission
javax.management.remote.SubjectDelegationPermission
"com.opencloud.rhino.security.principal.RhinoRole.view";
                  permission
javax.management.remote.SubjectDelegationPermission
"com.opencloud.rhino.security.principal.RhinoRole.admin";
                  permission
javax.management.remote.SubjectDelegationPermission
"com.opencloud.rhino.security.principal.RhinoRole.invoke";
                  permission
javax.management.remote.SubjectDelegationPermission
"com.opencloud.rhino.security.principal.RhinoRole.limited";
              };

              //admin role
              grant principal
com.opencloud.rhino.security.principal.RhinoRole "admin" {
                  permission javax.management.MBeanPermission
"*","getMBeanInfo,getAttribute,getObjectInstance,queryNames,queryMBean
s,setAttribute,invoke,unregisterMBean,instantiate,getDomains,getClassL
oader";
                  permission java.io.FilePermission
"@RHINO_CONFIG_DIR@/mlet.conf","read";
              };

              //invoke role
              grant principal
com.opencloud.rhino.security.principal.RhinoRole "invoke"{
                  permission javax.management.MBeanPermission
"*","getMBeanInfo,getAttribute,getObjectInstance,queryNames,queryMBean
s,setAttribute,invoke";
              };

              //view role
              grant principal
com.opencloud.rhino.security.principal.RhinoRole "view"{
                permission javax.management.MBeanPermission
"*","getMBeanInfo,getAttribute,getObjectInstance,queryNames,queryMBean
s";
              };

              //limited role members must be members of at least one
other service role below e.g. sip
              grant principal
com.opencloud.rhino.security.principal.RhinoRole "limited" {
                  permission javax.management.MBeanPermission
"com.opencloud.rhino.management.service.ServiceManagement#-
[SLEE:name=ServiceManagement]","getMBeanInfo,queryMBeans";
              };

              //service specific roles
              grant principal
com.opencloud.rhino.security.principal.RhinoRole "sip" {
                  //get access to the profile provisioning screen
```

```
                    permission javax.management.MBeanPermission
"com.opencloud.rhino.management.profile.ProfileProvisioning#-
[SLEE:name=ProfileProvisioning]","getMBeanInfo,queryMBeans,invoke";
                    permission javax.management.MBeanPermission
"com.opencloud.rhino.management.deployment.Deployment#getProfileSpecif
ications[SLEE:name=Deployment]","invoke";
                    permission javax.management.MBeanPermission
"com.opencloud.rhino.management.SleeManagement#DeploymentMBean[SLEE:na
me=SleeManagement]","getAttribute";

                    permission javax.management.MBeanPermission
"com.opencloud.rhino.deployed.profilespec.Open_Cloud.SipConfigProfile_
1_0.Profile#-
[SLEE/Profiles:name=conf,table=SipConfig]","queryNames,queryMBeans,get
MBeanInfo,invoke,getAttribute,setAttribute";
                  };
            </security-permission-spec>
      </classpath>
        <class>com.opencloud.slee.mlet.jmxr.JMXRAdaptor</class>
        <!-- jmx remote service protocol -->
        <arg>
            <type>java.lang.String</type>
            <value>rmi</value>
        </arg>
        <!--socket properties-->
        <arg>
            <type>java.lang.String</type>
            <value>jmxr-adaptor-ssl.properties</value>
        </arg>
        <!--login context-->
        <arg>
            <type>java.lang.String</type>
            <value>jmxr-adaptor</value>
        </arg>
        <!--the local rmi registry port -->
        <arg>
            <type>int</type>
            <value>@RMI_MBEAN_REGISTRY_PORT@</value>
        </arg>
        <!--the local jmx connector port -->
        <arg>
            <type>int</type>
            <value>@JMX_SERVICE_PORT@</value>
        </arg>
    </mlet>
    <mlet enabled="true">
        <classpath>
            <jar-url>@FILE_URL@@RHINO_BASE@/client/lib/web-console-
jmx.jar</jar-url>
            <jar-url>@FILE_URL@@RHINO_BASE@/client/lib/web-
console.war</jar-url>
            <jar-
url>@FILE_URL@@RHINO_BASE@/client/lib/javax.servlet.jar</jar-url>
            <jar-url>@FILE_URL@@RHINO_BASE@/client/lib/javamail-
api.jar</jar-url>
            <jar-
url>@FILE_URL@@RHINO_BASE@/client/lib/activation.jar</jar-url>
```

```
            <jar-
url>@FILE_URL@@RHINO_BASE@/client/lib/jmxremote.jar</jar-url>
            <jar-url>@FILE_URL@@RHINO_BASE@/client/lib/jmxri-
1.2.1.jar</jar-url>
            <jar-
url>@FILE_URL@@RHINO_BASE@/client/lib/endorsed/org.mortbay.jaas.jar</j
ar-url>
            <jar-
url>@FILE_URL@@RHINO_BASE@/client/lib/endorsed/org.mortbay.jetty.jar</
jar-url>
            <jar-
url>@FILE_URL@@RHINO_BASE@/client/lib/endorsed/org.mortbay.jetty.plus.
jar</jar-url>
            <!-- Uncomment to use xerces
            <jar-
url>@FILE_URL@@RHINO_BASE@/client/lib/endorsed/xercesImpl.jar</jar-
url>
            <jar-url>@FILE_URL@@RHINO_BASE@/client/lib/endorsed/xml-
apis.jar</jar-url>
            <jar-
url>@FILE_URL@@RHINO_BASE@/client/lib/endorsed/xmlParserAPIs.jar</jar-
url>
            -->

            <security-permission-spec>

              keystore "@FILE_URL@@RHINO_BASE@/web-console.keystore";

              grant codeBase "@FILE_URL@@RHINO_BASE@/client/lib/web-
console-jmx.jar"{
                  permission javax.management.MBeanPermission
"com.opencloud.slee.mlet.web.WebConsole#-
[name:=WebConsole]","registerMBean";
              };

              grant signedBy "web-console-lib" {

                permission java.security.SecurityPermission
"createAccessControlContext";
                permission java.lang.RuntimePermission
"setContextClassLoader";

                permission java.util.PropertyPermission
"org.mortbay.xml.XmlParser.NotValidating", "read,write";
                permission java.util.PropertyPermission
"org.mortbay.jetty.servlet.SessionCookie","read";
                permission java.util.PropertyPermission
"org.mortbay.jetty.servlet.SessionURL","read";
                permission java.util.PropertyPermission
"java.security.auth.login.config","read";
                permission java.util.PropertyPermission
"log4j.rootLogger","read";
                permission java.util.PropertyPermission
"com.opencloud.slee.mlet.web.ui.XSLTransformerFactory.base.url","read,
write";
                permission java.util.PropertyPermission
"*","read,write";
```

```
                permission java.io.FilePermission "client/lib/web-
console.war","read";
                permission java.io.FilePermission
"@RHINO_CONFIG_DIR@/rhino.passwd", "read";
                permission java.io.FilePermission
"${java.home}/lib/jaxp.properties", "read";
                //required for self operation as an mlet
                permission javax.management.MBeanPermission
"com.opencloud.slee.mlet.web.ui.UIParserServlet#-
[Adaptors:name=ui]","registerMBean";

                //uncomment to ignore role based security
                //permission javax.management.MBeanPermission "*","*";

                permission javax.security.auth.AuthPermission
"modifyPrincipals";
                permission javax.management.MBeanServerPermission
"findMBeanServer";
             };

             grant {
                permission java.security.SecurityPermission
"insertProvider.SunJSSE";
                permission javax.security.auth.AuthPermission
"doAs";
                permission javax.security.auth.AuthPermission
"doAsPrivileged";
                permission javax.security.auth.AuthPermission
"getSubject";
                permission javax.security.auth.AuthPermission
"createLoginContext.web-console";
                permission javax.security.auth.AuthPermission
"createLoginContext.other";

                //Let the code read everything, xml and xalan read a
lot of these things...
                permission java.util.PropertyPermission "*", "read";
                permission java.util.PropertyPermission
"javax.xml.parsers.SAXParserFactory", "read,write";
                permission java.util.PropertyPermission
"org.mortbay.xml.XmlParser.NotValidating", "read,write";

                permission java.lang.RuntimePermission
"setContextClassLoader";
                permission java.lang.RuntimePermission
"getClassLoader";
                permission java.lang.RuntimePermission
"createClassLoader";
                permission java.lang.RuntimePermission
"accessClassInPackage.sun.reflect";

                permission java.io.FilePermission
"${java.home}/lib/jaxp.properties", "read";
                permission java.io.FilePermission
"jar:@FILE_URL@@RHINO_BASE@/client/lib/web-console.war!/","read";
                permission java.io.FilePermission "@RHINO_BASE@/web-
console.keystore", "read";
```

```
                    permission java.io.FilePermission
"@RHINO_BASE@/client/lib/endorsed/-", "read";
                    permission java.io.FilePermission
"@RHINO_BASE@/client/lib/javax.servlet.jar", "read";
                    permission java.io.FilePermission
"@RHINO_BASE@/client/lib/web-console.war", "read";
                    permission java.io.FilePermission
"@RHINO_BASE@/client/lib/web-console-jmx.jar", "read";
                    permission java.io.FilePermission
"@RHINO_HOME@/logs", "read,write";
                    permission java.io.FilePermission
"@RHINO_HOME@/logs/*", "read,write";
                    permission java.io.FilePermission
"@RHINO_WORK_DIR@", "read,write";
                    permission java.io.FilePermission
"@RHINO_WORK_DIR@/-", "read,write,delete";
                    permission java.io.FilePermission
"@RHINO_CONFIG_DIR@/jetty.xml", "read";

                    // Allow HTTP connections to the adaptor from
localhost.
                    permission java.net.SocketPermission
"@LOCALIPS@:@WEB_CONSOLE_HTTPS_PORT@", "listen,resolve";
                    permission java.net.SocketPermission
"@LOCALIPS@:@WEB_CONSOLE_HTTP_PORT@", "listen,resolve";
                    permission java.net.SocketPermission "@LOCALIPS@",
"accept,connect,resolve";
                    //permission java.net.SocketPermission "remotehost",
"accept,resolve";
                permission java.net.SocketPermission "*",
"accept,resolve";

                    permission javax.management.MBeanPermission
"com.opencloud.slee.mlet.web.ui.UIParserServlet#-
[Adaptors:name=ui]","registerMBean";
                    permission javax.management.MBeanServerPermission
"findMBeanServer";

                    permission javax.management.MBeanPermission
"*","getAttribute";
                };

                //admin role
                grant principal
com.opencloud.rhino.security.principal.RhinoRole "admin" {
                    permission javax.management.MBeanPermission
"*","getMBeanInfo,getAttribute,getObjectInstance,queryNames,queryMBean
s,setAttribute,invoke,unregisterMBean,instantiate,getDomains,getClassL
oader";
                    permission java.io.FilePermission
"@RHINO_CONFIG_DIR@/mlet.conf","read";
                };

                //invoke role
                grant principal
com.opencloud.rhino.security.principal.RhinoRole "invoke"{
```

```
                  permission javax.management.MBeanPermission
"*","getMBeanInfo,getAttribute,getObjectInstance,queryNames,queryMBean
s,setAttribute,invoke";
              };

              //view role
              grant principal
com.opencloud.rhino.security.principal.RhinoRole "view"{
                permission javax.management.MBeanPermission
"*","getMBeanInfo,getAttribute,getObjectInstance,queryNames,queryMBean
s";
              };

            </security-permission-spec>
        </classpath>
        <class>com.opencloud.slee.mlet.web.WebConsole</class>
        <!--object name-->
        <arg>
            <type>java.lang.String</type>
            <value>ADAPTOR:name=WebConsole</value>
        </arg>
        <!--server configuration file-->
        <arg>
            <type>java.lang.String</type>
            <value>@FILE_URL@@RHINO_CONFIG_DIR@/jetty.xml</value>
        </arg>
        <!--remote host url - not used for local deployment-->
        <!--<arg>
            <type>java.lang.String</type>

<value>service:jmx:rmi:///jndi/rmi://127.0.0.1:1199/opencloud/rhino</v
alue>
        </arg>-->
    </mlet>
</mlets>
```

B.4. rhino-config.xml

```
<?xml version="1.0"?>

<!DOCTYPE rhino-config PUBLIC "-//Open Cloud Ltd.//DTD Rhino Config
0.9//EN" "http://www.opencloud.com/dtd/rhino-config_0_9.dtd">

<rhino-config>
  <file-locations>
    <temp-dir>@RHINO_WORK_DIR@/tmp</temp-dir>
    <deployment-dir
save="False">@RHINO_WORK_DIR@/deployments</deployment-dir>
    <state-dir>@RHINO_WORK_DIR@/state</state-dir>
    <per-node-mlet-conf>@FILE_URL@@RHINO_CONFIG_DIR@/mlet.conf</per-
node-mlet-conf>
    <notifications-
url>@FILE_URL@@RHINO_CONFIG_DIR@/notifications.xml</notifications-url>
    <savanna-base-url>@FILE_URL@@RHINO_CONFIG_DIR@/savanna</savanna-
base-url>
    <license-files>
```

```
      <license-url>@FILE_URL@@RHINO_HOME@/rhino-sdk.license</license-
url>
    </license-files>
  </file-locations>

  <cluster-admin-group>rhino-admin</cluster-admin-group>

  <activity-handler>
    <group-name>rhino-ah</group-name>
    <message-id>10000</message-id>
    <associated-memdb-name>ReplicatedMemoryDatabase</associated-memdb-
name>
  </activity-handler>

  <persistence-resources>
    <deployment>DeploymentFileDatabase</deployment>
    <replicated-sbbs>ReplicatedMemoryDatabase</replicated-sbbs>
    <non-replicated-sbbs>LocalMemoryDatabase</non-replicated-sbbs>
    <profiles>ProfileDatabase</profiles>
    <resource-adaptors>ReplicatedMemoryDatabase</resource-adaptors>
    <usage-state distributed="False"/>
  </persistence-resources>

  <rhino-monitoring>
    <group-name>rhino-monitoring</group-name>
    <message-id>19999</message-id>
  </rhino-monitoring>

  <ejb-resources>
    <!--
      Note: the names of the following resources must not be changed
as the
      core system depends on their presence:
        LocalMemoryDatabase
        ReplicatedMemoryDatabase
        ManagementDatabase

      Additionally, the SIP example services depend on the presence of
the
      JDBCResource resource.

      Other resource names are referenced only by other elements in
this
      configuration file and so may be modified as needed.
    -->


    <memdb-local>
      <jndi-name>LocalMemoryDatabase</jndi-name>
      <committed-size>50M</committed-size>
    </memdb-local>

    <memdb>
      <jndi-name>ReplicatedMemoryDatabase</jndi-name>
      <message-id>10002</message-id>
      <group-name>rhino-db</group-name>
      <committed-size>50M</committed-size>
    </memdb>
```

```
    <memdb>
      <jndi-name>ManagementDatabase</jndi-name>
      <message-id>10003</message-id>
      <group-name>rhino-db</group-name>
      <committed-size>10M</committed-size>
      <persistence>
        <persistence-instance>
          <datasource-
class>org.postgresql.jdbc3.Jdbc3SimpleDataSource</datasource-class>
          <dbid>rhino_sdk_management</dbid>
          <parameter>
            <param-name>serverName</param-name>
            <param-type>java.lang.String</param-type>
            <param-value>@MANAGEMENT_DATABASE_HOST@</param-value>
          </parameter>
          <parameter>
            <param-name>portNumber</param-name>
            <param-type>java.lang.Integer</param-type>
            <param-value>@MANAGEMENT_DATABASE_PORT@</param-value>
          </parameter>
          <parameter>
            <param-name>databaseName</param-name>
            <param-type>java.lang.String</param-type>
            <param-value>@MANAGEMENT_DATABASE_NAME@</param-value>
          </parameter>
          <parameter>
            <param-name>user</param-name>
            <param-type>java.lang.String</param-type>
            <param-value>@MANAGEMENT_DATABASE_USER@</param-value>
          </parameter>
          <parameter>
            <param-name>password</param-name>
            <param-type>java.lang.String</param-type>
            <param-value>@MANAGEMENT_DATABASE_PASSWORD@</param-value>
          </parameter>
          <parameter>
            <param-name>loginTimeout</param-name>
            <param-type>java.lang.Integer</param-type>
            <param-value>30</param-value>
          </parameter>
        </persistence-instance>
      </persistence>
    </memdb>

    <memdb>
      <jndi-name>ProfileDatabase</jndi-name>
      <message-id>10004</message-id>
      <group-name>rhino-db</group-name>
      <committed-size>50M</committed-size>
      <persistence>
        <persistence-instance>
          <datasource-
class>org.postgresql.jdbc3.Jdbc3SimpleDataSource</datasource-class>
          <dbid>rhino_sdk_profiles</dbid>
          <parameter>
            <param-name>serverName</param-name>
            <param-type>java.lang.String</param-type>
            <param-value>@MANAGEMENT_DATABASE_HOST@</param-value>
```

