| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/PBEKeySpec.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/spec/OAEPParameterSpec.html)   [**NEXT CLASS**](http://docs.google.com/javax/crypto/spec/PBEParameterSpec.html) | [**FRAMES**](http://docs.google.com/index.html?javax/crypto/spec/PBEKeySpec.html)    [**NO FRAMES**](http://docs.google.com/PBEKeySpec.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | FIELD | [CONSTR](#3znysh7) | [METHOD](#2et92p0) | DETAIL: FIELD | [CONSTR](#3dy6vkm) | [METHOD](#17dp8vu) |

## **javax.crypto.spec**

Class PBEKeySpec

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

**All Implemented Interfaces:** [KeySpec](http://docs.google.com/java/security/spec/KeySpec.html)

public class **PBEKeySpec**extends [Object](http://docs.google.com/java/lang/Object.html)implements [KeySpec](http://docs.google.com/java/security/spec/KeySpec.html)

A user-chosen password that can be used with password-based encryption (*PBE*).

The password can be viewed as some kind of raw key material, from which the encryption mechanism that uses it derives a cryptographic key.

Different PBE mechanisms may consume different bits of each password character. For example, the PBE mechanism defined in  [PKCS #5](http://www.ietf.org/rfc/rfc2898.txt) looks at only the low order 8 bits of each character, whereas PKCS #12 looks at all 16 bits of each character.

You convert the password characters to a PBE key by creating an instance of the appropriate secret-key factory. For example, a secret-key factory for PKCS #5 will construct a PBE key from only the low order 8 bits of each password character, whereas a secret-key factory for PKCS #12 will take all 16 bits of each character.

Also note that this class stores passwords as char arrays instead of String objects (which would seem more logical), because the String class is immutable and there is no way to overwrite its internal value when the password stored in it is no longer needed. Hence, this class requests the password as a char array, so it can be overwritten when done.

**Since:** 1.4 **See Also:**[SecretKeyFactory](http://docs.google.com/javax/crypto/SecretKeyFactory.html), [PBEParameterSpec](http://docs.google.com/javax/crypto/spec/PBEParameterSpec.html)

| **Constructor Summary** | |
| --- | --- |
| [**PBEKeySpec**](http://docs.google.com/javax/crypto/spec/PBEKeySpec.html#PBEKeySpec(char%5B%5D))(char[] password)            Constructor that takes a password. |
| [**PBEKeySpec**](http://docs.google.com/javax/crypto/spec/PBEKeySpec.html#PBEKeySpec(char%5B%5D,%20byte%5B%5D,%20int))(char[] password, byte[] salt, int iterationCount)            Constructor that takes a password, salt, iteration count for generating PBEKey of fixed-key-size PBE ciphers. |
| [**PBEKeySpec**](http://docs.google.com/javax/crypto/spec/PBEKeySpec.html#PBEKeySpec(char%5B%5D,%20byte%5B%5D,%20int,%20int))(char[] password, byte[] salt, int iterationCount, int keyLength)            Constructor that takes a password, salt, iteration count, and to-be-derived key length for generating PBEKey of variable-key-size PBE ciphers. |

| **Method Summary** | |
| --- | --- |
| void | [**clearPassword**](http://docs.google.com/javax/crypto/spec/PBEKeySpec.html#clearPassword())()            Clears the internal copy of the password. |
| int | [**getIterationCount**](http://docs.google.com/javax/crypto/spec/PBEKeySpec.html#getIterationCount())()            Returns the iteration count or 0 if not specified. |
| int | [**getKeyLength**](http://docs.google.com/javax/crypto/spec/PBEKeySpec.html#getKeyLength())()            Returns the to-be-derived key length or 0 if not specified. |
| char[] | [**getPassword**](http://docs.google.com/javax/crypto/spec/PBEKeySpec.html#getPassword())()            Returns a copy of the password. |
| byte[] | [**getSalt**](http://docs.google.com/javax/crypto/spec/PBEKeySpec.html#getSalt())()            Returns a copy of the salt or null if not specified. |

| **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** |
| --- |

### PBEKeySpec

public **PBEKeySpec**(char[] password)

Constructor that takes a password. An empty char[] is used if null is specified.

Note: password is cloned before it is stored in the new PBEKeySpec object.

**Parameters:**password - the password.

### PBEKeySpec

public **PBEKeySpec**(char[] password,  
 byte[] salt,  
 int iterationCount,  
 int keyLength)

Constructor that takes a password, salt, iteration count, and to-be-derived key length for generating PBEKey of variable-key-size PBE ciphers. An empty char[] is used if null is specified for password.

Note: the password and salt are cloned before they are stored in the new PBEKeySpec object.

**Parameters:**password - the password.salt - the salt.iterationCount - the iteration count.keyLength - the to-be-derived key length. **Throws:** [NullPointerException](http://docs.google.com/java/lang/NullPointerException.html) - if salt is null. [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if salt is empty, i.e. 0-length, iterationCount or keyLength is not positive.

### PBEKeySpec

public **PBEKeySpec**(char[] password,  
 byte[] salt,  
 int iterationCount)

Constructor that takes a password, salt, iteration count for generating PBEKey of fixed-key-size PBE ciphers. An empty char[] is used if null is specified for password.

Note: the password and salt are cloned before they are stored in the new PBEKeySpec object.

**Parameters:**password - the password.salt - the salt.iterationCount - the iteration count. **Throws:** [NullPointerException](http://docs.google.com/java/lang/NullPointerException.html) - if salt is null. [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if salt is empty, i.e. 0-length, or iterationCount is not positive.

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

### clearPassword

public final void **clearPassword**()

Clears the internal copy of the password.

### getPassword

public final char[] **getPassword**()

Returns a copy of the password.

Note: this method returns a copy of the password. It is the caller's responsibility to zero out the password information after it is no longer needed.

**Returns:**the password. **Throws:** [IllegalStateException](http://docs.google.com/java/lang/IllegalStateException.html) - if password has been cleared by calling clearPassword method.

### getSalt

public final byte[] **getSalt**()

Returns a copy of the salt or null if not specified.

Note: this method should return a copy of the salt. It is the caller's responsibility to zero out the salt information after it is no longer needed.

**Returns:**the salt.

### getIterationCount

public final int **getIterationCount**()

Returns the iteration count or 0 if not specified.

**Returns:**the iteration count.

### getKeyLength

public final int **getKeyLength**()

Returns the to-be-derived key length or 0 if not specified.

Note: this is used to indicate the preference on key length for variable-key-size ciphers. The actual key size depends on each provider's implementation.

**Returns:**the to-be-derived key length.

| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/PBEKeySpec.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/spec/OAEPParameterSpec.html)   [**NEXT CLASS**](http://docs.google.com/javax/crypto/spec/PBEParameterSpec.html) | [**FRAMES**](http://docs.google.com/index.html?javax/crypto/spec/PBEKeySpec.html)    [**NO FRAMES**](http://docs.google.com/PBEKeySpec.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | FIELD | [CONSTR](#3znysh7) | [METHOD](#2et92p0) | DETAIL: FIELD | [CONSTR](#3dy6vkm) | [METHOD](#17dp8vu) |

[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).