PEPPOL
PAN-EUROPEAN PUBLIC PROCUREMENT ONLINE

cip competitiveness and innovation framework programme 2007-2013

Guideline

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

Transport Infrastructure
ICT
Services-Components

PEPPOL-Silicone
How to deploy

Version: 2.2.1
Status: In use

Editors:
Philip Helger, PEPPOL.AT

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	

Revision History

Version	Date	Editor	Org	Description
2.21	2012-04-11	PH	BRZ	Initial version

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This deliverable contains original unpublished work except where clearly indicated otherwise.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

Contributors

Organisations

PEPPOL.AT/BRZ (Bundesrechenzentrum)[1], Austria, http://www.brz.gv.at/
IBX/Cap Gemini
dig.at

Persons

Philip Helger, BRZ
Alexandru Pislaru (Cap Gemini)
Dieter Dobersberger (dig.at)

[1] English: Austrian Federal Computing Centre

Table of Contents

1 Introduction

1.1 Objective and Scope

This document is the introduction on how to deploy and run components of the PEPPOL Silicone v2.2.1 package.

1.2 Audience

The audience for this document is organizations in need for a short introduction to the PEPPOL Silicone deployment process. 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:

- ICT Architects
- ICT Developers
- ICT Governing participants

2 Quick setup for deployment

If you are an experienced developer and only want to run the PEPPOL Silicone components you may use the following quick setup rules. For details see the following chapters.

1. Download and install the Sun Java JDK
    a. Set the environment variable `JAVA_HOME` to the base path of the Java installation
2. Download and install the latest Apache Tomcat
    a. Set the `CATALINA_HOME` environment variable
    b. Edit the file `%CATALINA_HOME%/conf/tomcat-users.xml` files
    c. Remove all example web applications
3. Download and install Apache Ant
    a. Set the `ANT_HOME` environment variable
    b. Append the `%ANT_HOME%/bin` directory to the `PATH` environment variable
4. Download and install Metro 2.1.1
    a. Set the `METRO_HOME` environment variable
    b. Install Metro into Tomcat: `ant -f Dtomcat.home=%CATALINA_HOME% -f %METRO_HOME%\metro-on-tomcat.xml install`
5. Download the latest PEPPOL Silicone binary distribution
6. If you want to run the SMP, adapt the SMP configuration – see the respective developer guide
7. If you want to run the START AP, adapt the START AP configuration – see the respective developer guide
8. If you want to run the LIME AP, adapt the LIME AP configuration – see the respective developer guide (work in progress)

3 Extended setup guideline

3.1 Sun JDK

You must have a JDK 1.6.x installed. This software has not yet been extensively tested with JDK 1.7.x but is assumed to work. The latest Sun JDK (1.6.0_31 at the time of writing) can be downloaded from the website http://www.oracle.com/technetwork/java/javase/downloads/index.html
For Linux distributions OpenJDK 1.6.x should also work, but has not been extensively tested.

3.2 Application server

Most PEPPOL users use Tomcat as their application server of choice but it has also been tested with Jetty 7.x. To grab the latest Tomcat 6.x or 7.x visit http://tomcat.apache.org/ and download the matching package. Afterwards set the environment variable `CATALINA_HOME` to the installation base directory.
To be able to use the Tomcat manager web application you must modify the `%CATALINA_HOME%/conf/tomcat-users.xml` file as described in http://tomcat.apache.org/tomcat-6.0-doc/manager-howto.html#Configuring_Manager_Application_Access (Tomcat 6.x) or http://tomcat.apache.org/tomcat-7.0-doc/manager-howto.html#Configuring_Manager_Application_Access (Tomcat 7.x).

3.3 Metro

The software provided by this project makes heavy use of the Java Metro libraries. Currently the version Metro 2.1.1 from http://metro.dev.java.net/2.1.1 is required. Download the ZIP file and extract it locally to perform further steps. The rest of the configuration depends on the application server you are using.

