SSL Certificate Formats

In connection with [**SSL certificates**](https://www.sslmentor.com/), a relatively large number of names such as PEM, CSR, KEY, DER, etc. are mentioned. These are files that are practically only "boxes" for the location of the certificate and its keys. A large number of formats were created gradually due to various implementations in operating systems or applications, some were standardized in RFC.

CSR (.csr)

A Certificate Signing Request (CSR) **is a certificate request that is passed to a certification authority for certification**. The request can be generated directly on the server, in the OpenSSL application or you can easily generate it in the order detail according to this manual, including the private key, after ordering the SSL certificate. The application format is according to PKCS # 10 (Public Key Cryptography Standards) and is defined in [**RFC 2986**](https://tools.ietf.org/html/rfc2986) (Certification Request Syntax Specification). The CSR application contains the necessary information for issuing the certificate. That is, the domain name, organization, state, and also the public key that the certification authority confirms. The encoding format of the CSR that is inserted into the order and sent to the certification authority is PEM. The information structure in the request is defined using ASN.1 (abstract syntax notation).  
After the certificate is issued and signed by the certification authority, the certificate is already delivered from the authority in other formats, such as CRT, p7b. It is often also sent directly by e-mail in txt PEM format, together with information about the issuance of the SSL certificate.

**We do not recommend creating** a certificate request and private key on unknown online sites.  
In our help we publish instructions on [**how to generate a CSR and private key in OpenSSL**](https://www.sslmentor.com/help/openssl-key-generation-and-certificate-request).

PEM (.pem)

One of the most used formats for storing SSL/TLS certificates. **It is a container for storing text-encoded cryptographic data (keys and certificates)** and allows easy sending by e-mail, it is defined in RFC 1421 to 1424. It can contain a separate public certificate but also a public certificate plus CA certificates or it can contain a whole set of certificates including public key, private key, and root certificates of the issuing certification authority. A Certificate Signing Request (CSR) is also supplied in PEM format, which is converted from PKCS10 format.  
The name originated from the abbreviation **Privacy-enhanced Electronic Mail (PEM)**, which was the standard for email security. The main essence of the PEM format is the recoding of the binary format (ie ones and zeros) by the base64 method and the addition of an informative header and footer of the -----BEGIN PRIVATE KEY----- and -----END PRIVATE KEY----- or -----BEGIN CERTIFICATE----- and -----END CERTIFICATE-----.

