

<https://knowledge.digicert.com/solution/SO26449.html>

How to convert a certificate into the appropriate format

Solution

If your server/device requires a different certificate format other than Base64 encoded X.509, a third party tool such as OpenSSL can be used to convert the certificates into the appropriate format.

For information on OpenSSL please visit: www.openssl.org

Note: OpenSSL is an open source tool that is not provided or supported by Thawte

Some common conversion commands are listed below:

Note: The PEM format is the most common format used for certificates. Extensions used for PEM certificates are cer, crt, and pem. They are Base64 encoded ASCII files. The DER format is the binary form of the certificate. DER formatted certificates do not contain the "BEGIN CERTIFICATE/END CERTIFICATE" statements. DER formatted certificates most often use the '.der' extension.

Convert x509 to PEM

```
openssl x509 -in certificatename.cer -outform PEM -out certificatename.pem
```

Convert PEM to DER

```
openssl x509 -outform der -in certificatename.pem -out certificatename.der
```

Convert DER to PEM

```
openssl x509 -inform der -in certificatename.der -out certificatename.pem
```

Convert PEM to P7B

Note: The PKCS#7 or P7B format is stored in Base64 ASCII format and has a file extension of .p7b or .p7c.

A P7B file only contains certificates and chain certificates (Intermediate CAs), not the private key. The most common platforms that support P7B files are Microsoft Windows and Java Tomcat.

```
openssl crl2pkcs7 -nocrl -certfile certificatename.pem -out certificatename.p7b -certfile CACert.cer
```

Convert PKCS7 to PEM

```
openssl pkcs7 -print_certs -in certificatename.p7b -out certificatename.pem
```

Convert pfx to PEM

Note: The PKCS#12 or PFX format is a binary format for storing the server certificate, intermediate certificates, and the private key in one encryptable file. PFX files usually have extensions such as .pfx and .p12. PFX files are typically used on Windows machines to import and export certificates and private keys.

```
openssl pkcs12 -in certificatename.pfx -out certificatename.pem
```

Convert PFX to PKCS#8

Note: This requires 2 commands

STEP 1: Convert PFX to PEM

```
openssl pkcs12 -in certificatename.pfx -nocerts -nodes -out certificatename.pem
```

STEP 2: Convert PEM to PKCS8

```
openSSL pkcs8 -in certificatename.pem -topk8 -nocrypt -out certificatename.pk8
```

Convert P7B to PFX

Note: This requires 2 commands

STEP 1: Convert P7B to CER

```
openssl pkcs7 -print_certs -in certificatename.p7b -out certificatename.cer
```

STEP 2: Convert CER and Private Key to PFX

```
openssl pkcs12 -export -in certificatename.cer -inkey privateKey.key -out  
certificatename.pfx -certfile cacert.cer
```