

# **mod17\_ex04: Creating Client Keystore**

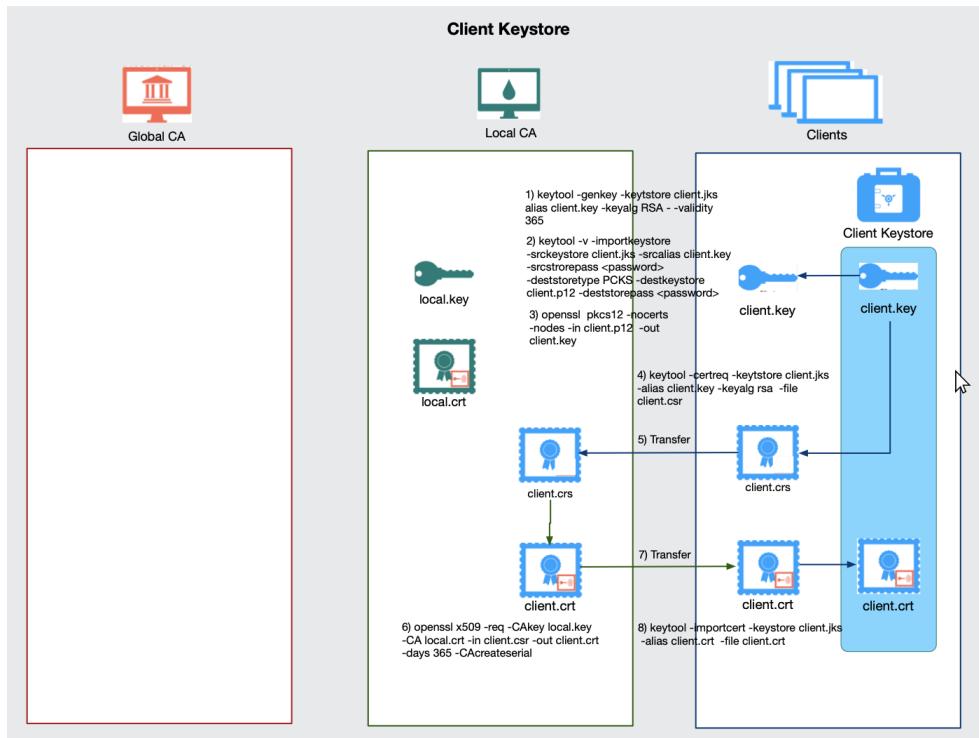
The purpose of this exercise is to create a host keystore for client. This is a repeat of [Creating a Local Keystore](#) with additional keytool commands.

## **Reference Information**

The following documents provide information related to this exercise.

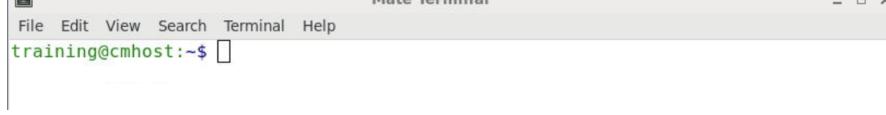
- [TBD](#)

## 1. Overview creating client keystore



1. The keytool command will create a client keystore with a private key.
2. The keytool command does not have a function for exporting the private key. The practice is to change the keystore format to PKCS12.
3. The openssl command will convert the PKCS12 private key into the PEM format.
4. The keytool command will create a certificate signing request (.csr).
5. Transfer the .csr file to the CA.
6. The CA uses the openssl command to sign the csr by creating the certificate file.
7. Transfer the .crt file to the client.
8. The keytool command imports the signed certificate file into the client keystore.

## 2. Open a Mate terminal as the user training.



## 3. Create the client keystore.

### 3.1 Change directory to client.

```
File Edit View Search Terminal Help
training@cmhost:~/tls/local$ cd ~/tls/client
training@cmhost:~/tls/client$
```

```
$ cd ~/tls/client
```

### 3.2 Create the local keystore with only a private key. Use password <password>. Use the information provided.

```
File Edit View Search Terminal Help
training@cmhost:~/tls/client$ keytool -genkey -keystore client.jks -keyalg RSA -alias client.key -validity 365
Enter keystore password:
Re-enter new password:
What is your first and last name?
[Unknown]: Ned Kelly
What is the name of your organizational unit?
[Unknown]: Edu
What is the name of your organization?
[Unknown]: Cloudride
What is the name of your City or Locality?
[Unknown]: Santa Clara
What is the name of your State or Province?
[Unknown]: CA
What is the two-letter country code for this unit?
[Unknown]: US
Is CN=Ned Kelly, OU=Edu, O=Cloudride, L=Santa Clara, ST=CA, C=US correct?
[no]: yes

Enter key password for <client.key>
(RETURN if same as keystore password):
```

```
$ keytool -genkey -keystore client.jks -keyalg RSA -alias client.key -validity 365
```

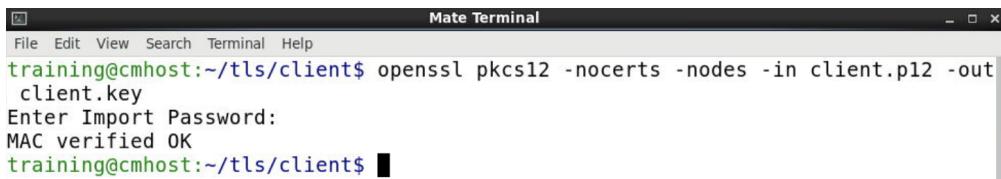
### 4. Export the private key from the local keystore.

