PEPPOL
PAN-EUROPEAN PUBLIC PROCUREMENT ONLINE

cip competitiveness and innovation framework programme 2007-2013

Documentation

Project Acronym: PEPPOL
Grant Agreement number: 224974
Project Title: Pan-European Public Procurement Online

PEPPOL Transport Infrastructure

Metro Java Sample Implementation Developer Documentation, Installation and Configuration Manual

Version: 2.1.3
Status: In Use

Editors:
Jan Victoir, Difi/Alfa1lab

Project co-funded by the European Commission within the ICT Policy Support Programme		
Dissemination Level		
P	Public	X
C	Confidential, only for members of the consortium and the Commission Services	

cc BY

Revision History

Version	Date dd.mm.yyyy	Editor	Org	Description
2.00	25.09.2011	Jose Gorvenia	Alfa1lab	Documentation for the first version of the PEPPOL START Sample Implementation (replaced the PEPPOL Reference Implementation)
2.10	13.01.2012	Jan Victoir	Alfa1lab	New updated release of the PEPPOL START Sample Implementation, see release notes
2.1.1	01.02.2012	Jorge Reátegui	Alfa1lab	Updated SVN location for the Sample Implementation source code to point to new PEPPOL EIA structure. Updated references to the START specification in the release notes.
2.1.2	14.03.2012	Jorge Reátegui	Alfa1lab	New updated release of the PEPPOL START Sample Implementation, see release notes
2.1.3	02.04.2012	Jan Victoir / Jorge Reátegui	Alfa1lab	Updated release for PEPPOL, including enhancements and corrections, see release notes for details

Statement of originality

This deliverable contains original unpublished work except where clearly indicated otherwise. Acknowledgement of previously published material and of the work of others has been made through appropriate citation, quotation or both.

Statement of copyright

This deliverable is released under the terms of the **Creative Commons Licence** accessed through the following link: http://creativecommons.org/licenses/by/3.0/.

In short, it is free to
Share — to copy, distribute and transmit the work
Remix — to adapt the work
Under the following conditions
Attribution — You must attribute the work in the manner specified by the author or licensor (but not in any way that suggests that they endorse you or your use of the work).

Contributors

Organisations

Difi (Direktoratet for forvaltning og IKT), Norway, www.difi.no

BRZ (Bundesrechenzentrum), Austria, www.brz.gv.at

Alfa1lab, Denmark, www.alfa1lab.com

Persons

Jan Victoir, Difi/Alfa1lab

Marcelo Tataje, Difi/Alfa1lab

George Reátegui, Difi/Alfa1lab

Philip Helger, BRZ

José Gorvenia, Difi/Alfa1lab

Kenneth Bengtsson, Difi/Alfa1lab

Table of Contents

1 Introduction

1.1 Objective and Scope

This document provides the necessary steps to download configure and build the PEPPOL Access Point application. This manual is intended for anyone that intends to setup the sample application provided by PEPPOL.

1.2 Audience

The audience for this document is organizations in need for a short introduction to the PEPPOL BIS. These may include the following PEPPOL Stakeholders:

- PEPPOL Community Governance
- Contracting Authorities
- Economic Operators
- ICT Providers
- Service Providers

More specific it is the following roles:

- Business Experts
- ICT Architects
- ICT Developers
- ICT Governing participants

1.3 Glossary and Abbreviations

Term	Description
AP	Access Point
API	Application Program Interface
ASCII	American Standard Code for Information Interchange
CA	Certificate Authority
FQN	Fully Qualified Domain Name
GUI	Graphical User Interface
http	Hyper Text Transfer Protocol
https	Hyper Text Transfer Protocol Secure
IP	Internet Protocol
J2EE	Java 2 Enterprise Edition (trade mark of SUN Microsystems)
JAR	Java Archive Repository
JDK	Java Development Kit
OS	Operating System
PC	Personal Computer
POM	Project Object Model
RM	Reliable Messaging
SDK	Software Development Kit
SOAP	Simple Object Access Protocol
SSL	Secure Socket Layer

Term	Description
START	Trusted Asynchronous Reliable Transport
URL	Uniform Resource Locator
WAR	Web Application Archive
XML	Extensible Markup Language

1.4 References

In addition to this manual, other manuals to be consulted:

Document Reference	Use and/or URL
ANT	http://ant.apache.org/

1.5 Document Typographical Conventions

The following table explains the typographical conventions used in this manual.

Designations, symbols, fonts	Description
Italics	Used e.g. for names of windows, dialog boxes, menus, menu items, fields, functions and properties in the text. Example: Enter the value in the *Parameter* field.
`Courier`	Used for entries, selection options for field entries and displays. Example: Select `annoC017` in the *Parameter* field
Boldface	Used for other emphasized elements in the text.
Square brackets []	Used to identify command buttons. Example: Acknowledge this message by clicking on [OK].
<Variable/parameter>	Variable/parameter to be substituted by an actual value. Example: `setup <Java VRM> -nodisplay`

This paragraph contains a **note**. Notes provide a more detailed explanation of a matter or describe particular features or related topics that have no effect on the basic state of affairs.

This paragraph contains a **warning**! Warnings draw attention to particular characteristics or to precautions to be taken. Failing to heed such a warning will have effects on the basic state of affairs!

