| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/X509CertSelector.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/X509Certificate.html)   [**NEXT CLASS**](http://docs.google.com/java/security/cert/X509CRL.html) | [**FRAMES**](http://docs.google.com/index.html?java/security/cert/X509CertSelector.html)    [**NO FRAMES**](http://docs.google.com/X509CertSelector.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | FIELD | [CONSTR](#3znysh7) | [METHOD](#2et92p0) | DETAIL: FIELD | [CONSTR](#3dy6vkm) | [METHOD](#4d34og8) |

## **java.security.cert**

Class X509CertSelector

[java.lang.Object](http://docs.google.com/java/lang/Object.html)  
 **java.security.cert.X509CertSelector**

**All Implemented Interfaces:** [Cloneable](http://docs.google.com/java/lang/Cloneable.html), [CertSelector](http://docs.google.com/java/security/cert/CertSelector.html)

public class **X509CertSelector**extends [Object](http://docs.google.com/java/lang/Object.html)implements [CertSelector](http://docs.google.com/java/security/cert/CertSelector.html)

A CertSelector that selects X509Certificates that match all specified criteria. This class is particularly useful when selecting certificates from a CertStore to build a PKIX-compliant certification path.

When first constructed, an X509CertSelector has no criteria enabled and each of the get methods return a default value (null, or -1 for the [getBasicConstraints](http://docs.google.com/java/security/cert/X509CertSelector.html#getBasicConstraints()) method). Therefore, the [match](http://docs.google.com/java/security/cert/X509CertSelector.html#match(java.security.cert.Certificate)) method would return true for any X509Certificate. Typically, several criteria are enabled (by calling [setIssuer](http://docs.google.com/java/security/cert/X509CertSelector.html#setIssuer(javax.security.auth.x500.X500Principal)) or [setKeyUsage](http://docs.google.com/java/security/cert/X509CertSelector.html#setKeyUsage(boolean%5B%5D)), for instance) and then the X509CertSelector is passed to [CertStore.getCertificates](http://docs.google.com/java/security/cert/CertStore.html#getCertificates(java.security.cert.CertSelector)) or some similar method.

Several criteria can be enabled (by calling [setIssuer](http://docs.google.com/java/security/cert/X509CertSelector.html#setIssuer(javax.security.auth.x500.X500Principal)) and [setSerialNumber](http://docs.google.com/java/security/cert/X509CertSelector.html#setSerialNumber(java.math.BigInteger)), for example) such that the match method usually uniquely matches a single X509Certificate. We say usually, since it is possible for two issuing CAs to have the same distinguished name and each issue a certificate with the same serial number. Other unique combinations include the issuer, subject, subjectKeyIdentifier and/or the subjectPublicKey criteria.

Please refer to RFC 2459 for definitions of the X.509 certificate extensions mentioned below.

**Concurrent Access**

Unless otherwise specified, the methods defined in this class are not thread-safe. Multiple threads that need to access a single object concurrently should synchronize amongst themselves and provide the necessary locking. Multiple threads each manipulating separate objects need not synchronize.

**Since:** 1.4 **See Also:**[CertSelector](http://docs.google.com/java/security/cert/CertSelector.html), [X509Certificate](http://docs.google.com/java/security/cert/X509Certificate.html)

| **Constructor Summary** | |
| --- | --- |
| [**X509CertSelector**](http://docs.google.com/java/security/cert/X509CertSelector.html#X509CertSelector())()            Creates an X509CertSelector. |