#### 4.1 Convert the keystore into a PKCS12 format.

```
File Edit View Search Terminal Help
training@cmhost:~/tls/client$ keytool -importkeystore -srckeystore client.jks -srcalias client.key -srcstorepass BadPass@1 -deststoretype PKCS12 -destkeystore client.p12 -deststorepass BadPass@1
Importing keystore client.jks to client.p12...
training@cmhost:~/tls/client$
```

```
$ keytool -importkeystore -srckeystore client.jks -srcalias client.key -srcstorepass <password> -deststoretype PKCS12 -destkeystore client.p12 -deststorepass <password>
```

## 4.2 Convert PKCS12 format to PEM format.



```
File Edit View Search Terminal Help
training@cmhost:~/tls/client$ openssl pkcs12 -nocerts -nodes -in client.p12 -out client.key
Enter Import Password:
MAC verified OK
training@cmhost:~/tls/client$
```

```
$ openssl pkcs12 -nocerts -nodes -in client.p12 -out client.key
```

## 5. Create a Certificate Signing Request.

### 5.1 Use the keytool to create a signing request.



```
File Edit View Search Terminal Help
training@cmhost:~/tls/client$ keytool -certreq -keystore client.jks -alias client.key -keyalg rsa -file client.csr
Enter keystore password:

Warning:
The JKS keystore uses a proprietary format. It is recommended to migrate to PKCS
12 which is an industry standard format using "keytool -importkeystore -srckeyst
ore client.jks -destkeystore client.jks -deststoretype pkcs12".
training@cmhost:~/tls/client$
```

```
$ keytool -certreq -keystore client.jks -alias client.key -keyalg rsa -file
client.csr
```

### 5.2 Transfer the signing request to the CA.



```
File Edit View Search Terminal Help
training@cmhost:~/tls/client$ ls
client.csr client.jks client.key client.p12 global.jks global.pem
training@cmhost:~/tls/client$ cp client.csr ~/tls/local/client.csr
training@cmhost:~/tls/client$
```

```
$ cp client.csr ~/tls/local/client.csr
```

## 6. Sign a Certificate Signing Request.

## 6.1 Change directory to local.



```
File Edit View Search Terminal Help
training@cmhost:~/tls/client$ cd ~/tls/local
training@cmhost:~/tls/local$
```

```
$ cd ~/tls/local
```

## 6.2 Use the openssl command and the CA private key and CA public certificate to sign the certificate signing request.



```
File Edit View Search Terminal Help
training@cmhost:~/tls/local$ ls
client.csr  global.p12  local.crt  local.jks  local.p12
global.jks  global.pem  local csr  local.key
training@cmhost:~/tls/local$ openssl x509 -req -CAkey local.key -CA local.crt -in client.csr -out client.crt -days 365 -CAcreateserial
Signature ok
subject=/C=US/ST=CA/L=Santa Clara/O=Cloudride/OU=Edu/CN=Ned Kelly
Getting CA Private Key
training@cmhost:~/tls/local$
```

```
$ openssl x509 -req -CAkey local.key -CA local.crt -in client.csr -out client.crt -days 365 -CAcreateserial
```

## 6.3 Transfer the signed certificates back to the client.



```
File Edit View Search Terminal Help
training@cmhost:~/tls/local$ cp client.crt ~/tls/client/client.crt
training@cmhost:~/tls/local$
```

```
$ cp client.crt ~/tls/client/client.crt
```

## 7. Validate the signed certificate.

## 7.1 List the contents of the keystore. Compare alias local to local.crt.

```
File Edit View Search Terminal Help
training@cmhost:~/tls/client$ keytool -keystore client.jks -list -v | less
```

```
% keytool -list -v -keystore client.jks | less
```

## 8. Import the signed certificate into the keystore.

### 8.1 Change directories to local.

```
File Edit View Search Terminal Help
training@cmhost:~/tls/local$ cd ~/tls/client
training@cmhost:~/tls/client$
```

```
$ cd ~/tls/client
```