Important Note: PEPPOL Silicone has not yet been tested with Metro >= 2.2 which was released on February 20th, 2012!

3.3.1 Automatic setup for Apache Tomcat

Metro comes with an Ant[2] script that installs itself into Tomcat. If you haven't installed Ant, install Ant and ensure that the `ANT_HOME` environment variable is correctly set and that `%ANT_HOME%/bin` is added to the `PATH` environment variable. Also the `METRO_HOME` environment variable must be set to the Metro base directory. The main script call than looks like this:

```
ant -f Dtomcat.home=%CATALINA_HOME% -f %METRO_HOME%\metro-on-tomcat.xml install
```

3.3.2 Manual setup for Apache Tomcat 6.x and 7.x

Copy the following files (without the directory structure of the Metro ZIP file) into your Tomcat endorsed directory (you may need to create this directory; it is e.g. `%CATALINA_HOME%/endorsed`):

- metro/lib/webservices-api.jar

Alternatively you can also copy the `webservice-api.jar` into the endorsed directory of your JRE but beware of the side effects as this affects all applications running on this JRE and this is probably not intended.
Copy the following files (without the directory structure of the Metro ZIP file) into your Tomcat lib directory (`%CATALINA_HOME%/lib`):

- metro/lib/stax-api.jar
- metro/lib/webservices-extra.jar
- metro/lib/webservices-extra-api.jar
- metro/lib/webservices-rt.jar
- metro/lib/webservices-tools.jar

[2] Apachae Ant: http://ant.apache.org/

Note: ensure that the webservices-api.jar is **only** contained in the `endorsed` directory and not in the `lib` directory as well!

Ensure that the endorsed libraries are loaded by Tomcat:

- Define the following system property on Tomcat start: `java.endorsed.dirs=/usr/share/tomcat/endorsed` (see the Tomcat 6 Classloader HOW-TO or the Tomcat 7 Classloader HOW-TO)
- For Tomcat 7 the parameter itself might by already present in the `/etc/init.d/tomcat7` file (in case you are using Linux) - just the endorsed directory itself might be missing

If you get start-up errors with Tomcat 7, maybe this post helps you solve the issues. The easiest thing is to remove all predefined non-PEPPOL web applications (like ROOT, examples, docs etc.) which are using Servlet API 3.0 in their `web.xml` files. If you can't do this, follow the instructions in the post (add `metadata-complete="true"` in the web.xml to all applications using Servlet API 3.0). Finally Tomcat must be restarted to make the changes work.

3.3.3 Setup for Jetty 7.x

Metro is handled as a separate "option" in Jetty. Assuming your Jetty installation is in `/opt/jetty` you need to do the following:

- Create a new directory `/opt/jetty/lib/metro`

Copy the following files into the created directory (`/opt/jetty/lib/metro`)

- metro/lib/webservices-rt.jar

Than modify `/opt/jetty/start.ini` and add `metro` to the `OPTIONS` property:

```
OPTIONS=Server,jsp,jmx,resources,websocket,ext,plus,annotations,metro
```

Finally a Jetty restart is required.

Important note: the name of the directory created under `/opt/jetty/lib` must match the name of the option you add!

3.4 Deployment

3.4.1 LIME

Especially for the LIME server it is recommended to not deploy the web application as a WAR file, because by default all data is stored inside the unpacked web application. So instead just create the appropriate context directory manually and copy the compiled web application inside. On update please be careful not to delete the stored documents!

3.4.2 Metro

For Metro compatibility ensure that none of the following files resides in any of your web applications `WEB-INF/lib` directory because they are provided by the application server:

- servlet-api.jar
- jsp-api.jar
- stax-api.jar
- webservices-api.jar
- webservices-extra.jar
- webservices-extra-api.jar

- webservices-rt.jar
- webservices-tools.jar

==> That's the reason why all `org.glassfish.metro` artefacts in the project `pom.xml` files are marked with the scope `provided`.

