ECS Test Drive SSL issues with Java

The version of Oracle’s Java for Windows does not use the certificate store for the operating system and instead uses its own certificate store. This causes problems because the root CA certificate that signs the ECS Test Drive certificate is included with Windows but not with Java. Because of this, you may see errors like the following:

If your application uses java.net.HttpUrlConnection:

Exception in thread "main" javax.net.ssl.SSLHandshakeException: sun.security.val

idator.ValidatorException: PKIX path building failed: sun.security.provider.cert

path.SunCertPathBuilderException: unable to find valid certification path to req

uested target

at sun.security.ssl.Alerts.getSSLException(Unknown Source)

If your application uses the Apache HTTP Client:

javax.net.ssl.SSLPeerUnverifiedException: peer not authenticated

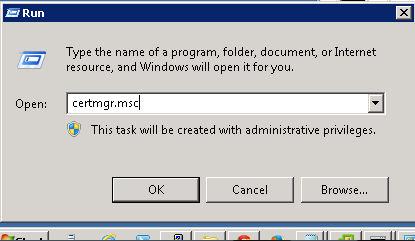
at sun.security.ssl.SSLSessionImpl.getPeerCertificates(Unknown Source)

at org.apache.http.conn.ssl.AbstractVerifier.verify(AbstractVerifier.java:128)

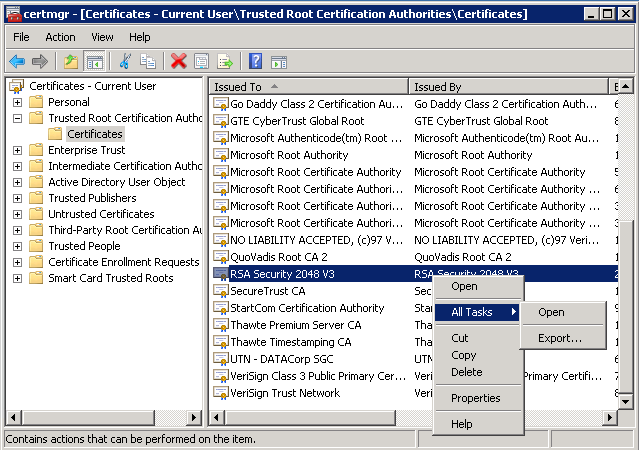
To fix this problem, we’ll export the certificate from the OS certificate store and import it into Java’s certificate store.

# Windows

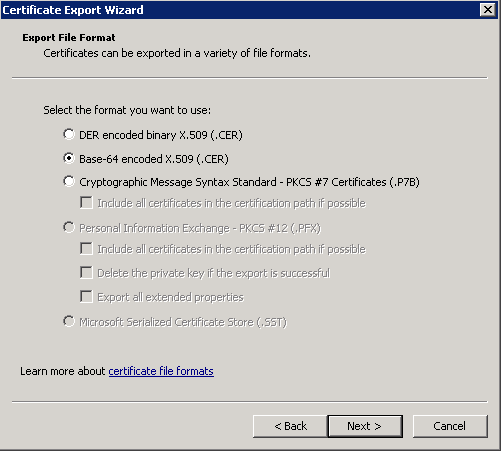
First, open the Windows certificate manager. You can do this by clicking the Start button, clicking Run and entering “certmgr.msc”:



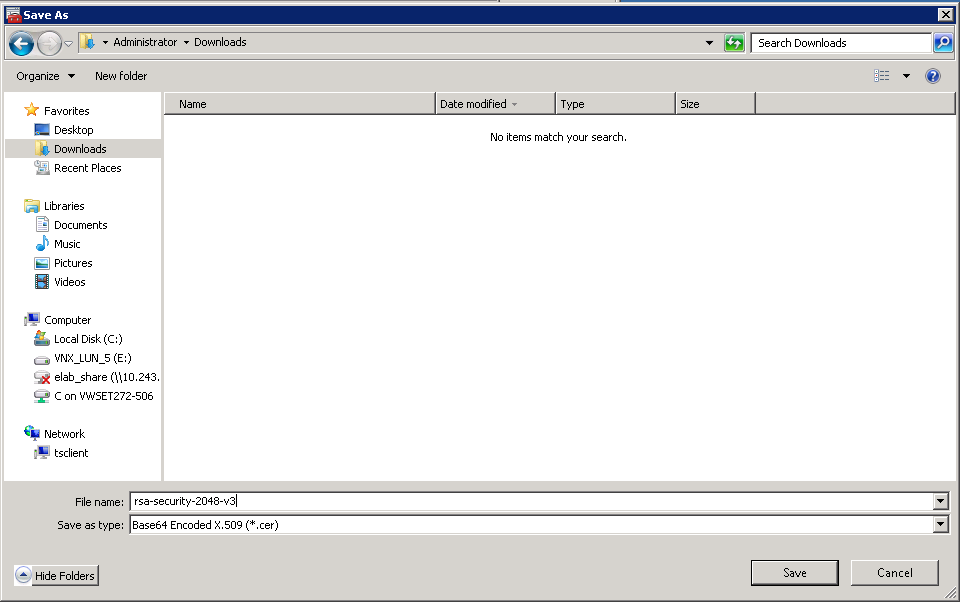
The certificate manager will load. Navigate to “Trusted Root Certification Authorities->Certificates” and locate the certificate named “RSA Security 2048 V3”. Right-click on the certificate and select “Export…”



