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

Class KeyPairGeneratorSpi

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

**Direct Known Subclasses:** [KeyPairGenerator](http://docs.google.com/java/security/KeyPairGenerator.html)

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

This class defines the *Service Provider Interface* (**SPI**) for the KeyPairGenerator class, which is used to generate pairs of public and private keys.

All the abstract methods in this class must be implemented by each cryptographic service provider who wishes to supply the implementation of a key pair generator for a particular algorithm.

In case the client does not explicitly initialize the KeyPairGenerator (via a call to an initialize method), each provider must supply (and document) a default initialization. For example, the *Sun* provider uses a default modulus size (keysize) of 1024 bits.

**See Also:**[KeyPairGenerator](http://docs.google.com/java/security/KeyPairGenerator.html), [AlgorithmParameterSpec](http://docs.google.com/java/security/spec/AlgorithmParameterSpec.html)

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

| **Method Summary** | |
| --- | --- |
| abstract  [KeyPair](http://docs.google.com/java/security/KeyPair.html) | [**generateKeyPair**](http://docs.google.com/java/security/KeyPairGeneratorSpi.html#generateKeyPair())()            Generates a key pair. |
| void | [**initialize**](http://docs.google.com/java/security/KeyPairGeneratorSpi.html#initialize(java.security.spec.AlgorithmParameterSpec,%20java.security.SecureRandom))([AlgorithmParameterSpec](http://docs.google.com/java/security/spec/AlgorithmParameterSpec.html) params, [SecureRandom](http://docs.google.com/java/security/SecureRandom.html) random)            Initializes the key pair generator using the specified parameter set and user-provided source of randomness. |
| abstract  void | [**initialize**](http://docs.google.com/java/security/KeyPairGeneratorSpi.html#initialize(int,%20java.security.SecureRandom))(int keysize, [SecureRandom](http://docs.google.com/java/security/SecureRandom.html) random)            Initializes the key pair generator for a certain keysize, using the default parameter set. |

| **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** |
| --- |

### KeyPairGeneratorSpi

public **KeyPairGeneratorSpi**()

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

### initialize

public abstract void **initialize**(int keysize,  
 [SecureRandom](http://docs.google.com/java/security/SecureRandom.html) random)

Initializes the key pair generator for a certain keysize, using the default parameter set.

**Parameters:**keysize - the keysize. This is an algorithm-specific metric, such as modulus length, specified in number of bits.random - the source of randomness for this generator. **Throws:** [InvalidParameterException](http://docs.google.com/java/security/InvalidParameterException.html) - if the keysize is not supported by this KeyPairGeneratorSpi object.

### initialize

public void **initialize**([AlgorithmParameterSpec](http://docs.google.com/java/security/spec/AlgorithmParameterSpec.html) params,  
 [SecureRandom](http://docs.google.com/java/security/SecureRandom.html) random)  
 throws [InvalidAlgorithmParameterException](http://docs.google.com/java/security/InvalidAlgorithmParameterException.html)

Initializes the key pair generator using the specified parameter set and user-provided source of randomness.

This concrete method has been added to this previously-defined abstract class. (For backwards compatibility, it cannot be abstract.) It may be overridden by a provider to initialize the key pair generator. Such an override is expected to throw an InvalidAlgorithmParameterException if a parameter is inappropriate for this key pair generator. If this method is not overridden, it always throws an UnsupportedOperationException.

**Parameters:**params - the parameter set used to generate the keys.random - the source of randomness for this generator. **Throws:** [InvalidAlgorithmParameterException](http://docs.google.com/java/security/InvalidAlgorithmParameterException.html) - if the given parameters are inappropriate for this key pair generator.**Since:** 1.2

### generateKeyPair

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

Generates a key pair. Unless an initialization method is called using a KeyPairGenerator interface, algorithm-specific defaults will be used. This will generate a new key pair every time it is called.

**Returns:**the newly generated KeyPair

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