| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/SignatureSpi.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/SignatureException.html)   [**NEXT CLASS**](http://docs.google.com/java/security/SignedObject.html) | [**FRAMES**](http://docs.google.com/index.html?java/security/SignatureSpi.html)    [**NO FRAMES**](http://docs.google.com/SignatureSpi.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | [FIELD](#3znysh7) | [CONSTR](#2et92p0) | [METHOD](#tyjcwt) | DETAIL: [FIELD](#1t3h5sf) | [CONSTR](#2s8eyo1) | [METHOD](#3rdcrjn) |

## **java.security**

Class SignatureSpi

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

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

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

This class defines the *Service Provider Interface* (**SPI**) for the Signature class, which is used to provide the functionality of a digital signature algorithm. Digital signatures are used for authentication and integrity assurance of digital data. .

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

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

| **Field Summary** | |
| --- | --- |
| protected  [SecureRandom](http://docs.google.com/java/security/SecureRandom.html) | [**appRandom**](http://docs.google.com/java/security/SignatureSpi.html#appRandom)            Application-specified source of randomness. |

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

| **Method Summary** | |
| --- | --- |
| [Object](http://docs.google.com/java/lang/Object.html) | [**clone**](http://docs.google.com/java/security/SignatureSpi.html#clone())()            Returns a clone if the implementation is cloneable. |
| protected abstract  [Object](http://docs.google.com/java/lang/Object.html) | [**engineGetParameter**](http://docs.google.com/java/security/SignatureSpi.html#engineGetParameter(java.lang.String))([String](http://docs.google.com/java/lang/String.html) param)  **Deprecated.** |
| protected  [AlgorithmParameters](http://docs.google.com/java/security/AlgorithmParameters.html) | [**engineGetParameters**](http://docs.google.com/java/security/SignatureSpi.html#engineGetParameters())()            This method is overridden by providers to return the parameters used with this signature engine, or null if this signature engine does not use any parameters. |
| protected abstract  void | [**engineInitSign**](http://docs.google.com/java/security/SignatureSpi.html#engineInitSign(java.security.PrivateKey))([PrivateKey](http://docs.google.com/java/security/PrivateKey.html) privateKey)            Initializes this signature object with the specified private key for signing operations. |
| protected  void | [**engineInitSign**](http://docs.google.com/java/security/SignatureSpi.html#engineInitSign(java.security.PrivateKey,%20java.security.SecureRandom))([PrivateKey](http://docs.google.com/java/security/PrivateKey.html) privateKey, [SecureRandom](http://docs.google.com/java/security/SecureRandom.html) random)            Initializes this signature object with the specified private key and source of randomness for signing operations. |
| protected abstract  void | [**engineInitVerify**](http://docs.google.com/java/security/SignatureSpi.html#engineInitVerify(java.security.PublicKey))([PublicKey](http://docs.google.com/java/security/PublicKey.html) publicKey)            Initializes this signature object with the specified public key for verification operations. |
| protected  void | [**engineSetParameter**](http://docs.google.com/java/security/SignatureSpi.html#engineSetParameter(java.security.spec.AlgorithmParameterSpec))([AlgorithmParameterSpec](http://docs.google.com/java/security/spec/AlgorithmParameterSpec.html) params)            This method is overridden by providers to initialize this signature engine with the specified parameter set. |
| protected abstract  void | [**engineSetParameter**](http://docs.google.com/java/security/SignatureSpi.html#engineSetParameter(java.lang.String,%20java.lang.Object))([String](http://docs.google.com/java/lang/String.html) param, [Object](http://docs.google.com/java/lang/Object.html) value)  **Deprecated.** *Replaced by* [*engineSetParameter*](http://docs.google.com/java/security/SignatureSpi.html#engineSetParameter(java.security.spec.AlgorithmParameterSpec))*.* |
| protected abstract  byte[] | [**engineSign**](http://docs.google.com/java/security/SignatureSpi.html#engineSign())()            Returns the signature bytes of all the data updated so far. |
| protected  int | [**engineSign**](http://docs.google.com/java/security/SignatureSpi.html#engineSign(byte%5B%5D,%20int,%20int))(byte[] outbuf, int offset, int len)            Finishes this signature operation and stores the resulting signature bytes in the provided buffer outbuf, starting at offset. |
| protected abstract  void | [**engineUpdate**](http://docs.google.com/java/security/SignatureSpi.html#engineUpdate(byte))(byte b)            Updates the data to be signed or verified using the specified byte. |
| protected abstract  void | [**engineUpdate**](http://docs.google.com/java/security/SignatureSpi.html#engineUpdate(byte%5B%5D,%20int,%20int))(byte[] b, int off, int len)            Updates the data to be signed or verified, using the specified array of bytes, starting at the specified offset. |
| protected  void | [**engineUpdate**](http://docs.google.com/java/security/SignatureSpi.html#engineUpdate(java.nio.ByteBuffer))([ByteBuffer](http://docs.google.com/java/nio/ByteBuffer.html) input)            Updates the data to be signed or verified using the specified ByteBuffer. |
| protected abstract  boolean | [**engineVerify**](http://docs.google.com/java/security/SignatureSpi.html#engineVerify(byte%5B%5D))(byte[] sigBytes)            Verifies the passed-in signature. |
| protected  boolean | [**engineVerify**](http://docs.google.com/java/security/SignatureSpi.html#engineVerify(byte%5B%5D,%20int,%20int))(byte[] sigBytes, int offset, int length)            Verifies the passed-in signature in the specified array of bytes, starting at the specified offset. |

