| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/CertificateFactory.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/CertificateExpiredException.html)   [**NEXT CLASS**](http://docs.google.com/java/security/cert/CertificateFactorySpi.html) | [**FRAMES**](http://docs.google.com/index.html?java/security/cert/CertificateFactory.html)    [**NO FRAMES**](http://docs.google.com/CertificateFactory.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 CertificateFactory

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

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

This class defines the functionality of a certificate factory, which is used to generate certificate, certification path (CertPath) and certificate revocation list (CRL) objects from their encodings.

For encodings consisting of multiple certificates, use generateCertificates when you want to parse a collection of possibly unrelated certificates. Otherwise, use generateCertPath when you want to generate a CertPath (a certificate chain) and subsequently validate it with a CertPathValidator.

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

The following example reads a file with Base64 encoded certificates, which are each bounded at the beginning by -----BEGIN CERTIFICATE-----, and bounded at the end by -----END CERTIFICATE-----. We convert the FileInputStream (which does not support mark and reset) to a BufferedInputStream (which supports those methods), so that each call to generateCertificate consumes only one certificate, and the read position of the input stream is positioned to the next certificate in the file:

FileInputStream fis = new FileInputStream(filename);  
 BufferedInputStream bis = new BufferedInputStream(fis);  
  
 CertificateFactory cf = CertificateFactory.getInstance("X.509");  
  
 while (bis.available() > 0) {  
 Certificate cert = cf.generateCertificate(bis);  
 System.out.println(cert.toString());  
 }

The following example parses a PKCS#7-formatted certificate reply stored in a file and extracts all the certificates from it:

FileInputStream fis = new FileInputStream(filename);  
 CertificateFactory cf = CertificateFactory.getInstance("X.509");  
 Collection c = cf.generateCertificates(fis);  
 Iterator i = c.iterator();  
 while (i.hasNext()) {  
 Certificate cert = (Certificate)i.next();  
 System.out.println(cert);  
 }

**Since:** 1.2 **See Also:**[Certificate](http://docs.google.com/java/security/cert/Certificate.html), [X509Certificate](http://docs.google.com/java/security/cert/X509Certificate.html), [CertPath](http://docs.google.com/java/security/cert/CertPath.html), [CRL](http://docs.google.com/java/security/cert/CRL.html), [X509CRL](http://docs.google.com/java/security/cert/X509CRL.html)

| **Constructor Summary** | |
| --- | --- |
| protected | [**CertificateFactory**](http://docs.google.com/java/security/cert/CertificateFactory.html#CertificateFactory(java.security.cert.CertificateFactorySpi,%20java.security.Provider,%20java.lang.String))([CertificateFactorySpi](http://docs.google.com/java/security/cert/CertificateFactorySpi.html) certFacSpi, [Provider](http://docs.google.com/java/security/Provider.html) provider, [String](http://docs.google.com/java/lang/String.html) type)            Creates a CertificateFactory object of the given type, and encapsulates the given provider implementation (SPI object) in it. |

