| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/X509CRL.html) | [**Tree**](http://docs.google.com/package-tree.html) | [**Deprecated**](http://docs.google.com/deprecated-list.html) | [**Index**](http://docs.google.com/index-files/index-1.html) | [**Help**](http://docs.google.com/help-doc.html) | | --- | --- | --- | --- | --- | --- | --- | --- | | | ***Java™ Platform***  ***Standard Ed. 6*** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| [**PREV CLASS**](http://docs.google.com/java/security/cert/X509CertSelector.html)   [**NEXT CLASS**](http://docs.google.com/java/security/cert/X509CRLEntry.html) | [**FRAMES**](http://docs.google.com/index.html?java/security/cert/X509CRL.html)    [**NO FRAMES**](http://docs.google.com/X509CRL.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | FIELD | [CONSTR](#3znysh7) | [METHOD](#2et92p0) | DETAIL: FIELD | [CONSTR](#4d34og8) | [METHOD](#17dp8vu) |

## **java.security.cert**

Class X509CRL

[java.lang.Object](http://docs.google.com/java/lang/Object.html)  
 [java.security.cert.CRL](http://docs.google.com/java/security/cert/CRL.html)  
 **java.security.cert.X509CRL**

**All Implemented Interfaces:** [X509Extension](http://docs.google.com/java/security/cert/X509Extension.html)

public abstract class **X509CRL**extends [CRL](http://docs.google.com/java/security/cert/CRL.html)implements [X509Extension](http://docs.google.com/java/security/cert/X509Extension.html)

Abstract class for an X.509 Certificate Revocation List (CRL). A CRL is a time-stamped list identifying revoked certificates. It is signed by a Certificate Authority (CA) and made freely available in a public repository.

Each revoked certificate is identified in a CRL by its certificate serial number. When a certificate-using system uses a certificate (e.g., for verifying a remote user's digital signature), that system not only checks the certificate signature and validity but also acquires a suitably- recent CRL and checks that the certificate serial number is not on that CRL. The meaning of "suitably-recent" may vary with local policy, but it usually means the most recently-issued CRL. A CA issues a new CRL on a regular periodic basis (e.g., hourly, daily, or weekly). Entries are added to CRLs as revocations occur, and an entry may be removed when the certificate expiration date is reached.

The X.509 v2 CRL format is described below in ASN.1:

CertificateList ::= SEQUENCE {  
 tbsCertList TBSCertList,  
 signatureAlgorithm AlgorithmIdentifier,  
 signature BIT STRING }

More information can be found in RFC 2459, "Internet X.509 Public Key Infrastructure Certificate and CRL Profile" at <http://www.ietf.org/rfc/rfc2459.txt>.

The ASN.1 definition of tbsCertList is:

TBSCertList ::= SEQUENCE {  
 version Version OPTIONAL,  
 -- if present, must be v2  
 signature AlgorithmIdentifier,  
 issuer Name,  
 thisUpdate ChoiceOfTime,  
 nextUpdate ChoiceOfTime OPTIONAL,  
 revokedCertificates SEQUENCE OF SEQUENCE {  
 userCertificate CertificateSerialNumber,  
 revocationDate ChoiceOfTime,  
 crlEntryExtensions Extensions OPTIONAL  
 -- if present, must be v2  
 } OPTIONAL,  
 crlExtensions [0] EXPLICIT Extensions OPTIONAL  
 -- if present, must be v2  
 }

CRLs are instantiated using a certificate factory. The following is an example of how to instantiate an X.509 CRL:

InputStream inStream = new FileInputStream("fileName-of-crl");  
 CertificateFactory cf = CertificateFactory.getInstance("X.509");  
 X509CRL crl = (X509CRL)cf.generateCRL(inStream);  
 inStream.close();