| **Methods inherited from class java.lang.**[**Object**](http://docs.google.com/java/lang/Object.html) |
| --- |
| [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)) |

| **Field Detail** |
| --- |

### appRandom

protected [SecureRandom](http://docs.google.com/java/security/SecureRandom.html) **appRandom**

Application-specified source of randomness.

| **Constructor Detail** |
| --- |

### SignatureSpi

public **SignatureSpi**()

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

### engineInitVerify

protected abstract void **engineInitVerify**([PublicKey](http://docs.google.com/java/security/PublicKey.html) publicKey)  
 throws [InvalidKeyException](http://docs.google.com/java/security/InvalidKeyException.html)

Initializes this signature object with the specified public key for verification operations.

**Parameters:**publicKey - the public key of the identity whose signature is going to be verified. **Throws:** [InvalidKeyException](http://docs.google.com/java/security/InvalidKeyException.html) - if the key is improperly encoded, parameters are missing, and so on.

### engineInitSign

protected abstract void **engineInitSign**([PrivateKey](http://docs.google.com/java/security/PrivateKey.html) privateKey)  
 throws [InvalidKeyException](http://docs.google.com/java/security/InvalidKeyException.html)

Initializes this signature object with the specified private key for signing operations.

**Parameters:**privateKey - the private key of the identity whose signature will be generated. **Throws:** [InvalidKeyException](http://docs.google.com/java/security/InvalidKeyException.html) - if the key is improperly encoded, parameters are missing, and so on.

### engineInitSign

protected void **engineInitSign**([PrivateKey](http://docs.google.com/java/security/PrivateKey.html) privateKey,  
 [SecureRandom](http://docs.google.com/java/security/SecureRandom.html) random)  
 throws [InvalidKeyException](http://docs.google.com/java/security/InvalidKeyException.html)

Initializes this signature object with the specified private key and source of randomness for signing operations.

This concrete method has been added to this previously-defined abstract class. (For backwards compatibility, it cannot be abstract.)

**Parameters:**privateKey - the private key of the identity whose signature will be generated.random - the source of randomness **Throws:** [InvalidKeyException](http://docs.google.com/java/security/InvalidKeyException.html) - if the key is improperly encoded, parameters are missing, and so on.

### engineUpdate

protected abstract void **engineUpdate**(byte b)  
 throws [SignatureException](http://docs.google.com/java/security/SignatureException.html)

Updates the data to be signed or verified using the specified byte.

**Parameters:**b - the byte to use for the update. **Throws:** [SignatureException](http://docs.google.com/java/security/SignatureException.html) - if the engine is not initialized properly.

### engineUpdate

protected abstract void **engineUpdate**(byte[] b,  
 int off,  
 int len)  
 throws [SignatureException](http://docs.google.com/java/security/SignatureException.html)

Updates the data to be signed or verified, using the specified array of bytes, starting at the specified offset.

