# Manejo de Certificados con openssl

## Identificación del tipo de certificado

**Question**

How do I find out if it is in PEM, DER, or pkcs12 format?

**Answer**

Open the certificate using a text editor like Notepad and see if it is in text format.

Example:

-----BEGIN CERTIFICATE-----   
MIIDijCCAvOgAwIBAgIJAKRvtQxONVZoMA0GCSqGSIb3DQEBBAUAMIGLMQswCQYD   
VQQGEwJVUzETMBEGA1UECBMKQ2FsaWZvcm5pYTESMBAGA1UEBxMJU3Vubnl2YWxl   
fQ6EaiXZVbrQt+PMqG0F80+4wxVXug9EW5Ob9M/opaCGI+cgtpLCwSf6CjsmAcUc

…………………………………………………………………………………………………………………………………………………………………………  
b6EjG/l4HW2BztYJfx15pk51M49TYS7okDKWYRT10y65xcyQdfUKvfDC1k5P9Q==   
-----END CERTIFICATE-----

If the certificate is in text format, then it is in PEM format.

You can read the contents of a PEM certificate (cert.crt) using the 'openssl' command on Linux or Windows as follows:

openssl x509 -in cert.crt -text

If the file content is binary, the certificate could be either DER or pkcs12/pfx. To find out which format, run the following 'openssl' commands to open the certificate:

To open a DER certificate:

openssl x509 -in cert.crt -inform DER -text

To display pkcs12 certificate information:

openssl pkcs12 -in cert.crt -info

## Comandos openssl Útiles

Create a private key

openssl genrsa -out server.key 4096

Generate a new private key and certificate signing request

openssl req -out server.csr -new -newkey rsa:4096 -nodes -keyout server.key

Generate a self-signed certificate

openssl req -x509 -sha256 -nodes -days 365 -newkey rsa:4096 -keyout server.key -out server.crt

Generate a certificate signing request (CSR) for an existing private key

openssl req -out server.csr -key server.key -new

Generate a certificate signing request based on an existing certificate

openssl x509 -x509toreq -in server.crt -out server.csr -signkey server.key

Remove a passphrase from a private key

openssl rsa -in server.pem -out newserver.pem

Parse a list of revoked serial numbers

openssl crl -inform DER -text -noout -in list.crl

Check a certificate signing request (CSR)

openssl req -text -noout -verify -in server.csr

Check a private key

openssl rsa -in server.key -check

Check a public key

openssl rsa -inform PEM -pubin -in pub.key -text -noout

openssl pkey -inform PEM -pubin -in pub.key -text -noout

Check a certificate

openssl x509 -in server.crt -text -noout

Check a PKCS#12 file (.pfx or .p12)

openssl pkcs12 -info -in server.p12

Verify a private key matches a certificate

#PEM

openssl x509 -noout -modulus -in server.crt | openssl md5

openssl rsa -noout -modulus -in server.key | openssl md5

openssl req -noout -modulus -in server.csr | openssl md5

#DER  
openssl x509 -noout -modulus -inform DER -in server.crt | openssl md5

#PKCS12  
openssl pkcs12 -in cert.crt -noout -modulus | openssl md5

Display all certificates including intermediates

openssl s\_client -connect www.paypal.com:443

Convert a DER file (.crt .cer .der) to PEM

openssl x509 -inform der -in server.cer -out server.pem

Convert a PEM file to DER

openssl x509 -outform der -in server.pem -out server.der

Convert a PKCS#12 file (.pfx .p12) containing a private key and certificates to PEM

openssl pkcs12 -in server.pfx -out server.pem -nodes

Convert a PEM certificate file and a private key to PKCS#12 (.pfx .p12)

openssl pkcs12 -export -out server.pfx -inkey server.key -in server.crt -certfile CACert.crt

Generate a Diffie Hellman key

openssl dhparam -out dhparam.pem 2048

Encrypt files with rsautl

openssl rsautl -encrypt -in plaintext.txt -out encrypted.txt -pubin -inkey pubkey.pem

Decrypt files with rsautl

openssl rsautl -decrypt -in encrypted.txt -out plaintext.txt -inkey privkey.pem

# 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 (see [How to Create a Self Signed Certificate using Java Keytool](https://www.sslshopper.com/article-how-to-create-a-self-signed-certificate-using-java-keytool.html)for more info)

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

If you need to move a certificate from Java Keytool to Apache or another type of system, check out these instructions for [converting a Java Keytool keystore using OpenSSL](https://www.sslshopper.com/article-move-your-java-keytool-ssl-certificate-to-openssl.html). For more information, check out the [Java Keytool documentation](http://java.sun.com/j2se/1.5.0/docs/tooldocs/solaris/keytool.html) or check out our [Tomcat SSL Installation Instructions which use Java Keytool](https://www.sslshopper.com/tomcat-ssl-installation-instructions.html).