| **Method Summary** | |
| --- | --- |
| void | [**addPathToName**](http://docs.google.com/java/security/cert/X509CertSelector.html#addPathToName(int,%20byte%5B%5D))(int type, byte[] name)            Adds a name to the pathToNames criterion. |
| void | [**addPathToName**](http://docs.google.com/java/security/cert/X509CertSelector.html#addPathToName(int,%20java.lang.String))(int type, [String](http://docs.google.com/java/lang/String.html) name)            Adds a name to the pathToNames criterion. |
| void | [**addSubjectAlternativeName**](http://docs.google.com/java/security/cert/X509CertSelector.html#addSubjectAlternativeName(int,%20byte%5B%5D))(int type, byte[] name)            Adds a name to the subjectAlternativeNames criterion. |
| void | [**addSubjectAlternativeName**](http://docs.google.com/java/security/cert/X509CertSelector.html#addSubjectAlternativeName(int,%20java.lang.String))(int type, [String](http://docs.google.com/java/lang/String.html) name)            Adds a name to the subjectAlternativeNames criterion. |
| [Object](http://docs.google.com/java/lang/Object.html) | [**clone**](http://docs.google.com/java/security/cert/X509CertSelector.html#clone())()            Returns a copy of this object. |
| byte[] | [**getAuthorityKeyIdentifier**](http://docs.google.com/java/security/cert/X509CertSelector.html#getAuthorityKeyIdentifier())()            Returns the authorityKeyIdentifier criterion. |
| int | [**getBasicConstraints**](http://docs.google.com/java/security/cert/X509CertSelector.html#getBasicConstraints())()            Returns the basic constraints constraint. |
| [X509Certificate](http://docs.google.com/java/security/cert/X509Certificate.html) | [**getCertificate**](http://docs.google.com/java/security/cert/X509CertSelector.html#getCertificate())()            Returns the certificateEquals criterion. |
| [Date](http://docs.google.com/java/util/Date.html) | [**getCertificateValid**](http://docs.google.com/java/security/cert/X509CertSelector.html#getCertificateValid())()            Returns the certificateValid criterion. |
| [Set](http://docs.google.com/java/util/Set.html)<[String](http://docs.google.com/java/lang/String.html)> | [**getExtendedKeyUsage**](http://docs.google.com/java/security/cert/X509CertSelector.html#getExtendedKeyUsage())()            Returns the extendedKeyUsage criterion. |
| [X500Principal](http://docs.google.com/javax/security/auth/x500/X500Principal.html) | [**getIssuer**](http://docs.google.com/java/security/cert/X509CertSelector.html#getIssuer())()            Returns the issuer criterion as an X500Principal. |
| byte[] | [**getIssuerAsBytes**](http://docs.google.com/java/security/cert/X509CertSelector.html#getIssuerAsBytes())()            Returns the issuer criterion as a byte array. |
| [String](http://docs.google.com/java/lang/String.html) | [**getIssuerAsString**](http://docs.google.com/java/security/cert/X509CertSelector.html#getIssuerAsString())()  **Denigrated**, use [getIssuer()](http://docs.google.com/java/security/cert/X509CertSelector.html#getIssuer()) or [getIssuerAsBytes()](http://docs.google.com/java/security/cert/X509CertSelector.html#getIssuerAsBytes()) instead. |
| boolean[] | [**getKeyUsage**](http://docs.google.com/java/security/cert/X509CertSelector.html#getKeyUsage())()            Returns the keyUsage criterion. |
| boolean | [**getMatchAllSubjectAltNames**](http://docs.google.com/java/security/cert/X509CertSelector.html#getMatchAllSubjectAltNames())()            Indicates if the X509Certificate must contain all or at least one of the subjectAlternativeNames specified in the [setSubjectAlternativeNames](http://docs.google.com/java/security/cert/X509CertSelector.html#setSubjectAlternativeNames(java.util.Collection)) or [addSubjectAlternativeName](http://docs.google.com/java/security/cert/X509CertSelector.html#addSubjectAlternativeName(int,%20java.lang.String)) methods. |
| byte[] | [**getNameConstraints**](http://docs.google.com/java/security/cert/X509CertSelector.html#getNameConstraints())()            Returns the name constraints criterion. |
| [Collection](http://docs.google.com/java/util/Collection.html)<[List](http://docs.google.com/java/util/List.html)<?>> | [**getPathToNames**](http://docs.google.com/java/security/cert/X509CertSelector.html#getPathToNames())()            Returns a copy of the pathToNames criterion. |
| [Set](http://docs.google.com/java/util/Set.html)<[String](http://docs.google.com/java/lang/String.html)> | [**getPolicy**](http://docs.google.com/java/security/cert/X509CertSelector.html#getPolicy())()            Returns the policy criterion. |
| [Date](http://docs.google.com/java/util/Date.html) | [**getPrivateKeyValid**](http://docs.google.com/java/security/cert/X509CertSelector.html#getPrivateKeyValid())()            Returns the privateKeyValid criterion. |
| [BigInteger](http://docs.google.com/java/math/BigInteger.html) | [**getSerialNumber**](http://docs.google.com/java/security/cert/X509CertSelector.html#getSerialNumber())()            Returns the serialNumber criterion. |
| [X500Principal](http://docs.google.com/javax/security/auth/x500/X500Principal.html) | [**getSubject**](http://docs.google.com/java/security/cert/X509CertSelector.html#getSubject())()            Returns the subject criterion as an X500Principal. |
| [Collection](http://docs.google.com/java/util/Collection.html)<[List](http://docs.google.com/java/util/List.html)<?>> | [**getSubjectAlternativeNames**](http://docs.google.com/java/security/cert/X509CertSelector.html#getSubjectAlternativeNames())()            Returns a copy of the subjectAlternativeNames criterion. |
| byte[] | [**getSubjectAsBytes**](http://docs.google.com/java/security/cert/X509CertSelector.html#getSubjectAsBytes())()            Returns the subject criterion as a byte array. |
| [String](http://docs.google.com/java/lang/String.html) | [**getSubjectAsString**](http://docs.google.com/java/security/cert/X509CertSelector.html#getSubjectAsString())()  **Denigrated**, use [getSubject()](http://docs.google.com/java/security/cert/X509CertSelector.html#getSubject()) or [getSubjectAsBytes()](http://docs.google.com/java/security/cert/X509CertSelector.html#getSubjectAsBytes()) instead. |
| byte[] | [**getSubjectKeyIdentifier**](http://docs.google.com/java/security/cert/X509CertSelector.html#getSubjectKeyIdentifier())()            Returns the subjectKeyIdentifier criterion. |
| [PublicKey](http://docs.google.com/java/security/PublicKey.html) | [**getSubjectPublicKey**](http://docs.google.com/java/security/cert/X509CertSelector.html#getSubjectPublicKey())()            Returns the subjectPublicKey criterion. |
| [String](http://docs.google.com/java/lang/String.html) | [**getSubjectPublicKeyAlgID**](http://docs.google.com/java/security/cert/X509CertSelector.html#getSubjectPublicKeyAlgID())()            Returns the subjectPublicKeyAlgID criterion. |
| boolean | [**match**](http://docs.google.com/java/security/cert/X509CertSelector.html#match(java.security.cert.Certificate))([Certificate](http://docs.google.com/java/security/cert/Certificate.html) cert)            Decides whether a Certificate should be selected. |
| void | [**setAuthorityKeyIdentifier**](http://docs.google.com/java/security/cert/X509CertSelector.html#setAuthorityKeyIdentifier(byte%5B%5D))(byte[] authorityKeyID)            Sets the authorityKeyIdentifier criterion. |
| void | [**setBasicConstraints**](http://docs.google.com/java/security/cert/X509CertSelector.html#setBasicConstraints(int))(int minMaxPathLen)            Sets the basic constraints constraint. |
| void | [**setCertificate**](http://docs.google.com/java/security/cert/X509CertSelector.html#setCertificate(java.security.cert.X509Certificate))([X509Certificate](http://docs.google.com/java/security/cert/X509Certificate.html) cert)            Sets the certificateEquals criterion. |
| void | [**setCertificateValid**](http://docs.google.com/java/security/cert/X509CertSelector.html#setCertificateValid(java.util.Date))([Date](http://docs.google.com/java/util/Date.html) certValid)            Sets the certificateValid criterion. |
| void | [**setExtendedKeyUsage**](http://docs.google.com/java/security/cert/X509CertSelector.html#setExtendedKeyUsage(java.util.Set))([Set](http://docs.google.com/java/util/Set.html)<[String](http://docs.google.com/java/lang/String.html)> keyPurposeSet)            Sets the extendedKeyUsage criterion. |
| void | [**setIssuer**](http://docs.google.com/java/security/cert/X509CertSelector.html#setIssuer(byte%5B%5D))(byte[] issuerDN)            Sets the issuer criterion. |
| void | [**setIssuer**](http://docs.google.com/java/security/cert/X509CertSelector.html#setIssuer(java.lang.String))([String](http://docs.google.com/java/lang/String.html) issuerDN)  **Denigrated**, use [setIssuer(X500Principal)](http://docs.google.com/java/security/cert/X509CertSelector.html#setIssuer(javax.security.auth.x500.X500Principal)) or [setIssuer(byte[])](http://docs.google.com/java/security/cert/X509CertSelector.html#setIssuer(byte%5B%5D)) instead. |
| void | [**setIssuer**](http://docs.google.com/java/security/cert/X509CertSelector.html#setIssuer(javax.security.auth.x500.X500Principal))([X500Principal](http://docs.google.com/javax/security/auth/x500/X500Principal.html) issuer)            Sets the issuer criterion. |
| void | [**setKeyUsage**](http://docs.google.com/java/security/cert/X509CertSelector.html#setKeyUsage(boolean%5B%5D))(boolean[] keyUsage)            Sets the keyUsage criterion. |
| void | [**setMatchAllSubjectAltNames**](http://docs.google.com/java/security/cert/X509CertSelector.html#setMatchAllSubjectAltNames(boolean))(boolean matchAllNames)            Enables/disables matching all of the subjectAlternativeNames specified in the [setSubjectAlternativeNames](http://docs.google.com/java/security/cert/X509CertSelector.html#setSubjectAlternativeNames(java.util.Collection)) or [addSubjectAlternativeName](http://docs.google.com/java/security/cert/X509CertSelector.html#addSubjectAlternativeName(int,%20java.lang.String)) methods. |
| void | [**setNameConstraints**](http://docs.google.com/java/security/cert/X509CertSelector.html#setNameConstraints(byte%5B%5D))(byte[] bytes)            Sets the name constraints criterion. |
| void | [**setPathToNames**](http://docs.google.com/java/security/cert/X509CertSelector.html#setPathToNames(java.util.Collection))([Collection](http://docs.google.com/java/util/Collection.html)<[List](http://docs.google.com/java/util/List.html)<?>> names)            Sets the pathToNames criterion. |
| void | [**setPolicy**](http://docs.google.com/java/security/cert/X509CertSelector.html#setPolicy(java.util.Set))([Set](http://docs.google.com/java/util/Set.html)<[String](http://docs.google.com/java/lang/String.html)> certPolicySet)            Sets the policy constraint. |
| void | [**setPrivateKeyValid**](http://docs.google.com/java/security/cert/X509CertSelector.html#setPrivateKeyValid(java.util.Date))([Date](http://docs.google.com/java/util/Date.html) privateKeyValid)            Sets the privateKeyValid criterion. |
| void | [**setSerialNumber**](http://docs.google.com/java/security/cert/X509CertSelector.html#setSerialNumber(java.math.BigInteger))([BigInteger](http://docs.google.com/java/math/BigInteger.html) serial)            Sets the serialNumber criterion. |
| void | [**setSubject**](http://docs.google.com/java/security/cert/X509CertSelector.html#setSubject(byte%5B%5D))(byte[] subjectDN)            Sets the subject criterion. |
| void | [**setSubject**](http://docs.google.com/java/security/cert/X509CertSelector.html#setSubject(java.lang.String))([String](http://docs.google.com/java/lang/String.html) subjectDN)  **Denigrated**, use [setSubject(X500Principal)](http://docs.google.com/java/security/cert/X509CertSelector.html#setSubject(javax.security.auth.x500.X500Principal)) or [setSubject(byte[])](http://docs.google.com/java/security/cert/X509CertSelector.html#setSubject(byte%5B%5D)) instead. |
| void | [**setSubject**](http://docs.google.com/java/security/cert/X509CertSelector.html#setSubject(javax.security.auth.x500.X500Principal))([X500Principal](http://docs.google.com/javax/security/auth/x500/X500Principal.html) subject)            Sets the subject criterion. |
| void | [**setSubjectAlternativeNames**](http://docs.google.com/java/security/cert/X509CertSelector.html#setSubjectAlternativeNames(java.util.Collection))([Collection](http://docs.google.com/java/util/Collection.html)<[List](http://docs.google.com/java/util/List.html)<?>> names)            Sets the subjectAlternativeNames criterion. |
| void | [**setSubjectKeyIdentifier**](http://docs.google.com/java/security/cert/X509CertSelector.html#setSubjectKeyIdentifier(byte%5B%5D))(byte[] subjectKeyID)            Sets the subjectKeyIdentifier criterion. |
| void | [**setSubjectPublicKey**](http://docs.google.com/java/security/cert/X509CertSelector.html#setSubjectPublicKey(byte%5B%5D))(byte[] key)            Sets the subjectPublicKey criterion. |
| void | [**setSubjectPublicKey**](http://docs.google.com/java/security/cert/X509CertSelector.html#setSubjectPublicKey(java.security.PublicKey))([PublicKey](http://docs.google.com/java/security/PublicKey.html) key)            Sets the subjectPublicKey criterion. |
| void | [**setSubjectPublicKeyAlgID**](http://docs.google.com/java/security/cert/X509CertSelector.html#setSubjectPublicKeyAlgID(java.lang.String))([String](http://docs.google.com/java/lang/String.html) oid)            Sets the subjectPublicKeyAlgID criterion. |
| [String](http://docs.google.com/java/lang/String.html) | [**toString**](http://docs.google.com/java/security/cert/X509CertSelector.html#toString())()            Return a printable representation of the CertSelector. |