**See Also:**[CRL](http://docs.google.com/java/security/cert/CRL.html), [CertificateFactory](http://docs.google.com/java/security/cert/CertificateFactory.html), [X509Extension](http://docs.google.com/java/security/cert/X509Extension.html)

| **Constructor Summary** | |
| --- | --- |
| protected | [**X509CRL**](http://docs.google.com/java/security/cert/X509CRL.html#X509CRL())()            Constructor for X.509 CRLs. |

| **Method Summary** | |
| --- | --- |
| boolean | [**equals**](http://docs.google.com/java/security/cert/X509CRL.html#equals(java.lang.Object))([Object](http://docs.google.com/java/lang/Object.html) other)            Compares this CRL for equality with the given object. |
| abstract  byte[] | [**getEncoded**](http://docs.google.com/java/security/cert/X509CRL.html#getEncoded())()            Returns the ASN.1 DER-encoded form of this CRL. |
| abstract  [Principal](http://docs.google.com/java/security/Principal.html) | [**getIssuerDN**](http://docs.google.com/java/security/cert/X509CRL.html#getIssuerDN())()  **Denigrated**, replaced by [getIssuerX500Principal()](http://docs.google.com/java/security/cert/X509CRL.html#getIssuerX500Principal()). |
| [X500Principal](http://docs.google.com/javax/security/auth/x500/X500Principal.html) | [**getIssuerX500Principal**](http://docs.google.com/java/security/cert/X509CRL.html#getIssuerX500Principal())()            Returns the issuer (issuer distinguished name) value from the CRL as an X500Principal. |
| abstract  [Date](http://docs.google.com/java/util/Date.html) | [**getNextUpdate**](http://docs.google.com/java/security/cert/X509CRL.html#getNextUpdate())()            Gets the nextUpdate date from the CRL. |
| abstract  [X509CRLEntry](http://docs.google.com/java/security/cert/X509CRLEntry.html) | [**getRevokedCertificate**](http://docs.google.com/java/security/cert/X509CRL.html#getRevokedCertificate(java.math.BigInteger))([BigInteger](http://docs.google.com/java/math/BigInteger.html) serialNumber)            Gets the CRL entry, if any, with the given certificate serialNumber. |
| [X509CRLEntry](http://docs.google.com/java/security/cert/X509CRLEntry.html) | [**getRevokedCertificate**](http://docs.google.com/java/security/cert/X509CRL.html#getRevokedCertificate(java.security.cert.X509Certificate))([X509Certificate](http://docs.google.com/java/security/cert/X509Certificate.html) certificate)            Get the CRL entry, if any, for the given certificate. |
| abstract  [Set](http://docs.google.com/java/util/Set.html)<? extends [X509CRLEntry](http://docs.google.com/java/security/cert/X509CRLEntry.html)> | [**getRevokedCertificates**](http://docs.google.com/java/security/cert/X509CRL.html#getRevokedCertificates())()            Gets all the entries from this CRL. |
| abstract  [String](http://docs.google.com/java/lang/String.html) | [**getSigAlgName**](http://docs.google.com/java/security/cert/X509CRL.html#getSigAlgName())()            Gets the signature algorithm name for the CRL signature algorithm. |
| abstract  [String](http://docs.google.com/java/lang/String.html) | [**getSigAlgOID**](http://docs.google.com/java/security/cert/X509CRL.html#getSigAlgOID())()            Gets the signature algorithm OID string from the CRL. |
| abstract  byte[] | [**getSigAlgParams**](http://docs.google.com/java/security/cert/X509CRL.html#getSigAlgParams())()            Gets the DER-encoded signature algorithm parameters from this CRL's signature algorithm. |
| abstract  byte[] | [**getSignature**](http://docs.google.com/java/security/cert/X509CRL.html#getSignature())()            Gets the signature value (the raw signature bits) from the CRL. |
| abstract  byte[] | [**getTBSCertList**](http://docs.google.com/java/security/cert/X509CRL.html#getTBSCertList())()            Gets the DER-encoded CRL information, the tbsCertList from this CRL. |
| abstract  [Date](http://docs.google.com/java/util/Date.html) | [**getThisUpdate**](http://docs.google.com/java/security/cert/X509CRL.html#getThisUpdate())()            Gets the thisUpdate date from the CRL. |
| abstract  int | [**getVersion**](http://docs.google.com/java/security/cert/X509CRL.html#getVersion())()            Gets the version (version number) value from the CRL. |
| int | [**hashCode**](http://docs.google.com/java/security/cert/X509CRL.html#hashCode())()            Returns a hashcode value for this CRL from its encoded form. |
| abstract  void | [**verify**](http://docs.google.com/java/security/cert/X509CRL.html#verify(java.security.PublicKey))([PublicKey](http://docs.google.com/java/security/PublicKey.html) key)            Verifies that this CRL was signed using the private key that corresponds to the given public key. |
| abstract  void | [**verify**](http://docs.google.com/java/security/cert/X509CRL.html#verify(java.security.PublicKey,%20java.lang.String))([PublicKey](http://docs.google.com/java/security/PublicKey.html) key, [String](http://docs.google.com/java/lang/String.html) sigProvider)            Verifies that this CRL was signed using the private key that corresponds to the given public key. |

