Reference Implementation

Project Acronym: PEPPOL

Grant Agreement number: 224974

Project Title: Pan-European Public Procurement Online

PEPPOL eSignature Infrastructure
XKMS Requester
Developer's Guide

Revision: 1.0

Authors:

France: Julien Pasquier (Lex Persona)

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

Revision	Date	Author	Organisation	Description
1.0	2011/02/28	Julien PASQUIER	Lex Persona	First 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This deliverable contains original unpublished work except where clearly indicated otherwise.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.

Table of Contents

1 Introduction

1.1 Objective

This document provides the documentation of the open source reference implementation PEPPOL XKMS Requester. It is written for developers that integrate and use the reference implementation to send XKMS requests to a PEPPOL XKMS Responder to validate X.509 certificates.

1.2 Description

The PEPPOL XKMS Requester is a Java API which requires a JRE 1.5 or greater for correct execution. It integrates also a simple command line application which enables users to test the API to validate X.509 certificates with remote XKMS responders.

The deliverable contains the following directories:

dist: contains the distribution files. Note that the XKMS Requester itself is contained in the XKMSRequester.jar file.

docs: contains the Javadoc documentation.

lib: contains the Java libraries used by the XKMS Requester.

resources: contains the files used to automatically generate the JAXB classes.

sources: contains the Java source code.

test: contains the XKMS Requester Command Line application and the configuration file (config.properties).

1.3 XKMS Requester Version

This documentation refers to the version 1.0.0 of the PEPPOL XKMS Requester implementation, released in February 2011 by Lex Persona.

The PEPPOL XKMS Requester is based on the version 1.95 of the *D1.3 Part 5 XKMS v2 Interface Specification*.

2 Presentation of the API

The following presentation of the API is didactic and describes the 3 steps which are required to validate an X.509 certificate (or a certificate path) with an XKMS Responder using the XKMS Requester API.

For an exhaustive example of the XKMS Requester API, see the **`eu.peppol.lsp.xkms.requester.cmd.XKMSRequesterCommand`** class which is contained in the "sources" directory.

For more details regarding the XKMS Requester API, see the Javadoc which is contained in the "docs/javadoc" directory.

2.1 Initializing the XKMSRequester

The following section describes how to initialize the XKMSRequester.

1- Create a new XKMSRequester object with the URL of the XKMS Responder to be used.

```
XKMSRequester xkmsRequester = new XKMSRequester(wsEndpointURL);
```

2- Set the X.509 certificate which is used by the XKMS Responder to sign XKMS responses. This certificate enables the requester to verify signature of the received XKMS responses.

```
xkmsRequester.setResponderCertificate(responderCert);
```

3- If the XKMS Responder is contacted with an HTTPS URL, you must set the SSL certificate public key of the XKMS server.

```
xkmsRequester.setSSLServerPublicKey(sslServerCert.getPublicKey());
```

4- If you want to sign XKMS requests that are sent to the responder, you must set the signer's private key, the signer's certificate, the digest algorithm (if null, SHA-1 will be used) and the JCE Signature provider (if null, the default provider will be used).

```
xkmsRequester.setRequestSignatureParams(
    signerPrivateKey, signerCert, null, null);
```

5- You can activate (or desactivate) the logs. By default, the logs are not activated.

```
xkmsRequester.setLogging(true, logDirectory);
```

6- You can specify the proxy server to be used.

```
xkmsRequester.setProxy(proxyHost, proxyPort);
```

2.2 Create and send the ValidateRequest

The following section describes how to create and send the request.

1- Create a new ValidateRequest object with the service name and the certificate path to be validated. Note that the certificate path can only contain the certificate to be validated.

```
ValidateRequest validateRequest = new ValidateRequest(service, certPath);
```

2- Add the "RespondWith" elements (the types of data the recipient requests to be sent in the response) to the XKMS request.

```
request.addRespondWith(PEPPOLConstants.RESPONDWITH_PEPPOL_EXTENDED);
request.addRespondWith(PEPPOLConstants.RESPONDWITH_EID_QUALITY);
request.addRespondWith(PEPPOLConstants.RESPONDWITH_VALIDATION_DETAILS);
request.addRespondWith(PEPPOLConstants.RESPONDWITH_TSL_SERVICE_INFORMATION);
request.addRespondWith(PEPPOLConstants.RESPONDWITH_OCSP);
```

3- You can specify the validation time which will be used by the responder to check the certificate validity.

```
validateRequest.setValidationTime(validationTime);
```

4- You can specify the usage of the certificate to be validated.

```
validateRequest.setSignatureKeyUsage(true);
validateRequest.setEncryptionKeyUsage(false);
validateRequest.setExchangeKeyUsage(false);
```

5- Send the validation request with the XKMSRequestor object.

```
ValidateResult validateResult = xkmsRequester.validate(validateRequest);
```

If the request cannot be sent to the responder, an XKMSRequesterException exception is thrown else the result is returned in a ValidateResult object.

2.3 Process the ValidateResult

The ValidateResult contains the following methods:

- String getId()

  Returns the Id attribute value of the result.

- String getService()

Returns the service name.

- byte[] getNonce()

  Returns the nonce value of the result.

- String getRequestId()

  Returns unique identifier Id specified in the request.

- ResultMajor getResultMajor()

  Returns the most significant component of the result code.

- ResultMinor getResultMinor()

  Returns the least significant component of the result code.

- List<KeyBindings> getKeyBindings()

  Returns the KeyBinding elements.

- PEPPOLValidateResultExt getPeppolValidateResultExt()

  Returns the extended validation information defined by PEPPOL. For more information, please consult the Javadoc of the eu.peppol.lsp.xkms.extensions.PEPPOLValidateResultExt class.