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

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

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

The AlgorithmParameterGenerator class is used to generate a set of parameters to be used with a certain algorithm. Parameter generators are constructed using the getInstance factory methods (static methods that return instances of a given class).

The object that will generate the parameters can be initialized in two different ways: in an algorithm-independent manner, or in an algorithm-specific manner:

* The algorithm-independent approach uses the fact that all parameter generators share the concept of a "size" and a source of randomness. The measure of size is universally shared by all algorithm parameters, though it is interpreted differently for different algorithms. For example, in the case of parameters for the *DSA* algorithm, "size" corresponds to the size of the prime modulus (in bits). When using this approach, algorithm-specific parameter generation values - if any - default to some standard values, unless they can be derived from the specified size.
* The other approach initializes a parameter generator object using algorithm-specific semantics, which are represented by a set of algorithm-specific parameter generation values. To generate Diffie-Hellman system parameters, for example, the parameter generation values usually consist of the size of the prime modulus and the size of the random exponent, both specified in number of bits.

In case the client does not explicitly initialize the AlgorithmParameterGenerator (via a call to an init method), each provider must supply (and document) a default initialization. For example, the Sun provider uses a default modulus prime size of 1024 bits for the generation of DSA parameters.

**Since:** 1.2 **See Also:**[AlgorithmParameters](http://docs.google.com/java/security/AlgorithmParameters.html), [AlgorithmParameterSpec](http://docs.google.com/java/security/spec/AlgorithmParameterSpec.html)

| **Constructor Summary** | |
| --- | --- |
| protected | [**AlgorithmParameterGenerator**](http://docs.google.com/java/security/AlgorithmParameterGenerator.html#AlgorithmParameterGenerator(java.security.AlgorithmParameterGeneratorSpi,%20java.security.Provider,%20java.lang.String))([AlgorithmParameterGeneratorSpi](http://docs.google.com/java/security/AlgorithmParameterGeneratorSpi.html) paramGenSpi, [Provider](http://docs.google.com/java/security/Provider.html) provider, [String](http://docs.google.com/java/lang/String.html) algorithm)            Creates an AlgorithmParameterGenerator object. |

| **Method Summary** | |
| --- | --- |
| [AlgorithmParameters](http://docs.google.com/java/security/AlgorithmParameters.html) | [**generateParameters**](http://docs.google.com/java/security/AlgorithmParameterGenerator.html#generateParameters())()            Generates the parameters. |
| [String](http://docs.google.com/java/lang/String.html) | [**getAlgorithm**](http://docs.google.com/java/security/AlgorithmParameterGenerator.html#getAlgorithm())()            Returns the standard name of the algorithm this parameter generator is associated with. |
| static [AlgorithmParameterGenerator](http://docs.google.com/java/security/AlgorithmParameterGenerator.html) | [**getInstance**](http://docs.google.com/java/security/AlgorithmParameterGenerator.html#getInstance(java.lang.String))([String](http://docs.google.com/java/lang/String.html) algorithm)            Returns an AlgorithmParameterGenerator object for generating a set of parameters to be used with the specified algorithm. |
| static [AlgorithmParameterGenerator](http://docs.google.com/java/security/AlgorithmParameterGenerator.html) | [**getInstance**](http://docs.google.com/java/security/AlgorithmParameterGenerator.html#getInstance(java.lang.String,%20java.security.Provider))([String](http://docs.google.com/java/lang/String.html) algorithm, [Provider](http://docs.google.com/java/security/Provider.html) provider)            Returns an AlgorithmParameterGenerator object for generating a set of parameters to be used with the specified algorithm. |
| static [AlgorithmParameterGenerator](http://docs.google.com/java/security/AlgorithmParameterGenerator.html) | [**getInstance**](http://docs.google.com/java/security/AlgorithmParameterGenerator.html#getInstance(java.lang.String,%20java.lang.String))([String](http://docs.google.com/java/lang/String.html) algorithm, [String](http://docs.google.com/java/lang/String.html) provider)            Returns an AlgorithmParameterGenerator object for generating a set of parameters to be used with the specified algorithm. |
| [Provider](http://docs.google.com/java/security/Provider.html) | [**getProvider**](http://docs.google.com/java/security/AlgorithmParameterGenerator.html#getProvider())()            Returns the provider of this algorithm parameter generator object. |
| void | [**init**](http://docs.google.com/java/security/AlgorithmParameterGenerator.html#init(java.security.spec.AlgorithmParameterSpec))([AlgorithmParameterSpec](http://docs.google.com/java/security/spec/AlgorithmParameterSpec.html) genParamSpec)            Initializes this parameter generator with a set of algorithm-specific parameter generation values. |
| void | [**init**](http://docs.google.com/java/security/AlgorithmParameterGenerator.html#init(java.security.spec.AlgorithmParameterSpec,%20java.security.SecureRandom))([AlgorithmParameterSpec](http://docs.google.com/java/security/spec/AlgorithmParameterSpec.html) genParamSpec, [SecureRandom](http://docs.google.com/java/security/SecureRandom.html) random)            Initializes this parameter generator with a set of algorithm-specific parameter generation values. |
| void | [**init**](http://docs.google.com/java/security/AlgorithmParameterGenerator.html#init(int))(int size)            Initializes this parameter generator for a certain size. |
| void | [**init**](http://docs.google.com/java/security/AlgorithmParameterGenerator.html#init(int,%20java.security.SecureRandom))(int size, [SecureRandom](http://docs.google.com/java/security/SecureRandom.html) random)            Initializes this parameter generator for a certain size and source of randomness. |

| **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** |
| --- |

### AlgorithmParameterGenerator

protected **AlgorithmParameterGenerator**([AlgorithmParameterGeneratorSpi](http://docs.google.com/java/security/AlgorithmParameterGeneratorSpi.html) paramGenSpi,  
 [Provider](http://docs.google.com/java/security/Provider.html) provider,  
 [String](http://docs.google.com/java/lang/String.html) algorithm)

Creates an AlgorithmParameterGenerator object.