4 Using Apache httpd

Apache httpd[3] is very often used in front of an application server for easier SSL handling and more flexible security configuration. There are several possibilities of connecting httpd with an application server, which are outlined in the following sections.
All the configuration examples in the following sections assume that you are running an SMP and a START AP on the same Tomcat instance that is not clustered and running on port 8080. The httpd configuration file is based on httpd v2.2.

The following configuration file snippets are only meant as a guideline and must be modified to match your local requirements. Please contact your local server administrator to ensure that your adopted configuration works with your environment and that all local security requirements are matched.
The given explanations to the configuration are meant for an easy understanding, and may not be complete or outlining all potential constraints and implications.

4.1 mod_proxy

When using mod_proxy, you are simply proxying all requests to a certain URL at the backend – in this case a Tomcat. See http://httpd.apache.org/docs/2.2/mod/mod_proxy.html for details of the configuration.
The following configuration file snippet shows one way how to configure an httpd virtual host on port 80 for an SMP using mod_proxy:

```
<VirtualHost 10.0.0.1:80>
  ServerAdmin webmaster@example.com
  ServerName peppol-smp.example.com

  DocumentRoot /data1/www/peppol-smp.example.com/htdocs
  <Directory />
    Options FollowSymLinks
    AllowOverride None
  </Directory>
  <Directory /data1/www/peppol-smp.example.com/htdocs/>
    Options Indexes FollowSymLinks MultiViews
    AllowOverride None
    Order allow,deny
    allow from all
  </Directory>

  ErrorLog /data1/www/peppol-smp.example.com/logs/error.log
  LogLevel warn
  CustomLog /data1/www/peppol-smp.example.com/logs/access.log combined

  <Proxy *>
    Order Allow,Deny
    Allow From All
  </Proxy>
```

[3] Apache httpd: http://httpd.apache.org/

```
  ProxyPass /accessPointService !
  ProxyPass /manager !
  ProxyPass / http://localhost:8080/
  ProxyPassReverse / http://localhost:8080/
</VirtualHost>
```

- `<VirtualHost 10.0.0.1:80>` assuming that the local IP address of your machine is 10.0.0.1 this indicates a host running on port 80 (default http port)
- `ServerAdmin webmaster@example.com` defines the email address of the webmaster to be displayed in error messages (if enabled)
- `ServerName peppol-smp.example.com` defines the public domain name to which this server applies
- `DocumentRoot /data1/www/peppol-smp.example.com/htdocs` defines the document root directory for static resources. Must point to an existing directory.
- `<Directory ...>` the following two directives specify access rights on the folders
- `ErrorLog /data1/www/peppol-smp.example.com/logs/error.log` defines the filename where errors should be logged. Please ensure that the directory exists
- `LogLevel warn` the mininum log level to log
- `CustomLog /data1/www/peppol-smp.example.com/logs/access.log combined` defines the filename where accesses are logged
- `<Proxy *>` defines access rules for the proxy configuration
- `ProxyPass /accessPointService !` means that the START AP accessPointService should not be proxied on port 80 (all URLs starting with /accessPointService). Note: this must of course match your START AP context name.
- `ProxyPass /manager !` means that the Tomcat Manager (mangement UI) should not be proxied on port 80
- `ProxyPass / http://localhost:8080/` means that all other incoming requests (”/”) should be proxied to the application server running on port 8080
- `ProxyPassReverse / http://localhost:8080/` means that all responses (answers to incoming requests) should be send back to the requestor at the base URL (”/”)