```
          </parameter>
          <parameter>
            <param-name>portNumber</param-name>
            <param-type>java.lang.Integer</param-type>
            <param-value>@MANAGEMENT_DATABASE_PORT@</param-value>
          </parameter>
          <parameter>
            <param-name>databaseName</param-name>
            <param-type>java.lang.String</param-type>
            <param-value>@MANAGEMENT_DATABASE_NAME@</param-value>
          </parameter>
          <parameter>
            <param-name>user</param-name>
            <param-type>java.lang.String</param-type>
            <param-value>@MANAGEMENT_DATABASE_USER@</param-value>
          </parameter>
          <parameter>
            <param-name>password</param-name>
            <param-type>java.lang.String</param-type>
            <param-value>@MANAGEMENT_DATABASE_PASSWORD@</param-value>
          </parameter>
          <parameter>
            <param-name>loginTimeout</param-name>
            <param-type>java.lang.Integer</param-type>
            <param-value>30</param-value>
          </parameter>
        </persistence-instance>
      </persistence>
    </memdb>

    <memdb>
      <jndi-name>DeploymentFileDatabase</jndi-name>
      <message-id>10005</message-id>
      <group-name>rhino-db</group-name>
      <committed-size>40M</committed-size>
      <persistence>
        <persistence-instance>
          <datasource-
class>org.postgresql.jdbc2.optional.SimpleDataSource</datasource-
class>
          <dbid>rhino_sdk_deployment_files</dbid>
          <parameter>
            <param-name>serverName</param-name>
            <param-type>java.lang.String</param-type>
            <param-value>@MANAGEMENT_DATABASE_HOST@</param-value>
          </parameter>
          <parameter>
            <param-name>portNumber</param-name>
            <param-type>java.lang.Integer</param-type>
            <param-value>@MANAGEMENT_DATABASE_PORT@</param-value>
          </parameter>
          <parameter>
            <param-name>databaseName</param-name>
            <param-type>java.lang.String</param-type>
            <param-value>@MANAGEMENT_DATABASE_NAME@</param-value>
          </parameter>
          <parameter>
            <param-name>user</param-name>
```

```
            <param-type>java.lang.String</param-type>
            <param-value>@MANAGEMENT_DATABASE_USER@</param-value>
          </parameter>
          <parameter>
            <param-name>password</param-name>
            <param-type>java.lang.String</param-type>
            <param-value>@MANAGEMENT_DATABASE_PASSWORD@</param-value>
          </parameter>
            <parameter>
              <param-name>loginTimeout</param-name>
              <param-type>java.lang.Integer</param-type>
              <param-value>30</param-value>
            </parameter>
        </persistence-instance>
      </persistence>
    </memdb>

    <jdbc>
      <jndi-name>JDBCResource</jndi-name>
      <datasource-
class>org.postgresql.jdbc2.optional.SimpleDataSource</datasource-
class>
      <parameter>
        <param-name>serverName</param-name>
        <param-type>java.lang.String</param-type>
        <param-value>@MANAGEMENT_DATABASE_HOST@</param-value>
      </parameter>
      <parameter>
        <param-name>portNumber</param-name>
        <param-type>java.lang.Integer</param-type>
        <param-value>@MANAGEMENT_DATABASE_PORT@</param-value>
      </parameter>
      <parameter>
        <param-name>databaseName</param-name>
        <param-type>java.lang.String</param-type>
        <param-value>@MANAGEMENT_DATABASE_NAME@</param-value>
      </parameter>
      <parameter>
        <param-name>user</param-name>
        <param-type>java.lang.String</param-type>
        <param-value>@MANAGEMENT_DATABASE_USER@</param-value>
      </parameter>
      <parameter>
        <param-name>password</param-name>
        <param-type>java.lang.String</param-type>
        <param-value>@MANAGEMENT_DATABASE_PASSWORD@</param-value>
      </parameter>
      <parameter>
        <param-name>loginTimeout</param-name>
        <param-type>java.lang.Integer</param-type>
        <param-value>30</param-value>
      </parameter>
      <connection-pool>
        <max-connections>15</max-connections>
        <idle-check-interval>0.0</idle-check-interval>
      </connection-pool>
    </jdbc>
```

```
    <jdbc>
      <jndi-name>ExternalDataSource</jndi-name>
      <datasource-
class>org.postgresql.jdbc2.optional.SimpleDataSource</datasource-
class>
      <parameter>
        <param-name>serverName</param-name>
        <param-type>java.lang.String</param-type>
        <param-value>localhost</param-value>
      </parameter>
      <parameter>
        <param-name>portNumber</param-name>
        <param-type>java.lang.Integer</param-type>
        <param-value>5432</param-value>
      </parameter>
      <parameter>
        <param-name>databaseName</param-name>
        <param-type>java.lang.String</param-type>
        <param-value>fmfm</param-value>
      </parameter>
      <parameter>
        <param-name>user</param-name>
        <param-type>java.lang.String</param-type>
        <param-value>fmfmslee</param-value>
      </parameter>
      <parameter>
      <param-name>password</param-name>
      <param-type>java.lang.String</param-type>
      <param-value>********</param-value>
      </parameter>
      <connection-pool>
        <max-connections>15</max-connections>
        <idle-check-interval>0.0</idle-check-interval>
      </connection-pool>
    </jdbc>
  </ejb-resources>
</rhino-config>
```

B.5. build.properties

```
# Location of Rhino client install
client.home=../../client

# Select location service implementation.
# If "usejdbclocation" property is true, JDBC location service will be
deployed.
# Default is to use Activity Context Naming implementation.
usejdbclocation=true

# Specify SIP RA config properties here, e.g.
ListeningPoints=10.0.0.1:5070/udp
sip.ra.properties=ListeningPoints=0.0.0.0:5060/udp;0.0.0.0:5060/tcp

# Default SBB trace level
# Possible values: Off, Severe, Warning, Config, Info, Fine, Finer,
Finest.
sbb.tracelevel=Finest
```

```
### Should not need to modify properties below ###

# Important paths
src=src
jars=jars
classes=classes
lib=lib

# Ant build properties
build.sysclasspath=ignore
failonerror=false

# SIP Resource Adaptor
sip.ra.jar=ocjainsip-1.2-ra.jar
sip.ra.type.jar=ocsip-resource-adaptor-type.jar
sip.ra.name=OCSIP 1.2, Open Cloud
sip.ra.entity=sipra

# Location Service - AC Naming or JDBC implementation
location.ac.jar=sip-ac-location-service.jar
location.ac.service=SIP AC Location Service 1.5, Open Cloud
location.ac.sbb=ACLocationSbb 1.5, Open Cloud
location.jdbc.jar=sip-jdbc-location-service.jar
location.jdbc.service=SIP JDBC Location Service 1.5, Open Cloud
location.jdbc.sbb=JDBCLocationSbb 1.5, Open Cloud

# Registrar Service
registrar.service.jar=sip-registrar-service.jar
registrar.service=SIP Registrar Service 1.5, Open Cloud
registrar.sbb=RegistrarSbb 1.5, Open Cloud

# Proxy Service
proxy.service.jar=sip-proxy-service.jar
proxy.service=SIP Proxy Service 1.5, Open Cloud
proxy.sbb=ProxySbb 1.5, Open Cloud

# FMFM
fmfm.service.jar=sip-fmfm-service.jar
fmfm.service=SIP FMFM Service 1.5, Open Cloud
fmfm.sbb=FindMeFollowMeSbb 1.5, Open Cloud
fmfm.proxy.sbb=ProxySbb 1.5-FMFM, Open Cloud

# B2BUA Service
b2bua.service.jar=sip-b2bua-service.jar
b2bua.service=B2BUA 1.0, Open Cloud
b2bua.sbb=B2BUA 1.0, Open Cloud
uac.sbb=UACDialogSbb 1.0, Open Cloud
uas.sbb=UASDialogSbb 1.0, Open Cloud
```

B.6. sip.properties

```
# This file contains properties that will be substituted
# into SBB deployment descriptors when building with Ant.

# Proxy SBB configuration
# Add names that the proxy host is known by. The first name in the
list
```

```
# will be treated as the Proxy's canonical hostname and will be used
in
# Via and Record-Route headers inserted by the proxy.
PROXY_HOSTNAMES=rhinoslee.proxy.sip,localhost,127.0.0.1
# Add domains that the proxy is authoritative for
PROXY_DOMAINS=rhinoslee.proxy.sip
PROXY_SIP_PORT=5060
PROXY_SIPS_PORT=5061
PROXY_LOOP_DETECTION=true

# FMFM SBB configuration
# Uses profiles for subscriber data by default, enable JDBC here.
FMFM_USE_JDBC=true
FMFM_PROFILE_TABLE_NAME=FMFMSubscribers
FMFM_DATASOURCE_NAME=jdbc/FMFMSubscribers

# Location service SBB impl - AC Naming or JDBC
# The SBB name will be set automatically in build.xml, do not set
here.
##LOCATIONSBB_NAME=ACLocationSbb
LOCATIONSBB_VENDOR=Open Cloud
LOCATIONSBB_VERSION=1.5

# Generic SIP SBB properties
SIP_RATYPE_NAME=OCSIP
SIP_RATYPE_VENDOR=Open Cloud
SIP_RATYPE_VERSION=1.2
SIP_PROVIDER=slee/resources/ocsip/1.2/provider
SIP_ACIFACTORY=slee/resources/ocsip/1.2/acifactory
SIP_LINKNAME=OCSIP
```

B.7. sbb-jar.xml (proxy)

```
<?xml version="1.0"?>

<!DOCTYPE sbb-jar PUBLIC "-//Sun Microsystems, Inc.//DTD JAIN SLEE SBB
1.0//EN" "http://java.sun.com/dtd/slee-sbb-jar_1_0.dtd">

<sbb-jar>

  <sbb>
    <description>SIP Proxy SBB</description>
    <sbb-name>ProxySbb</sbb-name>
    <sbb-vendor>Open Cloud</sbb-vendor>
    <sbb-version>1.5</sbb-version>
    <sbb-alias>PROXYSBB</sbb-alias>

    <sbb-ref>
      <sbb-name>@LOCATIONSBB_NAME@</sbb-name>
      <sbb-vendor>@LOCATIONSBB_VENDOR@</sbb-vendor>
      <sbb-version>@LOCATIONSBB_VERSION@</sbb-version>
      <sbb-alias>LOCATIONSBB</sbb-alias>
    </sbb-ref>

    <sbb-classes>

      <sbb-abstract-class reentrant="True">
```

```
        <sbb-abstract-class-name>
          com.opencloud.slee.services.sip.proxy.ProxySbb
        </sbb-abstract-class-name>

        <cmp-field>
          <cmp-field-name>proxyState</cmp-field-name>
        </cmp-field>
        <cmp-field>
          <cmp-field-name>forking</cmp-field-name>
        </cmp-field>
        <cmp-field>
          <cmp-field-name>recordRoute</cmp-field-name>
        </cmp-field>
        <cmp-field>
          <cmp-field-name>invite</cmp-field-name>
        </cmp-field>
        <cmp-field>
          <cmp-field-name>useLocationService</cmp-field-name>
        </cmp-field>
        <cmp-field>
          <cmp-field-name>proxyResponseListener</cmp-field-name>
          <sbb-alias-ref>PROXYSBB</sbb-alias-ref>
          <!-- Proxy refers to itself by default, this is intended for
when it
               is used standalone. If deployed with a separate
listener SBB,
               this field should be changed to refer to the listener.
-->
        </cmp-field>

        <get-child-relation-method>
          <sbb-alias-ref>
            LOCATIONSBB
          </sbb-alias-ref>
          <get-child-relation-method-name>
            getLocationServiceChildRelation
          </get-child-relation-method-name>
          <default-priority>0</default-priority>
        </get-child-relation-method>

      </sbb-abstract-class>

      <sbb-local-interface>
        <description>
          Used by other SBBs to proxy requests
        </description>
        <sbb-local-interface-name>
          com.opencloud.slee.services.sip.proxy.Proxy
        </sbb-local-interface-name>
      </sbb-local-interface>

      <sbb-activity-context-interface>
        <description>
          Used to store information about responses as they are
received
        </description>
        <sbb-activity-context-interface-name>
         com.opencloud.slee.services.sip.proxy.ProxyResponseContextACI
```

```
        </sbb-activity-context-interface-name>
      </sbb-activity-context-interface>

    </sbb-classes>

    <!-- SIP Requests -->
    <!-- Note Invite and Cancel use same initial event selector method
so they
         get same convergence name -->
    <event event-direction="Receive" initial-event="True">
      <event-name>InviteRequest</event-name>
      <event-type-ref>
        <event-type-name>javax.sip.message.Request.INVITE</event-type-
name>
        <event-type-vendor>javax.sip</event-type-vendor>
        <event-type-version>1.1</event-type-version>
      </event-type-ref>
      <initial-event-selector-method-name>initialEventSelect</initial-
event-selector-method-name>
    </event>

    <event event-direction="Receive" initial-event="True">
      <event-name>CancelRequest</event-name>
      <event-type-ref>
        <event-type-name>javax.sip.message.Request.CANCEL</event-type-
name>
        <event-type-vendor>javax.sip</event-type-vendor>
        <event-type-version>1.1</event-type-version>
      </event-type-ref>
      <initial-event-selector-method-name>initialEventSelect</initial-
event-selector-method-name>
    </event>

    <event event-direction="Receive" initial-event="True">
      <event-name>AckRequest</event-name>
      <event-type-ref>
        <event-type-name>javax.sip.message.Request.ACK</event-type-
name>
        <event-type-vendor>javax.sip</event-type-vendor>
        <event-type-version>1.1</event-type-version>
      </event-type-ref>
      <initial-event-select variable="ActivityContext"/>
    </event>

    <event event-direction="Receive" initial-event="True">
      <event-name>ByeRequest</event-name>
      <event-type-ref>
        <event-type-name>javax.sip.message.Request.BYE</event-type-
name>
        <event-type-vendor>javax.sip</event-type-vendor>
        <event-type-version>1.1</event-type-version>
      </event-type-ref>
      <initial-event-select variable="ActivityContext"/>
    </event>

    <event event-direction="Receive" initial-event="True">
      <event-name>OptionsRequest</event-name>
      <event-type-ref>
```

```
        <event-type-name>javax.sip.message.Request.OPTIONS</event-
type-name>
        <event-type-vendor>javax.sip</event-type-vendor>
        <event-type-version>1.1</event-type-version>
      </event-type-ref>
      <initial-event-select variable="ActivityContext"/>
    </event>

    <!-- TODO invoke SIP registrar as child
    <event event-direction="Receive" initial-event="True">
      <event-name>RegisterRequest</event-name>
      <event-type-ref>
        <event-type-name>javax.sip.message.Request.REGISTER</event-
type-name>
        <event-type-vendor>javax.sip</event-type-vendor>
        <event-type-version>1.1</event-type-version>
      </event-type-ref>
      <initial-event-select variable="ActivityContext"/>
    </event> -->

    <event event-direction="Receive" initial-event="True">
      <event-name>InfoRequest</event-name>
      <event-type-ref>
        <event-type-name>javax.sip.message.Request.INFO</event-type-
name>
        <event-type-vendor>javax.sip</event-type-vendor>
        <event-type-version>1.1</event-type-version>
      </event-type-ref>
      <initial-event-select variable="ActivityContext"/>
    </event>

    <event event-direction="Receive" initial-event="True">
      <event-name>UpdateRequest</event-name>
      <event-type-ref>
        <event-type-name>javax.sip.message.Request.UPDATE</event-type-
name>
        <event-type-vendor>javax.sip</event-type-vendor>
        <event-type-version>1.1</event-type-version>
      </event-type-ref>
      <initial-event-select variable="ActivityContext"/>
    </event>

    <event event-direction="Receive" initial-event="True">
      <event-name>PrackRequest</event-name>
      <event-type-ref>
        <event-type-name>javax.sip.message.Request.PRACK</event-type-
name>
        <event-type-vendor>javax.sip</event-type-vendor>
        <event-type-version>1.1</event-type-version>
      </event-type-ref>
      <initial-event-select variable="ActivityContext"/>
    </event>

    <event event-direction="Receive" initial-event="True">
      <event-name>SubscribeRequest</event-name>
      <event-type-ref>
        <event-type-name>javax.sip.message.Request.SUBSCRIBE</event-
type-name>
```

```
        <event-type-vendor>javax.sip</event-type-vendor>
        <event-type-version>1.1</event-type-version>
      </event-type-ref>
      <initial-event-select variable="ActivityContext"/>
    </event>

    <event event-direction="Receive" initial-event="True">
      <event-name>NotifyRequest</event-name>
      <event-type-ref>
        <event-type-name>javax.sip.message.Request.NOTIFY</event-type-
name>
        <event-type-vendor>javax.sip</event-type-vendor>
        <event-type-version>1.1</event-type-version>
      </event-type-ref>
      <initial-event-select variable="ActivityContext"/>
    </event>

    <event event-direction="Receive" initial-event="True">
      <event-name>ReferRequest</event-name>
      <event-type-ref>
        <event-type-name>javax.sip.message.Request.REFER</event-type-
name>
        <event-type-vendor>javax.sip</event-type-vendor>
        <event-type-version>1.1</event-type-version>
      </event-type-ref>
      <initial-event-select variable="ActivityContext"/>
    </event>

    <event event-direction="Receive" initial-event="True">
      <event-name>MessageRequest</event-name>
      <event-type-ref>
        <event-type-name>javax.sip.message.Request.MESSAGE</event-
type-name>
        <event-type-vendor>javax.sip</event-type-vendor>
        <event-type-version>1.1</event-type-version>
      </event-type-ref>
      <initial-event-select variable="ActivityContext"/>
    </event>