### 8.2 Import the signed certificate in the local keystore.

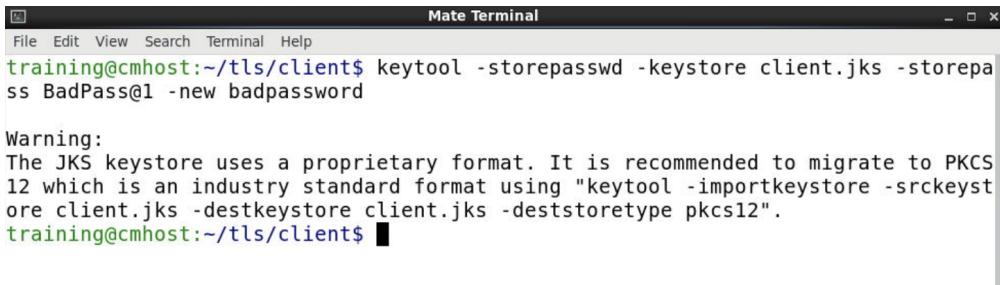
```
File Edit View Search Terminal Help
training@cmhost:~/tls/client$ keytool -importcert -keystore client.jks -alias client.crt -file client.crt
Enter keystore password:
Owner: CN=Ned Kelly, OU=Edu, O=Cloudride, L=Santa Clara, ST=CA, C=US
Issuer: CN=Ned Kelly, OU=Edu, O=Cloudride, L=Santa Clara, ST=CA, C=US
Serial number: 8b3d57767d7d61bb
Valid from: Sun Feb 27 11:52:56 PST 2022 until: Mon Feb 27 11:52:56 PST 2023
Certificate fingerprints:
    MD5: 99:61:AB:5C:53:2A:F5:63:93:3D:25:2E:EC:43:0C:DF
    SHA1: B8:AA:23:F9:AC:99:9F:41:15:0A:4C:9B:D0:AF:BE:52:C8:AB:2F:7F
    SHA256: 60:40:87:65:66:6E:2A:AA:72:78:85:9C:1A:E5:55:C9:5F:2F:DC:C3:3C:
32:4A:2A:46:8B:D0:8C:B9:27:80:4D
Signature algorithm name: SHA256withRSA
Subject Public Key Algorithm: 2048-bit RSA key
Version: 1
Trust this certificate? [no]: yes
Certificate was added to keystore

Warning:
The JKS keystore uses a proprietary format. It is recommended to migrate to PKCS
12 which is an industry standard format using "keytool -importkeystore -srckeyst
ore client.jks -destkeystore client.jks -deststoretype pkcs12".
training@cmhost:~/tls/client$
```

```
$ keytool -importcert -keystore client.jks -alias client.crt -file
client.crt
```

## 9. Managing Certificates in Keystore

## 9.1 Change the keystore password to badpassword.



```
Mate Terminal
File Edit View Search Terminal Help
training@cmhost:~/tls/client$ keytool -storepasswd -keystore client.jks -storepass BadPass@1 -new badpassword

Warning:
The JKS keystore uses a proprietary format. It is recommended to migrate to PKCS
12 which is an industry standard format using "keytool -importkeystore -srckeyst
ore client.jks -destkeystore client.jks -deststoretype pkcs12".
training@cmhost:~/tls/client$
```

```
% keytool -storepasswd -keystore client.jks -storepass <password> -new
badpassword
```

## 9.2 Change the password for the private key to badpassword.



```
Mate Terminal
File Edit View Search Terminal Help
training@cmhost:~/tls/client$ keytool -keypasswd -keystore client.jks -storepass
badpassword -keypass BadPass@1 -alias client.key
```

```
% keytool -keypasswd -keystore client.jks -storepass badpassword -keypass
BadPass@1 -alias client.key
```

## 9.3 Change the alias for the private key.

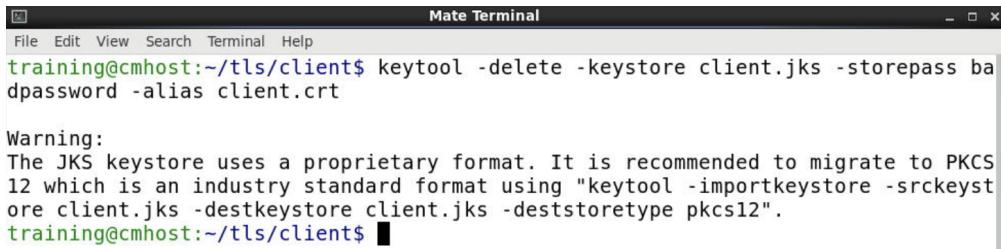


```
Mate Terminal
File Edit View Search Terminal Help
training@cmhost:~/tls/client$ keytool -changealias -keystore client.jks -storepa
ss badpassword -alias client.key -destalias badalias

Warning:
The JKS keystore uses a proprietary format. It is recommended to migrate to PKCS
12 which is an industry standard format using "keytool -importkeystore -srckeyst
ore client.jks -destkeystore client.jks -deststoretype pkcs12".
training@cmhost:~/tls/client$
```

```
% keytool -changealias -keystore client.jks -storepass badpassword -alias
client.key -destalias badalias
```

#### 9.4 Delete the public certificate, alias client.crt



```
Mate Terminal
File Edit View Search Terminal Help
training@cmhost:~/tls/client$ keytool -delete -keystore client.jks -storepass ba
dpassword -alias client.crt

Warning:
The JKS keystore uses a proprietary format. It is recommended to migrate to PKCS
12 which is an industry standard format using "keytool -importkeystore -srckeyst
ore client.jks -destkeystore client.jks -deststoretype pkcs12".
training@cmhost:~/tls/client$
```

```
% keytool -delete -keystore client.jks -storepass badpassword -alias
client.crt
```