| **Methods inherited from class java.lang.**[**Object**](http://docs.google.com/java/lang/Object.html) |
| --- |
| [equals](http://docs.google.com/java/lang/Object.html#equals(java.lang.Object)), [finalize](http://docs.google.com/java/lang/Object.html#finalize()), [getClass](http://docs.google.com/java/lang/Object.html#getClass()), [hashCode](http://docs.google.com/java/lang/Object.html#hashCode()), [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)) |

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

### X509CertSelector

public **X509CertSelector**()

Creates an X509CertSelector. Initially, no criteria are set so any X509Certificate will match.

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

### setCertificate

public void **setCertificate**([X509Certificate](http://docs.google.com/java/security/cert/X509Certificate.html) cert)

Sets the certificateEquals criterion. The specified X509Certificate must be equal to the X509Certificate passed to the match method. If null, then this check is not applied.

This method is particularly useful when it is necessary to match a single certificate. Although other criteria can be specified in conjunction with the certificateEquals criterion, it is usually not practical or necessary.

**Parameters:**cert - the X509Certificate to match (or null)**See Also:**[getCertificate()](http://docs.google.com/java/security/cert/X509CertSelector.html#getCertificate())

### setSerialNumber

public void **setSerialNumber**([BigInteger](http://docs.google.com/java/math/BigInteger.html) serial)

Sets the serialNumber criterion. The specified serial number must match the certificate serial number in the X509Certificate. If null, any certificate serial number will do.

**Parameters:**serial - the certificate serial number to match (or null)**See Also:**[getSerialNumber()](http://docs.google.com/java/security/cert/X509CertSelector.html#getSerialNumber())

### setIssuer

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

Sets the issuer criterion. The specified distinguished name must match the issuer distinguished name in the X509Certificate. If null, any issuer distinguished name will do.

**Parameters:**issuer - a distinguished name as X500Principal (or null)**Since:** 1.5

### setIssuer

public void **setIssuer**([String](http://docs.google.com/java/lang/String.html) issuerDN)  
 throws [IOException](http://docs.google.com/java/io/IOException.html)

**Denigrated**, use [setIssuer(X500Principal)](http://docs.google.com/java/security/cert/X509CertSelector.html#setIssuer(javax.security.auth.x500.X500Principal)) or [setIssuer(byte[])](http://docs.google.com/java/security/cert/X509CertSelector.html#setIssuer(byte%5B%5D)) instead. This method should not be relied on as it can fail to match some certificates because of a loss of encoding information in the RFC 2253 String form of some distinguished names.

Sets the issuer criterion. The specified distinguished name must match the issuer distinguished name in the X509Certificate. If null, any issuer distinguished name will do.

If issuerDN is not null, it should contain a distinguished name, in RFC 2253 format.

**Parameters:**issuerDN - a distinguished name in RFC 2253 format (or null) **Throws:** [IOException](http://docs.google.com/java/io/IOException.html) - if a parsing error occurs (incorrect form for DN)

### setIssuer

public void **setIssuer**(byte[] issuerDN)  
 throws [IOException](http://docs.google.com/java/io/IOException.html)

Sets the issuer criterion. The specified distinguished name must match the issuer distinguished name in the X509Certificate. If null is specified, the issuer criterion is disabled and any issuer distinguished name will do.

If issuerDN is not null, it should contain a single DER encoded distinguished name, as defined in X.501. The ASN.1 notation for this structure is as follows.

Name ::= CHOICE {  
 RDNSequence }  
  
 RDNSequence ::= SEQUENCE OF RelativeDistinguishedName  
  
 RelativeDistinguishedName ::=  
 SET SIZE (1 .. MAX) OF AttributeTypeAndValue  
  
 AttributeTypeAndValue ::= SEQUENCE {  
 type AttributeType,  
 value AttributeValue }  
  
 AttributeType ::= OBJECT IDENTIFIER  
  
 AttributeValue ::= ANY DEFINED BY AttributeType  
 ....  
 DirectoryString ::= CHOICE {  
 teletexString TeletexString (SIZE (1..MAX)),  
 printableString PrintableString (SIZE (1..MAX)),  
 universalString UniversalString (SIZE (1..MAX)),  
 utf8String UTF8String (SIZE (1.. MAX)),  
 bmpString BMPString (SIZE (1..MAX)) }

Note that the byte array specified here is cloned to protect against subsequent modifications.

**Parameters:**issuerDN - a byte array containing the distinguished name in ASN.1 DER encoded form (or null) **Throws:** [IOException](http://docs.google.com/java/io/IOException.html) - if an encoding error occurs (incorrect form for DN)

### setSubject

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

Sets the subject criterion. The specified distinguished name must match the subject distinguished name in the X509Certificate. If null, any subject distinguished name will do.

**Parameters:**subject - a distinguished name as X500Principal (or null)**Since:** 1.5

### setSubject

public void **setSubject**([String](http://docs.google.com/java/lang/String.html) subjectDN)  
 throws [IOException](http://docs.google.com/java/io/IOException.html)

**Denigrated**, use [setSubject(X500Principal)](http://docs.google.com/java/security/cert/X509CertSelector.html#setSubject(javax.security.auth.x500.X500Principal)) or [setSubject(byte[])](http://docs.google.com/java/security/cert/X509CertSelector.html#setSubject(byte%5B%5D)) instead. This method should not be relied on as it can fail to match some certificates because of a loss of encoding information in the RFC 2253 String form of some distinguished names.

Sets the subject criterion. The specified distinguished name must match the subject distinguished name in the X509Certificate. If null, any subject distinguished name will do.

If subjectDN is not null, it should contain a distinguished name, in RFC 2253 format.

**Parameters:**subjectDN - a distinguished name in RFC 2253 format (or null) **Throws:** [IOException](http://docs.google.com/java/io/IOException.html) - if a parsing error occurs (incorrect form for DN)

### setSubject

public void **setSubject**(byte[] subjectDN)  
 throws [IOException](http://docs.google.com/java/io/IOException.html)

Sets the subject criterion. The specified distinguished name must match the subject distinguished name in the X509Certificate. If null, any subject distinguished name will do.

If subjectDN is not null, it should contain a single DER encoded distinguished name, as defined in X.501. For the ASN.1 notation for this structure, see [setIssuer(byte [] issuerDN)](http://docs.google.com/java/security/cert/X509CertSelector.html#setIssuer(byte%5B%5D)).

**Parameters:**subjectDN - a byte array containing the distinguished name in ASN.1 DER format (or null) **Throws:** [IOException](http://docs.google.com/java/io/IOException.html) - if an encoding error occurs (incorrect form for DN)

### setSubjectKeyIdentifier

public void **setSubjectKeyIdentifier**(byte[] subjectKeyID)

Sets the subjectKeyIdentifier criterion. The X509Certificate must contain a SubjectKeyIdentifier extension for which the contents of the extension matches the specified criterion value. If the criterion value is null, no subjectKeyIdentifier check will be done.

