

# WSAMI Middleware Developer Guide

Author: Daniele Sacchetti – daniele.sacchetti@inria.fr

## Table of contents

1. INTRODUCTION .....	1
2. COMPILE, MAKEJAR, DIST AND RELEASE TARGETS.....	3
3. CSOAP.....	3
4. GATEWAY (GW) AND UNIVERSAL REPOSITORY (UR) SERVICES.....	3
5. SERVICE LOCATION PROTOCOL (SLP).....	4
6. CORE BROKER TOOLS.....	5

## 1. Introduction

Compiling and building a new release of WSAMI Middleware requires the following software installed on your machine:

- Java
- [ant](#) (version 1.6.2 or higher)

Once untarred `wsami-src-doc-1.0.0.tar.gz` in `$WSAMIDIR` directory, you have the following directory tree:

```
$WSAMIDIR  +--+ build.xml
           |
           + build +--+ class  (temporray directory to generate .class)
           |             + lib   (temporray directory to generate .jar)
           |
           + dist +--+ common
           |             + pda   (contain the final release for PDAs)
           |             + tomcat (contain the final release for PCs)
           |
           + docs (documentation)
           |
           + lib   (libraries used to compile the WSAMI Midelewre)
           |
           + samples (some basic samples showing basic functionalities)
           |
           + services +----+ ...
           |             + ... (Advanced examples of use of the Middleware)
           |             + ...
           |             + UR (UR client used to update the local Web services
           |                   directory - see [ARCHGUIDE] and [USERGUIDE])
           |
```

```

+ src +-+ CoreBroker +--+ CSoap
    |                               + Axis
    |                               + IFUtils
+ csoup
    |
+ WSAMI
    |
+ NDService
    |
+ GWService
    |
+ URService
    |
+ WSAMI

```

You can see the options offered by the default environment to recompile and create a new release of WSAMI Middleware by typing the following command from the \$WSAMIuuld.xml directory:

```
ant -projecthelp
```

You will get the following output:

```

Buildfile: build.xml
  Build file for wsami
Main targets:

backup                backup
clean                 Clean
compileCBIFUtils_dll  Generates shared library ifacemgr.dll (Windows, Intel)
compileCBIFUtils_so   Generates shared library libifacemgr.so (Linux, Intel)
compileCBIFUtils_so_pda Generates shared library libifacemgr.so (Linux, ARM)
compileCSoup          Compiles CSoup
compileLIBNDSLP_dll   Generates shared library NDslp.dll (Windows, Intel)
compileLIBNDSLP_so    Generates shared library libNDslp.so (Linux, Intel)
compileSLP_pda        Compiles slp (Linux, ARM)
compilewsamiAxis      Compiles wsami for Axis version
compilewsamiCSoup     Compiles wsami for CSoup version
distAxis              Generates distribution for Axis/Tomcat
distCSoup             Generates distribution for CSoup
init                  Initialization
makeApiDocs           Generates Java Api Documentation
makeDocs              Generates User Guide Documentation
makeJarCSoup          Generates CSoup Jar file
makeJarGWAxis         Generates GW Jar file for Axis version
makeJarURAxis         Generates UDDI Jar file for Axis version
makeJarswsamiAxis     Generates wsami Jar files for Axis version
makeJarswsamiCSoup    Generates wsami Jar files for CSoup version
release               Generates wsami Release
release-doc           Generates wsami Release with documentation
release-src           Generates wsami Release with source files
release-src-doc       Generates wsami Release with source files and documentation

Default target: release

```

If you want to build a new version of the WSAMI Middleware, you must apply the following modification to adapt the compilation scripts to your environment:

- \$WSAMIDIR/build.xml: update the wsami.version property with an increasing version number

- \$WSAMIDIR/commonLinux.xml and commonWindows.xml: set wsami.dir to \$WSAMIDIR

In the following sections we the above targets are examined in more detail.

