| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/ExemptionMechanism.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/javax/crypto/EncryptedPrivateKeyInfo.html)   [**NEXT CLASS**](http://docs.google.com/javax/crypto/ExemptionMechanismException.html) | [**FRAMES**](http://docs.google.com/index.html?javax/crypto/ExemptionMechanism.html)    [**NO FRAMES**](http://docs.google.com/ExemptionMechanism.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | FIELD | [CONSTR](#3znysh7) | [METHOD](#2et92p0) | DETAIL: FIELD | [CONSTR](#3dy6vkm) | [METHOD](#4d34og8) |

## **javax.crypto**

Class ExemptionMechanism

[java.lang.Object](http://docs.google.com/java/lang/Object.html)  
 **javax.crypto.ExemptionMechanism**

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

This class provides the functionality of an exemption mechanism, examples of which are *key recovery*, *key weakening*, and *key escrow*.

Applications or applets that use an exemption mechanism may be granted stronger encryption capabilities than those which don't.

**Since:** 1.4

| **Constructor Summary** | |
| --- | --- |
| protected | [**ExemptionMechanism**](http://docs.google.com/javax/crypto/ExemptionMechanism.html#ExemptionMechanism(javax.crypto.ExemptionMechanismSpi,%20java.security.Provider,%20java.lang.String))([ExemptionMechanismSpi](http://docs.google.com/javax/crypto/ExemptionMechanismSpi.html) exmechSpi, [Provider](http://docs.google.com/java/security/Provider.html) provider, [String](http://docs.google.com/java/lang/String.html) mechanism)            Creates a ExemptionMechanism object. |

| **Method Summary** | |
| --- | --- |
| protected  void | [**finalize**](http://docs.google.com/javax/crypto/ExemptionMechanism.html#finalize())()            Ensures that the key stored away by this ExemptionMechanism object will be wiped out when there are no more references to it. |
| byte[] | [**genExemptionBlob**](http://docs.google.com/javax/crypto/ExemptionMechanism.html#genExemptionBlob())()            Generates the exemption mechanism key blob. |
| int | [**genExemptionBlob**](http://docs.google.com/javax/crypto/ExemptionMechanism.html#genExemptionBlob(byte%5B%5D))(byte[] output)            Generates the exemption mechanism key blob, and stores the result in the output buffer. |
| int | [**genExemptionBlob**](http://docs.google.com/javax/crypto/ExemptionMechanism.html#genExemptionBlob(byte%5B%5D,%20int))(byte[] output, int outputOffset)            Generates the exemption mechanism key blob, and stores the result in the output buffer, starting at outputOffset inclusive. |
| static [ExemptionMechanism](http://docs.google.com/javax/crypto/ExemptionMechanism.html) | [**getInstance**](http://docs.google.com/javax/crypto/ExemptionMechanism.html#getInstance(java.lang.String))([String](http://docs.google.com/java/lang/String.html) algorithm)            Returns an ExemptionMechanism object that implements the specified exemption mechanism algorithm. |
| static [ExemptionMechanism](http://docs.google.com/javax/crypto/ExemptionMechanism.html) | [**getInstance**](http://docs.google.com/javax/crypto/ExemptionMechanism.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 ExemptionMechanism object that implements the specified exemption mechanism algorithm. |
| static [ExemptionMechanism](http://docs.google.com/javax/crypto/ExemptionMechanism.html) | [**getInstance**](http://docs.google.com/javax/crypto/ExemptionMechanism.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 ExemptionMechanism object that implements the specified exemption mechanism algorithm. |
| [String](http://docs.google.com/java/lang/String.html) | [**getName**](http://docs.google.com/javax/crypto/ExemptionMechanism.html#getName())()            Returns the exemption mechanism name of this ExemptionMechanism object. |
| int | [**getOutputSize**](http://docs.google.com/javax/crypto/ExemptionMechanism.html#getOutputSize(int))(int inputLen)            Returns the length in bytes that an output buffer would need to be in order to hold the result of the next [genExemptionBlob](http://docs.google.com/javax/crypto/ExemptionMechanism.html#genExemptionBlob(byte%5B%5D)) operation, given the input length inputLen (in bytes). |
| [Provider](http://docs.google.com/java/security/Provider.html) | [**getProvider**](http://docs.google.com/javax/crypto/ExemptionMechanism.html#getProvider())()            Returns the provider of this ExemptionMechanism object. |
| void | [**init**](http://docs.google.com/javax/crypto/ExemptionMechanism.html#init(java.security.Key))([Key](http://docs.google.com/java/security/Key.html) key)            Initializes this exemption mechanism with a key. |
| void | [**init**](http://docs.google.com/javax/crypto/ExemptionMechanism.html#init(java.security.Key,%20java.security.AlgorithmParameters))([Key](http://docs.google.com/java/security/Key.html) key, [AlgorithmParameters](http://docs.google.com/java/security/AlgorithmParameters.html) params)            Initializes this exemption mechanism with a key and a set of algorithm parameters. |
| void | [**init**](http://docs.google.com/javax/crypto/ExemptionMechanism.html#init(java.security.Key,%20java.security.spec.AlgorithmParameterSpec))([Key](http://docs.google.com/java/security/Key.html) key, [AlgorithmParameterSpec](http://docs.google.com/java/security/spec/AlgorithmParameterSpec.html) params)            Initializes this exemption mechanism with a key and a set of algorithm parameters. |
| boolean | [**isCryptoAllowed**](http://docs.google.com/javax/crypto/ExemptionMechanism.html#isCryptoAllowed(java.security.Key))([Key](http://docs.google.com/java/security/Key.html) key)            Returns whether the result blob has been generated successfully by this exemption mechanism. |