1.6 Release Notes

1.6.1 Changes

The following changes have been made since version 2.00:

- Merged set of properties files into one specific and complete config file
- Created a sample Main class for Sending a message to an Access Point
- Projects have been converted to Maven

CC BY

The following changes have been made since version 2.10

- Changed repository location for the source code and Documentation to reflect new PEPPOL EIA structure.
- Changed some references to the PEPPOL START specification in the Documentation.
- No changes in the source code have been made since version 2.10

The following changes have been made since version 2.11

- Simplified START client sending mechanism and removed dependencies from START server.
- Added support for WSDL callback handlers KeyStoreCallbackHandler and TrustStoreCallbackHandler.
- Changed service name to "accessPointService" to align with WSDL 2.0 example service name.
- Simplified structure of properties file.

The following changes have been made since version 2.12

- The build has been converted to a maven command line based tool, independent on the IDE. Remark that for convenience, within Netbeans one can import the POM.xml of the maven build and build from within the IDE.
- The Main class, entry for the client START, can be called from the command line.
- SSL class AccessPointX509TrustManager will be cleaned and left for proper implementation of the AP developer.

1.6.2 Bugs

Following bugs have been since version 2.00:

- Upgrade WSDL version 2.0
- Implemented a Constant for the authentication level at line 246 of the SAMLCallbackHandler.
- Upgraded Certificate/Metadata Validation. The common name is being validated now according START Specifications (Validation – Page 15).
- Implemented Certificate/Metadata Validation according to START Specifications (Validation – Page 15).
- Upgraded Certificate Reading Mechanism from SMP Public Metadata.
- Implemented Soap Fault Handling according START specifications (Faults – Page 12).
- Clean up for the General Code.

The following bugs have been solved in version 2.13

- The call to PING method (an optional component) has been removed from the code every time an attempt to send is performed. The validation of the certificate is performed within a customized tube as proposed by METRO.
- OCSP Validation included within the client.
- The redirection, as stated in the SMP has been implemented in the client AP.

1.6.3 Issues:

The following issues are known for the current version:

- The METRO library is closing the connection after 3,000 ms, indicated by the following statement in the log file: INFO: WSRM1157: Waiting for sequence [uuid:c03129fb-53ed-4672-82a0-2939e4dda1f3] state change to [CLOSED] has timed out after 3,000 milliseconds

  This is an informational warning message and the RM-METRO feature works as designed. In a production environment, WS-MakeConnection could be used as an alternative (depending on local architecture).

1.7 Software Solution and versions

A START Access Point (START AP) is an Access Point that implements the BUSDOX START profile (Secure Trusted Asynchronous Reliable Transport). The START code provided by PEPPOL covers the following:

- AccessPointService implementation (peppol-start-server)
- AccessPointClient implementation (peppol-start-client)
- Commons Transport Library (peppol-start-commons)

The START solution contains the following projects:

Project	Description
AccessPointService (peppol-start-server)	Contains the core types for building START Access Points and Clients, as well as the WSDL and Schemas used to create the client and server objects and deploy the application.
AccessPointClient (peppol-start-client)	Contains the implementation of the START Access Point that has been deployed in the PEPPOL infrastructure. Holds the logic needed for communicating with an Access Point.
Commons Transport Library (peppol-start-commons)	Contains intern logic to manage the document sending process.

Since the project is based upon maven, there is a general POM.xml (Project Object Model) available which will include the build of the three dependent projects.

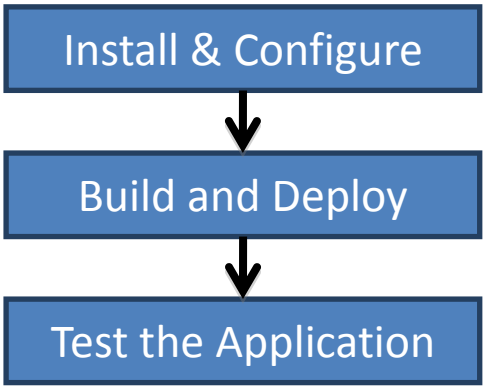
The START AP is delivered with the following software versions:

Software Component	**Version**
AP Server (peppol-start-server)	2.1.3
AP Client (peppol-start-client)	2.1.3
AP Commons (peppol-start-commons)	2.1.3

External dependencies are mentioned in the table below:

Software Component	**Version**
Tomcat	7.0.23
METRO libraries	2.1.1
Java JDK/JRE	1.7
Netbeans IDE *(optional since maven)*	7.0.0 (or higher)
Maven	2.2.1
Java	1.7.0_02

The procedure to get the application up and running is depicted in the next figure:



2 The Apache Tomcat Web Server

This section will describe how tio setup the Tomcat Web server for the application to be deployed.

2.1 Download Tomcat and METRO

The Tomcat web server can be downloaded from the repository http://tomcat.apache.org/. Select the version for your OS.

The METRO libraries can be downloaded from http://metro.java.net/ (within the section download).

2.2 Install Tomcat and METRO

Install tomcat in your destination directory.