| **Method Summary** | |
| --- | --- |
| [Certificate](http://docs.google.com/java/security/cert/Certificate.html) | [**generateCertificate**](http://docs.google.com/java/security/cert/CertificateFactory.html#generateCertificate(java.io.InputStream))([InputStream](http://docs.google.com/java/io/InputStream.html) inStream)            Generates a certificate object and initializes it with the data read from the input stream inStream. |
| [Collection](http://docs.google.com/java/util/Collection.html)<? extends [Certificate](http://docs.google.com/java/security/cert/Certificate.html)> | [**generateCertificates**](http://docs.google.com/java/security/cert/CertificateFactory.html#generateCertificates(java.io.InputStream))([InputStream](http://docs.google.com/java/io/InputStream.html) inStream)            Returns a (possibly empty) collection view of the certificates read from the given input stream inStream. |
| [CertPath](http://docs.google.com/java/security/cert/CertPath.html) | [**generateCertPath**](http://docs.google.com/java/security/cert/CertificateFactory.html#generateCertPath(java.io.InputStream))([InputStream](http://docs.google.com/java/io/InputStream.html) inStream)            Generates a CertPath object and initializes it with the data read from the InputStream inStream. |
| [CertPath](http://docs.google.com/java/security/cert/CertPath.html) | [**generateCertPath**](http://docs.google.com/java/security/cert/CertificateFactory.html#generateCertPath(java.io.InputStream,%20java.lang.String))([InputStream](http://docs.google.com/java/io/InputStream.html) inStream, [String](http://docs.google.com/java/lang/String.html) encoding)            Generates a CertPath object and initializes it with the data read from the InputStream inStream. |
| [CertPath](http://docs.google.com/java/security/cert/CertPath.html) | [**generateCertPath**](http://docs.google.com/java/security/cert/CertificateFactory.html#generateCertPath(java.util.List))([List](http://docs.google.com/java/util/List.html)<? extends [Certificate](http://docs.google.com/java/security/cert/Certificate.html)> certificates)            Generates a CertPath object and initializes it with a List of Certificates. |
| [CRL](http://docs.google.com/java/security/cert/CRL.html) | [**generateCRL**](http://docs.google.com/java/security/cert/CertificateFactory.html#generateCRL(java.io.InputStream))([InputStream](http://docs.google.com/java/io/InputStream.html) inStream)            Generates a certificate revocation list (CRL) object and initializes it with the data read from the input stream inStream. |
| [Collection](http://docs.google.com/java/util/Collection.html)<? extends [CRL](http://docs.google.com/java/security/cert/CRL.html)> | [**generateCRLs**](http://docs.google.com/java/security/cert/CertificateFactory.html#generateCRLs(java.io.InputStream))([InputStream](http://docs.google.com/java/io/InputStream.html) inStream)            Returns a (possibly empty) collection view of the CRLs read from the given input stream inStream. |
| [Iterator](http://docs.google.com/java/util/Iterator.html)<[String](http://docs.google.com/java/lang/String.html)> | [**getCertPathEncodings**](http://docs.google.com/java/security/cert/CertificateFactory.html#getCertPathEncodings())()            Returns an iteration of the CertPath encodings supported by this certificate factory, with the default encoding first. |
| static [CertificateFactory](http://docs.google.com/java/security/cert/CertificateFactory.html) | [**getInstance**](http://docs.google.com/java/security/cert/CertificateFactory.html#getInstance(java.lang.String))([String](http://docs.google.com/java/lang/String.html) type)            Returns a certificate factory object that implements the specified certificate type. |
| static [CertificateFactory](http://docs.google.com/java/security/cert/CertificateFactory.html) | [**getInstance**](http://docs.google.com/java/security/cert/CertificateFactory.html#getInstance(java.lang.String,%20java.security.Provider))([String](http://docs.google.com/java/lang/String.html) type, [Provider](http://docs.google.com/java/security/Provider.html) provider)            Returns a certificate factory object for the specified certificate type. |
| static [CertificateFactory](http://docs.google.com/java/security/cert/CertificateFactory.html) | [**getInstance**](http://docs.google.com/java/security/cert/CertificateFactory.html#getInstance(java.lang.String,%20java.lang.String))([String](http://docs.google.com/java/lang/String.html) type, [String](http://docs.google.com/java/lang/String.html) provider)            Returns a certificate factory object for the specified certificate type. |
| [Provider](http://docs.google.com/java/security/Provider.html) | [**getProvider**](http://docs.google.com/java/security/cert/CertificateFactory.html#getProvider())()            Returns the provider of this certificate factory. |
| [String](http://docs.google.com/java/lang/String.html) | [**getType**](http://docs.google.com/java/security/cert/CertificateFactory.html#getType())()            Returns the name of the certificate type associated with this certificate factory. |

| **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()), [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()), [toString](http://docs.google.com/java/lang/Object.html#toString()), [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** |
| --- |

### CertificateFactory

protected **CertificateFactory**([CertificateFactorySpi](http://docs.google.com/java/security/cert/CertificateFactorySpi.html) certFacSpi,  
 [Provider](http://docs.google.com/java/security/Provider.html) provider,  
 [String](http://docs.google.com/java/lang/String.html) type)

Creates a CertificateFactory object of the given type, and encapsulates the given provider implementation (SPI object) in it.

**Parameters:**certFacSpi - the provider implementation.provider - the provider.type - the certificate type.

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

### getInstance

public static final [CertificateFactory](http://docs.google.com/java/security/cert/CertificateFactory.html) **getInstance**([String](http://docs.google.com/java/lang/String.html) type)  
 throws [CertificateException](http://docs.google.com/java/security/cert/CertificateException.html)

Returns a certificate factory object that implements the specified certificate type.

This method traverses the list of registered security Providers, starting with the most preferred Provider. A new CertificateFactory object encapsulating the CertificateFactorySpi implementation from the first Provider that supports the specified type is returned.

