SSL - Commands

Tuesday, June 28, 2022 1:35 PM

Keytool

Java Keytool Commands for Creating and Importing

These commands allow you to generate a new Java Keytool keystore file, create a CSR, and import certificates. Any root or intermediate certificates will need to be imported before importing the primary certificate for your domain.

- **Generate a Java keystore and key pair**
  keytool -genkey -alias mydomain-keyalg RSA -keystore keystore.jks -keysize 2048
- **Generate a certificate signing request (CSR) for an existing Java keystore**
  keytool -certreq -alias mydomain-keystore keystore.jks-file mydomain.csr
- **Import a root or intermediate CA certificate to an existing Java keystore**
  keytool -import -trustcacerts -alias root -file Thawte.crt-keystore keystore.jks
- **Import a signed primary certificate to an existing Java keystore**
  keytool -import -trustcacerts -alias mydomain-file mydomain.crt-keystore keystore.jks
- **Generate a keystore and self-signed certificate**
  keytool -genkey -keyalg RSA -alias selfsigned -keystore keystore.jks-storepass password-validity 360 -keysize 2048

Java Keytool Commands for Checking

If you need to check the information within a certificate, or Java keystore, use these commands.

- **Check a stand-alone certificate**
  keytool -printcert -v -file mydomain.crt
- **Check which certificates are in a Java keystore**
  keytool -list -v -keystore keystore.jks
- **Check a particular keystore entry using an alias**
  keytool -list -v -keystore keystore.jks-alias mydomain

Other Java Keytool Commands

- **Delete a certificate from a Java Keytool keystore**
  keytool -delete -alias mydomain-keystore keystore.jks
- **Change a Java keystore password**
  keytool -storepasswd -new new_storepass -keystore keystore.jks
- **Export a certificate from a keystore**
  keytool -export -alias mydomain-file mydomain.crt-keystore keystore.jks
- **List Trusted CA Certs**
  keytool -list -v -keystore $JAVA_HOME/jre/lib/security/cacerts
- **Import New CA into Trusted Certs**
  keytool -import -trustcacerts -file /path/to/ca/ca.pem-alias CA_ALIAS-keystore $JAVA_HOME/jre/lib/security/cacerts

OpenSSL

General OpenSSL Commands

These commands allow you to generate CSRs, Certificates, Private Keys and do other miscellaneous tasks.

- **Generate a new private key and Certificate Signing Request**
  openssl req -out CSR.csr-new -newkey rsa:2048 -nodes -keyout privateKey.key
- **Generate a self-signed certificate**
  openssl req -x509 -sha256 -nodes -days 365 -newkey rsa:2048 -keyout privateKey.key-out certificate.crt
- **Generate a certificate signing request (CSR) for an existing private key**
  openssl req -out CSR.csr-key privateKey.key-new
- **Generate a certificate signing request based on an existing certificate**
  openssl x509 -x509toreq -in certificate.crt-out CSR.csr-signkey privateKey.key
- **Remove a passphrase from a private key**
  openssl rsa -in privateKey.pem-out newPrivateKey.pem

Checking Using OpenSSL

If you need to check the information within a Certificate, CSR or Private Key, use these commands.

- **Check a Certificate Signing Request (CSR)**
  openssl req -text -noout -verify -in CSR.csr
- **Check a private key**
  openssl rsa -in privateKey.key-check
- **Check a certificate**
  openssl x509 -in certificate.crt-text -noout
- **Check a PKCS#12 file (.pfx or .p12)**
  openssl pkcs12 -info -in keyStore.p12

Debugging Using OpenSSL

If you are receiving an error that the private doesn't match the certificate or that a certificate that you installed to a site is not trusted, try one of these commands.

- **Check an MD5 hash of the public key to ensure that it matches with what is in a CSR or private key**
  openssl x509 -noout -modulus -in certificate.crt| openssl md5
  openssl rsa -noout -modulus -in privateKey.key| openssl md5
  openssl req -noout -modulus -in CSR.csr| openssl md5
- **Check an SSL connection. All the certificates (including Intermediates) should be displayed**
  openssl s_client -connect www.paypal.com:443

Converting Using OpenSSL

These commands allow you to convert certificates and keys to different formats to make them compatible with specific types of servers or software. For example, you can convert a normal PEM file that would work with Apache to a PFX (PKCS#12) file and use it with Tomcat or IIS.