Example: to install tomcat in the /opt file system, exexute the following commands:

```
opt> gzunip apache-tomcat-7.0.23.tar.gz
opt> tar -xvf apache-tomcat-7.0.23.tar
Result:
apache-tomcat-7.0.23/bin/catalina.sh
apache-tomcat-7.0.23/bin/configtest.sh
apache-tomcat-7.0.23/bin/daemon.sh
apache-tomcat-7.0.23/bin/digest.sh
... ommited output lines....
apache-tomcat-7.0.23/webapps/manager/images/void.gif
apache-tomcat-7.0.23/webapps/manager/index.jsp
apache-tomcat-7.0.23/webapps/manager/status.xsd
apache-tomcat-7.0.23/webapps/manager/xform.xsl
```

Install the METRO libraries within the tomcat web server. One can use ant (refer to [ANT] for detailed information) or install the METRO libraries manually. The procedure is documented at http://metro.java.net/2.1.1/. In the next example, the libraries are installed invoking thr ant command.

```
jan@jan-ubuntu:~/metro$ ant -Dtomcat.home=/opt/apache-tomcat-7.0.23 -f
./metro/metro-on-tomcat.xml install
Unable to locate tools.jar. Expected to find it in /usr/lib/jvm/java-6-
openjdk/lib/tools.jar
Buildfile: /home/jan/metro/metro/metro-on-tomcat.xml
catalinahome-test:
init:
uninstall:
update-catalina-props:
test-api:
check-api:
```

```
install-api-tomcat5:
install-api-tomcat6:
    [mkdir] Created dir: /opt/apache-tomcat-7.0.23/endorsed
     [copy] Copying 1 file to /opt/apache-tomcat-7.0.23/endorsed
install-api-tomcat:
install:
     [echo] Installing Metro 2.1.1 for /opt/apache-tomcat-7.0.23 ...
    [mkdir] Created dir: /opt/apache-tomcat-7.0.23/shared/lib
     [copy] Copying 4 files to /opt/apache-tomcat-7.0.23/shared/lib
     [echo] ... installation complete.
BUILD SUCCESSFUL
Total time: 2 seconds
```

Check that in your installation of tomcat, the directories endorsed and shared have been created with the METRO libraries.

```
jan@jan-ubuntu:/opt/apache-tomcat-7.0.23$ ls -ltr endorsed/
total 252
-rw-r--r-- 1 jan jan 254458 2012-01-12 09:58 webservices-api.jar
```

And:

```
jan@jan-ubuntu:/opt/apache-tomcat-7.0.23$ ls -ltR shared
shared:
total 4
drwxr-xr-x 2 jan jan 4096 2012-01-12 09:58 lib
shared/lib:
total 18328
-rw-r--r-- 1 jan jan   842477 2012-01-12 09:58 webservices-extra.jar
-rw-r--r-- 1 jan jan   146216 2012-01-12 09:58 webservices-extra-api.jar
-rw-r--r-- 1 jan jan 14014133 2012-01-12 09:58 webservices-rt.jar
-rw-r--r-- 1 jan jan  3758324 2012-01-12 09:58 webservices-tools.jar
```

2.3 Additional Configuration

If necessary, define the JAVA_HOME variable to reflect the path of your JDK installation. The parameter can be set in the catalina.sh (for linux based OS) or catalina.bat (for windows based OS). The next example will define the value for linux OS:

Example:

JAVA_HOME=/opt/jdk1.7.0_02

export JAVA_HOME

The port on which secure traffic will be allowed within the tomcat web server is 8443 by default. Define the port 8443 with the keyStoreFile (location of the vertificate file), keyStorePass (corresponding password) of your company in the server.xml file (conf directory of the installation of tomcat).

Example:

```
<Connector port="8443" protocol="HTTP/1.1" SSLEnabled="true"
           maxThreads="150" scheme="https" secure="true"
           keystoreFile="/opt/apache-tomcat-7.0.23/cert/tomcat-ap.p12"
           keystorePass="12345" keystoreType="PKCS12"
           clientAuth="false" sslProtocol="TLS" />
```

2.4 Testing Tomcat

Start the tomcat server and check the catalina.out (logs directory of the tomcat server) for errors.

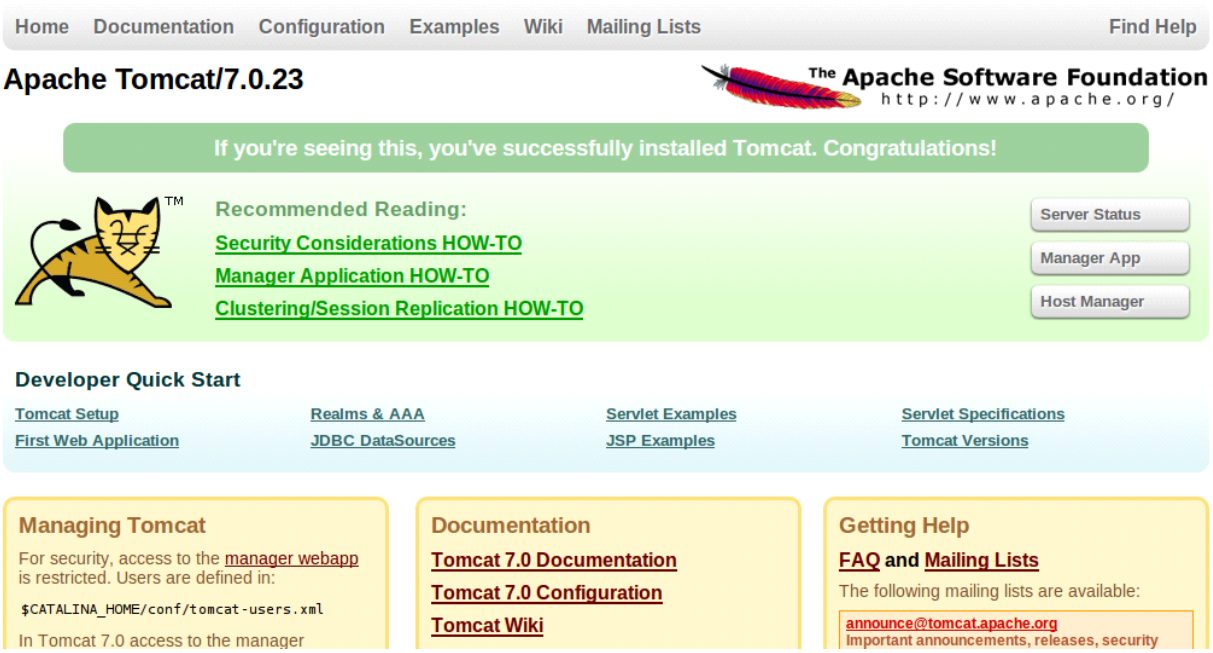
Connect to the URLs to verify your installation.

