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

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

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

This class defines the *Service Provider Interface* (**SPI**) for the ExemptionMechanism class. All the abstract methods in this class must be implemented by each cryptographic service provider who wishes to supply the implementation of a particular exemption mechanism.

**Since:** 1.4

| **Constructor Summary** | |
| --- | --- |
| [**ExemptionMechanismSpi**](http://docs.google.com/javax/crypto/ExemptionMechanismSpi.html#ExemptionMechanismSpi())() |

| **Method Summary** | |
| --- | --- |
| protected abstract  byte[] | [**engineGenExemptionBlob**](http://docs.google.com/javax/crypto/ExemptionMechanismSpi.html#engineGenExemptionBlob())()            Generates the exemption mechanism key blob. |
| protected abstract  int | [**engineGenExemptionBlob**](http://docs.google.com/javax/crypto/ExemptionMechanismSpi.html#engineGenExemptionBlob(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. |
| protected abstract  int | [**engineGetOutputSize**](http://docs.google.com/javax/crypto/ExemptionMechanismSpi.html#engineGetOutputSize(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 [engineGenExemptionBlob](http://docs.google.com/javax/crypto/ExemptionMechanismSpi.html#engineGenExemptionBlob(byte%5B%5D,%20int)) operation, given the input length inputLen (in bytes). |
| protected abstract  void | [**engineInit**](http://docs.google.com/javax/crypto/ExemptionMechanismSpi.html#engineInit(java.security.Key))([Key](http://docs.google.com/java/security/Key.html) key)            Initializes this exemption mechanism with a key. |
| protected abstract  void | [**engineInit**](http://docs.google.com/javax/crypto/ExemptionMechanismSpi.html#engineInit(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. |
| protected abstract  void | [**engineInit**](http://docs.google.com/javax/crypto/ExemptionMechanismSpi.html#engineInit(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. |

| **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** |
| --- |

### ExemptionMechanismSpi

public **ExemptionMechanismSpi**()

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

### engineGetOutputSize

protected abstract int **engineGetOutputSize**(int inputLen)

Returns the length in bytes that an output buffer would need to be in order to hold the result of the next [engineGenExemptionBlob](http://docs.google.com/javax/crypto/ExemptionMechanismSpi.html#engineGenExemptionBlob(byte%5B%5D,%20int)) operation, given the input length inputLen (in bytes).

The actual output length of the next [engineGenExemptionBlob](http://docs.google.com/javax/crypto/ExemptionMechanismSpi.html#engineGenExemptionBlob(byte%5B%5D,%20int)) 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)

### engineInit

protected abstract void **engineInit**([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.

### engineInit

protected abstract void **engineInit**([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.

### engineInit

protected abstract void **engineInit**([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.

### engineGenExemptionBlob

protected abstract byte[] **engineGenExemptionBlob**()  
 throws [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:** [ExemptionMechanismException](http://docs.google.com/javax/crypto/ExemptionMechanismException.html) - if problem(s) encountered in the process of generating.

### engineGenExemptionBlob

protected abstract int **engineGenExemptionBlob**(byte[] output,  
 int outputOffset)  
 throws [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 [engineGetOutputSize](http://docs.google.com/javax/crypto/ExemptionMechanismSpi.html#engineGetOutputSize(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:** [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.

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