Generating Java KeyStore (JKS) file – Script Process

### Version: 1.3

### Document ID: Generating Java KeyStore (JKS) file

### Document Owner: ConnectingOntario

#### Copyright Notice

Copyright © 2016, eHealth Ontario

#### All rights reserved

No part of this document may be reproduced in any form, including photocopying or transmission electronically to any computer, without prior written consent of eHealth Ontario. The information contained in this document is proprietary to eHealth Ontario and may not be used or disclosed except as expressly authorized in writing by eHealth Ontario.

#### Trademarks

Other product names mentioned in this document may be trademarks or registered trademarks of their respective companies and are hereby acknowledged.

## Document Control

The electronic version of this document is recognized as the only valid version.

#### Approval History

|  |  |  |
| --- | --- | --- |
| APPROVER(S) | TITLE/DEPARTMENT | APPROVED DATE |
|  |  | YYYY-MM-DD |
|  |  | YYYY-MM-DD |

#### Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| VERSION NO. | DATE | SUMMARY OF CHANGE | CHANGED BY |
| 1.0 | 2016-10-26 | Initial release | ConnectingOntario Program |
| 1.1 | 2017-05-31 | Updates to align with new scripts (V6) | ConnectingOntario Program |
| 1.2 | 2017-12-15 | Updates to align with new scripts (V8) | ConnectingOntario Program |
| 1.3 | 2018-05-01 | Updates to align with new scripts (V9) | ConnectingOntario Program |

## Document ID

Generating Java KeyStore (JKS) file

Table of Contents

[Version: 1.3 1](#_Toc512945538)

[Document ID: Generating Java KeyStore (JKS) file 1](#_Toc512945539)

[Document Owner: ConnectingOntario 1](#_Toc512945540)

[Document Control ii](#_Toc512945541)

[Document ID ii](#_Toc512945542)

[Overview 1](#_Toc512945543)

[Description 2](#_Toc512945544)

[Prerequisites 3](#_Toc512945545)

[Certificate Request & Installation Process 3](#_Toc512945546)

[Instructions 4](#_Toc512945547)

[Configure Certificates 4](#_Toc512945548)

[Copy the undersigned site certificate into certs directory 4](#_Toc512945549)

[Import Undersigned Site Certificate 4](#_Toc512945550)

[For Windows based Integration Engines 6](#_Toc512945551)

[Appendix A: Keystore instructions for sites using the HL7 over HTTP (HAPI) relay software 9](#_Toc512945552)

[Locate Installation Directory 9](#_Toc512945553)

[Locate keystore file and keystore password 10](#_Toc512945554)

[Sample copy of a config.xml file (mapped to the conformance test environment) 13](#_Toc512945555)

[Appendix B: Changing the Password (i.e. Storepass) associated to the Private Key 14](#_Toc512945556)

# Overview

Connection to ConnectingOntario infrastructure for contributing sites is secured via mutually authenticated Secured Sockets Layer (SSL) connection. For mutual authentication to work, both client and server present certificates to each other, validating them against a list of ‘trusted’ certificates. Within ConnectingOntario, PKI certificates are undersigned by the eHealth Ontario Certification Authority; eHealth issues the client (e.g. hospital) a unique PKI certificate as well as a copy of the eHealth Ontario Root CA which must both be installed in the trusted store in order for the client to authenticate.

Before a PKI certificate can be issued to the contributing site, they must first generate a Certificate Signing Request (CSR) file to eHealth Ontario.

This document will provide the steps to generate the CSR file using the custom batch scripts developed by eHealth Ontario that helps simplify and standardize this process across all contributing sites involved in the process.

