Setting up the SIP Registrar implementation
A step-by-step guide

by Michael Maretzke

26h July 2007

Introduction

The SIP Registrar implementation described in this document represents the outcome of an experiment to

- work on the RFC 3261 (SIP protocol)[1]
- work with SIP Servlets[2]
- increase the knowledge of the author

This document describes the installation and setup of the environment for the SIP Registrar implementation described in "Implementing a RFC 3261 SIP Registrar – A quick-starter's guide"[3].

The implementation is available from http://www.maretzke.com/pub/howtos/sip_registrar/index.html.

Setting up the Environment

The document describes the setup of the environment on a Microsoft Windows XP machine and an installation of the BEA WebLogic SIP Server 3.0. The BEA WebLogic SIP Server is available for free for test and development purposes[4]. The development environment utilized here is the BEA Workshop Studio 3.2[5] being based on the Eclipse development environment.

After the installation of the SIP server and the development environment, the source code of the SIP Registrar needs to be downloaded from http://www.maretzke.com/pub/howtos/sip_registrar/index.html. The zip archive needs to be unzipped.

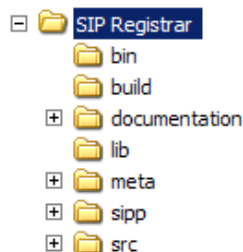The directory structure looks like follows:

Figure 1 Directory structure of SIP Registrar

The root directory of SIP Registrar contains two important files:

- `build.xml`
- `local.properties`

Adapt the values of the file `local.properties` to fit your local environment. Most important is to adapt the values for

- `DEPLOY-TARGETS`
- `SERVER-HOME`

```
# Add local environment specific properties on seperate
lines
SERVER=localhost
PORT=7001
USERNAME=weblogic
PASSWORD=weblogic
DEPLOY-TARGETS=AdminServer
SERVER-HOME=d:/beawlss30/sipserver30/server
```

Figure 2 Content of `local.properties`

The variable DEPLOY-TARGETS needs to hold the name of the server you specified during the installation of the SIP Server in the BEA WebLogic Configuration Wizard. Usually, given a default installation, the value holds `AdminServer`.

Verifying the Installation

After a successful deployment of the SIP Registrar application, the console of the SIP Server contains the following lines:

```
com.maretzke.sip.tools.registrar.Registrar --- Registrar
initialized. Serving domain: thesipdomain.de, minimum Expire
time is set to: 10

com.maretzke.sip.tools.proxy.LookupProxy --- LookupProxy
initialized. Serving domain: thesipdomain.de.
-------------------------------------
Bindings:
Expires:

-------------------------------------
```

If you find these line in the console log, the SIP Registrar is up and running.

To check if the application is up and running, point your browser to the web interface at http://localhost:7001/SIP_Registrar/listRegistrations.jsp

Your browser will show the current registrations known to the SIP Registrar. The screen should look like the screenshot shown below. Since there are currently no users registered, the page

doesn't contain any information. The page refreshes periodically.

To reset the SIP Registrar and remove all stored bindings in the Location Store, click the link "Clear bindings …".

Registrations

Clear bindings ...

Address	Contacts	Call-ID	CSeq	Expiration	Valid in seconds	Q

Figure 3 Screenshot of running Registrat

Command line – usage of ANT

The description below assumes you have a complete Java environment installed and properly configured. Furthermore, the tool ANT from the APACHE ANT Project[6] needs to be installed on your machine.

Start your command line tool (cmd.exe) and change the directory and drive to the location where your SIP Registrar is located at (e.g. `F:\JavaDev\SIP Registrar`). The system needs to be configured properly for the usage with the BEA WebLogic SIP Server 3.0. To do so, execute the `setWLSEnv.cmd` script found in `[BEAHOME]\sipserver30\server\bin`.

After setting up your environment properly, execute the command ant deploy to compile, package and deploy the SIP Registrar on the server.

```
F:\JavaDev\SIP Registrar>ant deploy

Buildfile: build.xml

compile:

    [javac] Compiling 7 source files to F:\JavaDev\SIP
Registrar\build

    [javac] Note: Some input files use unchecked or unsafe
operations.

    [javac] Note: Recompile with -Xlint:unchecked for
details.


package:

     [war] Building war: F:\JavaDev\SIP
Registrar\bin\SIP_Registrar.war


deploy:

 [wldeploy] weblogic.Deployer -nostage -verbose -noexit -
name SIP_Registrar -source F:\JavaDev\SIP
Registrar\bin\SIP_Registrar.war -targets AdminServer -
adminurl t3://localhost:7001 -user weblogic -password
******** -deploy

 [wldeploy] weblogic.Deployer invoked with options: -
nostage -verbose -noexit -name SIP_Registrar -source
F:\JavaDev\SIP Registrar\bin\SIP_Registrar.

war -targets AdminServer -adminurl t3://localhost:7001 -user
weblogic -deploy

 [wldeploy] <10.07.2007 16.04 Uhr CEST> <Info> <J2EE
Deployment SPI> <BEA-260121> <Initiating deploy operation
```

```
for application, SIP_Registrar [archive: F:\JavaDev\SIP
Registrar\bin\SIP_Registrar.war], to AdminServer .>

 [wldeploy] Task 7 initiated: [Deployer:149026]deploy
application SIP_Registrar on AdminServer.

 [wldeploy] Task 7 completed: [Deployer:149026]deploy
application SIP_Registrar on AdminServer.

 [wldeploy] Target state: deploy completed on Server
AdminServer

 [wldeploy]


BUILD SUCCESSFUL

Total time: 5 seconds
```