    <event event-direction="Receive" initial-event="True">
      <event-name>ExtensionRequest</event-name>
      <event-type-ref>
        <event-type-
name>javax.sip.message.Request.SIP_EXTENSION</event-type-name>
        <event-type-vendor>javax.sip</event-type-vendor>
        <event-type-version>1.1</event-type-version>
      </event-type-ref>
      <initial-event-select variable="ActivityContext"/>
    </event>

    <!-- SIP Responses -->

    <event event-direction="Receive" initial-event="False">
      <event-name>100Response</event-name>
      <event-type-ref>
        <event-type-
name>javax.sip.message.Response.INFORMATIONAL</event-type-name>
        <event-type-vendor>javax.sip</event-type-vendor>
```

```
        <event-type-version>1.1</event-type-version>
      </event-type-ref>
    </event>

    <event event-direction="Receive" initial-event="True">
      <!-- 2xx responses are initial events so that we can catch late
responses -->
      <event-name>200Response</event-name>
      <event-type-ref>
        <event-type-name>javax.sip.message.Response.SUCCESS</event-
type-name>
        <event-type-vendor>javax.sip</event-type-vendor>
        <event-type-version>1.1</event-type-version>
      </event-type-ref>
      <initial-event-select variable="ActivityContext"/>
      <initial-event-selector-method-name>initialEventSelect</initial-
event-selector-method-name>
    </event>

    <event event-direction="Receive" initial-event="False">
      <event-name>300Response</event-name>
      <event-type-ref>
        <event-type-
name>javax.sip.message.Response.REDIRECTION</event-type-name>
        <event-type-vendor>javax.sip</event-type-vendor>
        <event-type-version>1.1</event-type-version>
      </event-type-ref>
    </event>

    <event event-direction="Receive" initial-event="False">
      <event-name>400Response</event-name>
      <event-type-ref>
        <event-type-
name>javax.sip.message.Response.CLIENT_ERROR</event-type-name>
        <event-type-vendor>javax.sip</event-type-vendor>
        <event-type-version>1.1</event-type-version>
      </event-type-ref>
    </event>

    <event event-direction="Receive" initial-event="False">
      <event-name>500Response</event-name>
      <event-type-ref>
        <event-type-
name>javax.sip.message.Response.SERVER_ERROR</event-type-name>
        <event-type-vendor>javax.sip</event-type-vendor>
        <event-type-version>1.1</event-type-version>
      </event-type-ref>
    </event>

    <event event-direction="Receive" initial-event="False">
      <event-name>600Response</event-name>
      <event-type-ref>
        <event-type-
name>javax.sip.message.Response.GLOBAL_FAILURE</event-type-name>
        <event-type-vendor>javax.sip</event-type-vendor>
        <event-type-version>1.1</event-type-version>
      </event-type-ref>
    </event>
```

```
    <!-- TIMER C -->

    <event event-direction="Receive" initial-event="False">
      <event-name>TimerC</event-name>
      <event-type-ref>
        <event-type-name>javax.slee.facilities.TimerEvent</event-type-
name>
        <event-type-vendor>javax.slee</event-type-vendor>
        <event-type-version>1.0</event-type-version>
      </event-type-ref>
    </event>

    <!-- Transaction Timeout -->

    <event event-direction="Receive" initial-event="False">
      <event-name>TransactionTimeout</event-name>
      <event-type-ref>
        <event-type-name>javax.sip.Timeout.TRANSACTION</event-type-
name>
        <event-type-vendor>javax.sip</event-type-vendor>
        <event-type-version>1.1</event-type-version>
      </event-type-ref>
    </event>

    <env-entry>
      <description>
        Comma-separated list of hostnames, IP addresses that this
proxy
        is known by. The first name in the list will be used as the
server's
        canonical hostname in SIP headers.
      </description>
      <env-entry-name>hostnames</env-entry-name>
      <env-entry-type>java.lang.String</env-entry-type>
      <env-entry-value>@PROXY_HOSTNAMES@</env-entry-value>
    </env-entry>

    <env-entry>
      <description>
        Comma-separated list of domain names that this proxy is
authoritative for.
        The host part of the SIP URI must match one of these domain
names for the
        proxy to consider it local. Subdomains will not automatically
match, they
        must be specified here as well, eg.
"abc.com,a.abc.com,b.abc.com"
      </description>
      <env-entry-name>domains</env-entry-name>
      <env-entry-type>java.lang.String</env-entry-type>
      <env-entry-value>@PROXY_DOMAINS@</env-entry-value>
    </env-entry>

    <env-entry>
      <env-entry-name>sipPort</env-entry-name>
      <env-entry-type>java.lang.Integer</env-entry-type>
      <env-entry-value>@PROXY_SIP_PORT@</env-entry-value>
    </env-entry>
```

```
    <env-entry>
      <env-entry-name>sipsPort</env-entry-name>
      <env-entry-type>java.lang.Integer</env-entry-type>
      <env-entry-value>@PROXY_SIPS_PORT@</env-entry-value>
    </env-entry>

    <env-entry>
      <description>
        Proxy Timer C value, in seconds.
      </description>
      <env-entry-name>timerC</env-entry-name>
      <env-entry-type>java.lang.Integer</env-entry-type>
      <env-entry-value>180</env-entry-value>
    </env-entry>
    <env-entry>
      <description>
        Does the proxy detect loops?
      </description>
      <env-entry-name>loopDetection</env-entry-name>
      <env-entry-type>java.lang.Boolean</env-entry-type>
      <env-entry-value>@PROXY_LOOP_DETECTION@</env-entry-value>
    </env-entry>

    <env-entry>
      <env-entry-name>sipProviderName</env-entry-name>
      <env-entry-type>java.lang.String</env-entry-type>
      <env-entry-value>@SIP_PROVIDER@</env-entry-value>
    </env-entry>

    <env-entry>
      <env-entry-name>sipACIFactoryName</env-entry-name>
      <env-entry-type>java.lang.String</env-entry-type>
      <env-entry-value>@SIP_ACIFACTORY@</env-entry-value>
    </env-entry>

    <resource-adaptor-type-binding>
      <resource-adaptor-type-ref>
        <resource-adaptor-type-name>@SIP_RATYPE_NAME@</resource-
adaptor-type-name>
        <resource-adaptor-type-vendor>@SIP_RATYPE_VENDOR@</resource-
adaptor-type-vendor>
        <resource-adaptor-type-version>@SIP_RATYPE_VERSION@</resource-
adaptor-type-version>
      </resource-adaptor-type-ref>
      <activity-context-interface-factory-name>
        @SIP_ACIFACTORY@
      </activity-context-interface-factory-name>
      <resource-adaptor-entity-binding>
        <resource-adaptor-object-name>
          @SIP_PROVIDER@
        </resource-adaptor-object-name>
        <resource-adaptor-entity-link>
          @SIP_LINKNAME@
        </resource-adaptor-entity-link>
      </resource-adaptor-entity-binding>
    </resource-adaptor-type-binding>
  </sbb>
</sbb-jar>
```

B.8. sbb-jar.xml (registrar)

```
<?xml version="1.0"?>

<!DOCTYPE sbb-jar PUBLIC "-//Sun Microsystems, Inc.//DTD JAIN SLEE SBB
1.0//EN" "http://java.sun.com/dtd/slee-sbb-jar_1_0.dtd">

<sbb-jar>

  <sbb>
    <description>SIP Registrar SBB</description>
    <sbb-name>RegistrarSbb</sbb-name>
    <sbb-vendor>Open Cloud</sbb-vendor>
    <sbb-version>1.5</sbb-version>

    <!-- Location Service SBB - edit this to switch Location Service -
->
    <!-- implementations (AC Naming, JDBC, LDAP etc). -->
    <sbb-ref>
      <sbb-name>@LOCATIONSBB_NAME@</sbb-name>
      <sbb-vendor>@LOCATIONSBB_VENDOR@</sbb-vendor>
      <sbb-version>@LOCATIONSBB_VERSION@</sbb-version>
      <sbb-alias>LOCATIONSBB</sbb-alias>
    </sbb-ref>

    <sbb-classes>

      <sbb-abstract-class>
        <sbb-abstract-class-name>
          com.opencloud.slee.services.sip.registrar.RegistrarSbb
        </sbb-abstract-class-name>

        <get-child-relation-method>
          <sbb-alias-ref>
            LOCATIONSBB
          </sbb-alias-ref>
          <get-child-relation-method-name>
            getLocationServiceChildRelation
          </get-child-relation-method-name>
          <default-priority>0</default-priority>
        </get-child-relation-method>

      </sbb-abstract-class>

    </sbb-classes>

    <event event-direction="Receive" initial-event="True">
      <event-name>RegisterEvent</event-name>
      <event-type-ref>
        <event-type-name>javax.sip.message.Request.REGISTER</event-
type-name>
        <event-type-vendor>javax.sip</event-type-vendor>
        <event-type-version>1.1</event-type-version>
      </event-type-ref>
      <initial-event-select variable="ActivityContext"/>
    </event>
```

```
    <env-entry>
      <env-entry-name>sipProviderName</env-entry-name>
      <env-entry-type>java.lang.String</env-entry-type>
      <env-entry-value>@SIP_PROVIDER@</env-entry-value>
    </env-entry>

    <env-entry>
      <env-entry-name>sipACIFactoryName</env-entry-name>
      <env-entry-type>java.lang.String</env-entry-type>
      <env-entry-value>@SIP_ACIFACTORY@</env-entry-value>
    </env-entry>

    <resource-adaptor-type-binding>
      <resource-adaptor-type-ref>
        <resource-adaptor-type-name>@SIP_RATYPE_NAME@</resource-
adaptor-type-name>
        <resource-adaptor-type-vendor>@SIP_RATYPE_VENDOR@</resource-
adaptor-type-vendor>
        <resource-adaptor-type-version>@SIP_RATYPE_VERSION@</resource-
adaptor-type-version>
      </resource-adaptor-type-ref>
      <activity-context-interface-factory-name>
        @SIP_ACIFACTORY@
      </activity-context-interface-factory-name>
      <resource-adaptor-entity-binding>
        <resource-adaptor-object-name>
          @SIP_PROVIDER@
        </resource-adaptor-object-name>
        <resource-adaptor-entity-link>
          @SIP_LINKNAME@
        </resource-adaptor-entity-link>
      </resource-adaptor-entity-binding>
    </resource-adaptor-type-binding>

  </sbb>

</sbb-jar>
```

B.9. build.xml

```
<?xml version="1.0" encoding="UTF-8"?>

<!-- Ant script for SIP examples management. -->
<project name="Open Cloud Rhino SLEE SDK - SIP Examples"
default="deployexamples">

  <property file="${basedir}/build.properties"/>
  <property file="${basedir}/sip.properties"/>

  <!-- Setup properties for selecting the location service to use,
based
       on the "usejdbclocation" boolean property. -->
  <condition property="jdbc.location.selected">
    <istrue value="${usejdbclocation}"/>
  </condition>
  <condition property="locationsbb" value="JDBCLocationSbb">
    <isset property="jdbc.location.selected"/>
```

```
  </condition>
  <condition property="locationsbb" value="ACLocationSbb">
    <not><isset property="jdbc.location.selected"/></not>
  </condition>

  <!-- Import common Rhino SLEE tasks -->
  <import file="${client.home}/etc/common.xml"/>

  <target name="clean" description="Remove compiled class and jar
files">
    <delete dir="${jars}"/>
    <delete dir="${classes}"/>
  </target>

  <target name="init">
    <mkdir dir="${jars}"/>
    <mkdir dir="${classes}"/>
  </target>

  <path id="sip.classpath">
    <!-- JAIN SIP 1.1 (javax.sip) interfaces -->
    <pathelement location="${lib}/JainSipApi1.1.jar"/>
    <!-- SLEE SIP RA Type (javax.slee.sip) interfaces -->
    <pathelement location="${lib}/${sip.ra.type.jar}"/>
  </path>

  <!-- Run SIP SBB DDs through a filter to replace tokens -->
  <!-- This is so that RA Types and JNDI names etc can be changed
easily -->
  <macrodef name="filter-sip-dd">
    <attribute name="from"/> <!-- Source META-INF dir -->
    <attribute name="to"/>   <!-- Dest META-INF dir -->
    <attribute name="sip.properties"
default="${basedir}/sip.properties"/>
    <sequential>
      <copy toDir="@{to}" overwrite="true">
        <fileset dir="@{from}" includes="**/*"/>
        <filterset>
           <filtersfile file="@{sip.properties}"/>
           <filter token="LOCATIONSBB_NAME" value="${locationsbb}"/>
        </filterset>
      </copy>
    </sequential>
  </macrodef>

  <target name="deployexamples"
    description="Deploy SIP Resource Adaptor and example SIP services"
    depends="deployproxy"/>

  <target name="undeployexamples"
    description="Undeploy SIP Resource Adaptor and example SIP
services"
    depends="undeployproxy, undeployfmfm, undeployregistrar,
undeploylocationservice, undeploysipra"/>

  <target name="compile-sip-examples" depends="init"
description="Compile SIP example services">
    <mkdir dir="${classes}/sip-examples"/>
```

```
    <javac srcdir="${src}" destdir="${classes}/sip-examples"
source="1.4" debug="true"
      includes="com/opencloud/slee/services/sip/**/*.java">
      <classpath>
        <path refid="slee.classpath"/>
        <path refid="sip.classpath"/>
      </classpath>
    </javac>
  </target>

  <target name="build"
    depends="compile-sip-examples, sip-ac-location, sip-jdbc-location,
sip-registrar, sip-proxy, sip-fmfm, sip-b2bua"/>

  <target name="sip-ac-location" depends="compile-sip-examples">
    <deployablejar destfile="${jars}/sip-ac-location-service.jar"

metainfbase="${src}/com/opencloud/slee/services/sip/location/ac/META-
INF">
      <sbbjar destfile="${jars}/ac-location-sbb.jar"
classpath="${classes}/sip-examples">
        <fileset dir="${classes}/sip-examples"
          includes="com/opencloud/slee/services/sip/common/*.class,
          com/opencloud/slee/services/sip/location/*.class,
          com/opencloud/slee/services/sip/location/ac/*.class"/>
      </sbbjar>
    </deployablejar>
    <delete file="${jars}/ac-location-sbb.jar"/>
  </target>

  <target name="sip-jdbc-location" depends="compile-sip-examples">
    <deployablejar destfile="${jars}/sip-jdbc-location-service.jar"

metainfbase="${src}/com/opencloud/slee/services/sip/location/jdbc/META
-INF">
      <sbbjar destfile="${jars}/jdbc-location-sbb.jar" extjarxml="oc-
sbb-jar.xml" classpath="${classes}/sip-examples">
        <fileset dir="${classes}/sip-examples"
          includes="com/opencloud/slee/services/sip/common/*.class,
          com/opencloud/slee/services/sip/location/*.class,
          com/opencloud/slee/services/sip/location/jdbc/*.class"/>
      </sbbjar>
    </deployablejar>
    <delete file="${jars}/jdbc-location-sbb.jar"/>
  </target>

  <target name="sip-registrar" depends="compile-sip-examples">
    <filter-sip-dd
from="${src}/com/opencloud/slee/services/sip/registrar/META-INF"
to="${classes}/sip-examples/registrar-META-INF"/>
    <deployablejar destfile="${jars}/sip-registrar-service.jar"
      metainfbase="${classes}/sip-examples/registrar-META-INF">
      <sbbjar destfile="${jars}/registrar-sbb.jar"
classpath="${classes}/sip-examples">
        <fileset dir="${classes}/sip-examples"
          includes="com/opencloud/slee/services/sip/common/*.class,
          com/opencloud/slee/services/sip/registrar/*.class"/>
      </sbbjar>
```

```
    </deployablejar>
    <delete file="${jars}/registrar-sbb.jar"/>
  </target>

  <target name="sip-proxy" depends="compile-sip-examples">
    <filter-sip-dd
from="${src}/com/opencloud/slee/services/sip/proxy/META-INF"
to="${classes}/sip-examples/proxy-META-INF"/>
    <deployablejar destfile="${jars}/sip-proxy-service.jar"
      metainfbase="${classes}/sip-examples/proxy-META-INF">
      <sbbjar destfile="${jars}/proxy-sbb.jar"
classpath="${classes}/sip-examples" extjarxml="oc-sbb-jar.xml">
        <fileset dir="${classes}/sip-examples"
          includes="com/opencloud/slee/services/sip/common/*.class,
          com/opencloud/slee/services/sip/proxy/*.class"/>
      </sbbjar>
    </deployablejar>
    <delete file="${jars}/proxy-sbb.jar"/>
  </target>

  <target name="sip-fmfm" depends="sip-proxy">
    <filter-sip-dd
from="${src}/com/opencloud/slee/services/sip/fmfm/META-INF"
to="${classes}/sip-examples/fmfm-META-INF"/>
    <deployablejar destfile="${jars}/sip-fmfm-service.jar"
      metainfbase="${classes}/sip-examples/fmfm-META-INF">
      <profilespecjar destfile="${jars}/fmfm-profile.jar"
classpath="${classes}/sip-examples"/>
      <sbbjar destfile="${jars}/fmfm-sbb.jar"
classpath="${classes}/sip-examples" extjarxml="oc-sbb-jar.xml">
        <fileset dir="${classes}/sip-examples"
          includes="com/opencloud/slee/services/sip/common/*.class,
          com/opencloud/slee/services/sip/fmfm/**/*.class,
          com/opencloud/slee/services/sip/proxy/*.class"

excludes="com/opencloud/slee/services/sip/fmfm/profile/**/*"/>
      </sbbjar>
    </deployablejar>
    <delete file="${jars}/fmfm-profile.jar"/>
    <delete file="${jars}/fmfm-sbb.jar"/>
  </target>