Note that the list of registered providers may be retrieved via the [Security.getProviders()](http://docs.google.com/java/security/Security.html#getProviders()) method.

**Parameters:**type - the name of the requested certificate type. See Appendix A in the  [Java Cryptography Architecture API Specification & Reference](http://docs.google.com/technotes/guides/security/crypto/CryptoSpec.html#AppA)  for information about standard certificate types. **Returns:**a certificate factory object for the specified type. **Throws:** [CertificateException](http://docs.google.com/java/security/cert/CertificateException.html) - if no Provider supports a CertificateFactorySpi implementation for the specified type.**See Also:**[Provider](http://docs.google.com/java/security/Provider.html)

### getInstance

public static final [CertificateFactory](http://docs.google.com/java/security/cert/CertificateFactory.html) **getInstance**([String](http://docs.google.com/java/lang/String.html) type,  
 [String](http://docs.google.com/java/lang/String.html) provider)  
 throws [CertificateException](http://docs.google.com/java/security/cert/CertificateException.html),  
 [NoSuchProviderException](http://docs.google.com/java/security/NoSuchProviderException.html)

Returns a certificate factory object for the specified certificate type.

A new CertificateFactory object encapsulating the CertificateFactorySpi implementation from the specified provider is returned. The specified provider must be registered in the security provider list.

Note that the list of registered providers may be retrieved via the [Security.getProviders()](http://docs.google.com/java/security/Security.html#getProviders()) method.

**Parameters:**type - the certificate type. See Appendix A in the  [Java Cryptography Architecture API Specification & Reference](http://docs.google.com/technotes/guides/security/crypto/CryptoSpec.html#AppA)  for information about standard certificate types.provider - the name of the provider. **Returns:**a certificate factory object for the specified type. **Throws:** [CertificateException](http://docs.google.com/java/security/cert/CertificateException.html) - if a CertificateFactorySpi implementation for the specified algorithm is not available from the specified provider. [NoSuchProviderException](http://docs.google.com/java/security/NoSuchProviderException.html) - if the specified provider is not registered in the security provider list. [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if the provider name is null or empty.**See Also:**[Provider](http://docs.google.com/java/security/Provider.html)

### getInstance

public static final [CertificateFactory](http://docs.google.com/java/security/cert/CertificateFactory.html) **getInstance**([String](http://docs.google.com/java/lang/String.html) type,  
 [Provider](http://docs.google.com/java/security/Provider.html) provider)  
 throws [CertificateException](http://docs.google.com/java/security/cert/CertificateException.html)

Returns a certificate factory object for the specified certificate type.

A new CertificateFactory object encapsulating the CertificateFactorySpi implementation from the specified Provider object is returned. Note that the specified Provider object does not have to be registered in the provider list.

**Parameters:**type - the certificate type. See Appendix A in the  [Java Cryptography Architecture API Specification & Reference](http://docs.google.com/technotes/guides/security/crypto/CryptoSpec.html#AppA)  for information about standard certificate types.provider - the provider. **Returns:**a certificate factory object for the specified type. **Throws:** [CertificateException](http://docs.google.com/java/security/cert/CertificateException.html) - if a CertificateFactorySpi implementation for the specified algorithm is not available from the specified Provider object. [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if the provider is null.**Since:** 1.4 **See Also:**[Provider](http://docs.google.com/java/security/Provider.html)

### getProvider

public final [Provider](http://docs.google.com/java/security/Provider.html) **getProvider**()

Returns the provider of this certificate factory.

**Returns:**the provider of this certificate factory.

### getType

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

Returns the name of the certificate type associated with this certificate factory.

**Returns:**the name of the certificate type associated with this certificate factory.

### generateCertificate

public final [Certificate](http://docs.google.com/java/security/cert/Certificate.html) **generateCertificate**([InputStream](http://docs.google.com/java/io/InputStream.html) inStream)  
 throws [CertificateException](http://docs.google.com/java/security/cert/CertificateException.html)

Generates a certificate object and initializes it with the data read from the input stream inStream.

In order to take advantage of the specialized certificate format supported by this certificate factory, the returned certificate object can be typecast to the corresponding certificate class. For example, if this certificate factory implements X.509 certificates, the returned certificate object can be typecast to the X509Certificate class.