Figure 4 Console output for successful deployment

Using ANT build files in Eclipse

Eclipse supports ANT file controlled build processes. Open the view for ANT files by selecting "Window" from the menu bar. Select "Show View" and "Ant".
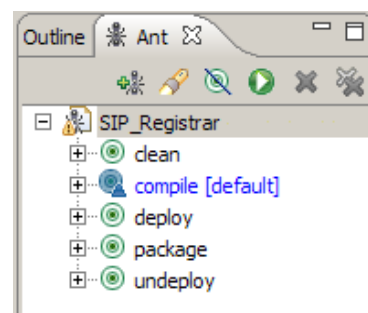
Figure 5 ANT view in Eclipse

Now, drag the `build.xml` file shown in the Package Explorer for the SIP Registrar into the ANT view of Eclipse. The various build targets of ANT become visible.

By clicking on deploy or undeploy, the application is deployed or undeployed from the actual server.

Ensure to edit the file `local.properties` in the SIP Registrar project prior to deploying and undeploying the application.

Open the project in Eclipse

Further development on the SIP Registrar might be done utilizing the Eclipse IDE. To import and open the project follow the steps described below.

Copy the content of the implementation's archive to a location of your choice. Afterwards, start Eclipse and select "Import" from the "File" menu. Choose "Existing Projects into Workspace" in the folder "General" and click "Next". In the "Import" dialog box, click on "Browse" to select the project's root directory. Navigate to the directory containing the unzipped SIP Registrar. Check the project "SIP Registrar" in the "Projects" area of the dialog

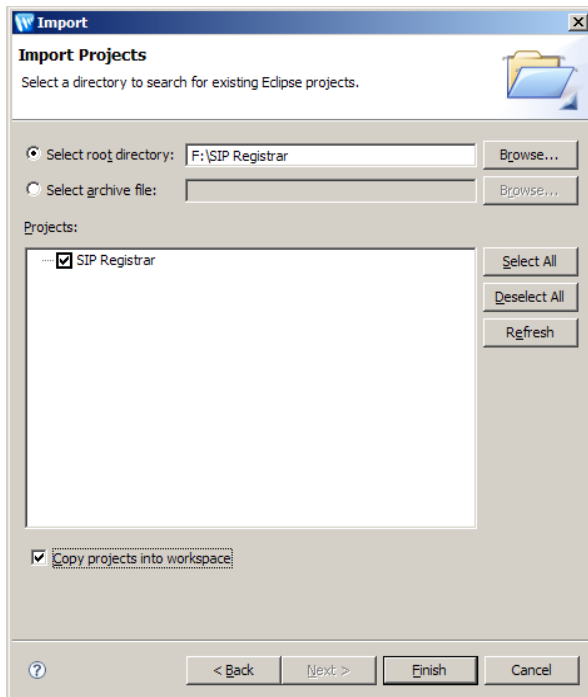box. Check "Copy Files into Workspace" and click "Finish" to start the import process.

Figure 6 Import dialog box of Eclipse - SIP Registrar

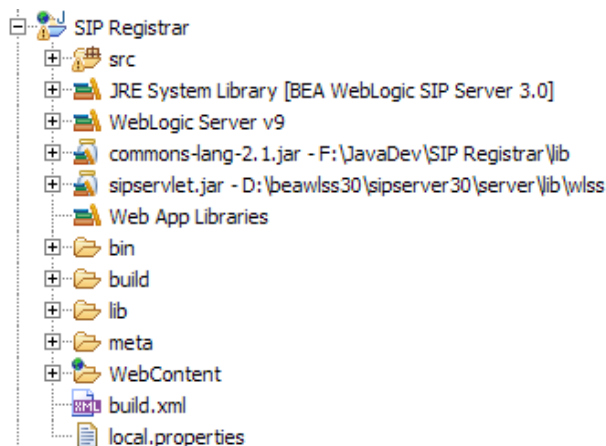After the import, the Package Explorer of Eclipse shows the following project structure.