- **Convert a DER file (.crt .cer .der) to PEM**
  openssl x509 -inform der -in certificate.cer-out certificate.pem
- **Convert a PEM file to DER**
  openssl x509 -outform der -in certificate.pem-out certificate.der
- **Convert a PKCS#12 file (.pfx .p12) containing a private key and certificates to PEM**
  openssl pkcs12 -in keyStore.pfx-out keyStore.pem-nodes
  You can add -nocerts to only output the private key or add -nokeys to only output the certificates.
- **Convert a PEM certificate file and a private key to PKCS#12 (.pfx .p12)**
  openssl pkcs12 -export -out certificate.pfx-inkey privateKey.key-in certificate.crt-certfile CACert.crt

GSK

Creating keystore by specifying password expiry

Sample 1

#gsk7cmd -keydb -create -db test.kdb -pw changeit -type kdb -expire 3400

The above command creates a keystore file (test.kdb) of kdb type and keep the password expiry to 7300 days

Sample 2

gsk7cmd -keydb -expiry -db test.kdb -pw changeit

This will list the password expiry of keystore test.kdb

Output:

Password expiry time: Aug 9, 2032 2:05:51 AM

Deleting the keystore

Sample 3

#gsk7cmd -keydb -delete -db test.kdb -pw changeit

This deletes the keystore file test.kdb

Creating a default keystore

Sample 4

#gsk7cmd -keydb -create -db testcacerts.jks -pw testit

The above command creates a keystore file with the name testcacerts.jks and the password testit in the current directory

Changing the keystore password

Sample 5

#gsk7cmd -keydb -changepw -db testcacerts.jks -pw testit -new_pw changeit

This changes the password from testit to changeit

Certificate Management (-cert)

Adding certificate to a keystore with out specifying label

Sample 6

#gsk7cmd -cert -add -file test.cer -db testcacerts.jks -pw changeit

This adds the certificate file test.cer in testcacerts.jks keystore, If label is not specified it will generate a label (kindly note the label details in example 7).

Sample 7

#gsk7cmd -cert -details -label "cn=TESTCERT, o=IBM, c=us" -db testcacerts.jks -pw changeit

This command will list the details of certificate with label "cn=TESTCERT, o=IBM, c=us" (The certificate which was added in example 6)

Output

Label: cn=TESTCERT, o=IBM, c=us
Key Size: 1024
Version: X509 V3
Serial Number: 12 57 4F 87 1B F8 69 DD
Issued by: CN=TESTCERT, O=IBM, C=US
Subject: CN=TESTCERT, O=IBM, C=US
Valid: From: Wednesday, May 12, 2010 2:01:04 AM IST To: Wednesday, May 8, 2030 2:01:04 AM IST
Fingerprint: BE:87:67:14:AD:FD:64:B9:CC:08:CF:3E:76:05:2A:DC:BB:EB:DF:69
Signature Algorithm: MD5withRSA (1.2.840.113549.1.1.4)
Trust Status: enabled

Deleting a certificate from the keystore

Sample 8

#gsk7cmd -cert -delete -label "cn=TESTCERT, o=IBM, c=us" -db testcacerts.jks -pw changeit

This command deletes the certificate with the label "cn=TESTCERT, o=IBM, c=us" (the certificate which was added in example 6)

Sample 9

#gsk7cmd -cert -details -label "cn=TESTCERT, o=IBM, c=us" -db testcacerts.jks -pw changeit

This commands confirms the delete operation in example 8, The below output says the certificate with the label 'cn=TESTCERT, o=IBM, c=us' does not exists

Output

The database doesn't contain an entry with label 'cn=TESTCERT, o=IBM, c=us'.
Check the label and try again.