| **Methods inherited from class java.security.cert.**[**CRL**](http://docs.google.com/java/security/cert/CRL.html) |
| --- |
| [getType](http://docs.google.com/java/security/cert/CRL.html#getType()), [isRevoked](http://docs.google.com/java/security/cert/CRL.html#isRevoked(java.security.cert.Certificate)), [toString](http://docs.google.com/java/security/cert/CRL.html#toString()) |

| **Methods inherited from class java.lang.**[**Object**](http://docs.google.com/java/lang/Object.html) |
| --- |
| [clone](http://docs.google.com/java/lang/Object.html#clone()), [finalize](http://docs.google.com/java/lang/Object.html#finalize()), [getClass](http://docs.google.com/java/lang/Object.html#getClass()), [notify](http://docs.google.com/java/lang/Object.html#notify()), [notifyAll](http://docs.google.com/java/lang/Object.html#notifyAll()), [wait](http://docs.google.com/java/lang/Object.html#wait()), [wait](http://docs.google.com/java/lang/Object.html#wait(long)), [wait](http://docs.google.com/java/lang/Object.html#wait(long,%20int)) |

| **Methods inherited from interface java.security.cert.**[**X509Extension**](http://docs.google.com/java/security/cert/X509Extension.html) |
| --- |
| [getCriticalExtensionOIDs](http://docs.google.com/java/security/cert/X509Extension.html#getCriticalExtensionOIDs()), [getExtensionValue](http://docs.google.com/java/security/cert/X509Extension.html#getExtensionValue(java.lang.String)), [getNonCriticalExtensionOIDs](http://docs.google.com/java/security/cert/X509Extension.html#getNonCriticalExtensionOIDs()), [hasUnsupportedCriticalExtension](http://docs.google.com/java/security/cert/X509Extension.html#hasUnsupportedCriticalExtension()) |

| **Constructor Detail** |
| --- |

### X509CRL

protected **X509CRL**()

Constructor for X.509 CRLs.

| **Method Detail** |
| --- |

### equals

public boolean **equals**([Object](http://docs.google.com/java/lang/Object.html) other)

Compares this CRL for equality with the given object. If the other object is an instanceof X509CRL, then its encoded form is retrieved and compared with the encoded form of this CRL.

**Overrides:**[equals](http://docs.google.com/java/lang/Object.html#equals(java.lang.Object)) in class [Object](http://docs.google.com/java/lang/Object.html) **Parameters:**other - the object to test for equality with this CRL. **Returns:**true iff the encoded forms of the two CRLs match, false otherwise.**See Also:**[Object.hashCode()](http://docs.google.com/java/lang/Object.html#hashCode()), [Hashtable](http://docs.google.com/java/util/Hashtable.html)

### hashCode

public int **hashCode**()

Returns a hashcode value for this CRL from its encoded form.