# Description

The following batch scripts are provided in the package in the **private certs – v9.zip** file:

* + **exportToPKCS12.cmd**
    - Exports all keys stored in the current JKS store into PKCS12 format. Password will remain the same as for JKS store
  + **generateCSR.cmd**
    - generates CSR request to get a new certificate from eHealth Ontario or renew existing one
  + **importRootCA.cmd**
    - imports root CA provided by eHealth Ontario; keystore might contain more than one root CA certificates. This is quite normal, especially when the root CA is updated / changed by the provider.
  + **importSignedCert.cmd**
    - imports undersigned site certificate chain provided by eHealth Ontario. Certificate chain must correspond to the private key used to generate CSR
  + **manageKeystore.cmd**
    - provides basic operations to manage keystore (list keys, get details for stored certificate, rename or delete the keys); users do not need to execute this file
  + **settings-pk.cmd**
    - file with all configuration settings
  + **keystoreops.cmd**
    - library file with high level commands; users do not need to execute this file
  + **signCSR.cmd**
  + **settings-rootca.cmd**
  + **generateSiteKey.cmd**
  + **tmpl folder –** with the sha1sha2\_prod\_rootca.jks embedded in the folder
  + **changePassword.cmd –** use this file to change the storepass associated to your private key (.jks)

# Prerequisites

The site needs to fulfill the following prerequisites before attempting to generate the Java KeyStore (JKS) file:

1. Windows 7 or higher
2. Oracle Java v7 or higher
3. Received the following certificate files (.txt format) from eHealth Ontario after submitting the CSR file:
   * Undersigned Site Certificate
   * Two or more Root and intermediate CA certificate(s)

# Certificate Request & Installation Process



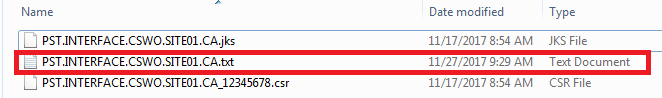
# Instructions

## Configure Certificates

* Once the site has successfully generated the CSR and sent it to eHealth Ontario, the site will be provided with the following certificate:
  + One **Undersigned site certificate** (public key)
* In order to get the interface working with the eHealth Ontario environments, you will need to execute the **ImportSignedCert.cmd** script

## Copy the undersigned site certificate into certs directory

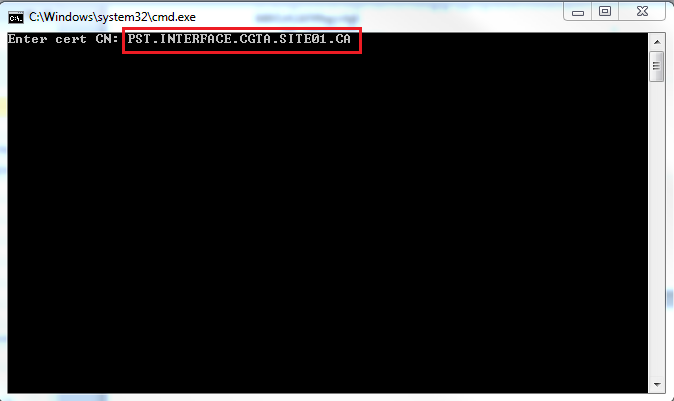
* Before you begin, place undersigned site certificate .txt files in the **Certs** folder. Make sure to use proper <CN> subdirectory corresponding to the CN of the cert you generated CSR for:



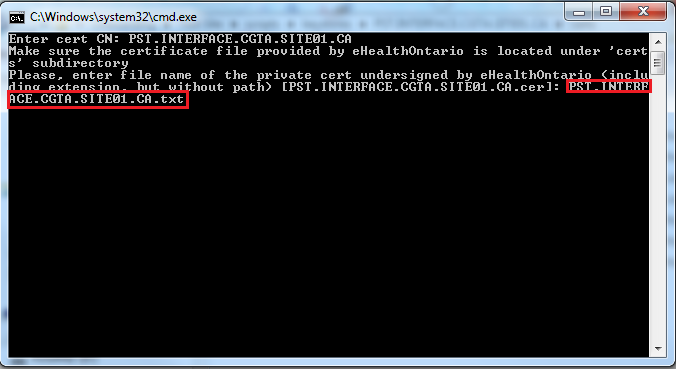
* In the example above, there is one undersigned site certificate (i.e. PST.INTERFACE.CGTA.SITE01.CA.txt).