In the case of a certificate factory for X.509 certificates, the certificate provided in inStream must be DER-encoded and may be supplied in binary or printable (Base64) encoding. If the certificate is provided in Base64 encoding, it must be bounded at the beginning by -----BEGIN CERTIFICATE-----, and must be bounded at the end by -----END CERTIFICATE-----.

Note that if the given input stream does not support [mark](http://docs.google.com/java/io/InputStream.html#mark(int)) and [reset](http://docs.google.com/java/io/InputStream.html#reset()), this method will consume the entire input stream. Otherwise, each call to this method consumes one certificate and the read position of the input stream is positioned to the next available byte after the inherent end-of-certificate marker. If the data in the input stream does not contain an inherent end-of-certificate marker (other than EOF) and there is trailing data after the certificate is parsed, a CertificateException is thrown.

**Parameters:**inStream - an input stream with the certificate data. **Returns:**a certificate object initialized with the data from the input stream. **Throws:** [CertificateException](http://docs.google.com/java/security/cert/CertificateException.html) - on parsing errors.

### getCertPathEncodings

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

Returns an iteration of the CertPath encodings supported by this certificate factory, with the default encoding first. See Appendix A in the  [Java Certification Path API Programmer's Guide](http://docs.google.com/technotes/guides/security/certpath/CertPathProgGuide.html#AppA) for information about standard encoding names and their formats.

Attempts to modify the returned Iterator via its remove method result in an UnsupportedOperationException.

**Returns:**an Iterator over the names of the supported CertPath encodings (as Strings)**Since:** 1.4

### generateCertPath

public final [CertPath](http://docs.google.com/java/security/cert/CertPath.html) **generateCertPath**([InputStream](http://docs.google.com/java/io/InputStream.html) inStream)  
 throws [CertificateException](http://docs.google.com/java/security/cert/CertificateException.html)

Generates a CertPath object and initializes it with the data read from the InputStream inStream. The data is assumed to be in the default encoding. The name of the default encoding is the first element of the Iterator returned by the [getCertPathEncodings](http://docs.google.com/java/security/cert/CertificateFactory.html#getCertPathEncodings()) method.

**Parameters:**inStream - an InputStream containing the data **Returns:**a CertPath initialized with the data from the InputStream **Throws:** [CertificateException](http://docs.google.com/java/security/cert/CertificateException.html) - if an exception occurs while decoding**Since:** 1.4

### generateCertPath

public final [CertPath](http://docs.google.com/java/security/cert/CertPath.html) **generateCertPath**([InputStream](http://docs.google.com/java/io/InputStream.html) inStream,  
 [String](http://docs.google.com/java/lang/String.html) encoding)  
 throws [CertificateException](http://docs.google.com/java/security/cert/CertificateException.html)

Generates a CertPath object and initializes it with the data read from the InputStream inStream. The data is assumed to be in the specified encoding. See Appendix A in the  [Java Certification Path API Programmer's Guide](http://docs.google.com/technotes/guides/security/certpath/CertPathProgGuide.html#AppA) for information about standard encoding names and their formats.

**Parameters:**inStream - an InputStream containing the dataencoding - the encoding used for the data **Returns:**a CertPath initialized with the data from the InputStream **Throws:** [CertificateException](http://docs.google.com/java/security/cert/CertificateException.html) - if an exception occurs while decoding or the encoding requested is not supported**Since:** 1.4

### generateCertPath

public final [CertPath](http://docs.google.com/java/security/cert/CertPath.html) **generateCertPath**([List](http://docs.google.com/java/util/List.html)<? extends [Certificate](http://docs.google.com/java/security/cert/Certificate.html)> certificates)  
 throws [CertificateException](http://docs.google.com/java/security/cert/CertificateException.html)

Generates a CertPath object and initializes it with a List of Certificates.

The certificates supplied must be of a type supported by the CertificateFactory. They will be copied out of the supplied List object.

**Parameters:**certificates - a List of Certificates **Returns:**a CertPath initialized with the supplied list of certificates **Throws:** [CertificateException](http://docs.google.com/java/security/cert/CertificateException.html) - if an exception occurs**Since:** 1.4

### generateCertificates

public final [Collection](http://docs.google.com/java/util/Collection.html)<? extends [Certificate](http://docs.google.com/java/security/cert/Certificate.html)> **generateCertificates**([InputStream](http://docs.google.com/java/io/InputStream.html) inStream)  
 throws [CertificateException](http://docs.google.com/java/security/cert/CertificateException.html)

Returns a (possibly empty) collection view of the certificates read from the given input stream inStream.