Figure 7 Project Explorer in Eclipse after successful import

Create the project in Eclipse

If you're having issues with the pre-configured Eclipse project, you may need to setup the project manually. Here's a step-by-step guide.
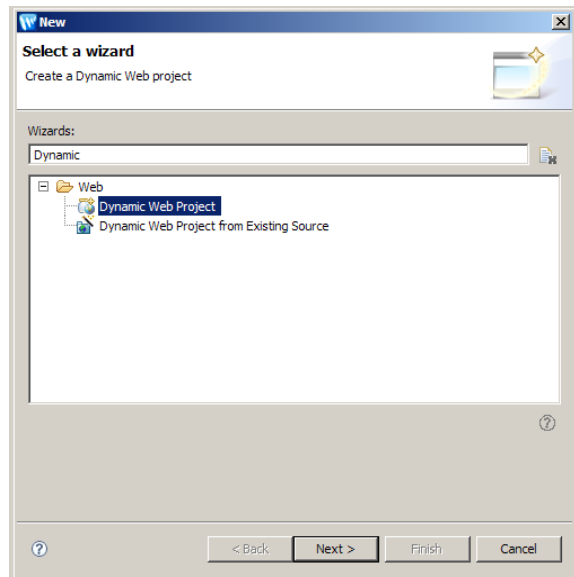
Create a new "Dynamic Web Project" project in Eclipse.

Figure 8 New Project wizard in Eclipse

Name the project "SIP Registrar" and select the target runtime as "WebLogic Server v9". If needed, create a new one by clicking on "New". Select "WebLogic Server v9" from the presented list. Remember, this description applies for the BEA Workshop which contains some deviations from a pure Eclipse installation. The asked for application server directory points to your installation of the BEA WebLogic SIP Server 3.0 (e.g. `d:/beawlss30/sipserver30`).
Click "Finish" and let the IDE create the project for you.

Now, copy the following folders:
- `bin`
- `build`
- `lib`
- `meta`
- `src`

and the following files:
- `build.xml`
- `local.properties`

from your downloaded SIP Registrar directory into the newly created SIP Registrar folder in the Eclipse workspace and refresh the view of the Package Explorer.
Unresolved import statements in the Java sources lead to errors in the project:
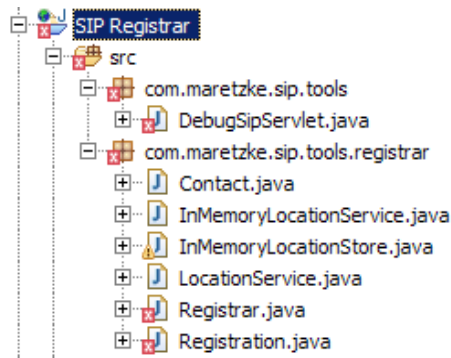
Figure 9 SIP Registrar with errors

Open the properties for the SIP Registrar project - right-click on SIP Registrar and select "Properties". Select "Java Build Path" and "Libraries". The following libraries need to be added:

- `sipservlet.jar`
- `commons-lang-2.1.jar`

The commons-lang library is part of the Jakarta Project[7]. The libraries are both stored in the SIP Registrar's `lib` directory.
Click on "Add External JARs" and maneuver to the `SIP Registrar\lib` directory. Add the libraries.
After adding the libraries to your project, the dialog should appear like the screenshot below.
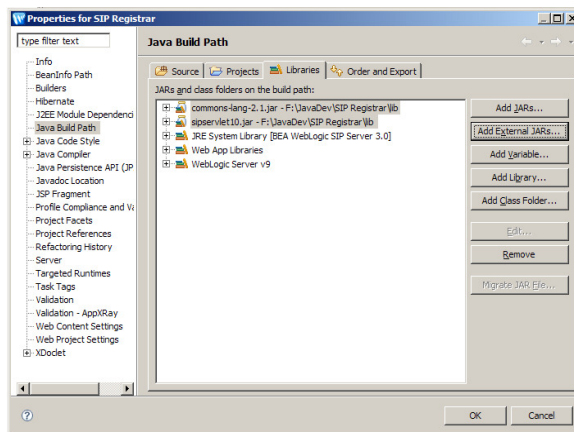

Figure 10 Properties for SIP Registrar after adding the libs

Now, the project should compile without any errors in Eclipse.

Important links:

SIP Registrar implementation download:
http://www.maretzke.com/pub/howtos/sip_registrar/index.html

Verifying the installation
http://localhost:7001/SIP_Registrar/listRegistrations.jsp

For questions and comments please contact me via michael@maretzke.com.

---

[1] RFC 3261 „SIP: Session Initiation Protocol",
see http://www.ietf.org/rfc/rfc3261.txt

[2] JSR-116 „SIP Servlet API",
see http://jcp.org/en/jsr/detail?id=116
and JSR-289 "SIP Servlet v1.1",
see http://jcp.org/en/jsr/detail?id=289

[3] "Implementing a RFC 3261 SIP Registrar – A quick starter's guide" by Michael Maretzke, 26th July 2007,
http://www.maretzke.com/pub/howtos/sip_registrar/index.html.

[4] BEA WebLogic SIP Server download,
see http://commerce.bea.com/showproduct.jsp?family=WLSS&major =3.0&minor=0

[5] BEA Workshop download,
see http://commerce.bea.com/showproduct.jsp?family=WLW&major =10.1&minor=0

[6] APACHE ANT Project see http://ant.apache.org/

[7] "commons lang" project at the Jakarta Project,
see http://jakarta.apache.org/commons/lang/