If subjectKeyID is not null, it should contain a single DER encoded value corresponding to the contents of the extension value (not including the object identifier, criticality setting, and encapsulating OCTET STRING) for a SubjectKeyIdentifier extension. The ASN.1 notation for this structure follows.

SubjectKeyIdentifier ::= KeyIdentifier  
  
 KeyIdentifier ::= OCTET STRING

Since the format of subject key identifiers is not mandated by any standard, subject key identifiers are not parsed by the X509CertSelector. Instead, the values are compared using a byte-by-byte comparison.

Note that the byte array supplied here is cloned to protect against subsequent modifications.

**Parameters:**subjectKeyID - the subject key identifier (or null)**See Also:**[getSubjectKeyIdentifier()](http://docs.google.com/java/security/cert/X509CertSelector.html#getSubjectKeyIdentifier())

### setAuthorityKeyIdentifier

public void **setAuthorityKeyIdentifier**(byte[] authorityKeyID)

Sets the authorityKeyIdentifier criterion. The X509Certificate must contain an AuthorityKeyIdentifier extension for which the contents of the extension value matches the specified criterion value. If the criterion value is null, no authorityKeyIdentifier check will be done.

If authorityKeyID is not null, it should contain a single DER encoded value corresponding to the contents of the extension value (not including the object identifier, criticality setting, and encapsulating OCTET STRING) for an AuthorityKeyIdentifier extension. The ASN.1 notation for this structure follows.

AuthorityKeyIdentifier ::= SEQUENCE {  
 keyIdentifier [0] KeyIdentifier OPTIONAL,  
 authorityCertIssuer [1] GeneralNames OPTIONAL,  
 authorityCertSerialNumber [2] CertificateSerialNumber OPTIONAL }  
  
 KeyIdentifier ::= OCTET STRING

Authority key identifiers are not parsed by the X509CertSelector. Instead, the values are compared using a byte-by-byte comparison.

When the keyIdentifier field of AuthorityKeyIdentifier is populated, the value is usually taken from the SubjectKeyIdentifier extension in the issuer's certificate. Note, however, that the result of X509Certificate.getExtensionValue(<SubjectKeyIdentifier Object Identifier>) on the issuer's certificate may NOT be used directly as the input to setAuthorityKeyIdentifier. This is because the SubjectKeyIdentifier contains only a KeyIdentifier OCTET STRING, and not a SEQUENCE of KeyIdentifier, GeneralNames, and CertificateSerialNumber. In order to use the extension value of the issuer certificate's SubjectKeyIdentifier extension, it will be necessary to extract the value of the embedded KeyIdentifier OCTET STRING, then DER encode this OCTET STRING inside a SEQUENCE. For more details on SubjectKeyIdentifier, see [setSubjectKeyIdentifier(byte[] subjectKeyID)](http://docs.google.com/java/security/cert/X509CertSelector.html#setSubjectKeyIdentifier(byte%5B%5D)).

Note also that the byte array supplied here is cloned to protect against subsequent modifications.

**Parameters:**authorityKeyID - the authority key identifier (or null)**See Also:**[getAuthorityKeyIdentifier()](http://docs.google.com/java/security/cert/X509CertSelector.html#getAuthorityKeyIdentifier())

### setCertificateValid

public void **setCertificateValid**([Date](http://docs.google.com/java/util/Date.html) certValid)

Sets the certificateValid criterion. The specified date must fall within the certificate validity period for the X509Certificate. If null, no certificateValid check will be done.

Note that the Date supplied here is cloned to protect against subsequent modifications.

**Parameters:**certValid - the Date to check (or null)**See Also:**[getCertificateValid()](http://docs.google.com/java/security/cert/X509CertSelector.html#getCertificateValid())

### setPrivateKeyValid

public void **setPrivateKeyValid**([Date](http://docs.google.com/java/util/Date.html) privateKeyValid)

Sets the privateKeyValid criterion. The specified date must fall within the private key validity period for the X509Certificate. If null, no privateKeyValid check will be done.

Note that the Date supplied here is cloned to protect against subsequent modifications.

**Parameters:**privateKeyValid - the Date to check (or null)**See Also:**[getPrivateKeyValid()](http://docs.google.com/java/security/cert/X509CertSelector.html#getPrivateKeyValid())

### setSubjectPublicKeyAlgID

public void **setSubjectPublicKeyAlgID**([String](http://docs.google.com/java/lang/String.html) oid)  
 throws [IOException](http://docs.google.com/java/io/IOException.html)

Sets the subjectPublicKeyAlgID criterion. The X509Certificate must contain a subject public key with the specified algorithm. If null, no subjectPublicKeyAlgID check will be done.

**Parameters:**oid - The object identifier (OID) of the algorithm to check for (or null). An OID is represented by a set of nonnegative integers separated by periods. **Throws:** [IOException](http://docs.google.com/java/io/IOException.html) - if the OID is invalid, such as the first component being not 0, 1 or 2 or the second component being greater than 39.**See Also:**[getSubjectPublicKeyAlgID()](http://docs.google.com/java/security/cert/X509CertSelector.html#getSubjectPublicKeyAlgID())

### setSubjectPublicKey

public void **setSubjectPublicKey**([PublicKey](http://docs.google.com/java/security/PublicKey.html) key)

Sets the subjectPublicKey criterion. The X509Certificate must contain the specified subject public key. If null, no subjectPublicKey check will be done.

**Parameters:**key - the subject public key to check for (or null)**See Also:**[getSubjectPublicKey()](http://docs.google.com/java/security/cert/X509CertSelector.html#getSubjectPublicKey())

### setSubjectPublicKey

public void **setSubjectPublicKey**(byte[] key)  
 throws [IOException](http://docs.google.com/java/io/IOException.html)

Sets the subjectPublicKey criterion. The X509Certificate must contain the specified subject public key. If null, no subjectPublicKey check will be done.

Because this method allows the public key to be specified as a byte array, it may be used for unknown key types.

If key is not null, it should contain a single DER encoded SubjectPublicKeyInfo structure, as defined in X.509. The ASN.1 notation for this structure is as follows.

SubjectPublicKeyInfo ::= SEQUENCE {  
 algorithm AlgorithmIdentifier,  
 subjectPublicKey BIT STRING }  
  
 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

Note that the byte array supplied here is cloned to protect against subsequent modifications.

**Parameters:**key - a byte array containing the subject public key in ASN.1 DER form (or null) **Throws:** [IOException](http://docs.google.com/java/io/IOException.html) - if an encoding error occurs (incorrect form for subject public key)**See Also:**[getSubjectPublicKey()](http://docs.google.com/java/security/cert/X509CertSelector.html#getSubjectPublicKey())

### setKeyUsage

public void **setKeyUsage**(boolean[] keyUsage)

Sets the keyUsage criterion. The X509Certificate must allow the specified keyUsage values. If null, no keyUsage check will be done. Note that an X509Certificate that has no keyUsage extension implicitly allows all keyUsage values.

Note that the boolean array supplied here is cloned to protect against subsequent modifications.

**Parameters:**keyUsage - a boolean array in the same format as the boolean array returned by [X509Certificate.getKeyUsage()](http://docs.google.com/java/security/cert/X509Certificate.html#getKeyUsage()). Or null.**See Also:**[getKeyUsage()](http://docs.google.com/java/security/cert/X509CertSelector.html#getKeyUsage())

### setExtendedKeyUsage

public void **setExtendedKeyUsage**([Set](http://docs.google.com/java/util/Set.html)<[String](http://docs.google.com/java/lang/String.html)> keyPurposeSet)  
 throws [IOException](http://docs.google.com/java/io/IOException.html)

Sets the extendedKeyUsage criterion. The X509Certificate must allow the specified key purposes in its extended key usage extension. If keyPurposeSet is empty or null, no extendedKeyUsage check will be done. Note that an X509Certificate that has no extendedKeyUsage extension implicitly allows all key purposes.

Note that the Set is cloned to protect against subsequent modifications.