Example (linux):

```
jan@jan-ubuntu:/opt/apache-tomcat-7.0.23/bin$ ./startup.sh
Using CATALINA_BASE:   /opt/apache-tomcat-7.0.23
Using CATALINA_HOME:   /opt/apache-tomcat-7.0.23
Using CATALINA_TMPDIR: /opt/apache-tomcat-7.0.23/temp
Using JRE_HOME:        /opt/jdk1.7.0_02
Using CLASSPATH:       /opt/apache-tomcat-7.0.23/bin/bootstrap.jar:/opt/apache-
tomcat-7.0.23/bin/tomcat-juli.jar
```

Example:

Connect to http://localhost:8080 and https://localhost:8443 . One must be able to connect to the server and be able to show the next screen:


Home Documentation Configuration Examples Wiki Mailing Lists Find Help
Apache Tomcat/7.0.23
The Apache Software Foundation
http://www.apache.org/
If you're seeing this, you've successfully installed Tomcat. Congratulations!
Recommended Reading:
Security Considerations HOW-TO
Manager Application HOW-TO
Clustering/Session Replication HOW-TO
Server Status
Manager App
Host Manager
Developer Quick Start
Tomcat Setup
First Web Application
Realms & AAA
JDBC DataSources
Servlet Examples
JSP Examples
Servlet Specifications
Tomcat Versions
Managing Tomcat
For security, access to the manager webapp is restricted. Users are defined in:
$CATALINA_HOME/conf/tomcat-users.xml
In Tomcat 7.0 access to the manager
Documentation
Tomcat 7.0 Documentation
Tomcat 7.0 Configuration
Tomcat Wiki
Getting Help
FAQ and Mailing Lists
The following mailing lists are available:
announce@tomcat.apache.org
Important announcements, releases, security

3 PEPPOL certificate configuration

This section will describe which part of the software uses the certificates, enabling you to install them accordingly.

The following table provides an overview, where to use what certificates/key pairs for configuring the PEPPOL OSS components:

	PKI Keystore	PKI Truststore	SSL Keystore	SSL Truststore
Access Point	Your AP Key Pair	Root Certificate Generic AP CA Certificate		
Access Point Host Server			Any Key Pair	Any matching Certificate
SMP	Your SMP Key Pair	Root Certificate Generic AP CA Certificate		
SML Client	Your SMP Key Pair	Root Certificate Generic AP CA Certificate		Thawte Root Certicicate
SML Server	Your SML Key Pair	Root Certificate Generic AP CA Certificate	Any valid SSL Key Pair	

When a Key Pair is referenced, it means **Certificate + Private Key** together. If Certificate is referenced, it means **Public Key** only

* The SML server is centrally operated, so normally you shouldn't care about this

Alternatively the following table shows which key to use in which component:

	AP Private Key	AP Certificate	AP CA Certificate	SMP Private Key	SMP Certificate	SML Private Key	SML Certificate	SMP CA Certificate	Root Certificate	SSL Private Key	SSL Certificate
AP Keystore	X	X									
AP Truststore			X						X		
AP WebApp Host										X[1]	X[1]
SMP Keystore				X	X						
SMP Truststore								X	X		
SML Client Keystore				X	X						
SML Client Truststore								X	X		X[2]
SML Server Keystore						X	X			X[2]	
SML Server Truststore								X	X		

CC BY

*: *This is the web application server (e.g. Tomcat, JBoss etc.) that is hosting your access point service.*

1: The Accesspoint must be reachable on https. The certificate can be self-signed.

2: The SML certificate is currently issued by Thawte. The Thwate root-certificate should be in your operating system root certificate store.

3: A Sample **Truststore** has been included in the source code and can be used for testing purposes. The **truststore** in mention can be found in the **certs** folder of the **peppol-start-client** and **peppol start-server** project.

*Use the following configuration:

```
name: sample-truststore.jks
password: peppol
alias:ca
```

4 The Access Point

This section will describe how to download the software, incorporate the changes to reflect your local configuration, build the software and deploy the application.

4.1 The AP source files

The source files of the AP application are stored into the next subversion repository: https://joinup.ec.europa.eu/svn/peppol/PEPPOL_EIA/1-ICT_Architecture/1-ICT-Transport_Infrastructure/14-ICT-Services-Components/ICT-Transport-Java_Metro_AP_SW-213/.

Within the Netbeans IDE, download the source files via the menu „Team“, than choose „Checkout“ and fill in the URL with the link mentioned above.

One can also donwnload the files directly with svn as follows (command line):

```
svn co  https://joinup.ec.europa.eu/svn/peppol/PEPPOL_EIA/1-ICT_Architecture/1-ICT-Transport_Infrastructure/14-ICT-Services-Components/ICT-Transport-Java_Metro_AP_SW-213/
```

It assumes you have the JKS certificate files generated for your local configuration.

