| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/Key.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/InvalidParameterException.html)   [**NEXT CLASS**](http://docs.google.com/java/security/KeyException.html) | [**FRAMES**](http://docs.google.com/index.html?java/security/Key.html)    [**NO FRAMES**](http://docs.google.com/Key.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | [FIELD](#3znysh7) | CONSTR | [METHOD](#2et92p0) | DETAIL: [FIELD](#tyjcwt) | CONSTR | [METHOD](#1t3h5sf) |

## **java.security**

Interface Key

**All Superinterfaces:** [Serializable](http://docs.google.com/java/io/Serializable.html) **All Known Subinterfaces:** [DHPrivateKey](http://docs.google.com/javax/crypto/interfaces/DHPrivateKey.html), [DHPublicKey](http://docs.google.com/javax/crypto/interfaces/DHPublicKey.html), [DSAPrivateKey](http://docs.google.com/java/security/interfaces/DSAPrivateKey.html), [DSAPublicKey](http://docs.google.com/java/security/interfaces/DSAPublicKey.html), [ECPrivateKey](http://docs.google.com/java/security/interfaces/ECPrivateKey.html), [ECPublicKey](http://docs.google.com/java/security/interfaces/ECPublicKey.html), [PBEKey](http://docs.google.com/javax/crypto/interfaces/PBEKey.html), [PrivateKey](http://docs.google.com/java/security/PrivateKey.html), [PublicKey](http://docs.google.com/java/security/PublicKey.html), [RSAMultiPrimePrivateCrtKey](http://docs.google.com/java/security/interfaces/RSAMultiPrimePrivateCrtKey.html), [RSAPrivateCrtKey](http://docs.google.com/java/security/interfaces/RSAPrivateCrtKey.html), [RSAPrivateKey](http://docs.google.com/java/security/interfaces/RSAPrivateKey.html), [RSAPublicKey](http://docs.google.com/java/security/interfaces/RSAPublicKey.html), [SecretKey](http://docs.google.com/javax/crypto/SecretKey.html) **All Known Implementing Classes:** [KerberosKey](http://docs.google.com/javax/security/auth/kerberos/KerberosKey.html), [SecretKeySpec](http://docs.google.com/javax/crypto/spec/SecretKeySpec.html)

public interface **Key**extends [Serializable](http://docs.google.com/java/io/Serializable.html)

The Key interface is the top-level interface for all keys. It defines the functionality shared by all key objects. All keys have three characteristics:

* An Algorithm  
  This is the key algorithm for that key. The key algorithm is usually an encryption or asymmetric operation algorithm (such as DSA or RSA), which will work with those algorithms and with related algorithms (such as MD5 with RSA, SHA-1 with RSA, Raw DSA, etc.) The name of the algorithm of a key is obtained using the [getAlgorithm](http://docs.google.com/java/security/Key.html#getAlgorithm()) method.
* An Encoded Form  
  This is an external encoded form for the key used when a standard representation of the key is needed outside the Java Virtual Machine, as when transmitting the key to some other party. The key is encoded according to a standard format (such as X.509 SubjectPublicKeyInfo or PKCS#8), and is returned using the [getEncoded](http://docs.google.com/java/security/Key.html#getEncoded()) method. Note: The syntax of the ASN.1 type SubjectPublicKeyInfo is defined as follows:  
   SubjectPublicKeyInfo ::= SEQUENCE {  
   algorithm AlgorithmIdentifier,  
   subjectPublicKey BIT STRING }  
    
   AlgorithmIdentifier ::= SEQUENCE {  
   algorithm OBJECT IDENTIFIER,  
   parameters ANY DEFINED BY algorithm OPTIONAL }  
   For more information, see [RFC 2459: Internet X.509 Public Key Infrastructure Certificate and CRL Profile](http://www.ietf.org/rfc/rfc2459.txt).
* A Format  
  This is the name of the format of the encoded key. It is returned by the [getFormat](http://docs.google.com/java/security/Key.html#getFormat()) method.