**Parameters:**keyPurposeSet - a Set of key purpose OIDs in string format (or null). Each OID is represented by a set of nonnegative integers separated by periods. **Throws:** [IOException](http://docs.google.com/java/io/IOException.html) - if the OID is invalid, such as the first component being not 0, 1 or 2 or the second component being greater than 39.**See Also:**[getExtendedKeyUsage()](http://docs.google.com/java/security/cert/X509CertSelector.html#getExtendedKeyUsage())

### setMatchAllSubjectAltNames

public void **setMatchAllSubjectAltNames**(boolean matchAllNames)

Enables/disables matching all of the subjectAlternativeNames specified in the [setSubjectAlternativeNames](http://docs.google.com/java/security/cert/X509CertSelector.html#setSubjectAlternativeNames(java.util.Collection)) or [addSubjectAlternativeName](http://docs.google.com/java/security/cert/X509CertSelector.html#addSubjectAlternativeName(int,%20java.lang.String)) methods. If enabled, the X509Certificate must contain all of the specified subject alternative names. If disabled, the X509Certificate must contain at least one of the specified subject alternative names.

The matchAllNames flag is true by default.

**Parameters:**matchAllNames - if true, the flag is enabled; if false, the flag is disabled.**See Also:**[getMatchAllSubjectAltNames()](http://docs.google.com/java/security/cert/X509CertSelector.html#getMatchAllSubjectAltNames())

### setSubjectAlternativeNames

public void **setSubjectAlternativeNames**([Collection](http://docs.google.com/java/util/Collection.html)<[List](http://docs.google.com/java/util/List.html)<?>> names)  
 throws [IOException](http://docs.google.com/java/io/IOException.html)

Sets the subjectAlternativeNames criterion. The X509Certificate must contain all or at least one of the specified subjectAlternativeNames, depending on the value of the matchAllNames flag (see [setMatchAllSubjectAltNames](http://docs.google.com/java/security/cert/X509CertSelector.html#setMatchAllSubjectAltNames(boolean))).

This method allows the caller to specify, with a single method call, the complete set of subject alternative names for the subjectAlternativeNames criterion. The specified value replaces the previous value for the subjectAlternativeNames criterion.

The names parameter (if not null) is a Collection with one entry for each name to be included in the subject alternative name criterion. Each entry is a List whose first entry is an Integer (the name type, 0-8) and whose second entry is a String or a byte array (the name, in string or ASN.1 DER encoded form, respectively). There can be multiple names of the same type. If null is supplied as the value for this argument, no subjectAlternativeNames check will be performed.

Each subject alternative name in the Collection may be specified either as a String or as an ASN.1 encoded byte array. For more details about the formats used, see [addSubjectAlternativeName(int type, String name)](http://docs.google.com/java/security/cert/X509CertSelector.html#addSubjectAlternativeName(int,%20java.lang.String)) and [addSubjectAlternativeName(int type, byte [] name)](http://docs.google.com/java/security/cert/X509CertSelector.html#addSubjectAlternativeName(int,%20byte%5B%5D)).

**Note:** for distinguished names, specify the byte array form instead of the String form. See the note in [addSubjectAlternativeName(int, String)](http://docs.google.com/java/security/cert/X509CertSelector.html#addSubjectAlternativeName(int,%20java.lang.String)) for more information.

Note that the names parameter can contain duplicate names (same name and name type), but they may be removed from the Collection of names returned by the [getSubjectAlternativeNames](http://docs.google.com/java/security/cert/X509CertSelector.html#getSubjectAlternativeNames()) method.

Note that a deep copy is performed on the Collection to protect against subsequent modifications.

**Parameters:**names - a Collection of names (or null) **Throws:** [IOException](http://docs.google.com/java/io/IOException.html) - if a parsing error occurs**See Also:**[getSubjectAlternativeNames()](http://docs.google.com/java/security/cert/X509CertSelector.html#getSubjectAlternativeNames())

### addSubjectAlternativeName

public void **addSubjectAlternativeName**(int type,  
 [String](http://docs.google.com/java/lang/String.html) name)  
 throws [IOException](http://docs.google.com/java/io/IOException.html)

Adds a name to the subjectAlternativeNames criterion. The X509Certificate must contain all or at least one of the specified subjectAlternativeNames, depending on the value of the matchAllNames flag (see [setMatchAllSubjectAltNames](http://docs.google.com/java/security/cert/X509CertSelector.html#setMatchAllSubjectAltNames(boolean))).

This method allows the caller to add a name to the set of subject alternative names. The specified name is added to any previous value for the subjectAlternativeNames criterion. If the specified name is a duplicate, it may be ignored.

The name is provided in string format. RFC 822, DNS, and URI names use the well-established string formats for those types (subject to the restrictions included in RFC 2459). IPv4 address names are supplied using dotted quad notation. OID address names are represented as a series of nonnegative integers separated by periods. And directory names (distinguished names) are supplied in RFC 2253 format. No standard string format is defined for otherNames, X.400 names, EDI party names, IPv6 address names, or any other type of names. They should be specified using the [addSubjectAlternativeName(int type, byte [] name)](http://docs.google.com/java/security/cert/X509CertSelector.html#addSubjectAlternativeName(int,%20byte%5B%5D)) method.

**Note:** for distinguished names, use [addSubjectAlternativeName(int, byte[])](http://docs.google.com/java/security/cert/X509CertSelector.html#addSubjectAlternativeName(int,%20byte%5B%5D)) instead. This method should not be relied on as it can fail to match some certificates because of a loss of encoding information in the RFC 2253 String form of some distinguished names.

**Parameters:**type - the name type (0-8, as specified in RFC 2459, section 4.2.1.7)name - the name in string form (not null) **Throws:** [IOException](http://docs.google.com/java/io/IOException.html) - if a parsing error occurs

### addSubjectAlternativeName

public void **addSubjectAlternativeName**(int type,  
 byte[] name)  
 throws [IOException](http://docs.google.com/java/io/IOException.html)

Adds a name to the subjectAlternativeNames criterion. The X509Certificate must contain all or at least one of the specified subjectAlternativeNames, depending on the value of the matchAllNames flag (see [setMatchAllSubjectAltNames](http://docs.google.com/java/security/cert/X509CertSelector.html#setMatchAllSubjectAltNames(boolean))).

This method allows the caller to add a name to the set of subject alternative names. The specified name is added to any previous value for the subjectAlternativeNames criterion. If the specified name is a duplicate, it may be ignored.

The name is provided as a byte array. This byte array should contain the DER encoded name, as it would appear in the GeneralName structure defined in RFC 2459 and X.509. The encoded byte array should only contain the encoded value of the name, and should not include the tag associated with the name in the GeneralName structure. The ASN.1 definition of this structure appears below.

GeneralName ::= CHOICE {  
 otherName [0] OtherName,  
 rfc822Name [1] IA5String,  
 dNSName [2] IA5String,  
 x400Address [3] ORAddress,  
 directoryName [4] Name,  
 ediPartyName [5] EDIPartyName,  
 uniformResourceIdentifier [6] IA5String,  
 iPAddress [7] OCTET STRING,  
 registeredID [8] OBJECT IDENTIFIER}

Note that the byte array supplied here is cloned to protect against subsequent modifications.

**Parameters:**type - the name type (0-8, as listed above)name - a byte array containing the name in ASN.1 DER encoded form **Throws:** [IOException](http://docs.google.com/java/io/IOException.html) - if a parsing error occurs

### setNameConstraints

public void **setNameConstraints**(byte[] bytes)  
 throws [IOException](http://docs.google.com/java/io/IOException.html)

Sets the name constraints criterion. The X509Certificate must have subject and subject alternative names that meet the specified name constraints.

The name constraints are specified as a byte array. This byte array should contain the DER encoded form of the name constraints, as they would appear in the NameConstraints structure defined in RFC 2459 and X.509. The ASN.1 definition of this structure appears below.