Sample of the issued certificate in PEM format

-----BEGIN CERTIFICATE-----  
MIIF+TCCBOGgAwIBAgIRAOUXUXsbB/LpS0VTQsz/HFcwDQYJKoZIhvcNAQELBQAw  
gY8xCzAJBgNVBAYTAkdCMRswGQYDVQQIExJHcmVhdGVyIE1hbmNoZXN0ZXIxEDAO  
BgNVBAcTB1NhbGZvcmQxGDAWBgNVBAoTD1NlY3RpZ28gTGltaXRlZDE3MDUGA1UE  
AxMuU2VjdGlnbyBSU0EgRG9tYWluIFZhbGlkYXRpb24gU2VjdXJlIFNlcnZlciBD  
QTAeFw0xOTAxMTcwMDAwMDBaFw0xOTA0MTcyMzU5NTlaMFAxITAfBgNVBAsTGERv  
bWFpbiBDb250cm9sIFZhbGlkYXRlZDERMA8GA1UECxMIRnJlZSBTU0wxGDAWBgNV  
BAMTD3dlYi1zZWN1cml0eS5jejCCASIwDQYJKoZIhvcNAQEBBQADggEPADCCAQoC  
ggEBAMNC5twUz78YyvW9Y+avpBZLZjGFLZbNZN3tukWL/1wuwLUrhuCju1IDXWnJ  
a7vu4IFA/m/fgD68Y+I6BEF/tdw94TGc/X0n+Q326ZB3ff8e5+GF2o2oXQCUEX60  
wGv17zIx8jCYZtaP9rWekUzWmkNPagImboWeYSLWkt7GvdJCU7VY8kpKm7Y/JF/P  
Qs4Z5+d4HMsfknJ+PofI7Ve3wT0aPE4aiQ3+MWryxcnZYzH7xNpeB7UbkfFIeDki  
4X1vkVFM2Do07IkY9dO8d0UNI3lDDJDpxCCW4kVOl8yQTRtmyPZmtXk5uoyFcCEh  
KP5/T2gxNr9KNzIornE0F7LZfpMCAwEAAaOCAowwggKIMB8GA1UdIwQYMBaAFI2M  
XsRUrYrhd+mb+ZsF4bgBjWHhMB0GA1UdDgQWBBSHdiVoz7sMpYkdFsQWYHoMcJZU  
IzAOBgNVHQ8BAf8EBAMCBaAwDAYDVR0TAQH/BAIwADAdBgNVHSUEFjAUBggrBgEF  
BQcDAQYIKwYBBQUHAwIwSQYDVR0gBEIwQDA0BgsrBgEEAbIxAQICBzAlMCMGCCsG  
AQUFBwIBFhdodHRwczovL3NlY3RpZ28uY29tL0NQUzAIBgZngQwBAgEwgYQGCCsG  
AQUFBwEBBHgwdjBPBggrBgEFBQcwAoZDaHR0cDovL2NydC5zZWN0aWdvLmNvbS9T  
ZWN0aWdvUlNBRG9tYWluVmFsaWRhdGlvblNlY3VyZVNlcnZlckNBLmNydDAjBggr  
BgEFBQcwAYYXaHR0cDovL29jc3Auc2VjdGlnby5jb20wLwYDVR0RBCgwJoIPd2Vi  
LXNlY3VyaXR5LmN6ghN3d3cud2ViLXNlY3VyaXR5LmN6MIIBBAYKKwYBBAHWeQIE  
AgSB9QSB8gDwAHYAu9nfvB+KcbWTlCOXqpJ7RzhXlQqrUugakJZkNo4e0YUAAAFo  
Ww+ePQAABAMARzBFAiEAtZoULgmDyEppYK9nmDWNRjRGcE+BBo/DLaaoaYWg7C8C  
IH+0UCda6Txcl05inAsMpzlePELyZ3hawgEdmMugK3bXAHYAdH7agzGtMxCRIZzO  
JU9CcMK//V5CIAjGNzV55hB7zFYAAAFoWw+eiwAABAMARzBFAiEAmdEbql+1pWWf  
HmpkY34JziCOQzaDlFVtUFen+blgIWwCIBVnwDD3vnwSsrO2T5Clo5Pjqa+xxU7O  
trLo/ZRGkICBMA0GCSqGSIb3DQEBCwUAA4IBAQAzOM9lS3RSU7rLy8T3BfixHvua  
ErZ+YOHCHpYhlCeSuFZ66jVHueYWvgfF8A+enRdMM0k0z0PC9enREnumNDq3msCf  
WYhSLd5lDXiEddg2GCrXkwhOFfOiG0tywS5CD+hLsTq1LQkWDQg7EKlIb6ddhaZO  
IYEQ9xwE7aehynQEvAjv3UyevMYfvw7glY+MW5bkMfsxPndDD1gDbnYt8kyenjcv  
odjnkvTw4ngnCy1gF9mVWkgQsE1j34FER1bVtR/FlspI0FB+ogV4Qhso1N23DwtF  
VDKxH8p+ddYh1LX4b6Oy3dZqzt4HOcunPKsFv36ABpeTs8FPOjgQueTWfHQ4  
-----END CERTIFICATE-----

You can decode this text string, for example [**on this page**](https://redkestrel.co.uk/products/decoder/), where you can find information about the certificate (validity, information in the certificate, authority, and much more).

PEM files are encoded in Base64 format, which is an encoding that converts binary data into a sequence of printable ASCII characters (a 64-element character set consisting of uppercase and lowercase letters of the English alphabet, numbers and plus signs ('+'), and a slash ('/')). PEM files are very easy to work with, as they have content in a readable text format and can be opened in any editor.The individual certificates are then clearly separated by a header and footer. More about the [**PEM format in WIKI ...**](https://en.wikipedia.org/wiki/Privacy-enhanced_Electronic_Mail)

