Working With Keystores

On this page:

Keystore Types

There are two types of keystores whicha re supported by Java

- JKS - Native Java archive, to be deprecated in favor of PKCS#12 standard
- PKCS#12 - archive format containing multiple cryptographic objects (also referred to as PFX)

Tools

There are two tools with which are used when working with keystores and certificates

- keytool - command line tool, part of the java distribution, for manipulating keystores (JKS and PKCS#12)
- openssl - client tool for manipulating certificates in multiple formats

All of the required tasks can be accomplished with keytool, so we will limit the scope of keystore management to keytool.

Create a keystore

Create a keystore in PKCS#12 format - the command below will create a keystore with a self-signed certificate for the given server. Please note that in order to have a signed certificate, the common name of the certificate cannot be an IP address.

Also, please note that in this example, we are also creating 3 subject alternative names: 1 for the common name (fqdn), 1 for the host name (hostname), and one for the IP address of the server.

```
keytool -genkeypair -keyalg RSA -sigalg SHA256withRSA -keysize 2048 -validity 3650 -keystore <keystore.p12> -
storetype  pkcs12 -storepass <storepass> -alias <aliasname> -dname "CN=<fqdn>,OU=<Org Unit>,O=<Company Name>,
L=<City>,S=<State>,C=<Country>" -ext BasicConstraints:critical=ca:false -ext SAN=dns:<fqdn>,dns:<hostname>,ip:
<ip_address>
```

View contents of a keystore

```
keytool -list -v -keystore <keystore.p12> -storepass <storepass>
```

Create a CSR

Create a CSR from an existing keystore, adding the subject alternative names. The alias is that containing the entry of type **PrivateKeyEntry**

```
keytool -noprompt -certreq -keystore <keystore.p12> -storepass <storepass> -alias <aliasname> -file <server.
csr> -ext SAN=dns:<fqdn>,dns:<hostname>,ip:<ipaddress> -ext BasicConstraints:critical=ca:false
```

View contents of a CSR

```
keytool -printcertreq -file <server.csr>
```

Import signed certificate into keystore

A signed server certificate must be imported into the keystore from which the CSR was generated, and into the same alias.

```
keytool -importcert -trustcacerts -keystore <keystore.p12> -storepass <storepass> -alias <aliasname> -file
<server.crt>
```

When you obtain your signed certificate, it may be provided in a variety of ways. One possibility is that it is a PKCS#7 chained certificate (contains the signed server certificate as well as the certificate chain). Another option is that it is that you were provided a single signed certificate, and a set of certificates comprising the certificate chain. If you were provided a PKCS#7, you will import into the PrivateKeyEnty alias. If you were provided separate certificates, you will import the server certificate into the PrivateKeyEntry alias, and then import each of the other certificates into a different alias - for example -alias intermediate1, -alias intermediate2, -alias caroot.