  <target name="sip-b2bua" depends="compile-sip-examples">
    <filter-sip-dd
from="${src}/com/opencloud/slee/services/sip/b2bua/META-INF"
to="${classes}/sip-examples/b2bua-META-INF"/>
    <deployablejar destfile="${jars}/sip-b2bua-service.jar"
      metainfbase="${classes}/sip-examples/b2bua-META-INF">
      <sbbjar destfile="${jars}/b2bua-sbb.jar"
classpath="${classes}/sip-examples" extjarxml="oc-sbb-jar.xml">
        <fileset dir="${classes}/sip-examples"
          includes="com/opencloud/slee/services/sip/common/*.class,
          com/opencloud/slee/services/sip/b2bua/**/*.class,
          com/opencloud/slee/services/sip/dialog/**/*.class,
          com/opencloud/slee/services/sip/proxy/*.class"/>
      </sbbjar>
    </deployablejar>
    <delete file="${jars}/b2bua-sbb.jar"/>
```

```
  </target>

  <!-- SIP Resource Adaptor -->

  <target name="deploysipra" depends="login" description="Deploy SIP
resource adaptor">
    <slee-management>
      <install srcfile="${lib}/${sip.ra.jar}"
url="file:${lib}/${sip.ra.jar}"/>
      <createraentity
        resourceadaptorid="${sip.ra.name}"
        entityname="${sip.ra.entity}"
        properties="${sip.ra.properties}"/>
      <bindralinkname entityname="${sip.ra.entity}"
linkname="${SIP_LINKNAME}"/>
      <activateraentity entityname="${sip.ra.entity}"/>
    </slee-management>
  </target>

  <target name="undeploysipra" depends="login" description="Undeploy
SIP resource adaptor">
    <slee-management>
      <deactivateraentity entityname="${sip.ra.entity}"/>
      <unbindralinkname linkname="${SIP_LINKNAME}"/>
      <waittilraentityisinactive entityname="${sip.ra.entity}"/>
      <removeraentity entityname="${sip.ra.entity}"/>
      <uninstall url="file:${lib}/${sip.ra.jar}"/>
    </slee-management>
  </target>

  <!-- SIP Location Service -->

  <!-- Will deploy either AC Naming or JDBC Location Service
implementation,
       depending on value of "usejdbclocation" boolean property. -->
  <target name="deploylocationservice" depends="deploy-jdbc-
locationservice, deploy-ac-locationservice" description="Deploy
Location Service"/>
  <target name="undeploylocationservice" depends="undeploy-jdbc-
locationservice, undeploy-ac-locationservice" description="Undeploy
Location Service"/>

  <target name="deploy-jdbc-locationservice" depends="build, login"
if="jdbc.location.selected"
    description="Deploy JDBC Location Service">
    <slee-management>
      <install srcfile="${jars}/${location.jdbc.jar}"
url="file:${jars}/${location.jdbc.jar}"/>
      <activateservice serviceid="${location.jdbc.service}"/>
      <settracelevel componentid="${location.jdbc.sbb}" type="sbb"
level="${sbb.tracelevel}"/>
    </slee-management>
  </target>

  <target name="deploy-ac-locationservice" depends="build, login"
unless="jdbc.location.selected"
    description="Undeploy AC Naming Location Service">
    <slee-management>
```

```
      <install srcfile="${jars}/${location.ac.jar}"
url="file:${jars}/${location.ac.jar}"/>
      <activateservice serviceid="${location.ac.service}"/>
      <settracelevel componentid="${location.ac.sbb}" type="sbb"
level="${sbb.tracelevel}"/>
    </slee-management>
  </target>

  <target name="undeploy-jdbc-locationservice" depends="login"
if="jdbc.location.selected"
    description="Deploy JDBC Location Service">
    <slee-management>
      <deactivateservice serviceid="${location.jdbc.service}"/>
      <waittilserviceisinactive serviceid="${location.jdbc.service}"/>
      <uninstall url="file:${jars}/${location.jdbc.jar}"/>
    </slee-management>
  </target>

  <target name="undeploy-ac-locationservice" depends="login"
unless="jdbc.location.selected"
    description="Undeploy AC Naming Location Service">
    <slee-management>
      <deactivateservice serviceid="${location.ac.service}"/>
      <waittilserviceisinactive serviceid="${location.ac.service}"/>
      <uninstall url="file:${jars}/${location.ac.jar}"/>
    </slee-management>
  </target>

  <!-- SIP Registrar Service -->

  <target name="deployregistrar" depends="build, deploysipra,
deploylocationservice"
    description="Deploy SIP Registrar service">
    <slee-management>
      <install srcfile="${jars}/${registrar.service.jar}"
url="file:${jars}/${registrar.service.jar}"/>
      <activateservice serviceid="${registrar.service}"/>
      <settracelevel componentid="${registrar.sbb}" type="sbb"
level="${sbb.tracelevel}"/>
    </slee-management>
  </target>

  <target name="undeployregistrar" depends="login"
    description="Undeploy SIP Registrar service">
    <slee-management>
      <deactivateservice serviceid="${registrar.service}"/>
      <waittilserviceisinactive serviceid="${registrar.service}"/>
      <uninstall url="file:${jars}/${registrar.service.jar}"/>
    </slee-management>
  </target>

  <!-- SIP Proxy Service -->

  <target name="deployproxy" depends="build, deployregistrar,
undeployfmfm"
    description="Deploy SIP Proxy service">
    <slee-management>
```

```
      <install srcfile="${jars}/${proxy.service.jar}"
url="file:${jars}/${proxy.service.jar}"/>
      <activateservice serviceid="${proxy.service}"/>
      <settracelevel componentid="${proxy.sbb}" type="sbb"
level="${sbb.tracelevel}"/>
    </slee-management>
  </target>

  <target name="undeployproxy" depends="login"
    description="Undeploy SIP Proxy service">
    <slee-management>
      <deactivateservice serviceid="${proxy.service}"/>
      <waittilserviceisinactive serviceid="${proxy.service}"/>
      <uninstall url="file:${jars}/${proxy.service.jar}"/>
    </slee-management>
  </target>

  <!-- SIP Find Me/Follow Me Service -->

  <target name="deployfmfm" depends="build, deployregistrar,
undeployproxy"
    description="Deploy Find Me/Follow Me service">
    <slee-management>
      <install srcfile="${jars}/${fmfm.service.jar}"
url="file:${jars}/${fmfm.service.jar}"/>
      <createprofiletable profilespec="FMFMSubscriberProfile 1.5, Open
Cloud" tablename="${FMFM_PROFILE_TABLE_NAME}"/>
      <activateservice serviceid="${fmfm.service}"/>
      <settracelevel componentid="${fmfm.sbb}" type="sbb"
level="${sbb.tracelevel}"/>
      <settracelevel componentid="${fmfm.proxy.sbb}" type="sbb"
level="${sbb.tracelevel}"/>
    </slee-management>
  </target>

  <target name="undeployfmfm" depends="login"
    description="Undeploy Find Me/Follow Me service">
    <slee-management>
      <removeprofiletable tablename="${FMFM_PROFILE_TABLE_NAME}"/>
      <deactivateservice serviceid="${fmfm.service}"/>
      <waittilserviceisinactive serviceid="${fmfm.service}"/>
      <uninstall url="file:${jars}/${fmfm.service.jar}"/>
    </slee-management>
  </target>

  <!-- SIP B2BUA Service -->

  <target name="deployb2bua" depends="build, deployregistrar"
    description="Deploy SIP B2BUA service">
    <slee-management>
      <install srcfile="${jars}/${b2bua.service.jar}"
url="file:${jars}/${b2bua.service.jar}"/>
      <activateservice serviceid="${b2bua.service}"/>
      <settracelevel componentid="${b2bua.sbb}" type="sbb"
level="${sbb.tracelevel}"/>
      <settracelevel componentid="${uac.sbb}" type="sbb"
level="${sbb.tracelevel}"/>
```

```
      <settracelevel componentid="${uas.sbb}" type="sbb"
level="${sbb.tracelevel}"/>
    </slee-management>
  </target>

  <target name="undeployb2bua" depends="login"
    description="Undeploy SIP B2BUA service">
    <slee-management>
      <deactivateservice serviceid="${b2bua.service}"/>
      <waittilserviceisinactive serviceid="${b2bua.service}"/>
      <uninstall url="file:${jars}/${b2bua.service.jar}"/>
    </slee-management>
  </target>

</project>
```

B.10. oc-sbb-jar.xml

```
<?xml version="1.0"?>

<!DOCTYPE oc-sbb-jar PUBLIC "-//Open Cloud Ltd.//DTD JAIN SLEE SBB
Extension 1.0//EN" "http://www.opencloud.com/slee/oc-sbb-jar_1_0.dtd">

<oc-sbb-jar>
  <sbb id="fmfm">
    <resource-ref>
      <res-ref-name>@FMFM_DATASOURCE_NAME@</res-ref-name>
      <res-type>javax.sql.DataSource</res-type>
      <res-auth>Container</res-auth>
      <res-sharing-scope>Shareable</res-sharing-scope>
      <res-jndi-name>jdbc/ExternalDataSource</res-jndi-name>
    </resource-ref>
  </sbb>
  <security-permission>
    <!-- The proxy may need to do a hostname lookup when checking
headers -->
    <security-permission-spec>
          grant {
            permission java.net.SocketPermission "*:1024-", "resolve";
          };
    </security-permission-spec>
  </security-permission>
</oc-sbb-jar>
```

B.11. FindMeFollowMeSbb.java

```
package com.opencloud.slee.services.sip.fmfm;

import com.opencloud.slee.services.sip.common.OCSipSbb;
import com.opencloud.slee.services.sip.proxy.Proxy;
import com.opencloud.slee.services.sip.proxy.ProxyResponseListener;
import
com.opencloud.slee.services.sip.fmfm.profile.FMFMSubscriberProfileCMP;
import com.opencloud.slee.services.sip.fmfm.jdbc.JDBCSubscriberLookup;

import javax.sip.RequestEvent;
```

```
import javax.sip.ServerTransaction;
import javax.sip.SipException;
import javax.sip.ClientTransaction;
import javax.sip.address.SipURI;
import javax.sip.address.URI;
import javax.sip.header.*;
import javax.sip.message.Message;
import javax.sip.message.Request;
import javax.sip.message.Response;
import javax.slee.*;
import javax.slee.profile.*;
import javax.slee.facilities.Level;
import javax.slee.facilities.NameAlreadyBoundException;
import javax.slee.nullactivity.NullActivity;
import javax.naming.InitialContext;
import javax.naming.Context;
import javax.naming.NamingException;
import javax.sql.DataSource;
import java.text.ParseException;
import java.util.ArrayList;
import java.util.List;

/**
 * Find Me/Follow Me service.
 *
 * Each SIP transaction is handled by a separate FMFM SBB Entity,
 * call state is in a shared ACI, named by call ID. Attach to the
 * ACI during each SIP transaction.
 */
public abstract class FindMeFollowMeSbb extends OCSipSbb {

    public void setSbbContext(SbbContext context) {
        super.setSbbContext(context);
        try {
            Context myEnv = (Context) new
InitialContext().lookup("java:comp/env");
            boolean useJDBC =
((Boolean)myEnv.lookup("fmfmUseJDBC")).booleanValue();

            if (useJDBC) {
            String dsName = (String)
myEnv.lookup("fmfmDataSourceName");
                DataSource ds = (DataSource) myEnv.lookup(dsName);
                lookup = new JDBCSubscriberLookup(ds, getSbbTracer());
            }
            else { // profiles
            String profileTableName = (String)
myEnv.lookup("fmfmProfileTableName");
                lookup = new FMFMProfileLookup(profileTableName);
            }
        } catch (NamingException ne) {
            severe("unable to set SBB context", ne);
        }
    }

    protected String getTraceMessageType() { return "FMFM"; }
```

```
    public void onInviteRequest(RequestEvent event,
ActivityContextInterface aci) {
        // Is this INVITE part of an existing call (re-INVITE) or
initiating
        // a new call?
        Request request = event.getRequest();
        if (isTraceable(Level.FINEST))
            finest("received INVITE request:\n" + request);
        CallStateACI state = lookupCallState(request);
        if (state == null) {
            // New INVITE request - go to connecting state and await
response
            finer("got INVITE for new call");
            setConnectingState(CONNECTING_INITIAL);
        }
        else {
            finer("got re-INVITE on existing call");
        }

createProxy().proxyRequest((ProxyResponseListener)getSbbLocalObject(),
request);
    }

    public void onAckRequest(RequestEvent event,
ActivityContextInterface aci) {
        aci.detach(getSbbLocalObject()); // detach from the ACK's
server txn
        if (isTraceable(Level.FINEST))
            finest("received ACK request:\n" + event.getRequest());

createProxy().proxyRequestStateless((Request)event.getRequest().clone(
));
    }

    public void onByeRequest(RequestEvent event,
ActivityContextInterface aci) {
        Request request = event.getRequest();
        if (isTraceable(Level.FINEST))
            finest("received BYE request:\n" + request);
        CallStateACI state = lookupCallState(request);
        if (state == null) {
            // BYE request for unknown call. Send error response.
            try {
                finer("got BYE for unknown call");
                Response response =
getSipMessageFactory().createResponse(Response.CALL_OR_TRANSACTION_DOE
S_NOT_EXIST, request);
                commitResponse(response);
            } catch (ParseException e) {
                warn("unable to send 481 response", e);
            }
        }
        else {
            finer("got BYE, terminate call");
            disconnect(state);

createProxy().proxyRequest((ProxyResponseListener)getSbbLocalObject(),
event.getRequest());
```

```
        }
    }

    public void onCancelRequest(RequestEvent event,
ActivityContextInterface aci) {
        aci.detach(getSbbLocalObject()); // detach from CANCEL server
txn

        if (isTraceable(Level.FINEST))
            finest("received CANCEL request:\n" + event.getRequest());

        try {
            // If we're in the middle of connecting a call, cancel any
in-progress txns.
            if (getConnectingState() == CONNECTING_INITIAL ||
                    getConnectingState() == CONNECTING_FMFM) {
                finer("got CANCEL for call");
                Response ok =
getSipMessageFactory().createResponse(Response.OK,
event.getRequest());
                if (isTraceable(Level.FINEST))
                    finest("sending response:\n"+ok);
                event.getServerTransaction().sendResponse(ok);
                getProxy().cancel(); // This will terminate any in-
progress txns
            }
            else {
                finer("got CANCEL for unknown call");
                Response err =
getSipMessageFactory().createResponse(Response.CALL_OR_TRANSACTION_DOE
S_NOT_EXIST, event.getRequest());
                if (isTraceable(Level.FINEST))
                    finest("sending response:\n"+err);
                event.getServerTransaction().sendResponse(err);
            }
        }
        catch (Exception e) {
            warn("unable to send CANCEL response", e);
        }
    }

    // Called by Proxy child when a response is ready for forwarding
    public void forwardResponse(Response response) {
        switch (response.getStatusCode()/100) {
            case 1:
                onProvisionalResponse(response);
                break;
            case 2:
                onOKResponse(response);
                break;
            default:
                onErrorResponse(response);
        }
    }

    public void receivedResponse(ClientTransaction ct, Response
response) {
        // nothing to do
```

```
    }

    private void onProvisionalResponse(Response response) {
        // nothing to do
        finer("forwarding provisional response");
        commitResponse(response);
    }

    private void onOKResponse(Response response) {
        if (getConnectingState() == CONNECTING_INITIAL ||
                getConnectingState() == CONNECTING_FMFM) {
            // got an OK response, so we are now connected
            finer("got OK, call connected, create call state");

            CallStateACI callState = createCallState(response);

            if (isTraceable(Level.FINE)) {
                fine("connected: orig=" +
callState.getOriginatingAddress() +
                        ", term=" + callState.getTerminatingAddress()
+
                        ", redirected=" +
callState.getRedirectedAddress()) ;
            }
        }
        finer("forwarding OK response");
        commitResponse(response);
    }

    private void onErrorResponse(Response response) {
        if (getConnectingState() == CONNECTING_INITIAL) {
            // Got an error response during first connect, try backup
addresses
            if (isTraceable(Level.FINER))
                finer("got " + response.getStatusCode() + " response,
trying backup addresses");
            String term =
((SipURI)((ToHeader)response.getHeader(ToHeader.NAME)).getAddress().ge
tURI()).toString();

            List targets = getBackupAddresses(term);
            if (targets == null || targets.isEmpty()) {
                if (isTraceable(Level.FINER))
                    finer("no backup addresses for " + term +
                            ", terminating call");
                commitResponse(response);
            }
            else {
                setConnectingState(CONNECTING_FMFM);
                // create new proxy and forward
                Proxy p = createProxy();
                p.setForking(true);

p.proxyRequest((ProxyResponseListener)getSbbLocalObject(),
                        getServerTransaction().getRequest(),
                        (URI[])targets.toArray(new
URI[targets.size()]));
            }
```