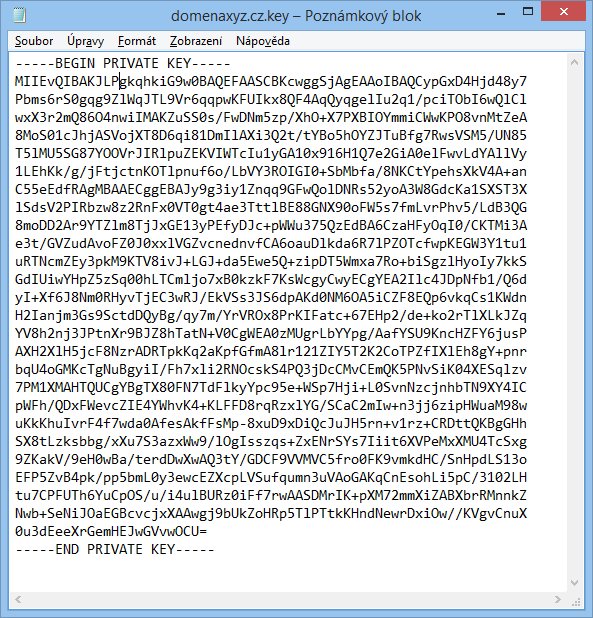
PFX (.pfx) / PKCS #12 format

.pfx, but also .p12 or .pkcs12 are formats defined in Public-Key Cryptography Standards (PKCS standards). It is a password container format that contains both public and private certificates. Unlike .pem files, the container is fully encrypted. **PKCS#12 (.p12)** was originally a private Microsoft standard that was later defined in [**RFC 7292**](https://tools.ietf.org/html/rfc7292). Provides improved security over the PEM text format.  
We will encounter the PFX format mainly on the Windows platform. If the certificate request is not generated directly in the Internet Information System (IIS), it is necessary to supply the server administrator with a certificate in PFX format for import into the server. For these cases, we publish in the help instructions on [**how to export the certificate to PFX using OpenSSL**](https://www.sslmentor.com/help/openssl-certificate-export-to-pfx).  
Code Signing certificates and electronic signatures are also exported to the .p12 / .pfx file.

The .pfx and .p12 files are de facto identical, and if you need the p12 file instead of the pfx, you may read that you just need to rename it. It doesn't always work that easily. You can learn more in the discussion at [**stackoverflow.com**](https://stackoverflow.com/questions/6819079/convert-pfx-format-to-p12).

KEY (.key)

The .key file contains the certificate in PEM format and contains only the private key of the certificate. The private key is enclosed in the strings ----- BEGIN PRIVATE KEY ----- and ----- END PRIVATE KEY -----. This file should go open in any text editor.  
There is no standardization for the .key format and it is de facto a designation of the file with the private key.

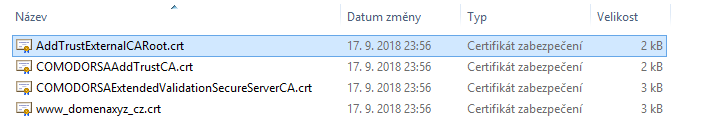


DER (.der)

DER (Distinguished Encoding Rules). A binary file (a string of zeros and ones) that contains the stored certificate information. It contains an [**SSL certificate**](https://www.sslmentor.com/) or the full root-chain path (intermediate certificates) and can also contain a private key. Used in the Unix world or on Java platforms, in Windows the .der file is automatically considered a certificate holder. DER is a defective binary version of a base64 encoded PEM file.

CRT (.crt)

The .crt file contains an SSL certificate in PEM format. They can be opened with any text editor and the certificate is enclosed in ----- BEGIN CERTIFICATE ----- and ----- END CERTIFICATE ----- tags.  
In Windows, when you double-click on a file and accept the warning, a window with the certificate details opens automatically. If you rename the .crt file to .txt, double-clicking opens a text editor with PEM content.



P7B (.p7b)

The P7B format contains the public key and intermediate certificates from the certification authority. Does not contain a private key. The P7B / PKCS # 7 format is saved in Base64 ASCII format and the file has a .p7b or .p7c extension. Defined in RFC 2315 as PKCS number 7. The format used by Windows. Java uses .keystore. It is possible to define a certificate hierarchy for these containers.

CER (.cer), CERT (.cert)

This is a different .pem file extension. Used to indicate the issued certificate. The stored certificate in PEM format is delimited by the header and footer ----- BEGIN CERTIFICATE ----- and ----- END CERTIFICATE -----.

Other file types and formats

CRL

Certificate Revocation List (CRL) - list of revoked certificates. Certification authorities publish lists of revoked certificates in these lists.

RFC 7468

The proposed standard [**RFC 7468**](https://tools.ietf.org/html/rfc7468) (Textual Encodings of PKIX, PKCS, and CMS Structures) describes and standardizes the text coding PKI (Public-Key Infrastructure X.509), PKCS (Public-Key Cryptography Standards) and CMS (Cryptographic Message Syntax).