Click next and when prompted select the “Base-64 encoded X.509 (.CER)” format then click Next.



Pick a location to save the certificate…





Change to your Java’s Bin directory to locate the keytool command. You’ll also have to locate the proper “cacerts” file that is used by the Java runtime you’re using to execute your application. This is generally in the jre\lib\security directory.

keytool.exe -importcert -file {Path to where you exported the certificate}\rsa-security-2048-v3.cer -keystore "{Path to your jre\lib\security directory}\cacerts" -storepass changeit

You will execute the command similar to the following:

C:\Program Files\Java\jdk1.8.0\_45\bin>**keytool.exe -importcert -file \Users\Administrator\Downloads\rsa-security-2048-v3.cer -keystore "C:\Program Files\Java\jdk1.8.0\_45\jre\lib\security\cacerts" -storepass changeit**

Owner: OU=RSA Security 2048 V3, O=RSA Security Inc

Issuer: OU=RSA Security 2048 V3, O=RSA Security Inc

Serial number: a0101010000027c0000000a00000002

Valid from: Thu Feb 22 15:39:23 EST 2001 until: Sun Feb 22 15:39:23 EST 2026

Certificate fingerprints:

         MD5:  77:0D:19:B1:21:FD:00:42:9C:3E:0C:A5:DD:0B:02:8E

         SHA1: 25:01:90:19:CF:FB:D9:99:1C:B7:68:25:74:8D:94:5F:30:93:95:42

         SHA256: AF:8B:67:62:A1:E5:28:22:81:61:A9:5D:5C:55:9E:E2:66:27:8F:75:D7:

9E:83:01:89:A5:03:50:6A:BD:6B:4C

         Signature algorithm name: SHA1withRSA

         Version: 3

Extensions:

#1: ObjectId: 2.5.29.35 Criticality=false

AuthorityKeyIdentifier [

KeyIdentifier [

0000: 07 C3 51 30 A4 AA E9 45   AE 35 24 FA FF 24 2C 33  ..Q0...E.5$..$,3

0010: D0 B1 9D 8C                                        ....

]

]

#2: ObjectId: 2.5.29.19 Criticality=true

BasicConstraints:[

  CA:true

  PathLen:2147483647

]

#3: ObjectId: 2.5.29.15 Criticality=true

KeyUsage [

  Key\_CertSign

  Crl\_Sign

]

#4: ObjectId: 2.5.29.14 Criticality=false

SubjectKeyIdentifier [

KeyIdentifier [

0000: 07 C3 51 30 A4 AA E9 45   AE 35 24 FA FF 24 2C 33  ..Q0...E.5$..$,3

0010: D0 B1 9D 8C                                        ....

]

]