Keys are generally obtained through key generators, certificates, or various Identity classes used to manage keys. Keys may also be obtained from key specifications (transparent representations of the underlying key material) through the use of a key factory (see [KeyFactory](http://docs.google.com/java/security/KeyFactory.html)).

A Key should use KeyRep as its serialized representation. Note that a serialized Key may contain sensitive information which should not be exposed in untrusted environments. See the  [Security Appendix](http://docs.google.com/platform/serialization/spec/security.html) of the Serialization Specification for more information.

**See Also:**[PublicKey](http://docs.google.com/java/security/PublicKey.html), [PrivateKey](http://docs.google.com/java/security/PrivateKey.html), [KeyPair](http://docs.google.com/java/security/KeyPair.html), [KeyPairGenerator](http://docs.google.com/java/security/KeyPairGenerator.html), [KeyFactory](http://docs.google.com/java/security/KeyFactory.html), [KeyRep](http://docs.google.com/java/security/KeyRep.html), [KeySpec](http://docs.google.com/java/security/spec/KeySpec.html), [Identity](http://docs.google.com/java/security/Identity.html), [Signer](http://docs.google.com/java/security/Signer.html)

| **Field Summary** | |
| --- | --- |
| static long | [**serialVersionUID**](http://docs.google.com/java/security/Key.html#serialVersionUID)            The class fingerprint that is set to indicate serialization compatibility with a previous version of the class. |

| **Method Summary** | |
| --- | --- |
| [String](http://docs.google.com/java/lang/String.html) | [**getAlgorithm**](http://docs.google.com/java/security/Key.html#getAlgorithm())()            Returns the standard algorithm name for this key. |
| byte[] | [**getEncoded**](http://docs.google.com/java/security/Key.html#getEncoded())()            Returns the key in its primary encoding format, or null if this key does not support encoding. |
| [String](http://docs.google.com/java/lang/String.html) | [**getFormat**](http://docs.google.com/java/security/Key.html#getFormat())()            Returns the name of the primary encoding format of this key, or null if this key does not support encoding. |

| **Field Detail** |
| --- |

### serialVersionUID

static final long **serialVersionUID**

The class fingerprint that is set to indicate serialization compatibility with a previous version of the class.

**See Also:**[Constant Field Values](http://docs.google.com/constant-values.html#java.security.Key.serialVersionUID)

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

### getAlgorithm

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

Returns the standard algorithm name for this key. For example, "DSA" would indicate that this key is a DSA key. 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 algorithm names.

**Returns:**the name of the algorithm associated with this key.

### getFormat

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

Returns the name of the primary encoding format of this key, or null if this key does not support encoding. The primary encoding format is named in terms of the appropriate ASN.1 data format, if an ASN.1 specification for this key exists. For example, the name of the ASN.1 data format for public keys is *SubjectPublicKeyInfo*, as defined by the X.509 standard; in this case, the returned format is "X.509". Similarly, the name of the ASN.1 data format for private keys is *PrivateKeyInfo*, as defined by the PKCS #8 standard; in this case, the returned format is "PKCS#8".

**Returns:**the primary encoding format of the key.

### getEncoded

byte[] **getEncoded**()

Returns the key in its primary encoding format, or null if this key does not support encoding.

**Returns:**the encoded key, or null if the key does not support encoding.

| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/Key.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/InvalidParameterException.html)   [**NEXT CLASS**](http://docs.google.com/java/security/KeyException.html) | [**FRAMES**](http://docs.google.com/index.html?java/security/Key.html)    [**NO FRAMES**](http://docs.google.com/Key.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | [FIELD](#3znysh7) | CONSTR | [METHOD](#2et92p0) | DETAIL: [FIELD](#tyjcwt) | CONSTR | [METHOD](#1t3h5sf) |

[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).