**Parameters:**b - the array of bytesoff - the offset to start from in the array of byteslen - the number of bytes to use, starting at offset **Throws:** [SignatureException](http://docs.google.com/java/security/SignatureException.html) - if the engine is not initialized properly

### engineUpdate

protected void **engineUpdate**([ByteBuffer](http://docs.google.com/java/nio/ByteBuffer.html) input)

Updates the data to be signed or verified using the specified ByteBuffer. Processes the data.remaining() bytes starting at at data.position(). Upon return, the buffer's position will be equal to its limit; its limit will not have changed.

**Parameters:**input - the ByteBuffer**Since:** 1.5

### engineSign

protected abstract byte[] **engineSign**()  
 throws [SignatureException](http://docs.google.com/java/security/SignatureException.html)

Returns the signature bytes of all the data updated so far. The format of the signature depends on the underlying signature scheme.

**Returns:**the signature bytes of the signing operation's result. **Throws:** [SignatureException](http://docs.google.com/java/security/SignatureException.html) - if the engine is not initialized properly or if this signature algorithm is unable to process the input data provided.

### engineSign

protected int **engineSign**(byte[] outbuf,  
 int offset,  
 int len)  
 throws [SignatureException](http://docs.google.com/java/security/SignatureException.html)

Finishes this signature operation and stores the resulting signature bytes in the provided buffer outbuf, starting at offset. The format of the signature depends on the underlying signature scheme.

The signature implementation is reset to its initial state (the state it was in after a call to one of the engineInitSign methods) and can be reused to generate further signatures with the same private key. This method should be abstract, but we leave it concrete for binary compatibility. Knowledgeable providers should override this method.

**Parameters:**outbuf - buffer for the signature result.offset - offset into outbuf where the signature is stored.len - number of bytes within outbuf allotted for the signature. Both this default implementation and the SUN provider do not return partial digests. If the value of this parameter is less than the actual signature length, this method will throw a SignatureException. This parameter is ignored if its value is greater than or equal to the actual signature length. **Returns:**the number of bytes placed into outbuf **Throws:** [SignatureException](http://docs.google.com/java/security/SignatureException.html) - if the engine is not initialized properly, if this signature algorithm is unable to process the input data provided, or if len is less than the actual signature length.**Since:** 1.2

### engineVerify

protected abstract boolean **engineVerify**(byte[] sigBytes)  
 throws [SignatureException](http://docs.google.com/java/security/SignatureException.html)

Verifies the passed-in signature.

**Parameters:**sigBytes - the signature bytes to be verified. **Returns:**true if the signature was verified, false if not. **Throws:** [SignatureException](http://docs.google.com/java/security/SignatureException.html) - if the engine is not initialized properly, the passed-in signature is improperly encoded or of the wrong type, if this signature algorithm is unable to process the input data provided, etc.

### engineVerify

protected boolean **engineVerify**(byte[] sigBytes,  
 int offset,  
 int length)  
 throws [SignatureException](http://docs.google.com/java/security/SignatureException.html)

Verifies the passed-in signature in the specified array of bytes, starting at the specified offset.

Note: Subclasses should overwrite the default implementation.

**Parameters:**sigBytes - the signature bytes to be verified.offset - the offset to start from in the array of bytes.length - the number of bytes to use, starting at offset. **Returns:**true if the signature was verified, false if not. **Throws:** [SignatureException](http://docs.google.com/java/security/SignatureException.html) - if the engine is not initialized properly, the passed-in signature is improperly encoded or of the wrong type, if this signature algorithm is unable to process the input data provided, etc.**Since:** 1.4

### engineSetParameter

[@Deprecated](http://docs.google.com/java/lang/Deprecated.html)  
protected abstract void **engineSetParameter**([String](http://docs.google.com/java/lang/String.html) param,  
 [Object](http://docs.google.com/java/lang/Object.html) value)  
 throws [InvalidParameterException](http://docs.google.com/java/security/InvalidParameterException.html)

**Deprecated.** *Replaced by* [*engineSetParameter*](http://docs.google.com/java/security/SignatureSpi.html#engineSetParameter(java.security.spec.AlgorithmParameterSpec))*.*

Sets the specified algorithm parameter to the specified value. This method supplies a general-purpose mechanism through which it is possible to set the various parameters of this object. A parameter may be any settable parameter for the algorithm, such as a parameter size, or a source of random bits for signature generation (if appropriate), or an indication of whether or not to perform a specific but optional computation. A uniform algorithm-specific naming scheme for each parameter is desirable but left unspecified at this time.