**Overrides:**[hashCode](http://docs.google.com/java/lang/Object.html#hashCode()) in class [Object](http://docs.google.com/java/lang/Object.html) **Returns:**the hashcode value.**See Also:**[Object.equals(java.lang.Object)](http://docs.google.com/java/lang/Object.html#equals(java.lang.Object)), [Hashtable](http://docs.google.com/java/util/Hashtable.html)

### getEncoded

public abstract byte[] **getEncoded**()  
 throws [CRLException](http://docs.google.com/java/security/cert/CRLException.html)

Returns the ASN.1 DER-encoded form of this CRL.

**Returns:**the encoded form of this certificate **Throws:** [CRLException](http://docs.google.com/java/security/cert/CRLException.html) - if an encoding error occurs.

### verify

public abstract void **verify**([PublicKey](http://docs.google.com/java/security/PublicKey.html) key)  
 throws [CRLException](http://docs.google.com/java/security/cert/CRLException.html),  
 [NoSuchAlgorithmException](http://docs.google.com/java/security/NoSuchAlgorithmException.html),  
 [InvalidKeyException](http://docs.google.com/java/security/InvalidKeyException.html),  
 [NoSuchProviderException](http://docs.google.com/java/security/NoSuchProviderException.html),  
 [SignatureException](http://docs.google.com/java/security/SignatureException.html)

Verifies that this CRL was signed using the private key that corresponds to the given public key.

**Parameters:**key - the PublicKey used to carry out the verification. **Throws:** [NoSuchAlgorithmException](http://docs.google.com/java/security/NoSuchAlgorithmException.html) - on unsupported signature algorithms. [InvalidKeyException](http://docs.google.com/java/security/InvalidKeyException.html) - on incorrect key. [NoSuchProviderException](http://docs.google.com/java/security/NoSuchProviderException.html) - if there's no default provider. [SignatureException](http://docs.google.com/java/security/SignatureException.html) - on signature errors. [CRLException](http://docs.google.com/java/security/cert/CRLException.html) - on encoding errors.

### verify

public abstract void **verify**([PublicKey](http://docs.google.com/java/security/PublicKey.html) key,  
 [String](http://docs.google.com/java/lang/String.html) sigProvider)  
 throws [CRLException](http://docs.google.com/java/security/cert/CRLException.html),  
 [NoSuchAlgorithmException](http://docs.google.com/java/security/NoSuchAlgorithmException.html),  
 [InvalidKeyException](http://docs.google.com/java/security/InvalidKeyException.html),  
 [NoSuchProviderException](http://docs.google.com/java/security/NoSuchProviderException.html),  
 [SignatureException](http://docs.google.com/java/security/SignatureException.html)

Verifies that this CRL was signed using the private key that corresponds to the given public key. This method uses the signature verification engine supplied by the given provider.

**Parameters:**key - the PublicKey used to carry out the verification.sigProvider - the name of the signature provider. **Throws:** [NoSuchAlgorithmException](http://docs.google.com/java/security/NoSuchAlgorithmException.html) - on unsupported signature algorithms. [InvalidKeyException](http://docs.google.com/java/security/InvalidKeyException.html) - on incorrect key. [NoSuchProviderException](http://docs.google.com/java/security/NoSuchProviderException.html) - on incorrect provider. [SignatureException](http://docs.google.com/java/security/SignatureException.html) - on signature errors. [CRLException](http://docs.google.com/java/security/cert/CRLException.html) - on encoding errors.

### getVersion

public abstract int **getVersion**()

Gets the version (version number) value from the CRL. The ASN.1 definition for this is:

version Version OPTIONAL,  
 -- if present, must be v2

Version ::= INTEGER { v1(0), v2(1), v3(2) }  
 -- v3 does not apply to CRLs but appears for consistency  
 -- with definition of Version for certs

**Returns:**the version number, i.e. 1 or 2.

### getIssuerDN

public abstract [Principal](http://docs.google.com/java/security/Principal.html) **getIssuerDN**()

**Denigrated**, replaced by [getIssuerX500Principal()](http://docs.google.com/java/security/cert/X509CRL.html#getIssuerX500Principal()). This method returns the issuer as an implementation specific Principal object, which should not be relied upon by portable code.