4.2 Configuration

In this section, the configuration of the AP will be handled.

4.2.1 Step 1: configuration file

At the root where the source files where downloaded, the start-ap-props.properties will contain the settings for the AP. The file contains the following properties:

```
ap.host=[YOUR_IP]
ap.port=[YOUR_PORT]
ap.name=[YOUR_AP_NAME]
server.keystore=[YOUR_KEYSTORE_FILE]
server.Keystore.password=[YOUR_KEYSTORE_PASS]
server.keystore.alias=[YOUR_KEYSTORE_ALIAS]
server.truststore=[YOUR_TRUSTSTORE_FILE]
server.truststore.password=[YOUR_TRUSTSTORE_PASS]
server.truststore.alias=[YOUR_TRUSTSTORE_ALIAS]
ap.logFile=[YOUR_LOG_PATH]
ap.inboxPath=[YOUR_INBOX_PATH]

document.scheme=[YOUR_DOC_ID_SCHEME]
process.scheme=[YOUR_PROCESS_ID_SCHEME]
participant.scheme=[YOUR_PARTICIPANT_ID_SCHEME]
```

Where [YOUR_IP] represents the IP or FQN where the AP will de deployed and [YOUR_PORT] the port (SSL) where the AP application will be listening for incoming requests.
The parameter [YOUR_AP_NAME] represents the name of the application, which will be set (in normal cases, except when you change the code to point elsewhere) to "*accessPointService*".
The parameter [YOUR_KEYSTORE_FILE] indicates where the keystore file is located, the corrersponding [YOUR_KEYSTORE_PASS] is the password for the keystore file and [YOUR_KEYSTORE_ALIAS] indicates the alias used within the file.
The parameter [YOUR_TRUSTSTORE_FILE] indicates where the trsutstore file is located, the corrersponding [YOUR_TRUSTSTORE_PASS] is the password for the truststore file and [YOUR_TRUSTSTORE_ALIAS] indicates the alias used within the file.
The parameter [YOUR_LOG_PATH] indicates the path where the tomcat server can store the client and the server logging.
The parameter [YOUR_INBOX_PATH] indicates the location where the incoming documents will be stored.

The AP application deployed within the Tomcat server should have write permissions on the YOUR_INBOX_PATH directory for the messages to be stored.

The parameter [YOUR_DOC_ID_SCHEME] represents your document scheme.
The parameter [YOUR_PROCESS_ID_SCHEME] represents your process scheme.
The parameter [YOUR_PARTICIPANT_ID_SCHEME] represents your participant scheme.

An example file is shown below:

```
ap.host=9.36.66.12
ap.port=8443
ap.name=accessPointService
server.keystore=/home/app/certs/my-ap-keystore.jks
server.Keystore.password=mypassword
server.keystore.alias=1
server.truststore=/home/app/certs/my-ap-truststore.jks
server.truststore.password=mypassword
server.truststore.alias=ca
ap.logFile=/home/app/logs
ap.inboxPath=/home/app/etc/peppol/userfolder

document.scheme=busdox-docid-qns
process.scheme=cenbii-procid-ubl
participant.scheme=iso6523-actorid-upis
```

4.2.2 Step 2: OCSP Validation

To use the OCSP validation, uncomment in the following files (Starting from the ROOT directory where you downloades the source files)

```
./peppol-start-server/src/wsdl/peppol-start-2.0.wsdl
./peppol-start-server/src/main/webapp/WEB-INF/wsdl/peppol-start-2.0.wsdl
./peppol-start-client/src/wsdl/peppol-start-2.0.wsdl
./peppol-start-client/src/peppol-start-2.0.xml
```

If you want to use the OCSP Validator uncomment the next section:

```
<sc:ValidatorConfiguration sc:revocationEnabled="true"
wspp:visibility="private">
           <sc:Validator
classname="org.busdox.transport.start.ocsp.Validator"
name="certificateValidator"/>
      </sc:ValidatorConfiguration>
```

4.3 Build the AP application

In this section, the build process for the AP application will be described.

The build process will generate a `ROOT.war` file in the target directory of the peppol-start-server project. The project can be build by simply executing the maven command as shown in the next example.

It supposes that you have installed maven and the JAVA_HOME points to the correct home directory of JAVA mentioned within the document. One can check with the following command:

```
jan@efact1:~/peppol$ mvn -version

Apache Maven 2.2.1 (rdebian-1)
Java version: 1.7.0_02
Java home: /usr/lib/jvm/java-7-oracle/jdk1.7.0_02/jre
Default locale: en_US, platform encoding: UTF-8
OS name: "linux" version: "2.6.35.4-rscloud" arch:
"amd64" Family: "unix"
```

