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

## **java.security.interfaces**

Interface DSAKeyPairGenerator

public interface **DSAKeyPairGenerator**

An interface to an object capable of generating DSA key pairs.

The initialize methods may each be called any number of times. If no initialize method is called on a DSAKeyPairGenerator, the default is to generate 1024-bit keys, using precomputed p, q and g parameters and an instance of SecureRandom as the random bit source.

Users wishing to indicate DSA-specific parameters, and to generate a key pair suitable for use with the DSA algorithm typically

1. Get a key pair generator for the DSA algorithm by calling the KeyPairGenerator getInstance method with "DSA" as its argument.
2. Initialize the generator by casting the result to a DSAKeyPairGenerator and calling one of the initialize methods from this DSAKeyPairGenerator interface.
3. Generate a key pair by calling the generateKeyPair method from the KeyPairGenerator class.

Note: it is not always necessary to do do algorithm-specific initialization for a DSA key pair generator. That is, it is not always necessary to call an initialize method in this interface. Algorithm-independent initialization using the initialize method in the KeyPairGenerator interface is all that is needed when you accept defaults for algorithm-specific parameters.

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

| **Method Summary** | |
| --- | --- |
| void | [**initialize**](http://docs.google.com/java/security/interfaces/DSAKeyPairGenerator.html#initialize(java.security.interfaces.DSAParams,%20java.security.SecureRandom))([DSAParams](http://docs.google.com/java/security/interfaces/DSAParams.html) params, [SecureRandom](http://docs.google.com/java/security/SecureRandom.html) random)            Initializes the key pair generator using the DSA family parameters (p,q and g) and an optional SecureRandom bit source. |
| void | [**initialize**](http://docs.google.com/java/security/interfaces/DSAKeyPairGenerator.html#initialize(int,%20boolean,%20java.security.SecureRandom))(int modlen, boolean genParams, [SecureRandom](http://docs.google.com/java/security/SecureRandom.html) random)            Initializes the key pair generator for a given modulus length (instead of parameters), and an optional SecureRandom bit source. |

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

### initialize

void **initialize**([DSAParams](http://docs.google.com/java/security/interfaces/DSAParams.html) params,  
 [SecureRandom](http://docs.google.com/java/security/SecureRandom.html) random)  
 throws [InvalidParameterException](http://docs.google.com/java/security/InvalidParameterException.html)

Initializes the key pair generator using the DSA family parameters (p,q and g) and an optional SecureRandom bit source. If a SecureRandom bit source is needed but not supplied, i.e. null, a default SecureRandom instance will be used.

**Parameters:**params - the parameters to use to generate the keys.random - the random bit source to use to generate key bits; can be null. **Throws:** [InvalidParameterException](http://docs.google.com/java/security/InvalidParameterException.html) - if the params value is invalid or null.

### initialize

void **initialize**(int modlen,  
 boolean genParams,  
 [SecureRandom](http://docs.google.com/java/security/SecureRandom.html) random)  
 throws [InvalidParameterException](http://docs.google.com/java/security/InvalidParameterException.html)

Initializes the key pair generator for a given modulus length (instead of parameters), and an optional SecureRandom bit source. If a SecureRandom bit source is needed but not supplied, i.e. null, a default SecureRandom instance will be used.

If genParams is true, this method generates new p, q and g parameters. If it is false, the method uses precomputed parameters for the modulus length requested. If there are no precomputed parameters for that modulus length, an exception will be thrown. It is guaranteed that there will always be default parameters for modulus lengths of 512 and 1024 bits.

**Parameters:**modlen - the modulus length in bits. Valid values are any multiple of 8 between 512 and 1024, inclusive.random - the random bit source to use to generate key bits; can be null.genParams - whether or not to generate new parameters for the modulus length requested. **Throws:** [InvalidParameterException](http://docs.google.com/java/security/InvalidParameterException.html) - if modlen is not between 512 and 1024, or if genParams is false and there are no precomputed parameters for the requested modulus length.

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

[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).