NameConstraints ::= SEQUENCE {  
 permittedSubtrees [0] GeneralSubtrees OPTIONAL,  
 excludedSubtrees [1] GeneralSubtrees OPTIONAL }  
  
 GeneralSubtrees ::= SEQUENCE SIZE (1..MAX) OF GeneralSubtree  
  
 GeneralSubtree ::= SEQUENCE {  
 base GeneralName,  
 minimum [0] BaseDistance DEFAULT 0,  
 maximum [1] BaseDistance OPTIONAL }  
  
 BaseDistance ::= INTEGER (0..MAX)  
  
 GeneralName ::= CHOICE {  
 otherName [0] OtherName,  
 rfc822Name [1] IA5String,  
 dNSName [2] IA5String,  
 x400Address [3] ORAddress,  
 directoryName [4] Name,  
 ediPartyName [5] EDIPartyName,  
 uniformResourceIdentifier [6] IA5String,  
 iPAddress [7] OCTET STRING,  
 registeredID [8] OBJECT IDENTIFIER}

Note that the byte array supplied here is cloned to protect against subsequent modifications.

**Parameters:**bytes - a byte array containing the ASN.1 DER encoding of a NameConstraints extension to be used for checking name constraints. Only the value of the extension is included, not the OID or criticality flag. Can be null, in which case no name constraints check will be performed. **Throws:** [IOException](http://docs.google.com/java/io/IOException.html) - if a parsing error occurs**See Also:**[getNameConstraints()](http://docs.google.com/java/security/cert/X509CertSelector.html#getNameConstraints())

### setBasicConstraints

public void **setBasicConstraints**(int minMaxPathLen)

Sets the basic constraints constraint. If the value is greater than or equal to zero, X509Certificates must include a basicConstraints extension with a pathLen of at least this value. If the value is -2, only end-entity certificates are accepted. If the value is -1, no check is done.

This constraint is useful when building a certification path forward (from the target toward the trust anchor. If a partial path has been built, any candidate certificate must have a maxPathLen value greater than or equal to the number of certificates in the partial path.

**Parameters:**minMaxPathLen - the value for the basic constraints constraint **Throws:** [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if the value is less than -2**See Also:**[getBasicConstraints()](http://docs.google.com/java/security/cert/X509CertSelector.html#getBasicConstraints())

### setPolicy

public void **setPolicy**([Set](http://docs.google.com/java/util/Set.html)<[String](http://docs.google.com/java/lang/String.html)> certPolicySet)  
 throws [IOException](http://docs.google.com/java/io/IOException.html)

Sets the policy constraint. The X509Certificate must include at least one of the specified policies in its certificate policies extension. If certPolicySet is empty, then the X509Certificate must include at least some specified policy in its certificate policies extension. If certPolicySet is null, no policy check will be performed.

Note that the Set is cloned to protect against subsequent modifications.

**Parameters:**certPolicySet - a Set of certificate policy OIDs in string format (or null). Each OID is represented by a set of nonnegative integers separated by periods. **Throws:** [IOException](http://docs.google.com/java/io/IOException.html) - if a parsing error occurs on the OID such as the first component is not 0, 1 or 2 or the second component is greater than 39.**See Also:**[getPolicy()](http://docs.google.com/java/security/cert/X509CertSelector.html#getPolicy())

### setPathToNames

public void **setPathToNames**([Collection](http://docs.google.com/java/util/Collection.html)<[List](http://docs.google.com/java/util/List.html)<?>> names)  
 throws [IOException](http://docs.google.com/java/io/IOException.html)

Sets the pathToNames criterion. The X509Certificate must not include name constraints that would prohibit building a path to the specified names.

This method allows the caller to specify, with a single method call, the complete set of names which the X509Certificates's name constraints must permit. The specified value replaces the previous value for the pathToNames criterion.

This constraint is useful when building a certification path forward (from the target toward the trust anchor. If a partial path has been built, any candidate certificate must not include name constraints that would prohibit building a path to any of the names in the partial path.

The names parameter (if not null) is a Collection with one entry for each name to be included in the pathToNames criterion. Each entry is a List whose first entry is an Integer (the name type, 0-8) and whose second entry is a String or a byte array (the name, in string or ASN.1 DER encoded form, respectively). There can be multiple names of the same type. If null is supplied as the value for this argument, no pathToNames check will be performed.

Each name in the Collection may be specified either as a String or as an ASN.1 encoded byte array. For more details about the formats used, see [addPathToName(int type, String name)](http://docs.google.com/java/security/cert/X509CertSelector.html#addPathToName(int,%20java.lang.String)) and [addPathToName(int type, byte [] name)](http://docs.google.com/java/security/cert/X509CertSelector.html#addPathToName(int,%20byte%5B%5D)).

**Note:** for distinguished names, specify the byte array form instead of the String form. See the note in [addPathToName(int, String)](http://docs.google.com/java/security/cert/X509CertSelector.html#addPathToName(int,%20java.lang.String)) for more information.

Note that the names parameter can contain duplicate names (same name and name type), but they may be removed from the Collection of names returned by the [getPathToNames](http://docs.google.com/java/security/cert/X509CertSelector.html#getPathToNames()) method.

Note that a deep copy is performed on the Collection to protect against subsequent modifications.

**Parameters:**names - a Collection with one entry per name (or null) **Throws:** [IOException](http://docs.google.com/java/io/IOException.html) - if a parsing error occurs**See Also:**[getPathToNames()](http://docs.google.com/java/security/cert/X509CertSelector.html#getPathToNames())

### addPathToName

public void **addPathToName**(int type,  
 [String](http://docs.google.com/java/lang/String.html) name)  
 throws [IOException](http://docs.google.com/java/io/IOException.html)

Adds a name to the pathToNames criterion. The X509Certificate must not include name constraints that would prohibit building a path to the specified name.

This method allows the caller to add a name to the set of names which the X509Certificates's name constraints must permit. The specified name is added to any previous value for the pathToNames criterion. If the name is a duplicate, it may be ignored.

The name is provided in string format. RFC 822, DNS, and URI names use the well-established string formats for those types (subject to the restrictions included in RFC 2459). IPv4 address names are supplied using dotted quad notation. OID address names are represented as a series of nonnegative integers separated by periods. And directory names (distinguished names) are supplied in RFC 2253 format. No standard string format is defined for otherNames, X.400 names, EDI party names, IPv6 address names, or any other type of names. They should be specified using the [addPathToName(int type, byte [] name)](http://docs.google.com/java/security/cert/X509CertSelector.html#addPathToName(int,%20byte%5B%5D)) method.

**Note:** for distinguished names, use [addPathToName(int, byte[])](http://docs.google.com/java/security/cert/X509CertSelector.html#addPathToName(int,%20byte%5B%5D)) instead. This method should not be relied on as it can fail to match some certificates because of a loss of encoding information in the RFC 2253 String form of some distinguished names.

**Parameters:**type - the name type (0-8, as specified in RFC 2459, section 4.2.1.7)name - the name in string form **Throws:** [IOException](http://docs.google.com/java/io/IOException.html) - if a parsing error occurs

### addPathToName

public void **addPathToName**(int type,  
 byte[] name)  
 throws [IOException](http://docs.google.com/java/io/IOException.html)

Adds a name to the pathToNames criterion. The X509Certificate must not include name constraints that would prohibit building a path to the specified name.

This method allows the caller to add a name to the set of names which the X509Certificates's name constraints must permit. The specified name is added to any previous value for the pathToNames criterion. If the name is a duplicate, it may be ignored.

The name is provided as a byte array. This byte array should contain the DER encoded name, as it would appear in the GeneralName structure defined in RFC 2459 and X.509. The ASN.1 definition of this structure appears in the documentation for [addSubjectAlternativeName(int type, byte [] name)](http://docs.google.com/java/security/cert/X509CertSelector.html#addSubjectAlternativeName(int,%20byte%5B%5D)).

Note that the byte array supplied here is cloned to protect against subsequent modifications.

**Parameters:**type - the name type (0-8, as specified in RFC 2459, section 4.2.1.7)name - a byte array containing the name in ASN.1 DER encoded form **Throws:** [IOException](http://docs.google.com/java/io/IOException.html) - if a parsing error occurs

### getCertificate

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

Returns the certificateEquals criterion. The specified X509Certificate must be equal to the X509Certificate passed to the match method. If null, this check is not applied.