## Import Undersigned Site Certificate

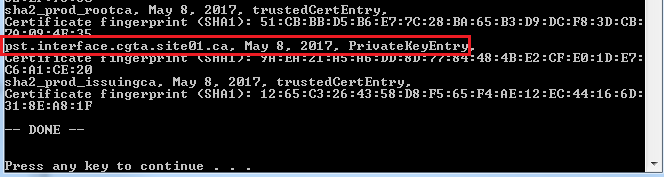
* Navigate to top level folder (i.e. where the scripts reside) and double click the **importSignedCert.cmd** script.
* You will be prompted for the cert CN. The exact CN value is required, and can be found in the confirmation email from eHealth Ontario after the CAF was submitted. Paste the cert CN value as show below and click ‘Enter’



* Once prompted, paste the file name of the undersigned site certificate (ensure the .txt extension is included).



* If execution was successful, you will see an entry corresponding to the certificate CN (i.e. pst.interface.cgta.site01.ca) with today’s date (i.e. script execution date).

****

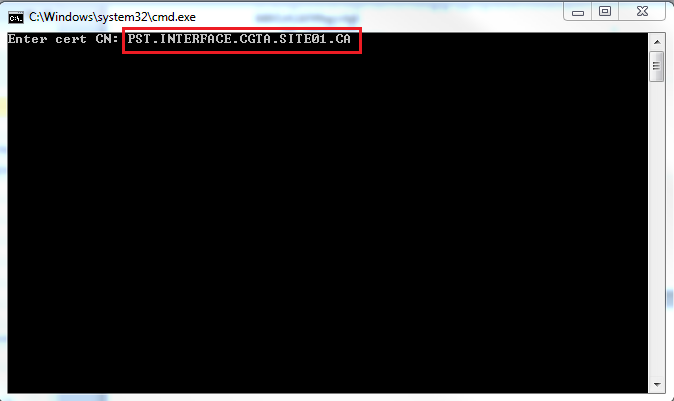
* The importSignedCert.log file will be generated under ‘logs’ directory. Review the content and compare it with the sample attached below (there should be no exceptions in the log file)



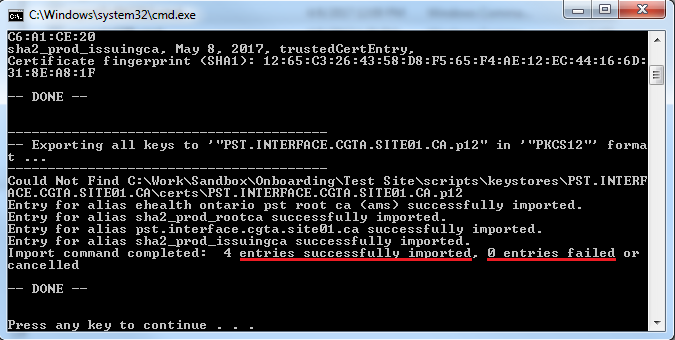
## For Windows based Integration Engines

If your certificate keystore requires a PKCS12 file format, follow these steps:

* Navigate to top level folder (i.e. where the scripts reside) and double click on the **exportToPKCS12.cmd** script
* You will be prompted for the cert CN. The exact CN value is required, and can be found in the confirmation email from eHealth Ontario after the CAF was submitted. Paste the cert CN value as show below and click ‘Enter’



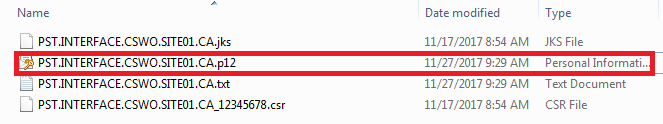
* Double click the **exportToPKCS12.cmd** script, and the following will be displayed:



* if you are using Java 7, you might see an errors saying that some entries failed to get exported. Java 7 does not support export of trusted keys (the ROOT CA certs imported earlier). Private key will still be exported correctly.
* Click the ‘enter’ key to close the command prompt window
* If your integration engine is using Microsoft certificate store to store the keys, you will need to import p12 file into MS Keystore. Contact vendor to make sure you are importing the key under correct branch of the MS Store.