```
        }
        else if (getConnectingState() == CONNECTING_FMFM) {
            // got an error response to FMFM, clean up the call
            if (isTraceable(Level.FINER))
                finer("got " + response.getStatusCode() +
                        " response, backup addresses unsuccessful,
terminating call");
            commitResponse(response);
        }
        else {
            if (isTraceable(Level.FINER))
                finer("got " + response.getStatusCode() +
                        " response, forwarding normally");
            commitResponse(response);
        }
    }

    /**
     * Get the call state that this message is associated with.
     * @param message
     * @return the ACI, or null if the message is not associated with
any call yet
     */
    private CallStateACI lookupCallState(Message message)  {
        String acName =
((CallIdHeader)message.getHeader(CallIdHeader.NAME)).getCallId();
        ActivityContextInterface aci =
getACNamingFacility().lookup(acName);
        if (aci == null) return null;
        if (isTraceable(Level.FINER))
            finer("lookupCallState: found call state ACI: " + acName);
        aci.attach(getSbbLocalObject());
        return asSbbActivityContextInterface(aci);
    }

    /**
     * Called when a call is connected. Create the call state ACI,
bind it into AC Naming,
     * and set initialise call data
     */
    private CallStateACI createCallState(Message message)  {
        assert getConnectingState() == CONNECTING_INITIAL ||
                getConnectingState() == CONNECTING_FMFM : "bug: may
only create call state when connecting";

        try {
            String acName =
((CallIdHeader)message.getHeader(CallIdHeader.NAME)).getCallId();
            String orig =
((SipURI)((FromHeader)message.getHeader(FromHeader.NAME)).getAddress()
.getURI()).toString();
            String term =
((SipURI)((ToHeader)message.getHeader(ToHeader.NAME)).getAddress().get
URI()).toString();
            String redirected =
((SipURI)((ContactHeader)message.getHeader(ContactHeader.NAME)).getAdd
ress().getURI()).toString();
            if (redirected == null || redirected.equals(term)) {
```

```
                // Actual address is same as original - don't set
redirectingAddress
                redirected = null;
            }

            NullActivity n =
getNullActivityFactory().createNullActivity();
            ActivityContextInterface aci =
getNullACIFactory().getActivityContextInterface(n);

            CallStateACI callState =
asSbbActivityContextInterface(aci);
            if (isTraceable(Level.FINER))
                finer("creating call state ACI: " + acName);

            getACNamingFacility().bind(callState, acName);

            callState.setCallID(acName);
            callState.setOriginatingAddress(orig);
            callState.setTerminatingAddress(term);
            if (redirected != null)
callState.setRedirectedAddress(redirected);
            callState.setCallStartTime(System.currentTimeMillis());

            return callState;
        } catch (UnrecognizedActivityException e) {
            throw new RuntimeException(e); // should never see this
for null ACs
        } catch (NameAlreadyBoundException n) {
            throw new RuntimeException("bug: name must not be already
bound", n);
        }
    }

    private void disconnect(CallStateACI callState) {
        // end the underlying activity, this will remove name binding
        if (isTraceable(Level.FINE)) {
            fine("disconnected: orig=" +
callState.getOriginatingAddress() +
                    ", term=" + callState.getTerminatingAddress() +
                    ", redirected=" + callState.getRedirectedAddress()
+
                    ", duration=" + (System.currentTimeMillis() -
callState.getCallStartTime()) + "ms");
            if (isTraceable(Level.FINER))
                finer("disconnect: removing call state ACI: " +
callState.getCallID());
        }

        callState.detach(getSbbLocalObject());
        ((NullActivity)callState.getActivity()).endActivity();
    }

    public ServerTransaction getServerTransaction() {
        ActivityContextInterface aci = getServerTransactionACI();
        return aci == null ? null : (ServerTransaction)
aci.getActivity();
    }
```

```
    public ActivityContextInterface getServerTransactionACI() {
        ActivityContextInterface[] acis =
getSbbContext().getActivities();
        for (int i = 0; i < acis.length; i++) {
            if (acis[i].getActivity() instanceof ServerTransaction)
                return acis[i];
        }
        return null;
    }

    private void commitResponse(Response response) {
        ActivityContextInterface aci = getServerTransactionACI();
        ServerTransaction st = (ServerTransaction) aci.getActivity();
        try {
            if (response.getStatusCode() >= 200) {
                finer("final response, detaching all activities");
                detachAllActivities();
            }
            if (isTraceable(Level.FINEST))
                finest("forwarding response: \n" + response);
            st.sendResponse(response);
        } catch (SipException e) {
            warn("unable to send response", e);
        }
    }
    private Proxy createProxy() {
        try {
            ChildRelation cr = getProxyChildRelation();
            Proxy p = (Proxy) cr.create();
            return p;
        } catch (CreateException e) {
            throw new RuntimeException(e);
        }
    }

    private Proxy getProxy() {
        // get the proxy we are attached to
        ChildRelation cr = getProxyChildRelation();
        return (Proxy) cr.iterator().next();
    }

    // returns a list of SIP URIs
    private List getBackupAddresses(String sipAddress) {
        FMFMSubscriber profile = lookup.lookup(new
Address(AddressPlan.SIP, sipAddress));
        if (profile == null) return null;

        Address[] addresses = profile.getBackupAddresses();
        ArrayList uris = new ArrayList(addresses.length);
        for (int i = 0; i < addresses.length; i++) {
            String address = addresses[i].getAddressString();
            try {
                SipURI uri = (SipURI)
getSipAddressFactory().createURI(address);
                uris.add(uri);
            } catch (ParseException e) {
                warn("unable to create uri for " + address);
            }
```

```
        }
        if (isTraceable(Level.FINER))
            finer("backup addresses for: " + sipAddress + ": " +
uris);
        return uris;
    }

    private class FMFMProfileLookup implements FMFMSubscriberLookup {
        FMFMProfileLookup(String profileTableName) {
            this.profileTableName = profileTableName;
        }

        public FMFMSubscriber lookup(Address address) {
            try {
                ProfileID id =
getProfileFacility().getProfileByIndexedAttribute(profileTableName,
"addresses", address);
                if (id == null) return null;
                final FMFMSubscriberProfileCMP profile =
getFMFMSubscriberProfile(id);
                return new FMFMSubscriber() {
                    public Address[] getAddresses() { return
profile.getAddresses(); }
                    public Address[] getBackupAddresses() { return
profile.getBackupAddresses(); }
                };
            } catch (Exception e) {
                if (isTraceable(Level.FINEST))
                    finest("error looking up profile for " + address,
e);
                return null;
            }
        }

        private final String profileTableName;
    }
    public abstract ChildRelation getProxyChildRelation();
    public abstract CallStateACI
asSbbActivityContextInterface(ActivityContextInterface aci);
    public abstract FMFMSubscriberProfileCMP
getFMFMSubscriberProfile(ProfileID id)
            throws UnrecognizedProfileTableNameException,
UnrecognizedProfileNameException;

    // Set if the SBB entity is in the process of connecting a call,
ie. it has
    // received an INVITE, and is waiting for a response (either to
the
    // initial proxied INVITE request, or a backup "follow-me"
request.
    public abstract int getConnectingState();
    public abstract void setConnectingState(int state);

    private FMFMSubscriberLookup lookup;

    private static final int CONNECTING_INITIAL = 1;
    private static final int CONNECTING_FMFM = 2;
}
```

B.12. FMFMSubscriberProfileManagement.java

```
package com.opencloud.slee.services.sip.fmfm.profile;

import javax.slee.profile.ProfileManagement;
import javax.slee.profile.ProfileVerificationException;
import javax.slee.Address;
import javax.slee.AddressPlan;

public abstract class FMFMSubscriberProfileManagement implements
FMFMSubscriberProfileCMP, ProfileManagement {
    public void profileInitialize() {
        setAddresses(new Address[0]);
        setBackupAddresses(new Address[0]);
    }

    public void profileLoad() {}

    public void profileStore() {}

    /**
     * Ensure all addresses are SIP addresses
     */
    public void profileVerify() throws ProfileVerificationException {
        Address[] addresses = getAddresses();
        for (int i = 0; i < addresses.length; i++)
verifyFMFMAddress(addresses[i]);
        Address[] backupAddresses = getBackupAddresses();
        for (int i = 0; i < backupAddresses.length; i++)
verifyFMFMAddress(backupAddresses[i]);
    }

    private void verifyFMFMAddress(Address address) throws
ProfileVerificationException {
        // Check address plan
        if (address.getAddressPlan() != AddressPlan.SIP)
            throw new ProfileVerificationException("Address \"" +
address + "\" is not a SIP address");
        // Check URI scheme - must be sip: or sips:
        String uri = address.getAddressString().toLowerCase();
        if (!(uri.startsWith("sip:") || uri.startsWith("sips:")))
            throw new ProfileVerificationException("Address \"" +
address + "\" is not a SIP address");
    }
}
```

B.13. postgres_findme_db.sql

```
create table findme (
  sipaddress varchar(80) not null,
  backup_sipaddress varchar(80) not null,
  seq_no integer,
  primary key (sipaddress, seq_no)
);
COMMENT ON TABLE findme IS 'FMFM Subscribers';
```

B.14. JDBCSubscriberLookup.java

```
package com.opencloud.slee.services.sip.fmfm.jdbc;

import com.opencloud.slee.services.sip.fmfm.FMFMSubscriber;
import com.opencloud.slee.services.sip.fmfm.FMFMSubscriberLookup;
import com.opencloud.slee.services.sip.common.Tracer;

import javax.sql.DataSource;
import javax.slee.Address;
import javax.slee.AddressPlan;
import javax.slee.facilities.Level;
import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.util.ArrayList;

/**
 * Utility class for looking up FMFM subscriber data.
 * <p>
 * DB schema:
 * <pre>
 * create table findme (                    -- for SIP FMFM service
 *  sipaddress varchar(80) not null,        -- subscriber's primary SIP
address
 *  backup_sipaddress varchar(80) not null,-- backup SIP address
 *  seq_no integer,                         -- sequence number for
ordering bacup addresses
 *  primary key (sipaddress, seq_no)
 * );
 * </pre>
 */
public class JDBCSubscriberLookup implements FMFMSubscriberLookup {
    public JDBCSubscriberLookup(DataSource ds, Tracer tracer) {
        this.ds = ds;
        this.tracer = tracer;
    }

    public FMFMSubscriber lookup(Address address) {
        try {
            return new JDBCFMFMSubscriber(address, ds);
        } catch (Exception e) {
            if (tracer.isTraceable(Level.FINEST))
                tracer.finest("JDBC query failed", e);
            return null;
        }
    }

    private static class JDBCFMFMSubscriber implements FMFMSubscriber
{
        JDBCFMFMSubscriber(Address address, DataSource ds) throws
SQLException {
            Connection conn = ds.getConnection();
            PreparedStatement query =
conn.prepareStatement(queryGetUser);
            query.setString(1, address.getAddressString());
```

```
            ResultSet result = query.executeQuery();
            ArrayList backups = new ArrayList(3); // not expecting
more than 3 entries
            while (result.next()) {
                String backupAddress =
result.getString("backup_sipaddress");
                if (backupAddress != null) backups.add(new
Address(AddressPlan.SIP, backupAddress.trim()));
            }
            result.close();
            query.close();

            this.publicAddress = address;
            this.backupAddresses = (Address[])backups.toArray(new
Address[backups.size()]);
        }

        public Address[] getAddresses() {
            // only one public address in DB, but SLEE address
profiles may have more
            return new Address[] { publicAddress };
        }

        public Address[] getBackupAddresses() {
            return backupAddresses;
        }
        private final Address publicAddress;
        private final Address[] backupAddresses;
    }

    private final DataSource ds;
    private final Tracer tracer;

    private static final String queryGetUser = "select * from findme
where sipaddress = ? order by seq_no";
}
```

Anexo C
Código Mobicents

C. Código Mobicents

En este anexo se incluirá el código correspondiente a la plataforma Mobicents que pueda resultar de utilidad para el seguimiento de la memoria.

C.1. build.xml

```
<!--
   The Mobicents build script

   Authors: Emil Ivov, Ranga, Ivelin -->
<project basedir="." default="make" name="mobicents.jpx">
    <!-- Use the -Doption=optval on the ant command line for these. --
>
    <property name="node" value="all" description="Controls the jboss
conrfiguration. Values are typically all or node1" />
    <property name="cachemode" value="local" description="Controls the
cache mode. Values are typically local or cluster" />
    <property name="debug" value="normal" description="Controls the
debug level. Values are typically debug or normal" />
    <property name="deploydir" value="deploy" />
    <!-- Node is the configuration. cachemode is either local or
cluster -->
    <!-- debug controls the logging. set it at either debug or normal
-->
    <!-- deploydir controls where you deploy it. Either ha-singleton
or deploy -->
    <property name="project.home" value="."/>
    <dirname  property="project.home.dir"  file="${project.home}" />
    <property name="dest" value="classes"/>
    <property name="build" value = "./build" />
    <property name="sipra" value = "./ra/sipra" />
    <property name="sipra-stage" value="${sipra}/stage"/>
    <property name="sipra-unit" value="sip-ra.jar"/>

    <property name="asteriskra" value = "./ra/asteriskra" />
    <property name="junit.reports" value="junit-reports"/>
    <property name="libs.home" value="lib"/>
    <property name="slee.tck.home" value="jain-slee-1.0-tck"/>
    <property name="classpath.prefix" value="org.mobicents" />
    <property name="dirpath.prefix" value="org/mobicents" />
    <property name="sip.services.home" value="thirdparty/sip-services"
/>

    <!-- Set up your system JBOSS_HOME variable and not as follows -->
    <!-- property name="jboss.home" value="/devtools/jboss-3.2.6" / --
>

    <property environment="system"/>

    <property name="jboss.home" value="${system.JBOSS_HOME}" />

    <property name="deployables" value="./deployable-unit" />
```

```
    <property name="mobicents.sar"
value="${jboss.home}/server/${node}/${deploydir}/mobicents.sar" />
    <property name="jboss.home.deploy"
value="${jboss.home}/server/${node}/deploy" />

    <property name="jboss.mobicents.deploy.dir"
value="${jboss.home}/server/${node}/deploy-mobicents" />

    <property name="slee-tck-tests-pattern"
value="${tests}"></property>

    <property name="src" value="src"/>
    <property file="${node}.port.properties" />
    <path id="project.class.path">
        <pathelement location="${dest}"/>
        <pathelement location="${libs.home}/commons-pool-1.2.jar"/>
        <pathelement location="${libs.home}/junit.jar"/>
        <pathelement location="${libs.home}/log4j.jar"/>
        <pathelement location="${libs.home}/commons-collections-
3.1.jar"/>
        <pathelement location="${libs.home}/commons-collections-
testframework-3.1.jar"/>
        <pathelement location="${libs.home}/commons-logging.jar"/>
        <pathelement location="${libs.home}/concurrent.jar"/>
        <pathelement location="${libs.home}/javassist.jar"/>
        <!--JBoss Cache -->
        <pathelement location="${jboss.home}/server/all/lib/jboss-
aop.jar"/>
        <pathelement location="${jboss.home}/server/all/lib/jboss-
cache.jar"/>
        <pathelement location="${jboss.home}/lib/jboss-common.jar"/>
        <pathelement location="${jboss.home}/lib/jboss-jmx.jar"/>
        <pathelement location="${jboss.home}/lib/jboss-system.jar"/>
        <pathelement
location="${jboss.home}/server/all/lib/jboss.jar"/>
        <pathelement location="${project.home}/lib/jgroups-all.jar"/>
        <!--Java Transacion Api-->
        <pathelement location="${libs.home}/jta.jar"/>
        <!--JMX REMOTE CLIENT-->
        <pathelement location="${jboss.home}/server/all/lib/jmx-
adaptor-plugin.jar"/>
        <pathelement
location="${jboss.home}/server/all/lib/jnpserver.jar"/>
        <pathelement location="${jboss.home}/server/all/lib/jboss-
transaction.jar"/>
        <!--TCK Tests -->
        <pathelement location="${slee.tck.home}/jars/sleetck.jar"/>
        <pathelement location="${slee.tck.home}/jars/sleetck-ra-
common.jar"/>

        <!-- JBoss HA for failover -->
        <pathelement
location="${jboss.home}/server/all/lib/jbossha.jar"/>

        <!-- Slee API jar file -->
        <pathelement location="${libs.home}/slee.jar"/>

        <!-- SIP API jar file -->
```

```
        <pathelement location="${sipra}/lib/JainSipApi1.2.jar"/>
        <pathelement location="${sipra}/lib/nist-sip-1.2.jar"/>

        <!-- For the SIP examples to compile -->
        <pathelement location="${sipra}/stage/sip-resource-
adaptor.jar"/>
        <pathelement location="${sipra}/stage/sip-resource-adaptor-
type.jar"/>

        <!-- J2EE jar - only required to compile the ejb-ref test case
-->
        <pathelement location="${jboss.home}/server/all/lib/jboss-
j2ee.jar"/>

        <!-- ANT junit jar file -->
        <pathelement location="${libs.home}/ant-junit.jar"/>
        <pathelement location="${libs.home}/sipunit.jar"/>