The output of the build is as follows (irrelevant lines are omitted to shorten the output)

```
jan@efact1:~/peppol$ mvn

[INFO] Scanning for projects...
[INFO] Reactor build order:
[INFO]   apconfig-plugin Maven Mojo
[INFO]   Unnamed - eu.peppol:peppol-start-sample-application:pom:2.1.3
[INFO]   peppol-start-commons
[INFO]   peppol-start-client
[INFO]   peppol-start-server
[INFO] ------------------------------------------------------------------------
```

```
[INFO] Building apconfig-plugin Maven Mojo
[INFO]    task-segment: [install]
[INFO] ------------------------------------------------------------------------
Downloading: http://repo1.maven.org/maven2/org/apache/maven/plugins/maven-
plugin-plugin/2.5/maven-plugin-plugin-2.5.pom
...
[INFO] Installing /home/jan/peppol/apconfig-plugin/target/apconfig-plugin-1.0-
SNAPSHOT.jar to
/home/jan/.m2/repository/org/sonatype/mavenbook/plugins/apconfig-plugin/1.0-
SNAPSHOT/apconfig-plugin-1.0-SNAPSHOT.jar
[INFO] ------------------------------------------------------------------------
[INFO] Building Unnamed - eu.peppol:peppol-start-sample-application:pom:2.1.3
[INFO]    task-segment: [install]
[INFO] ------------------------------------------------------------------------
[INFO] [site:attach-descriptor {execution: default-attach-descriptor}]
[INFO] [install:install {execution: default-install}]
[INFO] Installing /home/jan/peppol/pom.xml to
/home/jan/.m2/repository/eu/peppol/peppol-start-sample-application/2.1.3/peppol-
start-sample-application-2.1.3.pom
[INFO] ------------------------------------------------------------------------
[INFO] Building peppol-start-commons
[INFO]    task-segment: [install]
[INFO] ------------------------------------------------------------------------
...
[INFO] Parsing input schema(s)...
[INFO] Compiling input schema(s)...
...
commons-2.1.3.jar to /home/jan/.m2/repository/eu/peppol/peppol-start-
commons/2.1.3/peppol-start-commons-2.1.3.jar
[INFO] ------------------------------------------------------------------------
[INFO] Building peppol-start-client
[INFO]    task-segment: [install]
[INFO] ------------------------------------------------------------------------
[INFO] [apconfig:ap-config {execution: default}]
[INFO] Properties files loaded at: /home/jan/peppol/start-ap-props.properties
...
[INFO] jaxws:wsimport args: [-s, /home/jan/peppol/peppol-start-
client/target/generated-sources/jaxws-wsimport, -d, /home/jan/peppol/peppol-
start-client/target/classes, -verbose, -catalog, /home/jan/peppol/peppol-start-
client/src/jax-ws-catalog.xml, -wsdllocation, file:/home/jan/peppol/peppol-
start-client/target/src/wsdl/peppol-start-2.0.wsdl, -extension, -Xnocompile,
/home/jan/peppol/peppol-start-client/src/wsdl/peppol-start-2.0.wsdl]
parsing WSDL...

Using java.util.ServiceLoader
[WARNING] ignoring SOAP port "ResourceBindingPort": unrecognized transport. try
running wsimport with -extension switch.
  line 208 of file:/home/jan/peppol/peppol-start-client/src/wsdl/peppol-start-
2.0.wsdl

generating code...

org/w3/_2009/_02/ws_tra/AccessPointService.java
...
org/busdox/_2010/_02/channel/fault/StartException.java
...
[INFO] [antrun:run {execution: default}]
[INFO] Executing tasks
     [echo] ====== Building again: [project.artifactId] peppol-start-client
```

```
[INFO] Executed tasks
[INFO] [resources:testResources {execution: default-testResources}]
...
 [INFO] [install:install {execution: default-install}]
[INFO] Installing /home/jan/peppol/peppol-start-client/target/peppol-start-
client-2.1.3.jar to /home/jan/.m2/repository/eu/peppol/peppol-start-
client/2.1.3/peppol-start-client-2.1.3.jar
[INFO] ------------------------------------------------------------------------
[INFO] Building peppol-start-server
[INFO]    task-segment: [install]
[INFO] ------------------------------------------------------------------------
...
generating code...

org/w3/_2009/_02/ws_tra/AccessPointService.java
...
org/busdox/_2010/_02/channel/fault/package-info.java
...
[INFO] Tests are skipped.
[INFO] [war:war {execution: default-war}]
[INFO] Packaging webapp
[INFO] Assembling webapp [peppol-start-server] in [/home/jan/peppol/peppol-
start-server/target/ROOT]
[INFO] Processing war project
[INFO] Copying webapp webResources [/home/jan/peppol/peppol-start-server/src] to
[/home/jan/peppol/peppol-start-server/target/ROOT]
[INFO] Copying webapp resources [/home/jan/peppol/peppol-start-
server/src/main/webapp]
[INFO] Webapp assembled in [176 msecs]
[INFO] Building war: /home/jan/peppol/peppol-start-server/target/ROOT.war
[INFO] WEB-INF/web.xml already added, skipping
[INFO] [install:install {execution: default-install}]
[INFO] Installing /home/jan/peppol/peppol-start-server/target/ROOT.war to
/home/jan/.m2/repository/eu/peppol/peppol-start-server/2.1.3/peppol-start-
server-2.1.3.war
 [INFO]
[INFO] ------------------------------------------------------------------------
[INFO] Reactor Summary:
[INFO] ------------------------------------------------------------------------
[INFO] apconfig-plugin Maven Mojo ........................... SUCCESS
[1:15.457s]
[INFO] Unnamed - eu.peppol:peppol-start-sample-application:pom:2.1.3  SUCCESS
[23.441s]
[INFO] peppol-start-commons ............................. SUCCESS [28.635s]
[INFO] peppol-start-client .............................. SUCCESS [1:16.637s]
[INFO] peppol-start-server .............................. SUCCESS [7.695s]
[INFO] ------------------------------------------------------------------------
[INFO] ------------------------------------------------------------------------
[INFO] BUILD SUCCESSFUL
[INFO] ------------------------------------------------------------------------
[INFO] Total time: 3 minutes 32 seconds
[INFO] Finished at: Tue Apr 03 16:57:12 UTC 2012
[INFO] Final Memory: 82M/226M
[INFO] ------------------------------------------------------------------------
```

The duration of the build depends the first time on the connection speed to download the maven repository dependency files.

