

mod16_ex01: Creating the jssecacerts.pem File

The purpose of this exercise is to create a `jssecacerts.pem` file to be used by Auto-TLS. The `cacerts` file is a list of all approved Certificate Authorities. You will copy `cacerts` to `jssecacerts`. JSSE stands for Java Secure Socket Extension, the naming convention is legacy. You will then export the certificate file to a PKCS12 format and then use `openssl` to convert the PKCS12 file to a PEM format.

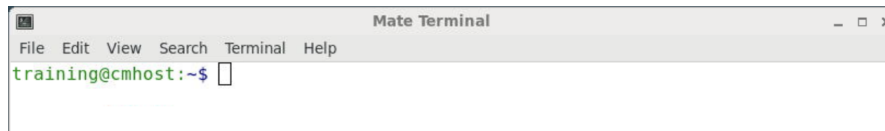
Reference Information

The following documents provide information related to this exercise.

[Use Cloudera Manager to generate internal CA and corresponding certificates](#)

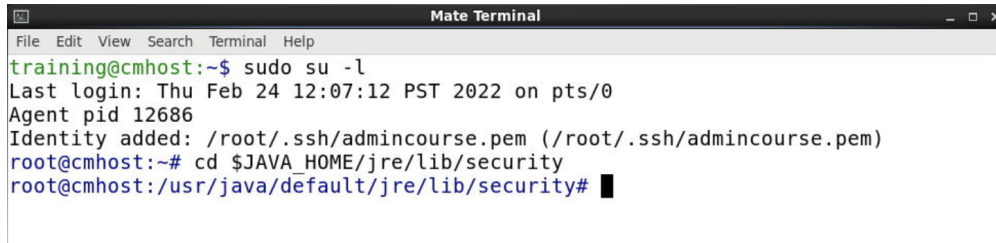
[Converting file encodings for TLS/SSL certificates and keys](#)

1. Open a Mate terminal as the local administrative user, training.



2. Create the jssecacerts keystore.

2.1 On the cmhost open a terminal window and switch to the root user. Change directory to the location of cacerts.



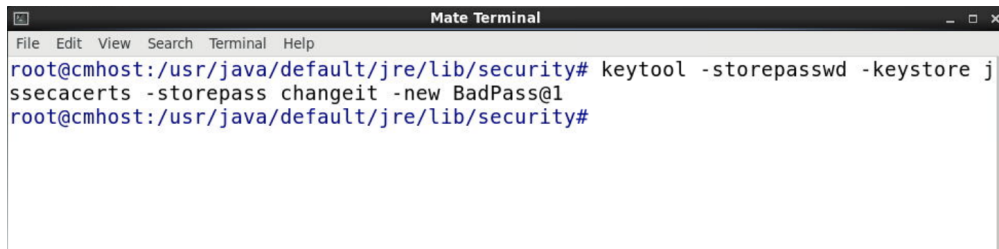
```
$ sudo su -l  
# cd $JAVA_HOME/jre/lib/security
```

2.2 Copy the cacerts file to jssecacerts. The jssecacerts will have the same password as cacerts.



```
# cp cacerts jssecacerts
```

2.3 Change the password on the jssecacerts file from the default of changeit to <password>.



```

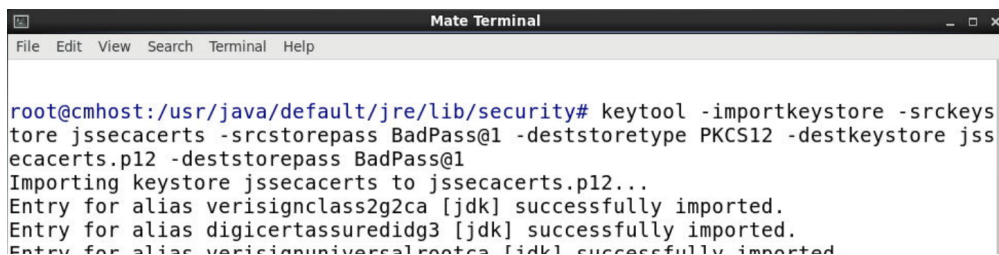
Mate Terminal
File Edit View Search Terminal Help
root@cmhost:/usr/java/default/jre/lib/security# keytool -storepasswd -keystore jssecacerts -storepass changeit -new BadPass@1
root@cmhost:/usr/java/default/jre/lib/security#