The following configuration file snippet shows one way how to configure an httpd virtual host on port 443 for a START AP using mod_proxy:

```
<VirtualHost 10.0.0.1:443>
  ServerAdmin webmaster@example.com
  ServerName peppol-ap.example.com

  DocumentRoot /data1/www/peppol-ap.example.com/ssl-htdocs
  <Directory />
    Options FollowSymLinks
    AllowOverride None
  </Directory>
  <Directory /data1/www/peppol-ap.example.com/ssl-htdocs/>
    Options Indexes FollowSymLinks MultiViews
    AllowOverride None
    Order allow,deny
    allow from all
```

```
  </Directory>

  ErrorLog /data1/www/peppol-ap.example.com/logs/ssl_error.log
  LogLevel warn
  CustomLog /data1/www/peppol-ap.example.com/logs/ssl_access.log combined

  SSLEngine on
  SSLCertificateFile /etc/apache2/ssl/peppol-ap.example.com.crt
  SSLCertificateKeyFile /etc/apache2/ssl/peppol-ap.example.com.key
  SSLCertificateChainFile /etc/apache2/ssl/CA.crt
  SetEnvIf User-Agent ".*MSIE.*" ssl-unclean-shutdown
  SSLCipherSuite ALL:!ADH:!EXPORT56:RC4+RSA:+HIGH:+MEDIUM:!LOW:+SSLv2:+EXP

  <Proxy *>
    Order Allow,Deny
    Allow From All
  </Proxy>
  ProxyPreserveHost on

  ProxyPass /manager/html http://localhost:8080/manager/html
  ProxyPassReverse /manager/html http://localhost:8080/manager/html

  ProxyPass /accessPointService http://localhost:8080/accessPointService
  ProxyPassReverse /accessPointService http://localhost:8080/accessPointService

  <Location /manager/html>
    Order Deny,Allow
    Deny from all
    Allow From 10.0.0.100
  </Location>
</VirtualHost>
```

- `<VirtualHost 10.0.0.1:443>` assuming that the local IP address of your machine is 10.0.0.1 this indicates a host running on port 443 (default https port)
- `ServerAdmin`, `ServerName`, `DocumentRoot`, `<Directory>`, `ErrorLog`, `LogLevel`, `CustomLog` and `<Proxy>` have been explained in the previous example.
- `SSLEngine on` enables SSL/TLS for that specific virtual host
- `SSLCertificateFile /etc/apache2/ssl/peppol-ap.example.com.crt` points to the PEM-encoded Certificate file for the server
- `SSLCertificateKeyFile /etc/apache2/ssl/peppol-ap.example.com.key` points to the PEM-encoded Private Key file for the server
- `SSLCertificateChainFile /etc/apache2/ssl/CA.crt` sets the optional *all-in-one* file where you can assemble the certificates of Certification Authorities (CA) which form the certificate chain of the server certificate
- `SetEnvIf User-Agent ".*MSIE.*" ssl-unclean-shutdown` workaround for some versions of Internet Explorer (see FAQ entry at http://httpd.apache.org/docs/2.2/ssl/ssl_faq.html#aboutssl)
- `SSLCipherSuite ALL:!ADH:!EXPORT56:RC4+RSA:+HIGH:+MEDIUM:!LOW:+SSLv2:+EXP` this complex directive uses a colon-separated *cipher-spec* string consisting of OpenSSL cipher

specifications to configure the Cipher Suite the client is permitted to negotiate in the SSL handshake phase.

- `ProxyPreserveHost on` will pass the Host: line from the incoming request to the proxied host, instead of the hostname specified in the `ProxyPass` line.
- `ProxyPass /manager/html http://localhost:8080/manager/html` ensures that requests to the Tomcat Manager are only available via https
- `ProxyPassReverse /manager/html http://localhost:8080/manager/html` send the responses back to the requestor
- `ProxyPass /accessPointService http://localhost:8080/accessPointService` ensures that the START AP is only accessed via https
- `ProxyPassReverse /accessPointService http://localhost:8080/accessPointService` send the responses back to the requestor
- `<Location /manager/html>` defines that the Tomcat manager can only be accessed from a certain IP address

4.2 mod_jk

"mod_jk" has a double unescaping issue when used with the SMP (because SMP URLs regularly contain colon characters). This issue can be solved be replacing the property `+ForwardURICompat` with `+ForwardURICompatUnparsed +RejectUnsafeURI`. See http://tomcat.apache.org/connectors-doc/reference/printer/apache.html for details.