**Parameters:**paramGenSpi - the delegateprovider - the provideralgorithm - the algorithm

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

### getAlgorithm

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

Returns the standard name of the algorithm this parameter generator is associated with.

**Returns:**the string name of the algorithm.

### getInstance

public static [AlgorithmParameterGenerator](http://docs.google.com/java/security/AlgorithmParameterGenerator.html) **getInstance**([String](http://docs.google.com/java/lang/String.html) algorithm)  
 throws [NoSuchAlgorithmException](http://docs.google.com/java/security/NoSuchAlgorithmException.html)

Returns an AlgorithmParameterGenerator object for generating a set of parameters to be used with the specified algorithm.

This method traverses the list of registered security Providers, starting with the most preferred Provider. A new AlgorithmParameterGenerator object encapsulating the AlgorithmParameterGeneratorSpi implementation from the first Provider that supports the specified algorithm 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:**algorithm - the name of the algorithm this parameter generator is associated with. 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 new AlgorithmParameterGenerator object. **Throws:** [NoSuchAlgorithmException](http://docs.google.com/java/security/NoSuchAlgorithmException.html) - if no Provider supports an AlgorithmParameterGeneratorSpi implementation for the specified algorithm.**See Also:**[Provider](http://docs.google.com/java/security/Provider.html)

### getInstance

public static [AlgorithmParameterGenerator](http://docs.google.com/java/security/AlgorithmParameterGenerator.html) **getInstance**([String](http://docs.google.com/java/lang/String.html) algorithm,  
 [String](http://docs.google.com/java/lang/String.html) provider)  
 throws [NoSuchAlgorithmException](http://docs.google.com/java/security/NoSuchAlgorithmException.html),  
 [NoSuchProviderException](http://docs.google.com/java/security/NoSuchProviderException.html)

Returns an AlgorithmParameterGenerator object for generating a set of parameters to be used with the specified algorithm.

A new AlgorithmParameterGenerator object encapsulating the AlgorithmParameterGeneratorSpi 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:**algorithm - the name of the algorithm this parameter generator is associated with. 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.provider - the string name of the Provider. **Returns:**the new AlgorithmParameterGenerator object. **Throws:** [NoSuchAlgorithmException](http://docs.google.com/java/security/NoSuchAlgorithmException.html) - if an AlgorithmParameterGeneratorSpi 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 [AlgorithmParameterGenerator](http://docs.google.com/java/security/AlgorithmParameterGenerator.html) **getInstance**([String](http://docs.google.com/java/lang/String.html) algorithm,  
 [Provider](http://docs.google.com/java/security/Provider.html) provider)  
 throws [NoSuchAlgorithmException](http://docs.google.com/java/security/NoSuchAlgorithmException.html)

Returns an AlgorithmParameterGenerator object for generating a set of parameters to be used with the specified algorithm.

A new AlgorithmParameterGenerator object encapsulating the AlgorithmParameterGeneratorSpi 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:**algorithm - the string name of the algorithm this parameter generator is associated with. 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.provider - the Provider object. **Returns:**the new AlgorithmParameterGenerator object. **Throws:** [NoSuchAlgorithmException](http://docs.google.com/java/security/NoSuchAlgorithmException.html) - if an AlgorithmParameterGeneratorSpi implementation for the specified algorithm is not available from the specified Provider object. [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if the specified 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 algorithm parameter generator object.

**Returns:**the provider of this algorithm parameter generator object

### init

public final void **init**(int size)

Initializes this parameter generator for a certain size. To create the parameters, the SecureRandom implementation of the highest-priority installed provider is used as the source of randomness. (If none of the installed providers supply an implementation of SecureRandom, a system-provided source of randomness is used.)

**Parameters:**size - the size (number of bits).

### init

public final void **init**(int size,  
 [SecureRandom](http://docs.google.com/java/security/SecureRandom.html) random)

Initializes this parameter generator for a certain size and source of randomness.

**Parameters:**size - the size (number of bits).random - the source of randomness.

### init

public final void **init**([AlgorithmParameterSpec](http://docs.google.com/java/security/spec/AlgorithmParameterSpec.html) genParamSpec)  
 throws [InvalidAlgorithmParameterException](http://docs.google.com/java/security/InvalidAlgorithmParameterException.html)

Initializes this parameter generator with a set of algorithm-specific parameter generation values. To generate the parameters, the SecureRandom implementation of the highest-priority installed provider is used as the source of randomness. (If none of the installed providers supply an implementation of SecureRandom, a system-provided source of randomness is used.)

**Parameters:**genParamSpec - the set of algorithm-specific parameter generation values. **Throws:** [InvalidAlgorithmParameterException](http://docs.google.com/java/security/InvalidAlgorithmParameterException.html) - if the given parameter generation values are inappropriate for this parameter generator.

### init

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

Initializes this parameter generator with a set of algorithm-specific parameter generation values.

**Parameters:**genParamSpec - the set of algorithm-specific parameter generation values.random - the source of randomness. **Throws:** [InvalidAlgorithmParameterException](http://docs.google.com/java/security/InvalidAlgorithmParameterException.html) - if the given parameter generation values are inappropriate for this parameter generator.

### generateParameters

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

Generates the parameters.

**Returns:**the new AlgorithmParameters object.

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