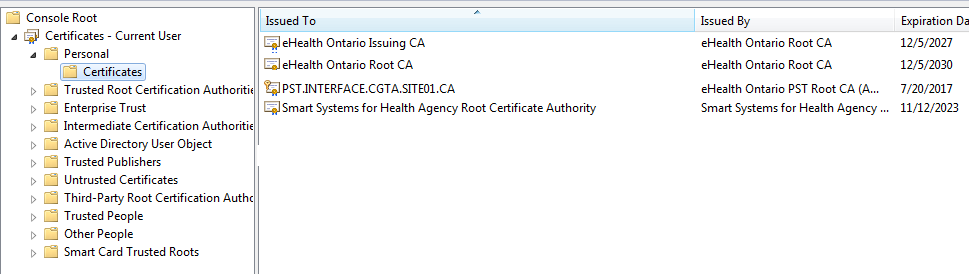
General steps are to import p12 file into MS Keystore are:

* Go to the **certs** directory and you will see the .p12 file created there



* Once imported, your keystore will have the private certificate and the Root/Intermediate CA certificates.
* If you are using Java 7, you will also need to import all of the Root CAs separately. The easiest way is to change extension of the SHAx\_PROD\_x.txt files from’ txt’ to ‘cer’ and then double click on the file. This would initiate the Wizard to import key into the keystore. Make sure you are importing it into the right branch. Contact your vendor if unsure.

**Note:** the default password associated to the .p12 file will be **changeme.**



* The exportToPKCS12.log file will be generated under **logs** directory. Review the content and compare it with the sample attached below (there should be no exceptions in the log file)



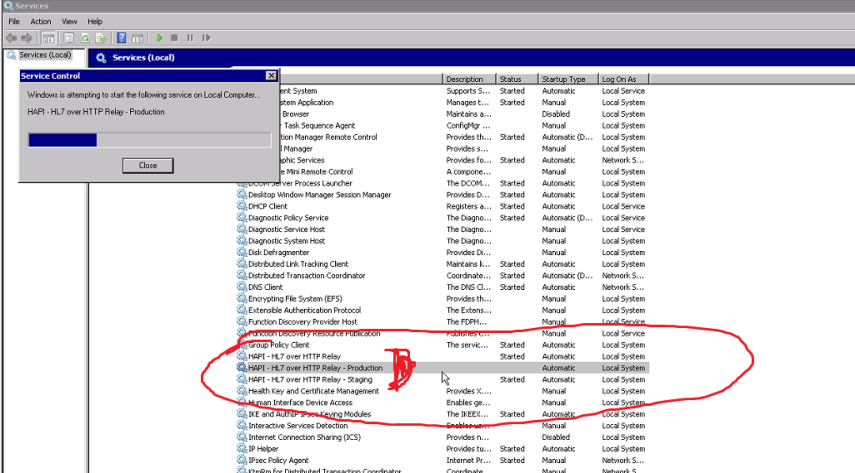
# Appendix A: Keystore instructions for sites using the HL7 over HTTP (HAPI) relay software

The location of keystore is specific to the integration software being used. This section provides details on locating keystore for ‘HAPI HL7 over HTTP Relay’ tool; other solutions will locate the keystore elsewhere. For existing installations, you would normally do this only once. After the keystore is located, it should be maintained in a separate directory with the scripts. For new installations (or when preservation of existing private key is not required), JKS file can be regenerated from scratch on every key renewal.