5 Deployment of the Access Point

In this section the deployment of the AP application will be described. In the previous section, we mentioned that the outcome of the build process resulted into the “ROOT.war” file.

5.1 Step1: Stop the Tomcat Web Server

Stop the tomcat application server.

Example (linux OS):

```
jan@jan-ubuntu:/opt/apache-tomcat-7.0.23/bin$ ./shutdown.sh
Using CATALINA_BASE:   /opt/apache-tomcat-7.0.23
Using CATALINA_HOME:   /opt/apache-tomcat-7.0.23
Using CATALINA_TMPDIR: /opt/apache-tomcat-7.0.23/temp
Using JRE_HOME:        /opt/jdk1.7.0_02
Using CLASSPATH:       /opt/apache-tomcat-7.0.23/bin/bootstrap.jar:/opt/apache-
tomcat-7.0.23/bin/tomcat-juli.jar
```

5.2 Step 2: Clean the Tomcat webapps directory

`Within /opt/ apache-tomcat-7.0.23/webapps` (exmple on linux OS), delete the ROOT directory and if present the ROOT.war file.

5.3 Step 3: Copy the ROOT.war file

Copy the ROOT.war file into the webapps direcory of the Tomcat web server.

5.4 Step 4: Start the Tomcat Web Server

Start the tomcat application server.

Example (linux OS):

```
 jan@jan-ubuntu:/opt/apache-tomcat-7.0.23/bin$ ./startup.sh
Using CATALINA_BASE:   /opt/apache-tomcat-7.0.23
Using CATALINA_HOME:   /opt/apache-tomcat-7.0.23
Using CATALINA_TMPDIR: /opt/apache-tomcat-7.0.23/temp
Using JRE_HOME:        /opt/jdk1.7.0_02
Using CLASSPATH:       /opt/apache-tomcat-7.0.23/bin/bootstrap.jar:/opt/apache-
tomcat-7.0.23/bin/tomcat-juli.jar
```

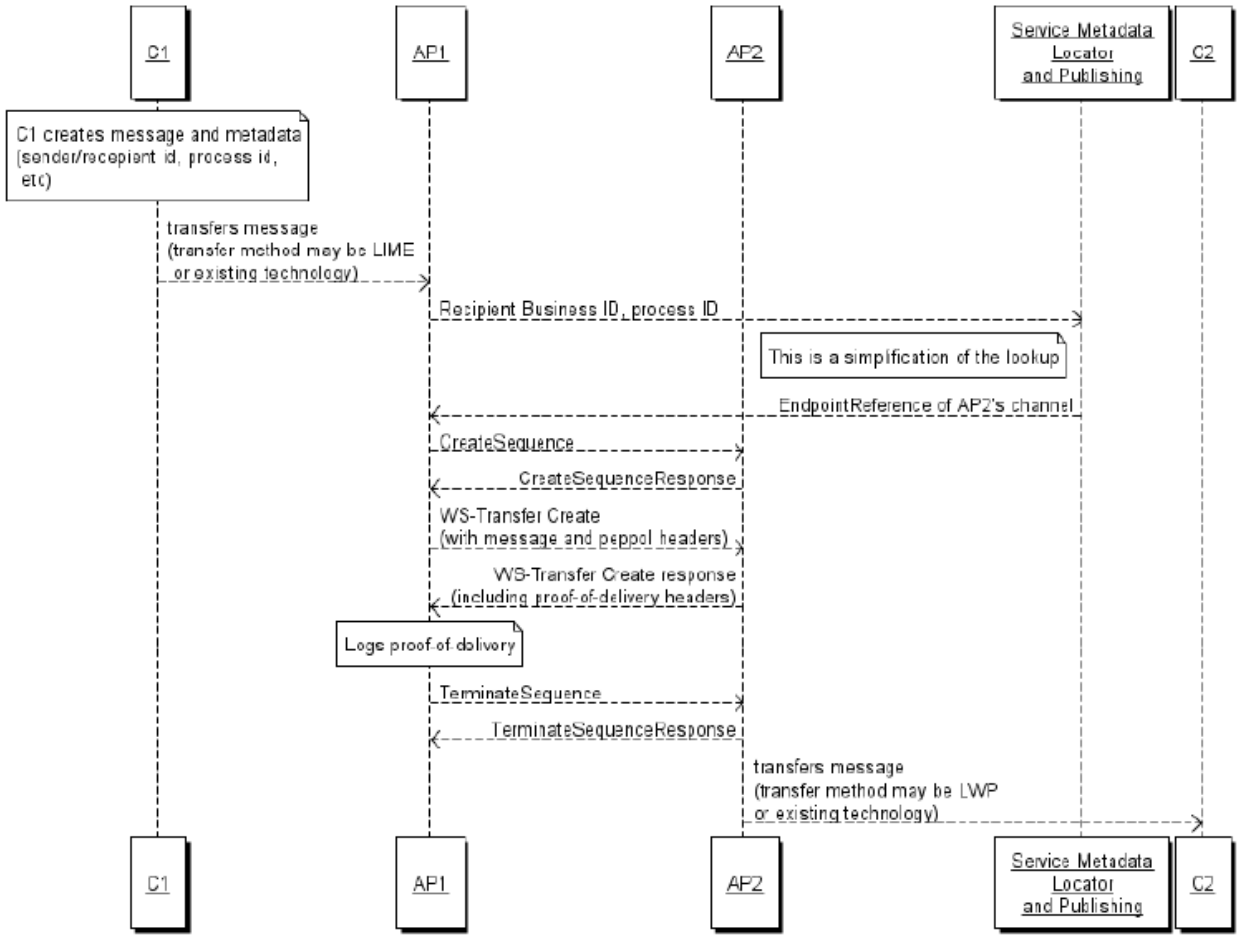
At this point, the AP application is ready to be used.