## 2. Compile, makeJar, dist and release targets

compilewsamiAxis	Compiles wsami for Axis version
compilewsamiCSoap	Compiles wsami for CSoap version
distAxis	Generates distribution for Axis/Tomcat
distCSoap	Generates distribution for CSoap
makeJarswsamiAxis	Generates wsami Jar files for Axis version
makeJarswsamiCSoap	Generates wsami Jar files for CSoap version
release	Generates wsami Release
release-doc	Generates wsami Release with documentation
release-src	Generates wsami Release with source files
release-src-doc	Generates wsami Release with source files and documentation

The targets `compilewsamiAxis` and `compilewsamiCSoap` are used to compile the Java files implementing the WSAMI Middleware into `.class` files in `$WSAMIDIR/build/class`.

The targets `makeJarswsamiAxis` and `makeJarswsamiCSoap` are used to create the jar file (libraries) from the compiled `.class` files in `$WSAMIDIR/build/lib`.

The targets `distAxis` and `distCSoap` are used to create a complete distribution of WSAMI Middleware (with all the files necessary to install on a machine: libraries, configuration files, ...) in `$WSAMIDIR/dist/tomcat` and `$WSAMIDIR/dist/pda`.

The targets `release`, `release-doc`, `release-src` and `release-src-doc` are used to create the in `$WSAMIDIR/dist/wsami-x.y.z.tar.gz` file to be distributed and containing all the distributions and the source code and/or the documentation.

For more details about the different distributions available for WSAMI Middleware (PDA/PC, Linux/Windows with Cygwin) see [USERGUIDE] and [ARCHGUIDE].

## 3. CSoap

compileCSoap	Compiles CSoap
makeJarCSoap	Generates CSoap Jar file

The WSAMI Middleware includes CSoap, a Java implementation of SOAP that can run over J2ME Personal Profile.

For more information about CSoap, see [CSOAPARCH] and [CSOAPUSER] and for more information about how WSAMI Middleware makes use of CSoap see [USERGUIDE] and [ARCHGUIDE].

The targets `compileCSoap` and `makeJarCSoap` are respectively used to compile CSoap and build the jar files in `$WSAMIDIR/build`.

## 4. Gateway (GW) and Universal Repository (UR) services

makeJarGWAxis	Generates GW Jar file for Axis version
makeJarURAxis	Generates UR Jar file for Axis version

These targets are used to build the jar files that implement the advanced services (Gateway and

Universal Repository) offered by the WSAMI Middleware (for more details about these services, see [USERGUIDE] and [ARCHGUIDE]).

The generated jar files will be included in the WSAMI version for PC but not for in the PDA version because these services are not designed to run on PDAs.

## 5. Service Location Protocol (SLP)

<code>compileLIBNDSLP_dll</code>	Generates shared library <code>NDslp.dll</code> (Windows, Intel)
<code>compileLIBNDSLP_so</code>	Generates shared library <code>libNDslp.so</code> (Linux, Intel)
<code>compileSLP_pda</code>	Compiles <code>slp</code> (Linux, ARM)

For service discovery, the WSAMI Middleware makes use of [OpenSLP](#) SLP implementation: a SLP server that must be running along with WSAMI Middleware and a SLP library implementing the client-side functionalities (i.e., search for services available on the local network using multicast requests). The interaction between the Naming and Discovery Service and SLP is described in more detail in [ARCHGUIDE].

The nodes running WSAMI Middleware can be divided in two categories: standard nodes with only one network interface and gateway nodes with two network interfaces where the Gateway Service is deployed. For mode details about gateway nodes, see [USERGUIDE] and [ARCHGUIDE].

OpenSLP provides the following releases:

- OpenSLP Java API 1.0.0 that provides client-side functionalities to search for services only on one network interface and that does not provide a SLP server.
- OpenSLP 1.2.0 (implemented in C) that provides both a SLP server and client-side functionalities to search for services on multiple network interfaces.

- For the server side, the WSAMI Middleware provides a SLP server for each supported platform:

- PDA/Linux: `$WSAMIDIR/dist/pda/bin/slpd`: version 1.0.11
- PC/Linux: `$WSAMIDIR/dist/tomcat/bin/slpd`: version 1.0.11
- PC/Cygwin+Windows: `$WSAMIDIR/dist/tomcat/bin/slpd.exe`: version 1.0.11

- If you want to update the version installed on the WSAMI Middleware for PC you can download from OpenSLP web site and copy the `slpd` (or `slpd.exe` for Windows) to the right directory (see above).