| **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)), [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** |
| --- |

### ExemptionMechanism

protected **ExemptionMechanism**([ExemptionMechanismSpi](http://docs.google.com/javax/crypto/ExemptionMechanismSpi.html) exmechSpi,  
 [Provider](http://docs.google.com/java/security/Provider.html) provider,  
 [String](http://docs.google.com/java/lang/String.html) mechanism)

Creates a ExemptionMechanism object.

**Parameters:**exmechSpi - the delegateprovider - the providermechanism - the exemption mechanism

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

### getName

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

Returns the exemption mechanism name of this ExemptionMechanism object.

This is the same name that was specified in one of the getInstance calls that created this ExemptionMechanism object.

**Returns:**the exemption mechanism name of this ExemptionMechanism object.

### getInstance

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

Returns an ExemptionMechanism object that implements the specified exemption mechanism algorithm.

This method traverses the list of registered security Providers, starting with the most preferred Provider. A new ExemptionMechanism object encapsulating the ExemptionMechanismSpi 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 standard name of the requested exemption mechanism. See Appendix A in the  [Java Cryptography Architecture Reference Guide](http://docs.google.com/technotes/guides/security/crypto/CryptoSpec.html#AppA) for information about standard exemption mechanism names. **Returns:**the new ExemptionMechanism object. **Throws:** [NullPointerException](http://docs.google.com/java/lang/NullPointerException.html) - if algorithm is null. [NoSuchAlgorithmException](http://docs.google.com/java/security/NoSuchAlgorithmException.html) - if no Provider supports an ExemptionMechanismSpi implementation for the specified algorithm.**See Also:**[Provider](http://docs.google.com/java/security/Provider.html)

### getInstance

public static final [ExemptionMechanism](http://docs.google.com/javax/crypto/ExemptionMechanism.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 ExemptionMechanism object that implements the specified exemption mechanism algorithm.

A new ExemptionMechanism object encapsulating the ExemptionMechanismSpi 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 standard name of the requested exemption mechanism. See Appendix A in the  [Java Cryptography Architecture Reference Guide](http://docs.google.com/technotes/guides/security/crypto/CryptoSpec.html#AppA) for information about standard exemption mechanism names.provider - the name of the provider. **Returns:**the new ExemptionMechanism object. **Throws:** [NullPointerException](http://docs.google.com/java/lang/NullPointerException.html) - if algorithm is null. [NoSuchAlgorithmException](http://docs.google.com/java/security/NoSuchAlgorithmException.html) - if an ExemptionMechanismSpi 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 is null or empty.**See Also:**[Provider](http://docs.google.com/java/security/Provider.html)

### getInstance

public static final [ExemptionMechanism](http://docs.google.com/javax/crypto/ExemptionMechanism.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 ExemptionMechanism object that implements the specified exemption mechanism algorithm.

A new ExemptionMechanism object encapsulating the ExemptionMechanismSpi 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 standard name of the requested exemption mechanism. See Appendix A in the  [Java Cryptography Architecture Reference Guide](http://docs.google.com/technotes/guides/security/crypto/CryptoSpec.html#AppA) for information about standard exemption mechanism names.provider - the provider. **Returns:**the new ExemptionMechanism object. **Throws:** [NullPointerException](http://docs.google.com/java/lang/NullPointerException.html) - if algorithm is null. [NoSuchAlgorithmException](http://docs.google.com/java/security/NoSuchAlgorithmException.html) - if an ExemptionMechanismSpi 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.**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 ExemptionMechanism object.

