**Java Certificate Factory**

The *Java CertificateFactory* class (java.security.cert.CertificateFactory) is capable of creating Java Certificate instances from binary certificate encodings like X.509 (ASN.1 DER). To read more about the Java Certificate class, see the [**Java Certificate tutorial**](http://tutorials.jenkov.com/java-cryptography/certificate.html).

The Java CertificateFactory can also create CertPath instances. A CertPath is a chain of certificates where each certificate in the chain is signed by the next certificate in the chain. See the [**Java CertPath tutorial**](http://tutorials.jenkov.com/java-cryptography/certpath.html) for more information about the CertPath class.

**Creating a CertificateFactory Instance**

Before you can create Certificate instances you must create a Java CertificateFactory instance. Here is an example of creating a CertificateFactory:

CertificateFactory certificateFactory = CertificateFactory.getInstance("X.509");

This example creates a CertificateFactory instance capable of creating X.509 certificate instances (X509Certificate - a subclass of Certificate).

**Creating a Certificate Instance**

Once you have created a CertificateFactory instance you can start creating Certificate instances. You do so via the generateCertificate() method. Here is a generateCertificate() example:

InputStream certificateInputStream = new FileInputStream("my-x509-certificate.crt");

Certificate certificate = certificateFactory.generateCertificate(certificateInputStream);

**Creating a CertPath Instance**

The Java CertificateFactory can also create a CertPath instance. You create a CertPath instance by calling the CertificateFactory generateCertPath() method. Here is a generateCertPath() example:

InputStream certificateInputStream = new FileInputStream("my-x509-certificate-chain.crt");

CertPath certPath = certificateFactory.generateCertPath(certificateInputStream);

## The CertificateFactory Class

The CertificateFactory class is an [engine class](https://docs.oracle.com/javase/7/docs/technotes/guides/security/crypto/CryptoSpec.html#Engine) that defines the functionality of a certificate factory, which is used to generate certificate and certificate revocation list (CRL) objects from their encodings.

A certificate factory for X.509 must return certificates that are an instance of java.security.cert.X509Certificate, and CRLs that are an instance of java.security.cert.X509CRL.

### Creating a CertificateFactory Object

CertificateFactory objects are obtained by using one of the [getInstance() static factory methods](https://docs.oracle.com/javase/7/docs/technotes/guides/security/crypto/CryptoSpec.html#ProviderImplReq).

### Generating Certificate Objects

To generate a certificate object and initialize it with the data read from an input stream, use the generateCertificate method:

final Certificate generateCertificate(InputStream inStream)

To return a (possibly empty) collection view of the certificates read from a given input stream, use the generateCertificates method:

final Collection generateCertificates(InputStream inStream)

### Generating CRL Objects

To generate a certificate revocation list (CRL) object and initialize it with the data read from an input stream, use the generateCRL method:

final CRL generateCRL(InputStream inStream)

To return a (possibly empty) collection view of the CRLs read from a given input stream, use the generateCRLs method:

final Collection generateCRLs(InputStream inStream)

### Generating CertPath Objects

The certificate path builder and validator for PKIX is defined by the Internet X.509 Public Key Infrastructure Certificate and CRL Profile, [RFC 3280](http://www.ietf.org/rfc/rfc3280.txt).

A certificate store implementation for retrieving certificates and CRLs from Collection and LDAP directories, using the PKIX LDAP V2 Schema is also available from the IETF as [RFC 2587](http://www.ietf.org/rfc/rfc2587.txt).

To generate a CertPath object and initialize it with data read from an input stream, use one of the following generateCertPath methods (with or without specifying the encoding to be used for the data):

final CertPath generateCertPath(InputStream inStream)

final CertPath generateCertPath(InputStream inStream,

String encoding)

To generate a CertPath object and initialize it with a list of certificates, use the following method:

final CertPath generateCertPath(List certificates)

To retrieve a list of the CertPath encodings supported by this certificate factory, you can call the getCertPathEncodings method:

final Iterator getCertPathEncodings()