**Returns:**the X509Certificate to match (or null)**See Also:**[setCertificate(java.security.cert.X509Certificate)](http://docs.google.com/java/security/cert/X509CertSelector.html#setCertificate(java.security.cert.X509Certificate))

### getSerialNumber

public [BigInteger](http://docs.google.com/java/math/BigInteger.html) **getSerialNumber**()

Returns the serialNumber criterion. The specified serial number must match the certificate serial number in the X509Certificate. If null, any certificate serial number will do.

**Returns:**the certificate serial number to match (or null)**See Also:**[setSerialNumber(java.math.BigInteger)](http://docs.google.com/java/security/cert/X509CertSelector.html#setSerialNumber(java.math.BigInteger))

### getIssuer

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

Returns the issuer criterion as an X500Principal. This distinguished name must match the issuer distinguished name in the X509Certificate. If null, the issuer criterion is disabled and any issuer distinguished name will do.

**Returns:**the required issuer distinguished name as X500Principal (or null)**Since:** 1.5

### getIssuerAsString

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

**Denigrated**, use [getIssuer()](http://docs.google.com/java/security/cert/X509CertSelector.html#getIssuer()) or [getIssuerAsBytes()](http://docs.google.com/java/security/cert/X509CertSelector.html#getIssuerAsBytes()) instead. This method should not be relied on as it can fail to match some certificates because of a loss of encoding information in the RFC 2253 String form of some distinguished names.

Returns the issuer criterion as a String. This distinguished name must match the issuer distinguished name in the X509Certificate. If null, the issuer criterion is disabled and any issuer distinguished name will do.

If the value returned is not null, it is a distinguished name, in RFC 2253 format.

**Returns:**the required issuer distinguished name in RFC 2253 format (or null)

### getIssuerAsBytes

public byte[] **getIssuerAsBytes**()  
 throws [IOException](http://docs.google.com/java/io/IOException.html)

Returns the issuer criterion as a byte array. This distinguished name must match the issuer distinguished name in the X509Certificate. If null, the issuer criterion is disabled and any issuer distinguished name will do.

If the value returned is not null, it is a byte array containing a single DER encoded distinguished name, as defined in X.501. The ASN.1 notation for this structure is supplied in the documentation for [setIssuer(byte [] issuerDN)](http://docs.google.com/java/security/cert/X509CertSelector.html#setIssuer(byte%5B%5D)).

Note that the byte array returned is cloned to protect against subsequent modifications.

**Returns:**a byte array containing the required issuer distinguished name in ASN.1 DER format (or null) **Throws:** [IOException](http://docs.google.com/java/io/IOException.html) - if an encoding error occurs

### getSubject

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

Returns the subject criterion as an X500Principal. This distinguished name must match the subject distinguished name in the X509Certificate. If null, the subject criterion is disabled and any subject distinguished name will do.

**Returns:**the required subject distinguished name as X500Principal (or null)**Since:** 1.5

### getSubjectAsString

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

**Denigrated**, use [getSubject()](http://docs.google.com/java/security/cert/X509CertSelector.html#getSubject()) or [getSubjectAsBytes()](http://docs.google.com/java/security/cert/X509CertSelector.html#getSubjectAsBytes()) instead. This method should not be relied on as it can fail to match some certificates because of a loss of encoding information in the RFC 2253 String form of some distinguished names.

Returns the subject criterion as a String. This distinguished name must match the subject distinguished name in the X509Certificate. If null, the subject criterion is disabled and any subject distinguished name will do.

If the value returned is not null, it is a distinguished name, in RFC 2253 format.

**Returns:**the required subject distinguished name in RFC 2253 format (or null)

### getSubjectAsBytes

public byte[] **getSubjectAsBytes**()  
 throws [IOException](http://docs.google.com/java/io/IOException.html)

Returns the subject criterion as a byte array. This distinguished name must match the subject distinguished name in the X509Certificate. If null, the subject criterion is disabled and any subject distinguished name will do.

If the value returned is not null, it is a byte array containing a single DER encoded distinguished name, as defined in X.501. The ASN.1 notation for this structure is supplied in the documentation for [setSubject(byte [] subjectDN)](http://docs.google.com/java/security/cert/X509CertSelector.html#setSubject(byte%5B%5D)).

Note that the byte array returned is cloned to protect against subsequent modifications.

**Returns:**a byte array containing the required subject distinguished name in ASN.1 DER format (or null) **Throws:** [IOException](http://docs.google.com/java/io/IOException.html) - if an encoding error occurs

### getSubjectKeyIdentifier

public byte[] **getSubjectKeyIdentifier**()

Returns the subjectKeyIdentifier criterion. The X509Certificate must contain a SubjectKeyIdentifier extension with the specified value. If null, no subjectKeyIdentifier check will be done.

Note that the byte array returned is cloned to protect against subsequent modifications.

**Returns:**the key identifier (or null)**See Also:**[setSubjectKeyIdentifier(byte[])](http://docs.google.com/java/security/cert/X509CertSelector.html#setSubjectKeyIdentifier(byte%5B%5D))

### getAuthorityKeyIdentifier

public byte[] **getAuthorityKeyIdentifier**()

Returns the authorityKeyIdentifier criterion. The X509Certificate must contain a AuthorityKeyIdentifier extension with the specified value. If null, no authorityKeyIdentifier check will be done.

Note that the byte array returned is cloned to protect against subsequent modifications.

**Returns:**the key identifier (or null)**See Also:**[setAuthorityKeyIdentifier(byte[])](http://docs.google.com/java/security/cert/X509CertSelector.html#setAuthorityKeyIdentifier(byte%5B%5D))

### getCertificateValid

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

Returns the certificateValid criterion. The specified date must fall within the certificate validity period for the X509Certificate. If null, no certificateValid check will be done.

Note that the Date returned is cloned to protect against subsequent modifications.

**Returns:**the Date to check (or null)**See Also:**[setCertificateValid(java.util.Date)](http://docs.google.com/java/security/cert/X509CertSelector.html#setCertificateValid(java.util.Date))

### getPrivateKeyValid

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

Returns the privateKeyValid criterion. The specified date must fall within the private key validity period for the X509Certificate. If null, no privateKeyValid check will be done.

Note that the Date returned is cloned to protect against subsequent modifications.

**Returns:**the Date to check (or null)**See Also:**[setPrivateKeyValid(java.util.Date)](http://docs.google.com/java/security/cert/X509CertSelector.html#setPrivateKeyValid(java.util.Date))

### getSubjectPublicKeyAlgID

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

Returns the subjectPublicKeyAlgID criterion. The X509Certificate must contain a subject public key with the specified algorithm. If null, no subjectPublicKeyAlgID check will be done.

**Returns:**the object identifier (OID) of the signature algorithm to check for (or null). An OID is represented by a set of nonnegative integers separated by periods.**See Also:**[setSubjectPublicKeyAlgID(java.lang.String)](http://docs.google.com/java/security/cert/X509CertSelector.html#setSubjectPublicKeyAlgID(java.lang.String))

### getSubjectPublicKey

public [PublicKey](http://docs.google.com/java/security/PublicKey.html) **getSubjectPublicKey**()

Returns the subjectPublicKey criterion. The X509Certificate must contain the specified subject public key. If null, no subjectPublicKey check will be done.

**Returns:**the subject public key to check for (or null)**See Also:**[setSubjectPublicKey(java.security.PublicKey)](http://docs.google.com/java/security/cert/X509CertSelector.html#setSubjectPublicKey(java.security.PublicKey))

### getKeyUsage

public boolean[] **getKeyUsage**()

Returns the keyUsage criterion. The X509Certificate must allow the specified keyUsage values. If null, no keyUsage check will be done.

Note that the boolean array returned is cloned to protect against subsequent modifications.

**Returns:**a boolean array in the same format as the boolean array returned by [X509Certificate.getKeyUsage()](http://docs.google.com/java/security/cert/X509Certificate.html#getKeyUsage()). Or null.**See Also:**[setKeyUsage(boolean[])](http://docs.google.com/java/security/cert/X509CertSelector.html#setKeyUsage(boolean%5B%5D))

### getExtendedKeyUsage

public [Set](http://docs.google.com/java/util/Set.html)<[String](http://docs.google.com/java/lang/String.html)> **getExtendedKeyUsage**()

Returns the extendedKeyUsage criterion. The X509Certificate must allow the specified key purposes in its extended key usage extension. If the keyPurposeSet returned is empty or null, no extendedKeyUsage check will be done. Note that an X509Certificate that has no extendedKeyUsage extension implicitly allows all key purposes.