**Returns:**the provider of this ExemptionMechanism object.

### isCryptoAllowed

public final boolean **isCryptoAllowed**([Key](http://docs.google.com/java/security/Key.html) key)  
 throws [ExemptionMechanismException](http://docs.google.com/javax/crypto/ExemptionMechanismException.html)

Returns whether the result blob has been generated successfully by this exemption mechanism.

The method also makes sure that the key passed in is the same as the one this exemption mechanism used in initializing and generating phases.

**Parameters:**key - the key the crypto is going to use. **Returns:**whether the result blob of the same key has been generated successfully by this exemption mechanism; false if key is null. **Throws:** [ExemptionMechanismException](http://docs.google.com/javax/crypto/ExemptionMechanismException.html) - if problem(s) encountered while determining whether the result blob has been generated successfully by this exemption mechanism object.

### getOutputSize

public final int **getOutputSize**(int inputLen)  
 throws [IllegalStateException](http://docs.google.com/java/lang/IllegalStateException.html)

Returns the length in bytes that an output buffer would need to be in order to hold the result of the next [genExemptionBlob](http://docs.google.com/javax/crypto/ExemptionMechanism.html#genExemptionBlob(byte%5B%5D)) operation, given the input length inputLen (in bytes).

The actual output length of the next [genExemptionBlob](http://docs.google.com/javax/crypto/ExemptionMechanism.html#genExemptionBlob(byte%5B%5D)) call may be smaller than the length returned by this method.

**Parameters:**inputLen - the input length (in bytes) **Returns:**the required output buffer size (in bytes) **Throws:** [IllegalStateException](http://docs.google.com/java/lang/IllegalStateException.html) - if this exemption mechanism is in a wrong state (e.g., has not yet been initialized)

### init

public final void **init**([Key](http://docs.google.com/java/security/Key.html) key)  
 throws [InvalidKeyException](http://docs.google.com/java/security/InvalidKeyException.html),  
 [ExemptionMechanismException](http://docs.google.com/javax/crypto/ExemptionMechanismException.html)

Initializes this exemption mechanism with a key.

If this exemption mechanism requires any algorithm parameters that cannot be derived from the given key, the underlying exemption mechanism implementation is supposed to generate the required parameters itself (using provider-specific default values); in the case that algorithm parameters must be specified by the caller, an InvalidKeyException is raised.

**Parameters:**key - the key for this exemption mechanism **Throws:** [InvalidKeyException](http://docs.google.com/java/security/InvalidKeyException.html) - if the given key is inappropriate for this exemption mechanism. [ExemptionMechanismException](http://docs.google.com/javax/crypto/ExemptionMechanismException.html) - if problem(s) encountered in the process of initializing.

### init

public final void **init**([Key](http://docs.google.com/java/security/Key.html) key,  
 [AlgorithmParameterSpec](http://docs.google.com/java/security/spec/AlgorithmParameterSpec.html) params)  
 throws [InvalidKeyException](http://docs.google.com/java/security/InvalidKeyException.html),  
 [InvalidAlgorithmParameterException](http://docs.google.com/java/security/InvalidAlgorithmParameterException.html),  
 [ExemptionMechanismException](http://docs.google.com/javax/crypto/ExemptionMechanismException.html)

Initializes this exemption mechanism with a key and a set of algorithm parameters.

If this exemption mechanism requires any algorithm parameters and params is null, the underlying exemption mechanism implementation is supposed to generate the required parameters itself (using provider-specific default values); in the case that algorithm parameters must be specified by the caller, an InvalidAlgorithmParameterException is raised.

**Parameters:**key - the key for this exemption mechanismparams - the algorithm parameters **Throws:** [InvalidKeyException](http://docs.google.com/java/security/InvalidKeyException.html) - if the given key is inappropriate for this exemption mechanism. [InvalidAlgorithmParameterException](http://docs.google.com/java/security/InvalidAlgorithmParameterException.html) - if the given algorithm parameters are inappropriate for this exemption mechanism. [ExemptionMechanismException](http://docs.google.com/javax/crypto/ExemptionMechanismException.html) - if problem(s) encountered in the process of initializing.

### init

public final void **init**([Key](http://docs.google.com/java/security/Key.html) key,  
 [AlgorithmParameters](http://docs.google.com/java/security/AlgorithmParameters.html) params)  
 throws [InvalidKeyException](http://docs.google.com/java/security/InvalidKeyException.html),  
 [InvalidAlgorithmParameterException](http://docs.google.com/java/security/InvalidAlgorithmParameterException.html),  
 [ExemptionMechanismException](http://docs.google.com/javax/crypto/ExemptionMechanismException.html)

Initializes this exemption mechanism with a key and a set of algorithm parameters.