6 Testing

In this section, a general description of how the AP application can be tested will be provided. To

6.1 Message Flow

The message flow between the various components in the software layering is depicted in the next figure:



6.2 Testing the application

To test the AP application, find the *peppol-start-client-2.1.3.jar* in the *target* folder of the Client project; finally run the Java command with the following parameters:

```
[0] = Sender Participant Identifier
[1] = Receiver Participant Identifier
[2] = Document Type
[3] = Process Type
[4] = File path of the message
```

CC BY

Example (Windows Seven OS):

```
java -jar peppol-start-client-2.1.3.jar 9902:DK12345678 9902:DK87654321
urn:oasis:names:specification:ubl:schema:xsd:Invoice-
2::Invoice##urn:www.cenbii.eu:transaction:biicoretrdm010:ver1.0:#urn:www.peppol.
eu:bis:peppol4a:ver1.0::2.0 urn:www.cenbii.eu:profile:bii04:ver1.0
C:\SubmitOrder.xml
```

7 Trouble Shooting

In this section, we shall discuss the solution to some common problems you might encounter when installing the AP.

Keystore Files

If tomcat is unable to find your keystore, following exception (catalina.out) will be shown. Make sure the files are available (in this example: /opt/apache-tomcat-7.0.23/cert/tomcat-ap.p12)

```
SEVERE: Failed to load keystore type PKCS12 with path /opt/apache-tomcat-7.0.23/cert/tomcat-ap.p12 due to /opt/apache-tomcat-7.0.23/cert/tomcat-ap.p12 (No such file or directory)
java.io.FileNotFoundException: /opt/apache-tomcat-7.0.23/cert/tomcat-ap.p12 (No such file or directory)
        at java.io.FileInputStream.open(Native Method)
        at java.io.FileInputStream.<init>(FileInputStream.java:138)
        at org.apache.tomcat.util.net.jsse.JSSESocketFactory.getStore(JSSESocketFactory.java:400)
        at org.apache.tomcat.util.net.jsse.JSSESocketFactory.getKeystore(JSSESocketFactory.java:306)
        at org.apache.tomcat.util.net.jsse.JSSESocketFactory.getKeyManagers(JSSESocketFactory.java:565)
        at org.apache.tomcat.util.net.jsse.JSSESocketFactory.getKeyManagers(JSSESocketFactory.java:505)
        at org.apache.tomcat.util.net.jsse.JSSESocketFactory.init(JSSESocketFactory.java:449)
        at org.apache.tomcat.util.net.jsse.JSSESocketFactory.createSocket(JSSESocketFactory.java:158)
        at org.apache.tomcat.util.net.JIoEndpoint.bind(JIoEndpoint.java:369)
        at org.apache.tomcat.util.net.AbstractEndpoint.init(AbstractEndpoint.java:553)
        at org.apache.coyote.AbstractProtocol.init(AbstractProtocol.java:369)
        at org.apache.coyote.http11.AbstractHttp11JsseProtocol.init(AbstractHttp11JsseProtocol.java:119)
        at org.apache.catalina.connector.Connector.initInternal(Connector.java:937)
        at org.apache.catalina.util.LifecycleBase.init(LifecycleBase.java:102)
        at org.apache.catalina.core.StandardService.initInternal(StandardService.java:559)
        at org.apache.catalina.util.LifecycleBase.init(LifecycleBase.java:102)
        at org.apache.catalina.core.StandardServer.initInternal(StandardServer.java:781)
        at org.apache.catalina.util.LifecycleBase.init(LifecycleBase.java:102)
        at org.apache.catalina.startup.Catalina.load(Catalina.java:573)
        at org.apache.catalina.startup.Catalina.load(Catalina.java:598)
        at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method)
        at sun.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:57)
        at sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:43)
        at java.lang.reflect.Method.invoke(Method.java:601)
        at org.apache.catalina.startup.Bootstrap.load(Bootstrap.java:281)
        at org.apache.catalina.startup.Bootstrap.main(Bootstrap.java:449)
```

Message Storage

If tomcat is unable to store the messages, one must verify the permissions on the directory where the messages are to be stored.

Build errors

If the build process returns an error, of might be that the option „skip tests“ is not enabled in your IDE. Make sure that tests are skipped when building the application.

If you are working with NetBeans under Windows, and get a ***premature end of file exception***, solve this by changing the location of the folder **“wsdl“**. (Copy/Paste it in a location outside the boundaries of the project).

Wsit-client

Because the assembles and configurations files (java generated sources and metro config files) for “START AccessPoint Java implementation” project made the WSDL for that binding (according START Profile) to require both “assymmetricBinding” and “transportBinding” assertions to be generated and this is not supported by JAX-WS framework, WS-SecurityPolicy requires METRO. This is the reason you need to specify an "wsit-client" in the “peppol-start-client” because if you remove it, you will get issues related to WS-Addressing, WS-Security and WSRM included in embedded policies.