```

```
# keytool -storepasswd -keystore jssecacerts
-storepass changeit -new <password>
```

3. Convert the Java keystore to PKCS12 format.

3.1 Use the keytool command to export from the jssecacerts keystore to the pkcs12 format.



```

Mate Terminal
File Edit View Search Terminal Help
root@cmhost:/usr/java/default/jre/lib/security# keytool -importkeystore -srckeystore jssecacerts -srcstorepass BadPass@1 -deststoretype PKCS12 -destkeystore jssecacerts.p12 -deststorepass BadPass@1
Importing keystore jssecacerts to jssecacerts.p12...
Entry for alias verisignclass2g2ca [jdk] successfully imported.
Entry for alias digicertassuredidg3 [jdk] successfully imported.
Entry for alias verisignuniversalrootca [jdk] successfully imported.

```

```
# keytool -importkeystore \
-srckeystore jssecacerts \
-srcstorepass <password> \
-deststoretype PKCS12 \
-destkeystore jssecacerts.p12 \
-deststorepass <password>
```

4. Convert PKCS12 file to PEM format.

4.1 Use the openssl command to convert the pkcs12 format to the pem format.



```

root@cmhost:/usr/java/default/jre/lib/security# openssl pkcs12 -in jssecacerts.p12 -passin pass:BadPass@1 -out jssecacerts.pem
MAC verified OK
root@cmhost:/usr/java/default/jre/lib/security#

```

```
# openssl pkcs12 -in jssecacerts.p12 \
-passin pass:<password> \
-out jssecacerts.pem
```

4.2 Use the keytool command to verify the certificates in jssecacerts.pem. Pipe the output to less and page through. Type q to quit.

```
Certificate[1]:
Owner: OU=VeriSign Trust Network, OU="(c) 1998 VeriSign, Inc. - For authorized use only", OU=Class 2 Public Primary Certification Authority - G2, O="VeriSign, Inc.", C=US
Issuer: OU=VeriSign Trust Network, OU="(c) 1998 VeriSign, Inc. - For authorized use only", OU=Class 2 Public Primary Certification Authority - G2, O="VeriSign, Inc.", C=US
Serial number: b92f60cc889fa17a4609b85b706c8aaf
Valid from: Sun May 17 17:00:00 PDT 1998 until: Tue Aug 01 16:59:59 PDT 2028
Certificate fingerprints:
    MD5:  2D:BB:E5:25:D3:D1:65:82:3A:B7:0E:FA:E6:EB:E2:E1
    SHA1: B3:EA:C4:47:76:C9:C8:1C:EA:F2:9D:95:B6:CC:A0:08:18:67:EC:9D
    SHA256: 3A:43:E2:20:FE:7F:3E:A9:65:3D:1E:21:74:2E:AC:2B:75:C2:0F:D8:98:03:05:BC:50:2C:AF:8C:2D:9B:41:A1
Signature algorithm name: SHA1withRSA
Subject Public Key Algorithm: 1024-bit RSA key
Version: 1

Certificate[2]:
Owner: CN=DigiCert Assured ID Root G3, OU=www.digicert.com, O=DigiCert Inc, C=US
Issuer: CN=DigiCert Assured ID Root G3, OU=www.digicert.com, O=DigiCert Inc, C=US
Serial number: ba15afa1ddfa0b54944afcd24a06cec
```

```
# keytool -printcert -v -file jssecacerts.pem | less
```

4.3 Type exit to logout as the user root and return to the user training.

```
# exit
```

5. Return to Cloudera Manager Home.