**Returns:**an immutable Set of key purpose OIDs in string format (or null)**See Also:**[setExtendedKeyUsage(java.util.Set)](http://docs.google.com/java/security/cert/X509CertSelector.html#setExtendedKeyUsage(java.util.Set))

### getMatchAllSubjectAltNames

public boolean **getMatchAllSubjectAltNames**()

Indicates if the X509Certificate must contain all or at least one of the subjectAlternativeNames specified in the [setSubjectAlternativeNames](http://docs.google.com/java/security/cert/X509CertSelector.html#setSubjectAlternativeNames(java.util.Collection)) or [addSubjectAlternativeName](http://docs.google.com/java/security/cert/X509CertSelector.html#addSubjectAlternativeName(int,%20java.lang.String)) methods. If true, the X509Certificate must contain all of the specified subject alternative names. If false, the X509Certificate must contain at least one of the specified subject alternative names.

**Returns:**true if the flag is enabled; false if the flag is disabled. The flag is true by default.**See Also:**[setMatchAllSubjectAltNames(boolean)](http://docs.google.com/java/security/cert/X509CertSelector.html#setMatchAllSubjectAltNames(boolean))

### getSubjectAlternativeNames

public [Collection](http://docs.google.com/java/util/Collection.html)<[List](http://docs.google.com/java/util/List.html)<?>> **getSubjectAlternativeNames**()

Returns a copy of the subjectAlternativeNames criterion. The X509Certificate must contain all or at least one of the specified subjectAlternativeNames, depending on the value of the matchAllNames flag (see [getMatchAllSubjectAltNames](http://docs.google.com/java/security/cert/X509CertSelector.html#getMatchAllSubjectAltNames())). If the value returned is null, no subjectAlternativeNames check will be performed.

If the value returned is not null, it is a Collection with one entry for each name to be included in the subject alternative name criterion. Each entry is a List whose first entry is an Integer (the name type, 0-8) and whose second entry is a String or a byte array (the name, in string or ASN.1 DER encoded form, respectively). There can be multiple names of the same type. Note that the Collection returned may contain duplicate names (same name and name type).

Each subject alternative name in the Collection may be specified either as a String or as an ASN.1 encoded byte array. For more details about the formats used, see [addSubjectAlternativeName(int type, String name)](http://docs.google.com/java/security/cert/X509CertSelector.html#addSubjectAlternativeName(int,%20java.lang.String)) and [addSubjectAlternativeName(int type, byte [] name)](http://docs.google.com/java/security/cert/X509CertSelector.html#addSubjectAlternativeName(int,%20byte%5B%5D)).

Note that a deep copy is performed on the Collection to protect against subsequent modifications.

**Returns:**a Collection of names (or null)**See Also:**[setSubjectAlternativeNames(java.util.Collection>)](http://docs.google.com/java/security/cert/X509CertSelector.html#setSubjectAlternativeNames(java.util.Collection))

### getNameConstraints

public byte[] **getNameConstraints**()

Returns the name constraints criterion. The X509Certificate must have subject and subject alternative names that meet the specified name constraints.

The name constraints are returned as a byte array. This byte array contains the DER encoded form of the name constraints, as they would appear in the NameConstraints structure defined in RFC 2459 and X.509. The ASN.1 notation for this structure is supplied in the documentation for [setNameConstraints(byte [] bytes)](http://docs.google.com/java/security/cert/X509CertSelector.html#setNameConstraints(byte%5B%5D)).

Note that the byte array returned is cloned to protect against subsequent modifications.

**Returns:**a byte array containing the ASN.1 DER encoding of a NameConstraints extension used for checking name constraints. null if no name constraints check will be performed.**See Also:**[setNameConstraints(byte[])](http://docs.google.com/java/security/cert/X509CertSelector.html#setNameConstraints(byte%5B%5D))

### getBasicConstraints

public int **getBasicConstraints**()

Returns the basic constraints constraint. If the value is greater than or equal to zero, the X509Certificates must include a basicConstraints extension with a pathLen of at least this value. If the value is -2, only end-entity certificates are accepted. If the value is -1, no basicConstraints check is done.

**Returns:**the value for the basic constraints constraint**See Also:**[setBasicConstraints(int)](http://docs.google.com/java/security/cert/X509CertSelector.html#setBasicConstraints(int))

### getPolicy

public [Set](http://docs.google.com/java/util/Set.html)<[String](http://docs.google.com/java/lang/String.html)> **getPolicy**()

Returns the policy criterion. The X509Certificate must include at least one of the specified policies in its certificate policies extension. If the Set returned is empty, then the X509Certificate must include at least some specified policy in its certificate policies extension. If the Set returned is null, no policy check will be performed.

**Returns:**an immutable Set of certificate policy OIDs in string format (or null)**See Also:**[setPolicy(java.util.Set)](http://docs.google.com/java/security/cert/X509CertSelector.html#setPolicy(java.util.Set))

### getPathToNames

public [Collection](http://docs.google.com/java/util/Collection.html)<[List](http://docs.google.com/java/util/List.html)<?>> **getPathToNames**()

Returns a copy of the pathToNames criterion. The X509Certificate must not include name constraints that would prohibit building a path to the specified names. If the value returned is null, no pathToNames check will be performed.

If the value returned is not null, it is a Collection with one entry for each name to be included in the pathToNames criterion. Each entry is a List whose first entry is an Integer (the name type, 0-8) and whose second entry is a String or a byte array (the name, in string or ASN.1 DER encoded form, respectively). There can be multiple names of the same type. Note that the Collection returned may contain duplicate names (same name and name type).

Each name in the Collection may be specified either as a String or as an ASN.1 encoded byte array. For more details about the formats used, see [addPathToName(int type, String name)](http://docs.google.com/java/security/cert/X509CertSelector.html#addPathToName(int,%20java.lang.String)) and [addPathToName(int type, byte [] name)](http://docs.google.com/java/security/cert/X509CertSelector.html#addPathToName(int,%20byte%5B%5D)).

Note that a deep copy is performed on the Collection to protect against subsequent modifications.

**Returns:**a Collection of names (or null)**See Also:**[setPathToNames(java.util.Collection>)](http://docs.google.com/java/security/cert/X509CertSelector.html#setPathToNames(java.util.Collection))

### toString

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

Return a printable representation of the CertSelector.

**Overrides:**[toString](http://docs.google.com/java/lang/Object.html#toString()) in class [Object](http://docs.google.com/java/lang/Object.html) **Returns:**a String describing the contents of the CertSelector

### match

public boolean **match**([Certificate](http://docs.google.com/java/security/cert/Certificate.html) cert)

Decides whether a Certificate should be selected.

**Specified by:**[match](http://docs.google.com/java/security/cert/CertSelector.html#match(java.security.cert.Certificate)) in interface [CertSelector](http://docs.google.com/java/security/cert/CertSelector.html) **Parameters:**cert - the Certificate to be checked **Returns:**true if the Certificate should be selected, false otherwise

### clone

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

Returns a copy of this object.

**Specified by:**[clone](http://docs.google.com/java/security/cert/CertSelector.html#clone()) in interface [CertSelector](http://docs.google.com/java/security/cert/CertSelector.html)**Overrides:**[clone](http://docs.google.com/java/lang/Object.html#clone()) in class [Object](http://docs.google.com/java/lang/Object.html) **Returns:**the copy**See Also:**[Cloneable](http://docs.google.com/java/lang/Cloneable.html)

| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/X509CertSelector.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/X509Certificate.html)   [**NEXT CLASS**](http://docs.google.com/java/security/cert/X509CRL.html) | [**FRAMES**](http://docs.google.com/index.html?java/security/cert/X509CertSelector.html)    [**NO FRAMES**](http://docs.google.com/X509CertSelector.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | FIELD | [CONSTR](#3znysh7) | [METHOD](#2et92p0) | DETAIL: FIELD | [CONSTR](#3dy6vkm) | [METHOD](#4d34og8) |

[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).