If this exemption mechanism requires any algorithm parameters and params is null, the underlying exemption mechanism implementation is supposed to generate the required parameters itself (using provider-specific default values); in the case that algorithm parameters must be specified by the caller, an InvalidAlgorithmParameterException is raised.

**Parameters:**key - the key for this exemption mechanismparams - the algorithm parameters **Throws:** [InvalidKeyException](http://docs.google.com/java/security/InvalidKeyException.html) - if the given key is inappropriate for this exemption mechanism. [InvalidAlgorithmParameterException](http://docs.google.com/java/security/InvalidAlgorithmParameterException.html) - if the given algorithm parameters are inappropriate for this exemption mechanism. [ExemptionMechanismException](http://docs.google.com/javax/crypto/ExemptionMechanismException.html) - if problem(s) encountered in the process of initializing.

### genExemptionBlob

public final byte[] **genExemptionBlob**()  
 throws [IllegalStateException](http://docs.google.com/java/lang/IllegalStateException.html),  
 [ExemptionMechanismException](http://docs.google.com/javax/crypto/ExemptionMechanismException.html)

Generates the exemption mechanism key blob.

**Returns:**the new buffer with the result key blob. **Throws:** [IllegalStateException](http://docs.google.com/java/lang/IllegalStateException.html) - if this exemption mechanism is in a wrong state (e.g., has not been initialized). [ExemptionMechanismException](http://docs.google.com/javax/crypto/ExemptionMechanismException.html) - if problem(s) encountered in the process of generating.

### genExemptionBlob

public final int **genExemptionBlob**(byte[] output)  
 throws [IllegalStateException](http://docs.google.com/java/lang/IllegalStateException.html),  
 [ShortBufferException](http://docs.google.com/javax/crypto/ShortBufferException.html),  
 [ExemptionMechanismException](http://docs.google.com/javax/crypto/ExemptionMechanismException.html)

Generates the exemption mechanism key blob, and stores the result in the output buffer.

If the output buffer is too small to hold the result, a ShortBufferException is thrown. In this case, repeat this call with a larger output buffer. Use [getOutputSize](http://docs.google.com/javax/crypto/ExemptionMechanism.html#getOutputSize(int)) to determine how big the output buffer should be.

**Parameters:**output - the buffer for the result **Returns:**the number of bytes stored in output **Throws:** [IllegalStateException](http://docs.google.com/java/lang/IllegalStateException.html) - if this exemption mechanism is in a wrong state (e.g., has not been initialized). [ShortBufferException](http://docs.google.com/javax/crypto/ShortBufferException.html) - if the given output buffer is too small to hold the result. [ExemptionMechanismException](http://docs.google.com/javax/crypto/ExemptionMechanismException.html) - if problem(s) encountered in the process of generating.

### genExemptionBlob

public final int **genExemptionBlob**(byte[] output,  
 int outputOffset)  
 throws [IllegalStateException](http://docs.google.com/java/lang/IllegalStateException.html),  
 [ShortBufferException](http://docs.google.com/javax/crypto/ShortBufferException.html),  
 [ExemptionMechanismException](http://docs.google.com/javax/crypto/ExemptionMechanismException.html)

Generates the exemption mechanism key blob, and stores the result in the output buffer, starting at outputOffset inclusive.

If the output buffer is too small to hold the result, a ShortBufferException is thrown. In this case, repeat this call with a larger output buffer. Use [getOutputSize](http://docs.google.com/javax/crypto/ExemptionMechanism.html#getOutputSize(int)) to determine how big the output buffer should be.

**Parameters:**output - the buffer for the resultoutputOffset - the offset in output where the result is stored **Returns:**the number of bytes stored in output **Throws:** [IllegalStateException](http://docs.google.com/java/lang/IllegalStateException.html) - if this exemption mechanism is in a wrong state (e.g., has not been initialized). [ShortBufferException](http://docs.google.com/javax/crypto/ShortBufferException.html) - if the given output buffer is too small to hold the result. [ExemptionMechanismException](http://docs.google.com/javax/crypto/ExemptionMechanismException.html) - if problem(s) encountered in the process of generating.

### finalize

protected void **finalize**()

Ensures that the key stored away by this ExemptionMechanism object will be wiped out when there are no more references to it.

**Overrides:**[finalize](http://docs.google.com/java/lang/Object.html#finalize()) in class [Object](http://docs.google.com/java/lang/Object.html)

| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/ExemptionMechanism.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/javax/crypto/EncryptedPrivateKeyInfo.html)   [**NEXT CLASS**](http://docs.google.com/javax/crypto/ExemptionMechanismException.html) | [**FRAMES**](http://docs.google.com/index.html?javax/crypto/ExemptionMechanism.html)    [**NO FRAMES**](http://docs.google.com/ExemptionMechanism.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).