**Note:** This appendix can be ignored if your site does not use the HL7 over HTTP (HAPI) relay software.

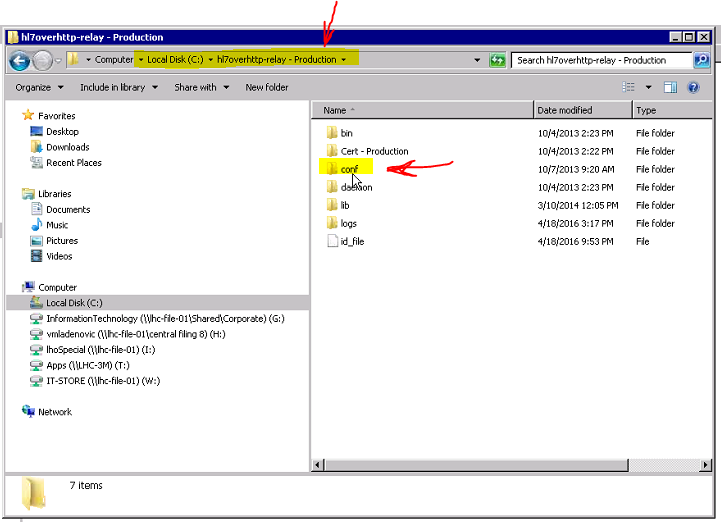
### Locate Installation Directory

The directory might be named as ‘hl7overhttp – relay’. If you are unsure about the location, check your windows services panel to locate the startup script home directory:

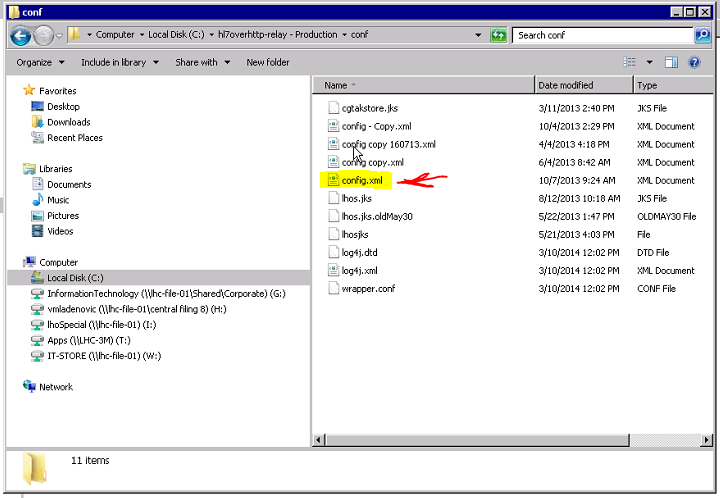


### Locate keystore file and keystore password

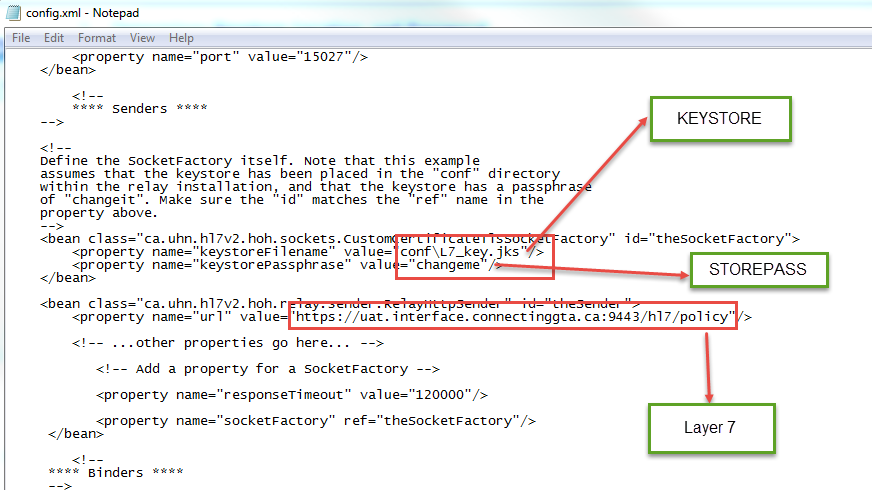
Within tool installation directory, locate configuration directory