        <!-- Slee 1.1 Resource Adaptor Interfaces -->
        <pathelement location="${libs.home}/sleeRA.zip"/>
        <!-- Required by test usage.types.2216 -->
        <pathelement location="${libs.home}/dom4j.jar"/>

        <pathelement location="${asteriskra}/lib/asterisk-java.jar"/>

        <!-- DTD Files-->
        <pathelement location="${libs.home}/dtd"/>
    </path>

    <path id="deployer.class.path">
        <pathelement location="${libs.home}/slee.jar"/>
      <pathelement location="${jboss.home}/client/client.jar" />
      <pathelement location="${jboss.home}/client/jbossall-client.jar"
/>
      <pathelement location="${jboss.home}/client/getopt.jar" />
      <pathelement location="${jboss.home}/lib/log4j.jar" />
      <pathelement location="${jboss.home}/lib/jboss-jmx.jar" />
      <pathelement location="${jboss.home}/lib/xml-apis.jar" />
      <pathelement location="${jboss.home}/lib/xercesImpl.jar" />
      <pathelement
location="${jboss.home}/server/all/deploy/mobicents.sar" />
    </path>

    <path id ="run.class.path" >
        <!-- Stuff necessary to launch jboss -->
        <pathelement location="${jboss.home}/bin/run.jar"/>
        <pathelement location="${java.home}/lib/tools.jar"/>
        <!-- pathelement location="${project.home}/classes"/ -->
    </path>



    <target name="javacompile" depends="init"
            description="Java compiles all java source files.">
        <javac classpathref="project.class.path" debug="true"
deprecation="true"
```

```
                destdir="${dest}" nowarn="false" target="1.4"
source="1.4">

            <src path="${src}"/>
        </javac>
    </target>

    <target name="rmicompile" depends="init"
            description="Compiles rmi classes.">
                <rmic classpathref="project.class.path"
classname="${classpath.prefix}.slee.test.suite.RMINotificationListener
Impl" base="${dest}" />
    </target>

    <target name="sipra" depends="javacompile"
            description="Succsesively calls precompile, compile,
postcompile, package, deploy.">
        <subant target="">
            <fileset dir="${sipra}" includes="build.xml"/>
        </subant>
    </target>

    <target name="sipraclean">
        <subant target="clean">
            <property name="stage-dir" value="${sipra-stage}"/>
            <property name="ra-unit" value="${sipra-unit}"/>
            <fileset dir="${sipra}" includes="build.xml"/>
        </subant>
    </target>

    <target name="sipradeploy">
            <subant target="sipradeploy">
            <fileset dir="${sipra}" includes="build.xml"/>
            </subant>
    </target>

    <target name="siprabuild" >
        <subant target="siprabuild">
            <fileset dir="${sipra}" includes="build.xml"/>
            <property name="sipra.target.dir"
value="${jboss.mobicents.deploy.dir}"/>
        </subant>
    </target>

    <target name="asteriskra" depends="javacompile"
            description="Succsesively calls precompile, compile,
postcompile, package, deploy.">
        <subant target="">
            <fileset dir="${asteriskra}" includes="build.xml"/>
        </subant>
    </target>

    <target name="asteriskraclean">
        <subant target="clean">
            <fileset dir="${asteriskra}" includes="build.xml"/>
        </subant>
    </target>
```

```
    <!-- Deploys the Asterisk RA to JBOSS. Requres JBOSS to be running
-->
    <target name="asteriskradeploy" depends="asteriskra"
        description="Deploys the Asterisk RA to JBOSS">

        <echo>${asteriskra}/bin/DeployAsteriskRA.bat</echo>
        <exec dir="${basedir}\${asteriskra}\bin" spawn="false"
vmlauncher="true"
            os="Windows XP, Windows_NT, Windows 2000"

executable="${basedir}\${asteriskra}\bin\DeployAsteriskRA.bat" >
        </exec>

        <exec dir="${basedir}/${asteriskra}/bin" spawn="false"
            os="Solaris, SunOS"

executable="${basedir}/${asteriskra}/bin/DeployAsteriskRA.sh" >
        </exec>
    </target>


    <target name="resource" description="Copys resource files to the
${dest} folder">
        <copy todir="${dest}">
            <fileset dir="${src}">
                <include name="**/*.xml"/>
                <include name="**/*.properties"/>
            </fileset>
        </copy>
    </target>

    <target name="javadoc">
        <!--
        <javadoc destdir="docs" package="true">
            <fileset dir="${src}"/>
            <tag description="To Do:" name="todo" scope="all"/>
        </javadoc>
        -->
    </target>

    <!-- The following are here just in case they'll be needed one
day-->
    <target name="precompile" description="Not Implemented."/>


    <target name="postcompile" description="Updates build number.">
        <buildnumber file="build.number"/>
    </target>

    <!-- <target name="compile" depends="javacompile,rmicompile,sipra"
description="A javacompile alias"/>-->
    <target name="compile" depends="javacompile,rmicompile"
description="A javacompile alias"/>

    <target name="package" depends="javadoc,resource"
                description="Creates javadocs and copys resource files
to the ${dest} folder"/>
```

```
    <target name="make"
depends="init,precompile,compile,postcompile,package,deploy, deploy-
rar"
            description="Successively calls precompile, compile,
postcompile, package, deploy."/>

    <target name="hotcode" depends="rmicompile,package,deploy"
                description="Does the same as rebuild except no
compile. Eclipse compiled classes are used so hot code replacement is
available."/>

    <target name="all" depends="make" description="An alias for
make."/>

    <!-- These targets compile and run the forwarder for failover
testing -->
    <target name="compileforwarder" >
            <javac classpathref="project.class.path" debug="true"
deprecation="true"
            destdir="${dest}" nowarn="false" target="1.4"
source="1.4">
            <src path="${sip.services.home}/src/util"/>
             </javac>
        <rmic classpathref="project.class.path"
classname="util.MyRouter" base="${dest}" />
    </target>
    <target name="runforwarder" depends="compileforwarder"
description="Start the SIP forwarder service. It listens on 5060 and
forwards messages to an available back-end SLEE server.">
    <delete failonerror="false" file="forwarderdebug.txt" />
    <delete failonerror="false" file="forwarderlog.txt" />
    <java classpathref="project.class.path" fork="true"
classname="util.UdpForwarderImpl">
         <arg value="127.0.0.1" />
         <arg value="5065" />
         <arg value="5060" />
    </java>
    </target>

    <!-- clean rebuild and reinstall the targets everything -->
    <target name="rebuild" depends="clean,make" description="Clean and
make."/>

    <target name="clean" depends="sipraclean,asteriskraclean,auto-
deploy-clean,clean-ra">
        <delete file="mobicents.sar"/>
        <delete failonerror="false" includeemptydirs="true">
            <fileset dir="${dest}"/>
            <fileset dir="${junit.reports}"/>
        </delete>
        <delete failonerror="false" includeemptydirs="true">
            <fileset dir="${mobicents.sar}"/>
        </delete>
        <delete failonerror="false" includeemptydirs="true">
          <fileset
file="${jboss.home}/server/${node}/deploy/connector-test-ejb.jar" />
          <fileset file="${jboss.home}/server/${node}/deploy/ejbref-
test-ejb.jar" />
```

```
        </delete>
        <delete failonerror="false" includeemptydirs="true">
            <fileset dir="${jboss.home}/server/${node}/tmp/deploy">
                <include name="tmpDUJars**/*.*"/>

                <include name="tmpDUJars*"/>
            </fileset>
        </delete>
    </target>

    <target name="init"
        description="Creates target directories (e.g. ${dest},
${junit.reports} ...)">
        <antcall target="copy-jboss-config"/>
        <mkdir dir="${mobicents.sar}" />
        <mkdir dir="${dest}"/>
        <mkdir dir="${junit.reports}"/>
        <mkdir dir="${jboss.mobicents.deploy.dir}"/>
        <mkdir dir="${jboss.mobicents.deploy.dir}/scripts"/>
      <mkdir dir="${sipra-stage}"/>
    </target>

    <target name="copy-jboss-config">
        <echo>${node} is new JBoss deployment configuration</echo>
        <mkdir dir="${jboss.home}/server/${node}"/>
        <sync todir="${jboss.home}/server/${node}"
includeEmptyDirs="true">
            <fileset file="${jboss.home}/server/all/**" />
        </sync>
        <!-- delete this file to make sure that the smart copy of the
node specific config will work -->
        <delete file="${jboss.home}/server/${node}/conf/jboss-
service.xml"/>
    </target>


    <target name="buildsar"
        description="Creates a jboss sar archive from the ${dest}
directory">
        <jar destfile="mobicents.sar">
            <metainf file="xml/jboss-service.xml"/>

            <fileset dir="${dest}">
                <include name="**/*.*"/>
            </fileset>
        </jar>
    </target>



    <!-- Make the ejb-jar files needed for the ejb-ref and jca-
connector tests -->
    <target name="deploy-ejb-jars">

        <echo>dirpath is ${dirpath.prefix}</echo>

        <!-- Build the ejb-jar for the ejb-ref test -->
        <delete failonerror="false" includeemptydirs="true">
```

```
            <fileset file="${build}/ejbref-test-ejb.jar"/>
            <fileset file="${build}/connector-test-ejb.jar"/>
        </delete>
        <mkdir dir="${build}" />

        <jar destfile="${build}/ejbref-test-ejb.jar" >
            <metainf
file="${src}/${dirpath.prefix}/slee/test/env/ejbref/ejb/META-INF/ejb-
jar.xml"/>

            <metainf
file="${src}/${dirpath.prefix}/slee/test/env/ejbref/ejb/META-
INF/jboss.xml"/>
            <fileset dir="${dest}" >
                    <include name =
"${dirpath.prefix}/slee/test/env/ejbref/ejb/**/*.class" />
            </fileset>
        </jar>
        <copy todir="${jboss.home.deploy}" file="${build}/ejbref-test-
ejb.jar" />

        <!-- Build the ejb-jar for the jca connector test -->
        <jar destfile="${build}/connector-test-ejb.jar" >
            <metainf
file="${src}/${dirpath.prefix}/slee/test/connector/ejb/META-INF/ejb-
jar.xml"/>
            <metainf
file="${src}/${dirpath.prefix}/slee/test/connector/ejb/META-
INF/jboss.xml"/>
            <fileset dir="${dest}" >
                    <include name =
"${dirpath.prefix}/slee/test/connector/ejb/ConnectorTestHome.class" />
                    <include name =
"${dirpath.prefix}/slee/test/connector/ejb/**/*.class" />
                <include name =
"${dirpath.prefix}/slee/test/connector/TestEvent.class" />
            </fileset>
        </jar>
        <copy todir="${jboss.home.deploy}" file="${build}/connector-
test-ejb.jar" />



        <!-- delete failonerror="false" includeemptydirs="true">
            <fileset dir="${build}"/>
        </delete -->
    </target>

    <target name="build-test-connector-du">
         <!-- Build the test event jar -->
       <jar destfile="${build}/connector-test-event.jar" >
            <metainf
file="${src}/${dirpath.prefix}/slee/test/connector/META-INF/event-
jar.xml"/>
            <fileset dir="${dest}" >
                <include name =
"${dirpath.prefix}/slee/test/connector/TestEvent.class" />
            </fileset>
```

```
      </jar>
       <jar destfile="${build}/connector-test-event-du.jar" >
            <metainf
file="${src}/${dirpath.prefix}/slee/test/connector/META-
INF/deployable-unit.xml"/>
            <fileset dir="${build}" >
                <include name = "connector-test-event.jar" />
            </fileset>
       </jar>
         <!-- Build the test event jar -->
       <jar destfile="${build}/connector-test-event.jar" >
            <metainf
file="${src}/${dirpath.prefix}/slee/test/connector/META-INF/event-
jar.xml"/>
            <fileset dir="${dest}" >
                <include name =
"${dirpath.prefix}/slee/test/connector/TestEvent.class" />
            </fileset>

      </jar>
       <jar destfile="${build}/connector-test-event-du.jar" >
            <metainf
file="${src}/${dirpath.prefix}/slee/test/connector/META-
INF/deployable-unit.xml"/>
            <fileset dir="${build}" >
                <include name = "connector-test-event.jar" />
            </fileset>
       </jar>
    </target>

    <!-- Build the test event jar -->
    <!-- Run one junit test without starting the container -->
    <target name="test-connector"
            description="Runs ejb-connector test">

         <!-- Install the test event jar -->
         <java classpathref="deployer.class.path"
            fork="true"

      classname="org.mobicents.slee.container.management.jmx.SleeComma
ndInterface" >
          <arg value="-deploy" />
          <arg
value="file://${project.home.dir}/mobicents/build/connector-test-
event-du.jar" />
      </java>
        <junit fork="yes" showoutput="true">
            <formatter type="xml" />
            <classpath refid="project.class.path"/>
            <test
name="org.mobicents.slee.test.connector.ConnectorTest"
todir="${junit.reports}"/>
        </junit>
    </target>

    <!-- Make and deploy the JCA resource adaptor rar file -->
    <target name="deploy-rar">
```

```
        <mkdir dir="${build}/rar" />

        <!-- Build the ra.jar file -->
        <jar destfile="${build}/rar/ra.jar" >
            <fileset  dir="${dest}" >
                <include name =
"${dirpath.prefix}/slee/connector/adaptor/**/*.class" />
            </fileset>

        </jar>

        <!-- If the calling j2ee app is not deployed in JBoss it will
still need jboss-ha.jar
        <copy todir="${build}/rar" file="${libs.home}/jbossha.jar" />
        -->

        <!-- Build the mobicents.rar file -->
        <jar destfile="${build}/mobicents.rar">
            <metainf
file="${src}/${dirpath.prefix}/slee/connector/adaptor/META-
INF/ra.xml"/>
            <fileset dir="build/rar">
                <include name="**/*.*" />
            </fileset>
        </jar>

        <copy todir="${mobicents.sar}" file="${build}/mobicents.rar"
/>
        <copy todir="${mobicents.sar}"
file="${src}/${dirpath.prefix}/slee/test/connector/ejb/slee-ds.xml" />

        <delete failonerror="false" includeemptydirs="true">
            <fileset dir="${build}/rar"/>
        </delete>
    </target>

    <target name="clean-ra">
        <delete failonerror="false" includeemptydirs="true">
            <fileset file="${build}/mobicents.rar"/>
        </delete>
    </target>

    <target name="deploy" depends="copylibs"
        description="Creates a JBoss sar archive and deploys it to a
jboss installation">

        <copy todir="${mobicents.sar}" >
            <fileset dir="classes">
                <include name="org/mobicents/**/*.*"/>
                <!-- Really we shouldn't be deploying all the test
stuff here since -->
                <exclude
name="${dirpath.prefix}/slee/test/connector/**/*.*"/>
                <exclude
name="${dirpath.prefix}/slee/test/env/**/*.*"/>
                <exclude
name="${dirpath.prefix}/slee/connector/adaptor/**/*.*"/>
```

```
                <exclude
name="${dirpath.prefix}/slee/resource/sip/**/*.*"/>
            </fileset>
        </copy>

        <copy todir="${mobicents.sar}/dtd" >
            <fileset dir="dtd">
                <include name="**/*.*"/>
            </fileset>
        </copy>

        <copy  file="README.txt" todir="${jboss.home}"/>

        <copy  file="xml/log4j.xml.${debug}"
tofile="${jboss.home}/server/${node}/conf/log4j.xml" overwrite="true"
/>
        <copy  file="xml/cluster-service.xml"
todir="${jboss.home.deploy}" />
        <copy  file="xml/cache-invalidation-service.xml"
todir="${jboss.home.deploy}" />
        <copy  file="xml/jboss-service.xml"
todir="${mobicents.sar}/META-INF" />
        <copy  file="xml/slee-event-jar.xml"
todir="${mobicents.sar}/xml" />
        <copy  file="xml/slee-profile-spec-jar.xml"
todir="${mobicents.sar}/xml" />
        <copy  file="xml/event-jar.xml" todir="${mobicents.sar}/xml"
/>
        <copy  file="xml/hsqldb-ds.xml.${node}"
tofile="${jboss.home.deploy}/hsqldb-ds.xml" overwrite="true" />
        <copy  file="xml/jboss-treecache-service.xml.${cachemode}"
tofile="${jboss.home.deploy}/jboss-treecache-service.xml"
overwrite="true" />
        <copy  file="xml/jboss-treecache-profile-
service.xml.${cachemode}" tofile="${mobicents.sar}/jboss-treecache-
profile-service.xml" overwrite="true" />
        <copy  file="xml/jboss-treecache-deployment-
service.xml.${cachemode}" tofile="${mobicents.sar}/jboss-treecache-
deployment-service.xml" overwrite="true" />
        <copy  file="xml/jboss-treecache-runtime-
service.xml.${cachemode}" tofile="${mobicents.sar}/jboss-treecache-
runtime-service.xml" overwrite="true"/>
        <copy  file="bin/SleeCommandInterface.bat"
todir="${jboss.home}/bin" />
        <copy  file="bin/SleeCommandInterface.sh"
todir="${jboss.home}/bin" />
        <chmod file="${jboss.home}/bin/SleeCommandInterface.sh"
perm="755"> </chmod>
        <copy  file="bin/java.policy" todir="${jboss.home}/bin" />
        <copy  file="${libs.home}/jgroups-all.jar"
todir="${jboss.home.deploy}/../lib"/>
        <copy  file="docs/pictures/mobicents-logo.gif"
tofile="${jboss.home.deploy}/jmx-console.war/images/logo.gif" />
        <copy  todir="${jboss.home}/server/">
            <fileset file="xml/server/**" />
        </copy>
```