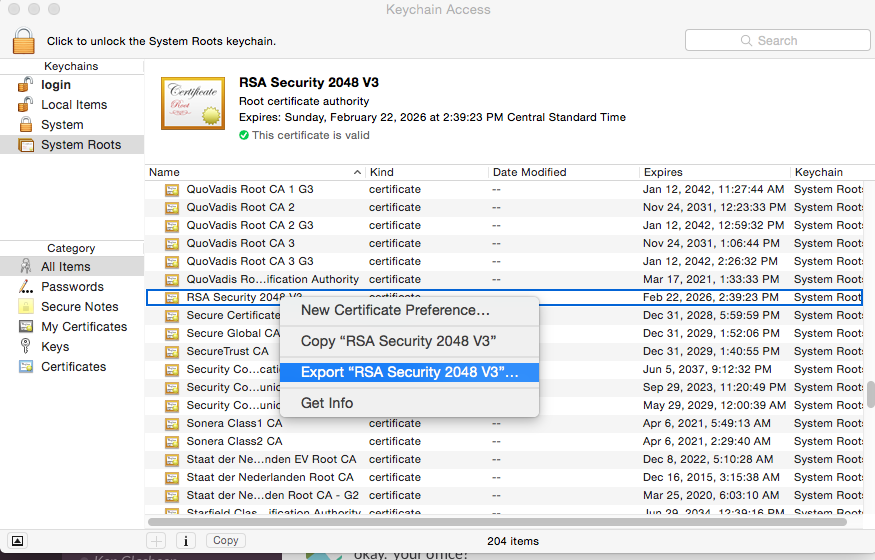
Trust this certificate? [no]:  **yes**

Certificate was added to keystore

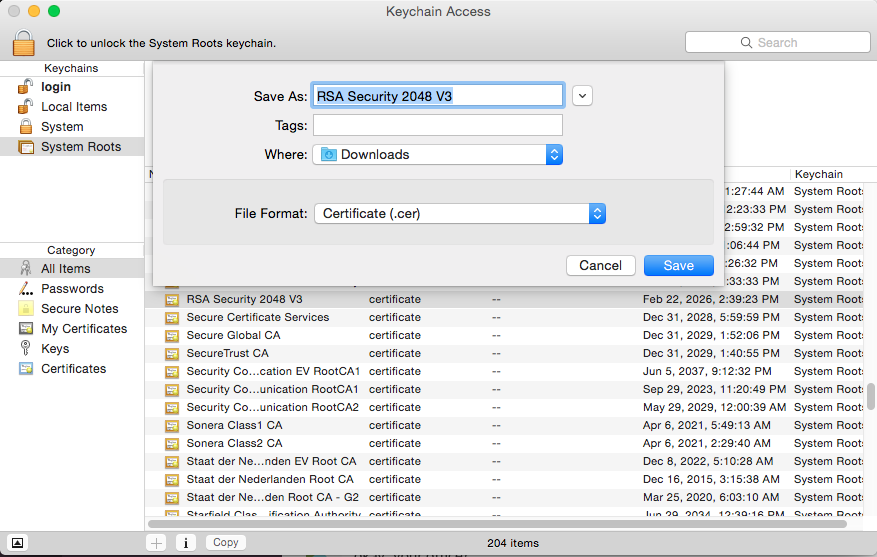
Retest your application now and the issue should be resolved.

# Apple OSX

First, open up the “Keychain Access” application. You can find this by clicking on the spotlight search icon in the upper right corner of the screen (it’s a magnifying glass) and typing “Keychain Access”. Inside Keychain Access, select the “System Roots” keychain and find the “RSA Security 2048 V3” certificate. Right-click on the certificate and select “Export RSA Security 2048 V3…”



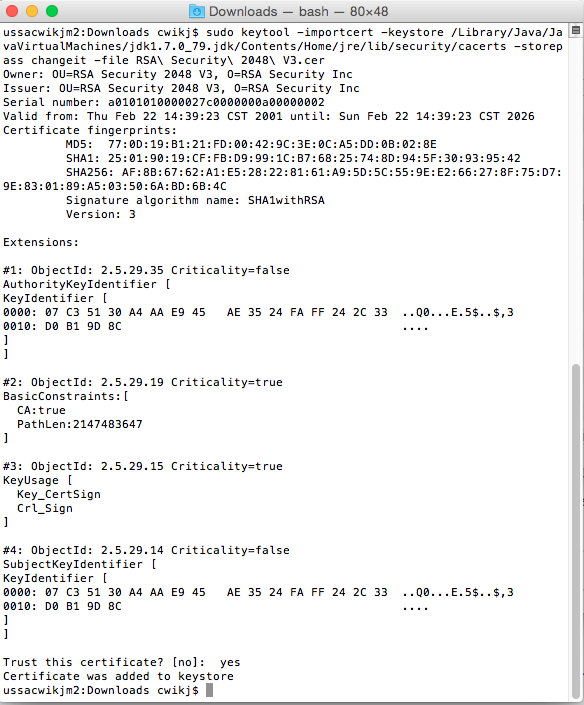
When prompted, save the certificate somewhere (e.g. Downloads) and make sure it’s in Certificate (.cer) format.



Finally, open up a Terminal and use the keytool application to import the certificate into your Java certificate store. Your certificate store should be under /Library/Java/JavaVirtualMachines***/{version}/***Contents/Home/jre/lib/security/cacerts

The command will be:

sudo keytool –importcert -keystore "{Path to your jre/lib/security directory}/cacerts" -storepass changeit -file {Path to where you exported the certificate}/RSA\ Security\ 2048\ V3.cer



Retest your application now and the issue should be resolved.