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

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

**All Implemented Interfaces:** [Cloneable](http://docs.google.com/java/lang/Cloneable.html)

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

This class provides the functionality of a "Message Authentication Code" (MAC) algorithm.

A MAC provides a way to check the integrity of information transmitted over or stored in an unreliable medium, based on a secret key. Typically, message authentication codes are used between two parties that share a secret key in order to validate information transmitted between these parties.

A MAC mechanism that is based on cryptographic hash functions is referred to as HMAC. HMAC can be used with any cryptographic hash function, e.g., MD5 or SHA-1, in combination with a secret shared key. HMAC is specified in RFC 2104.

**Since:** 1.4

| **Constructor Summary** | |
| --- | --- |
| protected | [**Mac**](http://docs.google.com/javax/crypto/Mac.html#Mac(javax.crypto.MacSpi,%20java.security.Provider,%20java.lang.String))([MacSpi](http://docs.google.com/javax/crypto/MacSpi.html) macSpi, [Provider](http://docs.google.com/java/security/Provider.html) provider, [String](http://docs.google.com/java/lang/String.html) algorithm)            Creates a MAC object. |

| **Method Summary** | |
| --- | --- |
| [Object](http://docs.google.com/java/lang/Object.html) | [**clone**](http://docs.google.com/javax/crypto/Mac.html#clone())()            Returns a clone if the provider implementation is cloneable. |
| byte[] | [**doFinal**](http://docs.google.com/javax/crypto/Mac.html#doFinal())()            Finishes the MAC operation. |
| byte[] | [**doFinal**](http://docs.google.com/javax/crypto/Mac.html#doFinal(byte%5B%5D))(byte[] input)            Processes the given array of bytes and finishes the MAC operation. |
| void | [**doFinal**](http://docs.google.com/javax/crypto/Mac.html#doFinal(byte%5B%5D,%20int))(byte[] output, int outOffset)            Finishes the MAC operation. |
| [String](http://docs.google.com/java/lang/String.html) | [**getAlgorithm**](http://docs.google.com/javax/crypto/Mac.html#getAlgorithm())()            Returns the algorithm name of this Mac object. |
| static [Mac](http://docs.google.com/javax/crypto/Mac.html) | [**getInstance**](http://docs.google.com/javax/crypto/Mac.html#getInstance(java.lang.String))([String](http://docs.google.com/java/lang/String.html) algorithm)            Returns a Mac object that implements the specified MAC algorithm. |
| static [Mac](http://docs.google.com/javax/crypto/Mac.html) | [**getInstance**](http://docs.google.com/javax/crypto/Mac.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 a Mac object that implements the specified MAC algorithm. |
| static [Mac](http://docs.google.com/javax/crypto/Mac.html) | [**getInstance**](http://docs.google.com/javax/crypto/Mac.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 a Mac object that implements the specified MAC algorithm. |
| int | [**getMacLength**](http://docs.google.com/javax/crypto/Mac.html#getMacLength())()            Returns the length of the MAC in bytes. |
| [Provider](http://docs.google.com/java/security/Provider.html) | [**getProvider**](http://docs.google.com/javax/crypto/Mac.html#getProvider())()            Returns the provider of this Mac object. |
| void | [**init**](http://docs.google.com/javax/crypto/Mac.html#init(java.security.Key))([Key](http://docs.google.com/java/security/Key.html) key)            Initializes this Mac object with the given key. |
| void | [**init**](http://docs.google.com/javax/crypto/Mac.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 Mac object with the given key and algorithm parameters. |
| void | [**reset**](http://docs.google.com/javax/crypto/Mac.html#reset())()            Resets this Mac object. |
| void | [**update**](http://docs.google.com/javax/crypto/Mac.html#update(byte))(byte input)            Processes the given byte. |
| void | [**update**](http://docs.google.com/javax/crypto/Mac.html#update(byte%5B%5D))(byte[] input)            Processes the given array of bytes. |
| void | [**update**](http://docs.google.com/javax/crypto/Mac.html#update(byte%5B%5D,%20int,%20int))(byte[] input, int offset, int len)            Processes the first len bytes in input, starting at offset inclusive. |
| void | [**update**](http://docs.google.com/javax/crypto/Mac.html#update(java.nio.ByteBuffer))([ByteBuffer](http://docs.google.com/java/nio/ByteBuffer.html) input)            Processes input.remaining() bytes in the ByteBuffer input, starting at input.position(). |

