Import Private Key and SSL Certificate into Java Keystore (JKS)

BACKGROUND:

Apache Tomcat and many other Java applications expect to retrieve SSL/TLS certificates from a Java Keystore (JKS).

Keytool application (shipped as part of Java) helps you to:

- Create a new JKS with a new Private Key.
- Generate a Certificate Signing Request (CSR) for the Private Key in this JKS.
- Import a Certificate that you received for this CSR into your JKS.

However, Keytool does not let you directly import a pre-existing Private Key for which you already have a Certificate. So you need to do this yourself, here's how:

The solution is to convert your existing Certificate and Private Key into a PKCS12 file, and then use the keytool functionality to merge one keystore with another one.

PURPOSE:

The purpose of this document is to provide you with steps to import RSA Private Key and x509 SSL Certificate into Java Keystore.

SCOPE:

This document covers step-by-step procedure for importing RSA Private Key and SSL Certificate into a Java Keystore, in such a way that it is ready for use with Apache Tomcat.

This document does NOT cover installation and configuration of base components like Apache Web Server (httpd), JDK/JRE, Apache Tomcat, OpenSSL etc.

AUDIENCE:

All personnel from Implementation Team, Seclore Support Team and Amazon Maintenance Team are the primary target audience for this document.

Any person having reasonable experience of Web Server technology and its components, can easily follow and understand instructions in this document and achieve the desired outcome as a result.

PRE-REQUISITES:

1. Java Development Kit (JDK) **-OR-** Java Runtime Environment (JRE) version 6 update 27 or later.
2. JDK **or** JRE **“bin”** folder added to **PATH** Environment Variable.
3. 32-bit / 64-bit OpenSSL for Windows (v0.9.8y / v1.0.1c / v1.0.1e).
4. OpenSSL **“bin”** folder added to **PATH** Environment Variable.

DOCUMENT SUBMISSION DETAILS

Date	4 January 2014
Classification	Restricted Internal Use
Document Type	Knowledge Base Article
Submitted To	
Designation	
Address	
Contact Number	
E-Mail	

DOCUMENT DISTRIBUTION LIST

Sr. No.	Name	Organization	Purpose
1	PSG Team (Anand Chorera)	Seclore	Document Preparation
2	Manjul Kubde	Seclore	Document Review
3	Manjul Kubde	Seclore	Document Approval
4			
5			

DOCUMENT REVISION HISTORY

Revision	Date	Name	Description
1.0	18 November 2013	Anand Chorera	Initial Draft
1.1	4 January 2014	Anand Chorera	Revision as per Reviewer’s comments

Step-1: Create a *.crt file* containing Intermediate and TrustedRoot (CACerts Bundle) Certificates

1. Create a folder and collect all your certificates in one place.

   Intermediate (CA_Intermediate.crt), Root (TrustedRoot.crt), and Primary Certificates (your_domain_name.crt).

2. Open a text editor (such as Notepad++ OR Wordpad) and paste the entire body of each certificate into one text file in the following order:

   a. The Intermediate Certificate – **CA_Intermediate.crt**

   b. The Root Certificate - **TrustedRoot.crt**

   Make sure to include the beginning and end tags on each certificate. The result should look like this:

```
-----BEGIN CERTIFICATE-----
(Your Intermediate certificate: CA_Intermediate.crt)
-----END CERTIFICATE-----
-----BEGIN CERTIFICATE-----
(Your Root certificate: TrustedRoot.crt)
-----END CERTIFICATE-----
```

   Save the combined file as **CACerts.crt**. This **.crt file** is now ready to use in next step.

Step-2: Convert SSL Certificate and Private Key to PKCS12 format using OpenSSL

```
openssl pkcs12 -export -in Cert.crt -inkey PrivKey.key -certfile CACerts.crt -name tomcat -out
keystore.p12 -passout pass:Seclore@123
```

Note: Make sure you put a password on the **.p12 file** - otherwise you'll get a null reference exception when you try to import it in next step.

Step-3: Convert the PKCS12 (.p12 file) to a JAVA KeyStore (using KeyTool)

```
keytool -importkeystore -deststorepass Seclore@123 -destkeypass Seclore@123 -destkeystore ps.keystore
-srckeystore keystore.p12 -srcstoretype PKCS12 -srcstorepass Seclore@123 -alias tomcat
```

This will create a Java KeyStore file named **ps.keystore** and it can be used in Tomcat with the following details:

Alias: tomcat

Password: Seclore@123

References:

http://www.digicert.com/ssl-support/pem-ssl-creation.htm

http://stackoverflow.com/questions/906402/importing-an-existing-x509-certificate-and-private-key-in-java-keystore-to-use-i

http://cunning.sharp.fm/2008/06/importing_private_keys_into_a.html

http://www.cammckenzie.com/blog/index.php/2013/01/14/import-private-key-and-certificate-into-java-keystore/