Gets the issuer (issuer distinguished name) value from the CRL. The issuer name identifies the entity that signed (and issued) the CRL.

The issuer name field contains an X.500 distinguished name (DN). The ASN.1 definition for this is:

issuer Name  
  
 Name ::= CHOICE { RDNSequence }  
 RDNSequence ::= SEQUENCE OF RelativeDistinguishedName  
 RelativeDistinguishedName ::=  
 SET OF AttributeValueAssertion  
  
 AttributeValueAssertion ::= SEQUENCE {  
 AttributeType,  
 AttributeValue }  
 AttributeType ::= OBJECT IDENTIFIER  
 AttributeValue ::= ANY

The Name describes a hierarchical name composed of attributes, such as country name, and corresponding values, such as US. The type of the AttributeValue component is determined by the AttributeType; in general it will be a directoryString. A directoryString is usually one of PrintableString, TeletexString or UniversalString.

**Returns:**a Principal whose name is the issuer distinguished name.

### getIssuerX500Principal

public [X500Principal](http://docs.google.com/javax/security/auth/x500/X500Principal.html) **getIssuerX500Principal**()

Returns the issuer (issuer distinguished name) value from the CRL as an X500Principal.

It is recommended that subclasses override this method.

**Returns:**an X500Principal representing the issuer distinguished name**Since:** 1.4

### getThisUpdate

public abstract [Date](http://docs.google.com/java/util/Date.html) **getThisUpdate**()

Gets the thisUpdate date from the CRL. The ASN.1 definition for this is:

thisUpdate ChoiceOfTime  
 ChoiceOfTime ::= CHOICE {  
 utcTime UTCTime,  
 generalTime GeneralizedTime }

**Returns:**the thisUpdate date from the CRL.

### getNextUpdate

public abstract [Date](http://docs.google.com/java/util/Date.html) **getNextUpdate**()

Gets the nextUpdate date from the CRL.

**Returns:**the nextUpdate date from the CRL, or null if not present.

### getRevokedCertificate

public abstract [X509CRLEntry](http://docs.google.com/java/security/cert/X509CRLEntry.html) **getRevokedCertificate**([BigInteger](http://docs.google.com/java/math/BigInteger.html) serialNumber)

Gets the CRL entry, if any, with the given certificate serialNumber.

**Parameters:**serialNumber - the serial number of the certificate for which a CRL entry is to be looked up **Returns:**the entry with the given serial number, or null if no such entry exists in this CRL.**See Also:**[X509CRLEntry](http://docs.google.com/java/security/cert/X509CRLEntry.html)

### getRevokedCertificate

public [X509CRLEntry](http://docs.google.com/java/security/cert/X509CRLEntry.html) **getRevokedCertificate**([X509Certificate](http://docs.google.com/java/security/cert/X509Certificate.html) certificate)

Get the CRL entry, if any, for the given certificate.

This method can be used to lookup CRL entries in indirect CRLs, that means CRLs that contain entries from issuers other than the CRL issuer. The default implementation will only return entries for certificates issued by the CRL issuer. Subclasses that wish to support indirect CRLs should override this method.

**Parameters:**certificate - the certificate for which a CRL entry is to be looked up **Returns:**the entry for the given certificate, or null if no such entry exists in this CRL. **Throws:** [NullPointerException](http://docs.google.com/java/lang/NullPointerException.html) - if certificate is null**Since:** 1.5

### getRevokedCertificates

public abstract [Set](http://docs.google.com/java/util/Set.html)<? extends [X509CRLEntry](http://docs.google.com/java/security/cert/X509CRLEntry.html)> **getRevokedCertificates**()

Gets all the entries from this CRL. This returns a Set of X509CRLEntry objects.

**Returns:**all the entries or null if there are none present.**See Also:**[X509CRLEntry](http://docs.google.com/java/security/cert/X509CRLEntry.html)

### getTBSCertList

public abstract byte[] **getTBSCertList**()  
 throws [CRLException](http://docs.google.com/java/security/cert/CRLException.html)

Gets the DER-encoded CRL information, the tbsCertList from this CRL. This can be used to verify the signature independently.