| **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)) |

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

### Mac

protected **Mac**([MacSpi](http://docs.google.com/javax/crypto/MacSpi.html) macSpi,  
 [Provider](http://docs.google.com/java/security/Provider.html) provider,  
 [String](http://docs.google.com/java/lang/String.html) algorithm)

Creates a MAC object.

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

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

### getAlgorithm

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

Returns the algorithm name of this Mac object.

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

**Returns:**the algorithm name of this Mac object.

### getInstance

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

Returns a Mac object that implements the specified MAC algorithm.

This method traverses the list of registered security Providers, starting with the most preferred Provider. A new Mac object encapsulating the MacSpi 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 MAC algorithm. 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 algorithm names. **Returns:**the new Mac object. **Throws:** [NoSuchAlgorithmException](http://docs.google.com/java/security/NoSuchAlgorithmException.html) - if no Provider supports a MacSpi implementation for the specified algorithm.**See Also:**[Provider](http://docs.google.com/java/security/Provider.html)

### getInstance

public static final [Mac](http://docs.google.com/javax/crypto/Mac.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 a Mac object that implements the specified MAC algorithm.

A new Mac object encapsulating the MacSpi 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 MAC algorithm. 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 algorithm names.provider - the name of the provider. **Returns:**the new Mac object. **Throws:** [NoSuchAlgorithmException](http://docs.google.com/java/security/NoSuchAlgorithmException.html) - if a MacSpi 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 [Mac](http://docs.google.com/javax/crypto/Mac.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 a Mac object that implements the specified MAC algorithm.

A new Mac object encapsulating the MacSpi 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 MAC algorithm. 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 algorithm names.provider - the provider. **Returns:**the new Mac object. **Throws:** [NoSuchAlgorithmException](http://docs.google.com/java/security/NoSuchAlgorithmException.html) - if a MacSpi 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 Mac object.

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

### getMacLength

public final int **getMacLength**()

Returns the length of the MAC in bytes.

**Returns:**the MAC length in bytes.

### 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)

Initializes this Mac object with the given key.

**Parameters:**key - the key. **Throws:** [InvalidKeyException](http://docs.google.com/java/security/InvalidKeyException.html) - if the given key is inappropriate for initializing this MAC.

### 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)

Initializes this Mac object with the given key and algorithm parameters.

**Parameters:**key - the key.params - the algorithm parameters. **Throws:** [InvalidKeyException](http://docs.google.com/java/security/InvalidKeyException.html) - if the given key is inappropriate for initializing this MAC. [InvalidAlgorithmParameterException](http://docs.google.com/java/security/InvalidAlgorithmParameterException.html) - if the given algorithm parameters are inappropriate for this MAC.

### update

public final void **update**(byte input)  
 throws [IllegalStateException](http://docs.google.com/java/lang/IllegalStateException.html)

Processes the given byte.

**Parameters:**input - the input byte to be processed. **Throws:** [IllegalStateException](http://docs.google.com/java/lang/IllegalStateException.html) - if this Mac has not been initialized.

### update

public final void **update**(byte[] input)  
 throws [IllegalStateException](http://docs.google.com/java/lang/IllegalStateException.html)

Processes the given array of bytes.

**Parameters:**input - the array of bytes to be processed. **Throws:** [IllegalStateException](http://docs.google.com/java/lang/IllegalStateException.html) - if this Mac has not been initialized.

### update

public final void **update**(byte[] input,  
 int offset,  
 int len)  
 throws [IllegalStateException](http://docs.google.com/java/lang/IllegalStateException.html)

Processes the first len bytes in input, starting at offset inclusive.