```
        <copy  file="lib/jboss-cache.jar"
todir="${jboss.home}/server/${node}/lib/" />

        <copy  todir="${jboss.home}/docs/licenses/">
            <fileset file="LICENSES/*" />
        </copy>

        <echo>Touching ${mobicents.sar}/META-INF/jboss-service.xml to
cause redeploy</echo>
        <touch>
            <fileset file="${mobicents.sar}/META-INF/jboss-
service.xml" />
        </touch>
    </target>

    <!-- Starts the JBoss Container-->
    <target name="run-jboss" description="Runs the JBoss container">
        <java spawn="true" fork="true" dir="${jboss.home}/bin"
jar="${jboss.home}/bin/run.jar"  classpathref="run.class.path">
            <arg value="-c all"/>
            <arg value="-b 127.0.0.1"/>
        </java>
    </target>

    <target name="shutdown-jboss" description="Shuts down the jboss
server">
        <java  dir="${jboss.home}/bin"
jar="${jboss.home}/bin/shutdown.jar" classpathref="run.class.path"
fork="true" >
            <arg value="-S"/>
        </java>
    </target>

    <!-- Run one junit test without starting the container -->
    <target name="test"
            description="Runs one test specified by the command line
parameter test.name (ant ... -Dtest.name=MyTest)">
        <junit fork="yes" showoutput="true">
            <formatter type="xml" />
            <classpath refid="project.class.path"/>
            <test name="${test.name}" todir="${junit.reports}"/>
        </junit>
    </target>


    <!-- Copies all lib files to the sar directory of the local jboss
installation -->
    <target name="copylibs" description = "copy project libs not part
of jboss into jboss" >
        <copy todir="${mobicents.sar}" flatten="yes">
            <fileset dir="${libs.home}">
                <include name="commons-beanutils.jar"/>
                <include name="commons-lang.jar"/>
                <include name="JainSipApi1.1.jar"/>
                <include name="javassist.jar"/>
                <include name="sleeRA.zip"/>
                <include name="commons-beanutils.jar"/>
                <include name="jboss-jmx.jar"/>
```

```
                <include name="sleetck-minimal.jar"/>
                <include name="nist-sip-1.2.jar"/>
                <include name="sleetck-ra-common.jar"/>
                <include name="commons-logging.jar"/>
                <include name="commons-pool-1.2.jar"/>
                <include name="slee-tasks.jar"/>
                <include name="tcktools.jar"/>
                <include name="slee.jar"/>
            </fileset>
            <fileset dir="${slee.tck.home}">
                <include name="jars/sleetck-ra-common.jar"/>
                <include name="jars/slee-tasks.jar"/>
                <include name="jars/tcktools.jar"/>
            </fileset>
        </copy>
        <copy todir="${mobicents.sar}/META-INF" file="xml/jboss-
service.xml"/>

    </target>

    <target name="cc-build-loop" depends="rebuild" />

    <!-- running jmx remote client standalone  -->
    <target name="jmx" depends="compile">
        <java fork="true" classpathref="project.class.path"
classname="${classpath.prefix}.slee.test.ConnectorClient"/>
    </target>

    <target name="tck-batchtest" depends="rmicompile"
        description="Runs the full suite of TCK tests. The server has
to be running when this target is used.">
        <junit printsummary="yes" haltonfailure="no">
            <classpath refid="project.class.path">
            </classpath>

            <formatter type="xml"/>

            <batchtest fork="yes" todir="${junit.reports}">
                <fileset dir="${src}"
excludesfile="exclude_batchtest.list">
                    <include
name="org/mobicents/slee/test/suite/tckwrapper/**/*Test.java" />
                </fileset>
            </batchtest>
        </junit>
        <junitreport todir="${junit.reports}">
          <fileset dir="${junit.reports}">
            <include name="TEST-*.xml"/>
          </fileset>
          <report format="frames" todir="${junit.reports}/html"/>
        </junitreport>

        <echo>>>> Summary Test report can be found at:
${junit.reports}/html/index.html</echo>
    </target>

     <!-- Build the pinger service and copy over -->
    <target name="build-pinger-jars">
```

```
        <mkdir dir="${build}" />
            <javac classpathref="project.class.path" debug="true"
deprecation="true"
                destdir="${dest}" nowarn="false" target="1.4"
source="1.4">
                <src path="${sip.services.home}/src"/>
         </javac>
            <jar destfile="${build}/pinger.jar">
                <metainf
file="${sip.services.home}/src/util/pinger/META-INF/sbb-jar.xml"/>
                <fileset  dir="${dest}">
                <include name="util/**/*.class"/>
                </fileset>
            </jar>
            <jar destfile="${build}/event.jar">
                <metainf
file="${sip.services.home}/src/util/pinger/META-INF/event-jar.xml"/>
                <fileset  dir="${dest}">
                <include name="util/Ping.class"/>
                </fileset>
            </jar>
            <copy todir="${build}"
file="${sip.services.home}/src/util/pinger/service.xml" />
            <jar destfile="${build}/pingerdu.jar">
                <metainf
file="${sip.services.home}/src/util/pinger/META-INF/deployable-
unit.xml"/>
                <fileset dir="${build}">
                    <include name="**/*.*" />
                </fileset>
            </jar>
            <move todir="${sip.services.home}/examples"
file="${build}/pingerdu.jar" />
            <delete failonerror="false" includeemptydirs="true">
                <fileset dir="${build}"/>
            </delete>
    </target>

    <target name="build-sip-proxy-service-jars">
        <!-- Build the jar file for enventrytest.jar and copy over -->
        <property name="build.sip.dir"
value="${build}/sip"></property>
        <mkdir dir="${build.sip.dir}" />
           <javac classpathref="project.class.path" debug="true"
deprecation="true"
                destdir="${dest}" nowarn="false" target="1.4"
source="1.4">
                <src path="${sip.services.home}/src"/>
         </javac>

        <jar destfile="${build.sip.dir}/registrarsbb.jar">
            <metainf
file="${sip.services.home}/src/org/mobicents/slee/services/sip/registr
ar/META-INF/sbb-jar.xml"/>
            <fileset  dir="${dest}">
            <include
name="org/mobicents/slee/services/sip/registrar/**/*.class"/>
            </fileset>
```

```
        </jar>
        <jar destfile="${build.sip.dir}/proxysbb.jar">
            <metainf
file="${sip.services.home}/src/org/mobicents/slee/services/sip/proxy/M
ETA-INF/sbb-jar.xml"/>
            <fileset  dir="${dest}">
            <include
name="org/mobicents/slee/services/sip/proxy/**/*.class"/>
            <include
name="org/mobicents/slee/services/sip/common/**/*.class"/>
            <include name="util/**/*.class"/>
        </fileset>
        </jar>
        <copy todir="${build.sip.dir}"
file="${sip.services.home}/src/org/mobicents/slee/services/sip/proxy/s
ervice.xml" />
        <jar destfile="${build.sip.dir}/proxyservice.jar">
            <metainf
file="${sip.services.home}/src/org/mobicents/slee/services/sip/proxy/M
ETA-INF/deployable-unit.xml"/>
            <fileset dir="${build.sip.dir}">
                    <include name="proxysbb.jar" />
                    <include name="registrarsbb.jar" />
                    <include name="service.xml" />
                </fileset>
        </jar>
        <move todir="${sip.services.home}/examples"
file="${build.sip.dir}/proxyservice.jar" />
        <delete failonerror="false" includeemptydirs="true">
            <fileset dir="${build.sip.dir}"/>
        </delete>
    </target>

    <!-- Deploys the SIP proxy to JBOSS. -->
    <target name="proxy-service-deploy"  depends="build-sip-proxy-
service-jars"
      description = "Deploys the SIP Proxy Service">
        <path id="proxyservice">
            <pathelement location="./thirdparty/sip-
services/examples/proxyservice.jar"/>
        </path>
        <property name="proxyservice" refid="proxyservice"/>
       <java classpathref="deployer.class.path" fork="true"
classname="org.mobicents.slee.container.management.jmx.SleeCommandInte
rface" >
        <arg value="-jnpPort"/>
        <arg value="${jnpPort}"/>
          <arg value="-deploy" />
          <arg value="file://${proxyservice}" />
       </java>
       <echo>Proxy Service deployed</echo>
      <!-- Activate the proxy service -->
       <java classpathref="deployer.class.path" fork="true"
classname="org.mobicents.slee.container.management.jmx.SleeCommandInte
rface" >
       <arg value="-jnpPort"/>
       <arg value="${jnpPort}"/>
         <arg value="-activateService" />
```

```
         <arg value="ServiceID[JAIN SIP Proxy Service#NIST#1.0]"/>
       </java>
      <echo> proxy service activated </echo>
    </target>

    <target name="tests-slee-tck" description="Invoke this target with
a file pattern for TCK test. For example -Dtests=tests/sbb or -
Dtests=tests/sbb/abstractclass/Test522Test">
        <!-- If all tests pass the result of the Java process is 0
otherwise != 0. This is useful for cruisecontrol integrations -->

        <!-- To debug TCK tests from Eclipse, run the class below with
the specified command line and test pattern -->

        <path id="slee.tck.du.path.id">
                <pathelement
location="${slee.tck.home}/jars/deployable-units"/>
        </path>
        <pathconvert targetos="unix" property="slee.tck.du.path"
refid="slee.tck.du.path.id"/>
        <property name="tck.params" value=" -batch 'testsuite
${slee.tck.home}/testsuite;workDirectory -create -overwrite
${slee.tck.home}/tckwork;open ${slee.tck.home}/slee-tck-
mobicents.jti;set sleetck.env.connections.tckResourceHostIP
127.0.0.1;set sleetck.env.connections.jmxAgentHostIP 127.0.0.1;set
sleetck.tests.needTests Yes; set
sleetck.env.connections.componentUrlPrefix
file\:${slee.tck.du.path};set sleetck.env.user-options.defaultTimeout
25000; concurrency 1;tests ${slee-tck-tests-pattern};
runtests;writereport ${slee.tck.home}/reports;' "/>
        <echo>Starting TCK with params: ${tck.params}</echo>
        <echo>Excluding tests listed in file:
${slee.tck.home}/testsuite/jain-slee-tck-1_0.jtx</echo>
           <java classname="com.sun.javatest.tool.Main"
             fork="yes" maxmemory="256m"
           >
            <sysproperty key="java.security.manager" value=""/>
            <sysproperty key="javatest.security.allowPropertiesAccess"
value="true"/>
            <sysproperty key="java.security.policy"
value="${slee.tck.home}/config/tck-security.policy"/>
            <sysproperty key="org.xml.sax.driver"
value="org.apache.xerces.parsers.SAXParser"/>
            <jvmarg line="-Xdebug -Xnoagent -
Xrunjdwp:transport=dt_socket,address=8484,server=y,suspend=n"/>
            <arg line="${tck.params}"/>
            <classpath refid="project.class.path"/>
               <classpath>
                   <pathelement
location="${slee.tck.home}/jars/sleetck.jar"/>
                   <pathelement
location="${slee.tck.home}/lib/javatest.jar"/>
                   <pathelement
location="${slee.tck.home}/lib/xerces.jar"/>
                   <pathelement
location="${slee.tck.home}/lib/sigtest.jar"/>
               </classpath>
           </java>
```

```
        <path id="slee.tck.reports.path.id">
                <pathelement
location="${slee.tck.home}/reports/report.html"/>
        </path>
        <property name="slee.tck.reports.path"
refid="slee.tck.reports.path.id"/>
        <echo>SLEE TCK report available at
${slee.tck.home}/reports/report.html</echo>
    </target>

    <target name="cc-dailytckrun" depends="clean,make"
        description="Used for daily tck runs executed by
Cruisecontrol. You won't need using this target and besides you won't
be able to since you're lacking the jboss launch script.">

        <!-- The following script will run jboss-->
        <exec dir="${jboss.home}" spawn="true" os="Linux"
            executable="${jboss.home}/bin/cc-run.sh"/>
        <echo message="Running JBOSS."/>

        <!-- Wait for jboss to completely start -->
        <sleep minutes="5"/>

        <!-- run the TCK -->
        <echo message="Starting the TCK over test dir ${tests}"/>

        <exec executable="ant" os="Linux">
            <arg value="tests-slee-tck"/>
            <arg value="-Dtests=tests"/>
        </exec>

        <echo message="TCK testing done, preparing to shutdown
JBOSS"/>


        <!-- Kill JBoss -->
        <exec dir="${jboss.home}" spawn="false" os="Linux"
            executable="${jboss.home}/bin/shutdown.sh">
            <arg value="--shutdown"/>
        </exec>

    </target>

    <target name="runPMD" description="Runs PMD Report">
        <ant antfile="runPMD.xml" target="pmd" inheritAll="true"
inheritRefs="true"/>
    </target>

    <target name="runPMDOnPackage" description="Runs PMD Report for a
package.">
        <ant antfile="runPMD.xml" target="pmdForPackage"
inheritAll="true" inheritRefs="true"/>
    </target>

    <target name="auto-deploy-sip" depends="init,make,siprabuild,
build-sip-proxy-service-jars"
        description = "Copies all SIP components to the server
directories. The server will consequently deploy or redeploy them.">
```

```
        <copy todir="${jboss.mobicents.deploy.dir}"
file="${sip.services.home}/examples/proxyservice.jar" />
        <copy todir="${jboss.mobicents.deploy.dir}/scripts"
file="${sip.services.home}/sip-deploy.bsh" />
    </target>

    <target name="auto-deploy-clean"
        description = "Deletes all Mobicents components placed in the
server directories.">
        <delete failonerror="false" includeemptydirs="true">
            <fileset dir="${jboss.mobicents.deploy.dir}"/>
            <fileset dir="${jboss.mobicents.deploy.dir}/scripts/"/>
        </delete>
    </target>
</project>
```

C.2. sipra.properties

```
#default IP address 0.0.0.0 can pickup the IP address automatically --
this is not a good idea
#if you have multiple interfaces.
#javax.sip.IP_ADDRESS=0.0.0.0
#
# When javax.sip.IP_ADDRESS is not specified, the SIP RA
# uses the default binding address of the SLEE container
#
#javax.sip.IP_ADDRESS=127.0.0.1
javax.sip.RETRANSMISSION_FILTER=on
javax.sip.STACK_NAME=SipResourceAdapter
javax.sip.PORT=5060
javax.sip.TRANSPORT=udp
javax.sip.STACK_PREFIX=gov.nist
gov.nist.javax.sip.TRACE_LEVEL=32
gov.nist.javax.sip.DEBUG_LOG=sipdebug.txt
gov.nist.slee.resource.sip.AUTHENTICATION=false
gov.nist.slee.resource.sip.AUTH_FILE=passwords.xml
gov.nist.javax.sip.THREAD_POOL_SIZE=1
```

C.3. wakeup-deployable-unit.xml

```
<?xml version="1.0" encoding="utf-8"?>
<!DOCTYPE deployable-unit PUBLIC "-//Sun Microsystems, Inc.//DTD JAIN
SLEE Deployable Unit 1.0//EN"
                                  "http://java.sun.com/dtd/slee-
deployable-unit_1_0.dtd">
<deployable-unit>
    <jar>jars/WakeUp-sbb.jar</jar>
    <service-xml>wakeup-service.xml</service-xml>
</deployable-unit>
```

C.4. WakeUpSbb.java

```
package org.mobicents.slee.examples.wakeup;

import java.text.ParseException;
```

```
import java.util.ArrayList;
import java.util.Iterator;
import java.util.Map;

import javax.naming.Context;
import javax.naming.InitialContext;
import javax.naming.NamingException;
import javax.sip.ClientTransaction;
import javax.sip.InvalidArgumentException;
import javax.sip.ServerTransaction;
import javax.sip.SipException;
import javax.sip.SipProvider;
import javax.sip.TransactionUnavailableException;
import javax.sip.address.Address;
import javax.sip.address.AddressFactory;
import javax.sip.address.SipURI;
import javax.sip.address.URI;
import javax.sip.header.ContactHeader;
import javax.sip.header.ContentTypeHeader;
import javax.sip.header.FromHeader;
import javax.sip.header.Header;
import javax.sip.header.HeaderFactory;
import javax.sip.header.MaxForwardsHeader;
import javax.sip.header.ToHeader;
import javax.sip.header.ViaHeader;
import javax.sip.message.MessageFactory;
import javax.sip.message.Request;
import javax.sip.message.Response;
import javax.slee.ActivityContextInterface;
import javax.slee.ComponentID;
import javax.slee.RolledBackContext;
import javax.slee.SbbContext;
import javax.slee.facilities.AlarmFacility;
import javax.slee.facilities.Level;
import javax.slee.facilities.TimerEvent;
import javax.slee.facilities.TimerFacility;
import javax.slee.facilities.TimerOptions;
import javax.slee.facilities.TraceFacility;
import javax.slee.nullactivity.NullActivity;
import javax.slee.nullactivity.NullActivityContextInterfaceFactory;
import javax.slee.nullactivity.NullActivityFactory;

import
org.mobicents.slee.resource.sip.SipActivityContextInterfaceFactory;
import org.mobicents.slee.resource.sip.SipFactoryProvider;
import
org.mobicents.slee.services.sip.common.SipSendErrorResponseException;
import org.mobicents.slee.services.sip.registrar.LocationService;
import
org.mobicents.slee.services.sip.registrar.LocationServiceException;
import org.mobicents.slee.services.sip.registrar.RegistrationBinding;



public abstract class WakeUpSbb implements javax.slee.Sbb {
```

```
      private LocationService locationService;

      public void onMessageEvent(javax.sip.RequestEvent event,
ActivityContextInterface aci) {
            Request request = event.getRequest();

            try {
            // Notifiy the client that we received the SIP MESSAGE
request
                  ServerTransaction st = (ServerTransaction)
aci.getActivity();
                  Response response =
messageFactory.createResponse(Response.OK, request);
            st.sendResponse(response);