In order to take advantage of the specialized certificate format supported by this certificate factory, each element in the returned collection view can be typecast to the corresponding certificate class. For example, if this certificate factory implements X.509 certificates, the elements in the returned collection can be typecast to the X509Certificate class.

In the case of a certificate factory for X.509 certificates, inStream may contain a sequence of DER-encoded certificates in the formats described for [generateCertificate](http://docs.google.com/java/security/cert/CertificateFactory.html#generateCertificate(java.io.InputStream)). In addition, inStream may contain a PKCS#7 certificate chain. This is a PKCS#7 *SignedData* object, with the only significant field being *certificates*. In particular, the signature and the contents are ignored. This format allows multiple certificates to be downloaded at once. If no certificates are present, an empty collection is returned.

Note that if the given input stream does not support [mark](http://docs.google.com/java/io/InputStream.html#mark(int)) and [reset](http://docs.google.com/java/io/InputStream.html#reset()), this method will consume the entire input stream.

**Parameters:**inStream - the input stream with the certificates. **Returns:**a (possibly empty) collection view of java.security.cert.Certificate objects initialized with the data from the input stream. **Throws:** [CertificateException](http://docs.google.com/java/security/cert/CertificateException.html) - on parsing errors.

### generateCRL

public final [CRL](http://docs.google.com/java/security/cert/CRL.html) **generateCRL**([InputStream](http://docs.google.com/java/io/InputStream.html) inStream)  
 throws [CRLException](http://docs.google.com/java/security/cert/CRLException.html)

Generates a certificate revocation list (CRL) object and initializes it with the data read from the input stream inStream.

In order to take advantage of the specialized CRL format supported by this certificate factory, the returned CRL object can be typecast to the corresponding CRL class. For example, if this certificate factory implements X.509 CRLs, the returned CRL object can be typecast to the X509CRL class.

Note that if the given input stream does not support [mark](http://docs.google.com/java/io/InputStream.html#mark(int)) and [reset](http://docs.google.com/java/io/InputStream.html#reset()), this method will consume the entire input stream. Otherwise, each call to this method consumes one CRL and the read position of the input stream is positioned to the next available byte after the the inherent end-of-CRL marker. If the data in the input stream does not contain an inherent end-of-CRL marker (other than EOF) and there is trailing data after the CRL is parsed, a CRLException is thrown.

**Parameters:**inStream - an input stream with the CRL data. **Returns:**a CRL object initialized with the data from the input stream. **Throws:** [CRLException](http://docs.google.com/java/security/cert/CRLException.html) - on parsing errors.

### generateCRLs

public final [Collection](http://docs.google.com/java/util/Collection.html)<? extends [CRL](http://docs.google.com/java/security/cert/CRL.html)> **generateCRLs**([InputStream](http://docs.google.com/java/io/InputStream.html) inStream)  
 throws [CRLException](http://docs.google.com/java/security/cert/CRLException.html)

Returns a (possibly empty) collection view of the CRLs read from the given input stream inStream.

In order to take advantage of the specialized CRL format supported by this certificate factory, each element in the returned collection view can be typecast to the corresponding CRL class. For example, if this certificate factory implements X.509 CRLs, the elements in the returned collection can be typecast to the X509CRL class.

In the case of a certificate factory for X.509 CRLs, inStream may contain a sequence of DER-encoded CRLs. In addition, inStream may contain a PKCS#7 CRL set. This is a PKCS#7 *SignedData* object, with the only significant field being *crls*. In particular, the signature and the contents are ignored. This format allows multiple CRLs to be downloaded at once. If no CRLs are present, an empty collection is returned.

Note that if the given input stream does not support [mark](http://docs.google.com/java/io/InputStream.html#mark(int)) and [reset](http://docs.google.com/java/io/InputStream.html#reset()), this method will consume the entire input stream.

**Parameters:**inStream - the input stream with the CRLs. **Returns:**a (possibly empty) collection view of java.security.cert.CRL objects initialized with the data from the input stream. **Throws:** [CRLException](http://docs.google.com/java/security/cert/CRLException.html) - on parsing errors.

| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/CertificateFactory.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/CertificateExpiredException.html)   [**NEXT CLASS**](http://docs.google.com/java/security/cert/CertificateFactorySpi.html) | [**FRAMES**](http://docs.google.com/index.html?java/security/cert/CertificateFactory.html)    [**NO FRAMES**](http://docs.google.com/CertificateFactory.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).