**Parameters:**input - the input buffer.offset - the offset in input where the input starts.len - the number of bytes to process. **Throws:** [IllegalStateException](http://docs.google.com/java/lang/IllegalStateException.html) - if this Mac has not been initialized.

### update

public final void **update**([ByteBuffer](http://docs.google.com/java/nio/ByteBuffer.html) input)

Processes input.remaining() bytes in the ByteBuffer input, starting at input.position(). Upon return, the buffer's position will be equal to its limit; its limit will not have changed.

**Parameters:**input - the ByteBuffer **Throws:** [IllegalStateException](http://docs.google.com/java/lang/IllegalStateException.html) - if this Mac has not been initialized.**Since:** 1.5

### doFinal

public final byte[] **doFinal**()  
 throws [IllegalStateException](http://docs.google.com/java/lang/IllegalStateException.html)

Finishes the MAC operation.

A call to this method resets this Mac object to the state it was in when previously initialized via a call to init(Key) or init(Key, AlgorithmParameterSpec). That is, the object is reset and available to generate another MAC from the same key, if desired, via new calls to update and doFinal. (In order to reuse this Mac object with a different key, it must be reinitialized via a call to init(Key) or init(Key, AlgorithmParameterSpec).

**Returns:**the MAC result. **Throws:** [IllegalStateException](http://docs.google.com/java/lang/IllegalStateException.html) - if this Mac has not been initialized.

### doFinal

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

Finishes the MAC operation.

A call to this method resets this Mac object to the state it was in when previously initialized via a call to init(Key) or init(Key, AlgorithmParameterSpec). That is, the object is reset and available to generate another MAC from the same key, if desired, via new calls to update and doFinal. (In order to reuse this Mac object with a different key, it must be reinitialized via a call to init(Key) or init(Key, AlgorithmParameterSpec).

The MAC result is stored in output, starting at outOffset inclusive.

**Parameters:**output - the buffer where the MAC result is storedoutOffset - the offset in output where the MAC is stored **Throws:** [ShortBufferException](http://docs.google.com/javax/crypto/ShortBufferException.html) - if the given output buffer is too small to hold the result [IllegalStateException](http://docs.google.com/java/lang/IllegalStateException.html) - if this Mac has not been initialized.

### doFinal

public final byte[] **doFinal**(byte[] input)  
 throws [IllegalStateException](http://docs.google.com/java/lang/IllegalStateException.html)

Processes the given array of bytes and finishes the MAC operation.

A call to this method resets this Mac object to the state it was in when previously initialized via a call to init(Key) or init(Key, AlgorithmParameterSpec). That is, the object is reset and available to generate another MAC from the same key, if desired, via new calls to update and doFinal. (In order to reuse this Mac object with a different key, it must be reinitialized via a call to init(Key) or init(Key, AlgorithmParameterSpec).

**Parameters:**input - data in bytes **Returns:**the MAC result. **Throws:** [IllegalStateException](http://docs.google.com/java/lang/IllegalStateException.html) - if this Mac has not been initialized.

### reset

public final void **reset**()

Resets this Mac object.

A call to this method resets this Mac object to the state it was in when previously initialized via a call to init(Key) or init(Key, AlgorithmParameterSpec). That is, the object is reset and available to generate another MAC from the same key, if desired, via new calls to update and doFinal. (In order to reuse this Mac object with a different key, it must be reinitialized via a call to init(Key) or init(Key, AlgorithmParameterSpec).

### clone

public final [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 provider 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 provider implementation is cloneable. **Throws:** [CloneNotSupportedException](http://docs.google.com/java/lang/CloneNotSupportedException.html) - if this is called on a delegate 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/Mac.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/KeyGeneratorSpi.html)   [**NEXT CLASS**](http://docs.google.com/javax/crypto/MacSpi.html) | [**FRAMES**](http://docs.google.com/index.html?javax/crypto/Mac.html)    [**NO FRAMES**](http://docs.google.com/Mac.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).