Adding certificate to a keystore with the label

Sample 10

#gsk7cmd -cert -add -file test.cer -label "This is a cert" -db testcacerts.jks -pw changeit

This adds the certificate 'test.cer' with the label "This is a cert". (in example 6 we have added the certificate without specifying the label)

Sample 11

#gsk7cmd -cert -details -label "This is a cert" -db testcacerts.jks -pw changeit

This confirms that the certificate test.cer has been added with the label "This is a cert", check the output below/

Output

Label: this is a cert
Key Size: 1024
Version: X509 V3
Serial Number: 12 57 4F 87 1B F8 69 DD
Issued by: CN=TESTCERT, O=IBM, C=US
Subject: CN=TESTCERT, O=IBM, C=US
Valid: From: Wednesday, May 12, 2010 2:01:04 AM IST To: Wednesday, May 8, 2030 2:01:04 AM IST
Fingerprint: BE:87:67:14:AD:FD:64:B9:CC:08:CF:3E:76:05:2A:DC:BB:EB:DF:69
Signature Algorithm: MD5withRSA (1.2.840.113549.1.1.4)
Trust Status: enabled

Renaming the label of a certificate

Sample 12

#gsk7cmd -cert -rename -label "This is a cert" -new_label "The_new_label" -db testcacerts.jks -pw changeit

This renames the lable "This is a cert" with new name "The_new_label".

Sample 13

#gsk7cmd -cert -details -label "The_new_label" -db testcacerts.jks -pw changeit

Example 13 and Example 14 confirms example 12,Check the output below.

Output

Label: the_new_label
Key Size: 1024
Version: X509 V3
Serial Number: 12 57 4F 87 1B F8 69 DD
Issued by: CN=TESTCERT, O=IBM, C=US
Subject: CN=TESTCERT, O=IBM, C=US
Valid: From: Wednesday, May 12, 2010 2:01:04 AM IST To: Wednesday, May 8, 2030 2:01:04 AM IST
Fingerprint: BE:87:67:14:AD:FD:64:B9:CC:08:CF:3E:76:05:2A:DC:BB:EB:DF:69
Signature Algorithm: MD5withRSA (1.2.840.113549.1.1.4)
Trust Status: enabled

Sample 14

#gsk7cmd -cert -details -label "This is a cert" -db testcacerts.jks -pw changeit

Example 14 and Example 13 confirms example 12, because in the output of example 13 testcacerts.jks keystore contains a certificate with the label "The_new_label" and the output of example 14 says the testcacerts.jks keystore does not have an with the label "This is a cert" (label name before rename).

Output

The database doesn't contain an entry with label 'This is a cert'.
Check the label and try again.

Extracting a certificate from the keyfile

Sample 15

#gsk7cmd -cert -extract -label "The_new_label" -target "this_is_extracted_cert.cer" -db testcacerts.jks -pw changeit

This will extracrt the certificate with label "The_new_label" into a file this_is_extracted_cert.cer, check the below output for file confirmation

#ls this_is_extracted_cert.cer
this_is_extracted_cert.cer

Creating a self signed certificate

Sample 16

gsk7cmd -cert -create -db testcacerts.jks -pw changeit -label 'New_Self_Signed' -dn CN=geeksidea,O=ibm,C=in -expire 7300 -size 1024 -x509version 3

This creates a self signed certificate with the label 'New_Self_Signed'

Sample 17

gsk7cmd -cert -details -label 'New_Self_Signed' -db testcacerts.jks -pw changeit

This confirms the self signed certificate creation ,Verify the certificate in the below output

Output

Label: new_self_signed
Key Size: 1024
Version: X509 V3
Serial Number: 50 29 68 22
Issued by: CN=geeksidea, O=ibm, C=in
Subject: CN=geeksidea, O=ibm, C=in
Valid: From: Tuesday, August 14, 2012 2:18:34 AM IST To: Monday, August 9, 2032 2:18:34 AM IST
Fingerprint: 0C:D5:A0:6A:54:76:6B:3E:D0:3E:2E:42:1C:D0:32:43:66:82:FE:70

Signature Algorithm: SHA1withRSA (1.2.840.113549.1.1.5)
Trust Status: enabled