**Parameters:**param - the string identifier of the parameter.value - the parameter value. **Throws:** [InvalidParameterException](http://docs.google.com/java/security/InvalidParameterException.html) - if param is an invalid parameter for this signature algorithm engine, the parameter is already set and cannot be set again, a security exception occurs, and so on.

### engineSetParameter

protected void **engineSetParameter**([AlgorithmParameterSpec](http://docs.google.com/java/security/spec/AlgorithmParameterSpec.html) params)  
 throws [InvalidAlgorithmParameterException](http://docs.google.com/java/security/InvalidAlgorithmParameterException.html)

This method is overridden by providers to initialize this signature engine with the specified parameter set.

**Parameters:**params - the parameters **Throws:** [UnsupportedOperationException](http://docs.google.com/java/lang/UnsupportedOperationException.html) - if this method is not overridden by a provider [InvalidAlgorithmParameterException](http://docs.google.com/java/security/InvalidAlgorithmParameterException.html) - if this method is overridden by a provider and the the given parameters are inappropriate for this signature engine

### engineGetParameters

protected [AlgorithmParameters](http://docs.google.com/java/security/AlgorithmParameters.html) **engineGetParameters**()

This method is overridden by providers to return the parameters used with this signature engine, or null if this signature engine does not use any parameters.

The returned parameters may be the same that were used to initialize this signature engine, or may contain a combination of default and randomly generated parameter values used by the underlying signature implementation if this signature engine requires algorithm parameters but was not initialized with any.

**Returns:**the parameters used with this signature engine, or null if this signature engine does not use any parameters **Throws:** [UnsupportedOperationException](http://docs.google.com/java/lang/UnsupportedOperationException.html) - if this method is not overridden by a provider**Since:** 1.4

### engineGetParameter

[@Deprecated](http://docs.google.com/java/lang/Deprecated.html)  
protected abstract [Object](http://docs.google.com/java/lang/Object.html) **engineGetParameter**([String](http://docs.google.com/java/lang/String.html) param)  
 throws [InvalidParameterException](http://docs.google.com/java/security/InvalidParameterException.html)

**Deprecated.**

Gets the value of the specified algorithm parameter. This method supplies a general-purpose mechanism through which it is possible to get the various parameters of this object. A parameter may be any settable parameter for the algorithm, such as a parameter size, or a source of random bits for signature generation (if appropriate), or an indication of whether or not to perform a specific but optional computation. A uniform algorithm-specific naming scheme for each parameter is desirable but left unspecified at this time.

**Parameters:**param - the string name of the parameter. **Returns:**the object that represents the parameter value, or null if there is none. **Throws:** [InvalidParameterException](http://docs.google.com/java/security/InvalidParameterException.html) - if param is an invalid parameter for this engine, or another exception occurs while trying to get this parameter.

### clone

public [Object](http://docs.google.com/java/lang/Object.html) **clone**()  
 throws [CloneNotSupportedException](http://docs.google.com/java/lang/CloneNotSupportedException.html)

Returns a clone if the implementation is cloneable.

**Overrides:**[clone](http://docs.google.com/java/lang/Object.html#clone()) in class [Object](http://docs.google.com/java/lang/Object.html) **Returns:**a clone if the implementation is cloneable. **Throws:** [CloneNotSupportedException](http://docs.google.com/java/lang/CloneNotSupportedException.html) - if this is called on an implementation that does not support Cloneable.**See Also:**[Cloneable](http://docs.google.com/java/lang/Cloneable.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/SignatureSpi.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/SignatureException.html)   [**NEXT CLASS**](http://docs.google.com/java/security/SignedObject.html) | [**FRAMES**](http://docs.google.com/index.html?java/security/SignatureSpi.html)    [**NO FRAMES**](http://docs.google.com/SignatureSpi.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | [FIELD](#3znysh7) | [CONSTR](#2et92p0) | [METHOD](#tyjcwt) | DETAIL: [FIELD](#1t3h5sf) | [CONSTR](#2s8eyo1) | [METHOD](#3rdcrjn) |

[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).