            //   CREATE A NEW NULL ACTIVITIY
            NullActivity timerBus =
this.nullActivityFactory.createNullActivity();
            // ATTACH ITSELF TO THE NULL ACTIVITY
            // BY USING THE ACTIVITY CONTEXT INTERFACE
            ActivityContextInterface timerBusACI =

      this.nullACIFactory.getActivityContextInterface(timerBus);
                  timerBusACI.attach(sbbContext.getSbbLocalObject());

                  // PARSING THE MESSAGE BODY
                  String body = new String(request.getRawContent());
                  int i = body.indexOf(" ");
                  String timerValue = body.substring(0,i);
                  int timer = Integer.parseInt(timerValue);
                  String bodyMessage = body.substring(i+1);

                  // SETTING VALUES ON THE ACTIVITY CONTEXT
                  // USING THE SBB CUSTOM ACI
                  WakeUpSbbActivityContextInterface myViewOfTimerBusACI
=

      this.asSbbActivityContextInterface(timerBusACI);
                  myViewOfTimerBusACI.setBody(bodyMessage);
                  // The From field of each SIP MESSAGE has the UA
Address of Record (logical address),
                  // which can be mapped to a current physical contact
address. The mapping is provided by the LocationService,
                  // which works together with the SIP Registrar
service.
                  FromHeader fromHeader =
(FromHeader)request.getHeader(FromHeader.NAME);
                  Address fromAddress = fromHeader.getAddress();
                  URI contactURI =
findLocalTarget(fromHeader.getAddress().getURI());
                  Address contactAddress =
addressFactory.createAddress(contactURI);
                  ContactHeader contactHeader =
headerFactory.createContactHeader(contactAddress);

                  myViewOfTimerBusACI.setContact(contactHeader);

                  // SETTING THE TIMER BY USING THE VALUE
```

```
                  // IN THE SIP MESSAGE BODY
                  TimerOptions options = new TimerOptions();
                  options.setPersistent(true);
                  this.timerFacility.setTimer(timerBusACI,
                              null,
                              System.currentTimeMillis()+timer*1000,
                              options);

            } catch (Exception e) {
                  this.trace(Level.WARNING,
                              "Exception while retrieveing Activity
Context Interface: ", e);
            }
      }


      public void onTimerEvent(TimerEvent event,
ActivityContextInterface aci) {
            // RETRIEVING STORED VALUE FROM THE ACTIVITY CONTEXT
INTERFACE
            WakeUpSbbActivityContextInterface myViewOfACI =
                  this.asSbbActivityContextInterface(aci);
            Header contact = myViewOfACI.getContact();
            String body = myViewOfACI.getBody();

            // SENDING BACK THE WAKE UP CALL
            sendWakeUpCall(contact, body);
      }



      private void sendWakeUpCall(Header toContact, String body) {
            String strContact = toContact.toString();

            int beginIndex = strContact.indexOf('<');
            int endIndex = strContact.indexOf('>');
            String toAddressStr = strContact.substring(beginIndex+1,
endIndex);
            try {
            SipURI fromAddress =
                  addressFactory.createSipURI("wakeup", "nist.gov");

            javax.sip.address.Address fromNameAddress =
addressFactory.createAddress(fromAddress);
            fromNameAddress.setDisplayName("WakeUp");
            FromHeader fromHeader =
                  headerFactory.createFromHeader(fromNameAddress,
"12345SomeTagID6789");

            javax.sip.address.Address toNameAddress =
addressFactory.createAddress(toAddressStr);
            toNameAddress.setDisplayName("Some Sleepy User");

            ToHeader toHeader =
                  headerFactory.createToHeader(toNameAddress, null);

            ArrayList viaHeaders = new ArrayList();
            ViaHeader viaHeader =
```

```
                  headerFactory.createViaHeader(
                        provider.getSipStack().getIPAddress(),
                        provider.getListeningPoints()[0].getPort(),

      provider.getListeningPoints()[0].getTransport(),
                        null);

            // add via headers
            viaHeaders.add(viaHeader);

            MaxForwardsHeader maxForwards =
                  this.headerFactory.createMaxForwardsHeader(70);


                  URI uri =
fp.getAddressFactory().createURI(toAddressStr);
                  Request req = messageFactory.createRequest(uri,
                              Request.MESSAGE,
                              this.provider.getNewCallId(),
                              headerFactory.createCSeqHeader(1,
                              Request.MESSAGE),
                              fromHeader,
                              toHeader,
                              viaHeaders,
                              maxForwards);
                  ContentTypeHeader contentType =
headerFactory.createContentTypeHeader("text", "plain");
                  req.setContent(body,contentType);
                  ClientTransaction ct =
provider.getNewClientTransaction(req);
                  ct.sendRequest();

            } catch (ParseException e) {
                  // TODO Auto-generated catch block
                  e.printStackTrace();
            } catch (InvalidArgumentException e) {
                  // TODO Auto-generated catch block
                  e.printStackTrace();
            } catch (TransactionUnavailableException e) {
                  // TODO Auto-generated catch block
                  e.printStackTrace();
            } catch (SipException e) {
                  // TODO Auto-generated catch block
                  e.printStackTrace();
            }


      }

      public MessageFactory getMessageFactory() { return
messageFactory; }


      /**
       *  Initialize the component
       */
      public void setSbbContext(SbbContext context) {
            this.sbbContext = context;
```

```
            try {
            Context myEnv = (Context) new
InitialContext().lookup("java:comp/env");
            // Storing Sbb Component ID
            id = sbbContext.getSbb();

            // Getting SLEE Factility
            traceFacility = (TraceFacility)
myEnv.lookup("slee/facilities/trace");
            timerFacility = (TimerFacility)
myEnv.lookup("slee/facilities/timer");
            alarmFacility = (AlarmFacility)
myEnv.lookup("slee/facilities/alarm");
            nullACIFactory =
(NullActivityContextInterfaceFactory)myEnv.

      lookup("slee/nullactivity/activitycontextinterfacefactory");
            nullActivityFactory =
(NullActivityFactory)myEnv.lookup("slee/nullactivity/factory");

            // Getting JAIN SIP Resource Adaptor interfaces
            fp = (SipFactoryProvider)
myEnv.lookup("slee/resources/jainsip/1.1/provider");

            provider = fp.getSipProvider();

            addressFactory = fp.getAddressFactory();
            headerFactory = fp.getHeaderFactory();
            messageFactory = fp.getMessageFactory();

            acif = (SipActivityContextInterfaceFactory) myEnv.

      lookup("slee/resources/jainsip/1.1/acifactory");

        } catch (NamingException ne) {
            this.trace(Level.WARNING, "Exception During
setSbbContext", ne);

        }

      }
    public void unsetSbbContext() { this.sbbContext = null; }

    // TODO: Implement the lifecycle methods if required
    public void sbbCreate() throws javax.slee.CreateException {}
    public void sbbPostCreate() throws javax.slee.CreateException {
setup(); }
    public void sbbActivate() {}
    public void sbbPassivate() {}
    public void sbbRemove() {}
    public void sbbLoad() { setup(); }
    public void sbbStore() {}
    public void sbbExceptionThrown(Exception exception, Object event,
ActivityContextInterface activity) {}
    public void sbbRolledBack(RolledBackContext context) {}
```

```
      public abstract
org.mobicents.slee.examples.wakeup.WakeUpSbbActivityContextInterface
asSbbActivityContextInterface(ActivityContextInterface aci);

    private void setup() {
      locationService = new LocationService();
    }

    /**
     * Attempts to find a locally registered contact address for the
given URI,
     * using the location service interface.
     */
    public URI findLocalTarget(URI uri) throws
SipSendErrorResponseException {
        String addressOfRecord = uri.toString();

        Map bindings = null;
        try {
            bindings = locationService.getBindings(addressOfRecord);
        } catch (LocationServiceException lse) {
            lse.printStackTrace();
        }

        if (bindings == null) {
            throw new SipSendErrorResponseException("User not found",
                    Response.NOT_FOUND);
        }
        if (bindings.isEmpty()) {
            throw new SipSendErrorResponseException(
                    "User temporarily unavailable",
                    Response.TEMPORARILY_UNAVAILABLE);
        }

        Iterator it = bindings.values().iterator();
        URI target = null;
        while (it.hasNext()) {
            RegistrationBinding binding = (RegistrationBinding)
it.next();
            System.out.println("BINDINGS: " + binding);
            ContactHeader header =
binding.getContactHeader(addressFactory, headerFactory);
            System.out.println("CONTACT HEADER: " + header);
            if (header == null) { // entry expired
                continue; // see if there are any more contacts...
            }
            Address na = header.getAddress();
            System.out.println("Address: " + na);
            target = na.getURI();
            break;
        }
        if (target == null) {
            System.err.println("findLocalTarget: No contacts for "
                    + addressOfRecord + " found.");
            throw new SipSendErrorResponseException(
                    "User temporarily unavailable",
                    Response.TEMPORARILY_UNAVAILABLE);
        }
```

```
        return target;
    }


      /**
       * Convenience method to retrieve the SbbContext object stored
in setSbbContext.
       *
       * TODO: If your SBB doesn't require the SbbContext object you
may remove this
       * method, the sbbContext variable and the variable assignment
in setSbbContext().
       *
       * @return this SBB's SbbContext object
       */

      protected SbbContext getSbbContext() {

            return sbbContext;
      }

      private SbbContext sbbContext; // This SBB's SbbContext

      protected final void trace(Level level, String message) {
        try {
            traceFacility.createTrace(id, level, "WakeUpSbb", message,
System.currentTimeMillis());
       } catch (Exception e) { }
    }

      protected final void trace(Level level, String message,
Throwable t) {
        try {
            traceFacility.createTrace(id, level, "WakeUpSbb", message,
t, System.currentTimeMillis());
        } catch (Exception e) { }
    }

      private MessageFactory messageFactory;
      private SipProvider provider;
      private SipFactoryProvider fp;
      private SipActivityContextInterfaceFactory acif;
      private TraceFacility traceFacility;
      private TimerFacility timerFacility;
      private AlarmFacility alarmFacility;
      private NullActivityContextInterfaceFactory nullACIFactory;
      private NullActivityFactory nullActivityFactory;
      private ComponentID id;
      private AddressFactory addressFactory;
      private HeaderFactory headerFactory;

}
```

C.5. WakeUpSbbActivityContextInterface.java

```
package org.mobicents.slee.examples.wakeup;
```

```
import javax.sip.header.Header;

public interface WakeUpSbbActivityContextInterface extends
javax.slee.ActivityContextInterface {
      public void setContact(Header header);
      public Header getContact();
      public void setBody(String body);
      public String getBody();

}
```

C.6. WakeUp-sbb-jar.xml

```
<?xml version="1.0" encoding="utf-8"?>
<!DOCTYPE sbb-jar PUBLIC "-//Sun Microsystems, Inc.//DTD JAIN SLEE SBB
1.0//EN"
                         "http://java.sun.com/dtd/slee-sbb-
jar_1_0.dtd">
<sbb-jar>
    <sbb>
        <description>This Sbb send a wake up call to the
user.</description>
        <sbb-name>Wake Up Sbb</sbb-name>
        <sbb-vendor>NIST</sbb-vendor>
        <sbb-version>1.0</sbb-version>
        <sbb-classes>
            <sbb-abstract-class>
                <sbb-abstract-class-
name>org.mobicents.slee.examples.wakeup.WakeUpSbb</sbb-abstract-class-
name>
            </sbb-abstract-class>
            <sbb-activity-context-interface>
                <sbb-activity-context-interface-
name>org.mobicents.slee.examples.wakeup.WakeUpSbbActivityContextInterf
ace</sbb-activity-context-interface-name>
            </sbb-activity-context-interface>
        </sbb-classes>

        <event event-direction="Receive" initial-event="True">
            <event-name>MessageEvent</event-name>
            <event-type-ref>
                <event-type-
name>javax.sip.message.Request.MESSAGE</event-type-name>
                <event-type-vendor>javax.sip</event-type-vendor>
                <event-type-version>1.1</event-type-version>
            </event-type-ref>
            <initial-event-select variable="ActivityContext"/>
        </event>

        <event event-direction="Receive" initial-event="False">
            <event-name>TimerEvent</event-name>
            <event-type-ref>
                <event-type-
name>javax.slee.facilities.TimerEvent</event-type-name>
                <event-type-vendor>javax.slee</event-type-vendor>
                <event-type-version>1.0</event-type-version>
            </event-type-ref>
```

```
        </event>

        <resource-adaptor-type-binding>
            <resource-adaptor-type-ref>
                <resource-adaptor-type-name>jain-sip</resource-
adaptor-type-name>
                <resource-adaptor-type-vendor>javax.sip</resource-
adaptor-type-vendor>
                <resource-adaptor-type-version>1.1</resource-adaptor-
type-version>
            </resource-adaptor-type-ref>
            <activity-context-interface-factory-name>
                slee/resources/jainsip/1.1/acifactory
            </activity-context-interface-factory-name>
            <resource-adaptor-entity-binding>
                <resource-adaptor-object-name>
                    slee/resources/jainsip/1.1/provider
                </resource-adaptor-object-name>
                <resource-adaptor-entity-link>
                                SipRA
                        </resource-adaptor-entity-link>
            </resource-adaptor-entity-binding>
        </resource-adaptor-type-binding>
    </sbb>
</sbb-jar>
```

C.7. wakeup-service.xml

```
<?xml version="1.0" encoding="utf-8"?>
<!DOCTYPE service-xml PUBLIC "-//Sun Microsystems, Inc.//DTD JAIN SLEE
Service 1.0//EN"
                            "http://java.sun.com/dtd/slee-
service_1_0.dtd">
<service-xml>
    <service>
        <description>Wake up service</description>
        <service-name>Wake Up Service</service-name>
        <service-vendor>NIST</service-vendor>
        <service-version>1.0</service-version>
        <root-sbb>
            <description>This Sbb send a wake up call to the
user.</description>
            <sbb-name>Wake Up Sbb</sbb-name>
            <sbb-vendor>NIST</sbb-vendor>
            <sbb-version>1.0</sbb-version>
        </root-sbb>
        <default-priority>0</default-priority>
    </service>
</service-xml>
```

Anexo D
Contrato del contribuyente

D. Contrato del contribuyente

```
http://www.mobicents.org

Software Contributor Agreement

This agreement is between you and CocoonHive LLC
("Mobicents"), represented by Ivelin Ivanov, owner of
the Mobicents project
hosted at http://www.mobicents.org . In this agreement,
"You" shall mean the
owner of the Contribution, as defined below, including
any individual or any corporation or entity which has
authorized the signatory of this agreement to act on
its behalf.  The purpose of this license is to set forth
the terms and conditions under which we may use
software that you wish to contribute to us for the
purpose of use within one or more of our software
development projects.  Nothing in this license changes
your ongoing right to use your contributions as you
wish for any purpose.

You accept and agree to the following terms and
conditions for your
present and future Contributions submitted to Mobicents:

1.    As used in this agreement, the term "Contribution"
shall mean any software, including source code
and/or object code, documentation, or modifications to
the foregoing, which You make available or
submit to Mobicents in any form.  Contributions shall
not include any software or documentation which
has been explicitly marked to indicate that it is not a

contribution to Mobicents.
2.    You hereby grant to Mobicents, its successors, and
assigns, the non-exclusive, transferable,
irrevocable, perpetual, royalty-free right to use,
modify, copy, sell, and distribute the Contributions
under the terms of any version of the GNU General
Public License, or any version of the GNU
Lesser General Public License.  Without limitation,
this grant is made with respect to any
copyright, patent, or other intellectual property or
moral rights You may have in or to the
Contributions.
3.    You represent that You are legally able and entitled
under the laws of your jurisdiction to enter
into this agreement and to grant Mobicents the rights
described in this license.
4.    You represent that You are the original author,
creator, or inventor of each of the Contributions to
Mobicents.
5.    Unless agreed to in writing, or required by
applicable law, your Contributions are licensed on an
"AS IS" basis and except as described in this
agreement, You do not make any warranty or
```

representation of any kind with regard to the Contributions.
6. You agree to notify Mobicents if any circumstance should arise which would make any of the foregoing representations inaccurate in any respect.
Signature:

E-mail:

Complete name:

Telephone:

Address:

Company/Title:

[Please send an e-mail indicating that you accept this agreement along with the full text of the agreement to ivelin@mobicents.org.]