**Returns:**the DER-encoded CRL information. **Throws:** [CRLException](http://docs.google.com/java/security/cert/CRLException.html) - if an encoding error occurs.

### getSignature

public abstract byte[] **getSignature**()

Gets the signature value (the raw signature bits) from the CRL. The ASN.1 definition for this is:

signature BIT STRING

**Returns:**the signature.

### getSigAlgName

public abstract [String](http://docs.google.com/java/lang/String.html) **getSigAlgName**()

Gets the signature algorithm name for the CRL signature algorithm. An example is the string "SHA-1/DSA". The ASN.1 definition for this is:

signatureAlgorithm AlgorithmIdentifier

AlgorithmIdentifier ::= SEQUENCE {  
 algorithm OBJECT IDENTIFIER,  
 parameters ANY DEFINED BY algorithm OPTIONAL }  
 -- contains a value of the type  
 -- registered for use with the  
 -- algorithm object identifier value

The algorithm name is determined from the algorithm OID string.

**Returns:**the signature algorithm name.

### getSigAlgOID

public abstract [String](http://docs.google.com/java/lang/String.html) **getSigAlgOID**()

Gets the signature algorithm OID string from the CRL. An OID is represented by a set of nonnegative whole numbers separated by periods. For example, the string "1.2.840.10040.4.3" identifies the SHA-1 with DSA signature algorithm, as per RFC 2459.

See [getSigAlgName](http://docs.google.com/java/security/cert/X509CRL.html#getSigAlgName()) for relevant ASN.1 definitions.

**Returns:**the signature algorithm OID string.

### getSigAlgParams

public abstract byte[] **getSigAlgParams**()

Gets the DER-encoded signature algorithm parameters from this CRL's signature algorithm. In most cases, the signature algorithm parameters are null; the parameters are usually supplied with the public key. If access to individual parameter values is needed then use [AlgorithmParameters](http://docs.google.com/java/security/AlgorithmParameters.html) and instantiate with the name returned by [getSigAlgName](http://docs.google.com/java/security/cert/X509CRL.html#getSigAlgName()).

See [getSigAlgName](http://docs.google.com/java/security/cert/X509CRL.html#getSigAlgName()) for relevant ASN.1 definitions.

**Returns:**the DER-encoded signature algorithm parameters, or null if no parameters are present.

| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/X509CRL.html) | [**Tree**](http://docs.google.com/package-tree.html) | [**Deprecated**](http://docs.google.com/deprecated-list.html) | [**Index**](http://docs.google.com/index-files/index-1.html) | [**Help**](http://docs.google.com/help-doc.html) | | --- | --- | --- | --- | --- | --- | --- | --- | | | ***Java™ Platform***  ***Standard Ed. 6*** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| [**PREV CLASS**](http://docs.google.com/java/security/cert/X509CertSelector.html)   [**NEXT CLASS**](http://docs.google.com/java/security/cert/X509CRLEntry.html) | [**FRAMES**](http://docs.google.com/index.html?java/security/cert/X509CRL.html)    [**NO FRAMES**](http://docs.google.com/X509CRL.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | FIELD | [CONSTR](#3znysh7) | [METHOD](#2et92p0) | DETAIL: FIELD | [CONSTR](#4d34og8) | [METHOD](#17dp8vu) |

[Submit a bug or feature](http://bugs.sun.com/services/bugreport/index.jsp)

For further API reference and developer documentation, see [Java SE Developer Documentation](http://docs.google.com/webnotes/devdocs-vs-specs.html). That documentation contains more detailed, developer-targeted descriptions, with conceptual overviews, definitions of terms, workarounds, and working code examples.

Copyright 2006 Sun Microsystems, Inc. All rights reserved. Use is subject to [license terms](http://docs.google.com/legal/license.html). Also see the [documentation redistribution policy](http://java.sun.com/docs/redist.html).