- If you want to update the version installed on the WSAMI Middleware for PDA you must download the source code and recompile using the ant target `compileSLP_pda`. You must install the OpenSLP source on the directory specified in `$WSAMIDIR/build.xml` by the property `openslp.dir.linux`. You need the toolchain cross-compiler for ARM architecture (that you can download at <http://www.handhelds.org>).

Once you have obtained the `slpd` server for the ARM architecture, you can copy it to the `$WSAMIDIR/dist/pda/bin` directory

- On the client side, the nodes that do not act as Gateway make use of OpenSLP Java API 1.0.0 because they have only one network interface and they only need to search for services on this interface.

Gateway nodes make use of OpenSLP 1.2.0 because they have two network interfaces and they need to search for services on both of them.

The OpenSLP 1.2.0 library is only provided in C, so the WSAMI Middleware (ND service) makes use of it is through Java Native Interface (JNI).

The two ant targets `compileLIBNDSLP_dll` and `compileLIBNDSLP_so` produce a shared library respectively for Windows (`$WSAMIDIR/build/class/NDSservice/Axis/NDslp.dll`) and Linux (`$WSAMIDIR/build/class/NDSservice/Axis/libNDslp.so`) for PC. These targets require OpenSLP to be installed and built in the directory specified in `$WSAMIDIR/build.xml` by the property `openslp.dir.linux` or `openslp.dir.Windows` (see below). The generated library contains the JNI and C functions that act as interface and allow the WSAMI Middleware to make calls from Java to the C OpenSLP library provided by OpenSLP.

When you execute “`ant distAxis`”, the `libNDslp.so` (or `NDslp.dll`) will be copied from `$WSAMIDIR/build/class/NDSservice/Axis` to `$WSAMIDIR/dist/common/tomcat/shared/lib`.

If you want to update the OpenSLP library used on the client-side (only PC/Intel, not PDA) by the WSAMI Middleware:

- On Linux:
  - install OpenSLP on the directory specified in `$WSAMIDIR/build.xml` by the property `openslp.dir.linux`
  - build the library `${openslp.dir.linux}/libslp/.libs/libslp.so` (if it doesn't exist) following OpenSLP instructions
  - type “`ant distAxis`”. The files `${openslp.dir.linux}/libslp/.libs/libslp.so` and `$WSAMIDIR/build/class/NDSservice/Axis/libNDslp.so` will be copied to the directory `$WSAMIDIR/dist/tomcat/shared/lib`
- On Windows+Cygwin:
  - install OpenSLP on the directory specified in `$WSAMIDIR/build.xml` by the property `openslp.dir.Windows`
  - build the library `${openslp.dir.Windows}/win32/libslp/Release/slp.dll` (if it doesn't exist) following OpenSLP instructions
  - type “`ant distAxis`”. The files `$WSAMIDIR/build/class/NDSservice/Axis/slp.dll` and `${openslp.dir.Windows}/win32/libslp/Release/slp.dll` will be copied to the directory `$WSAMIDIR/dist/tomcat/shared/lib`

## 6. Core Broker Tools

<code>compileCBIFUtils_dll</code>	Generates shared library <code>ifacemgr.dll</code> (Windows, Intel)
<code>compileCBIFUtils_so</code>	Generates shared library <code>libifacemgr.so</code> (Linux, Intel)
<code>compileCBIFUtils_so_pda</code>	Generates shared library <code>libifacemgr.so</code> (Linux, ARM)

These targets allow to modify and rebuild the platform-dependent tools (power plugging, network interfaces, routing table) described in [ARCHGUIDE].

The source code is available in `$WSAMI/src/CoreBroker/IFUtils` both for Linux and for Windows (only for power plugging) and it includes the JNI code used to make calls from Java code to the shared libraries, some libraries and source code files used to compile the tools. After the execution of the ant targets, the shared library `libifacemgr.so` (of `ifacemgr.dll` for Windows) will be available in directory `$WSAMIDIR/dist/common/tomcatsharedlib` or in directory `$WSAMIDIR/dist/common/pdasharedlib`.

## **Bibliography**

### USERGUIDE

WSAMI Middleware User's guide

### ARCHGUIDE:

WSAMI Middleware Architecture guide

### CSOAPGUIDE

CSoap User Guide

### CSOAPARCH

CSoap Architecture Guide