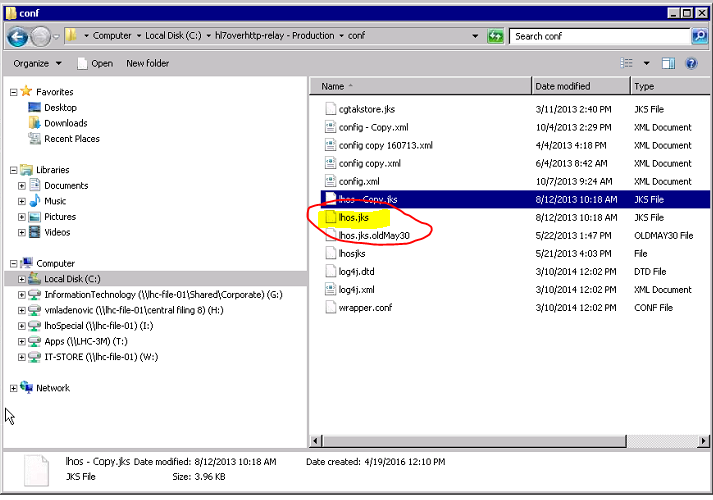
and open **config.xml** configuration file



In the configuration file, confirm ‘keystore’ filename and password to ensure you are updating the **correct** keystore



(keystore file path is relative to the startup directory of the HTTP Relay)



* Replace existing jks file with new one and update keystore password in the ‘config.xml’ file if necessary.

### Sample copy of a config.xml file (mapped to the conformance test environment)



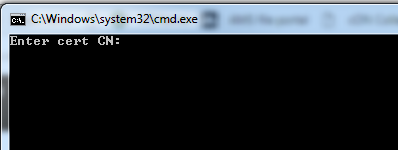
# Appendix B: Changing the Password (i.e. Storepass) associated to the Private Key

You have the option to change the password associated to the private key (i.e. jks file) after it has been generated by using the **changePassword.cmd** batch script.

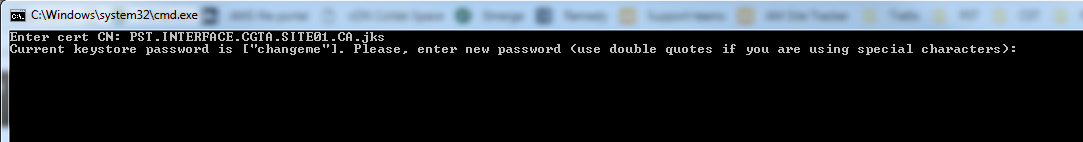
**Note:** This script can only be run after the CSR file has been generated.

To change the password, carry out these steps:

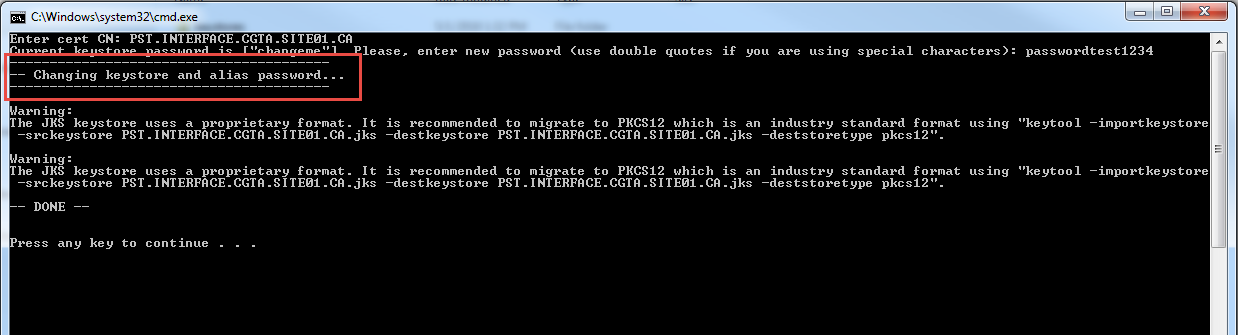
1. Click the **changePassword.cmd**
2. Enter the Certificate Common Name (CN) assigned to your site and click the ‘enter’ key



1. Enter the new password and click the ‘enter’ key



1. You will see a message of “Changing keystore and alias password” when the new password is accepted



1. The new password is stored in a file called **storepass.txt** and is located in the same directory as your private key 