| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/RC5ParameterSpec.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/RC2ParameterSpec.html)   [**NEXT CLASS**](http://docs.google.com/javax/crypto/spec/SecretKeySpec.html) | [**FRAMES**](http://docs.google.com/index.html?javax/crypto/spec/RC5ParameterSpec.html)    [**NO FRAMES**](http://docs.google.com/RC5ParameterSpec.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 RC5ParameterSpec

[java.lang.Object](http://docs.google.com/java/lang/Object.html)  
 **javax.crypto.spec.RC5ParameterSpec**

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

public class **RC5ParameterSpec**extends [Object](http://docs.google.com/java/lang/Object.html)implements [AlgorithmParameterSpec](http://docs.google.com/java/security/spec/AlgorithmParameterSpec.html)

This class specifies the parameters used with the [*RC5*](http://www.ietf.org/rfc/rfc2040.txt) algorithm.

The parameters consist of a version number, a rounds count, a word size, and optionally an initialization vector (IV) (only in feedback mode).

This class can be used to initialize a Cipher object that implements the *RC5* algorithm as supplied by [RSA Security Inc.](http://www.rsasecurity.com), or any parties authorized by RSA Security.

**Since:** 1.4

| **Constructor Summary** | |
| --- | --- |
| [**RC5ParameterSpec**](http://docs.google.com/javax/crypto/spec/RC5ParameterSpec.html#RC5ParameterSpec(int,%20int,%20int))(int version, int rounds, int wordSize)            Constructs a parameter set for RC5 from the given version, number of rounds and word size (in bits). |
| [**RC5ParameterSpec**](http://docs.google.com/javax/crypto/spec/RC5ParameterSpec.html#RC5ParameterSpec(int,%20int,%20int,%20byte%5B%5D))(int version, int rounds, int wordSize, byte[] iv)            Constructs a parameter set for RC5 from the given version, number of rounds, word size (in bits), and IV. |
| [**RC5ParameterSpec**](http://docs.google.com/javax/crypto/spec/RC5ParameterSpec.html#RC5ParameterSpec(int,%20int,%20int,%20byte%5B%5D,%20int))(int version, int rounds, int wordSize, byte[] iv, int offset)            Constructs a parameter set for RC5 from the given version, number of rounds, word size (in bits), and IV. |

| **Method Summary** | |
| --- | --- |
| boolean | [**equals**](http://docs.google.com/javax/crypto/spec/RC5ParameterSpec.html#equals(java.lang.Object))([Object](http://docs.google.com/java/lang/Object.html) obj)            Tests for equality between the specified object and this object. |
| byte[] | [**getIV**](http://docs.google.com/javax/crypto/spec/RC5ParameterSpec.html#getIV())()            Returns the IV or null if this parameter set does not contain an IV. |
| int | [**getRounds**](http://docs.google.com/javax/crypto/spec/RC5ParameterSpec.html#getRounds())()            Returns the number of rounds. |
| int | [**getVersion**](http://docs.google.com/javax/crypto/spec/RC5ParameterSpec.html#getVersion())()            Returns the version. |
| int | [**getWordSize**](http://docs.google.com/javax/crypto/spec/RC5ParameterSpec.html#getWordSize())()            Returns the word size in bits. |
| int | [**hashCode**](http://docs.google.com/javax/crypto/spec/RC5ParameterSpec.html#hashCode())()            Calculates a hash code value for the object. |

| **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()), [finalize](http://docs.google.com/java/lang/Object.html#finalize()), [getClass](http://docs.google.com/java/lang/Object.html#getClass()), [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** |
| --- |

### RC5ParameterSpec

public **RC5ParameterSpec**(int version,  
 int rounds,  
 int wordSize)

Constructs a parameter set for RC5 from the given version, number of rounds and word size (in bits).

**Parameters:**version - the version.rounds - the number of rounds.wordSize - the word size in bits.

### RC5ParameterSpec

public **RC5ParameterSpec**(int version,  
 int rounds,  
 int wordSize,  
 byte[] iv)

Constructs a parameter set for RC5 from the given version, number of rounds, word size (in bits), and IV.

Note that the size of the IV (block size) must be twice the word size. The bytes that constitute the IV are those between iv[0] and iv[2\*(wordSize/8)-1] inclusive.

**Parameters:**version - the version.rounds - the number of rounds.wordSize - the word size in bits.iv - the buffer with the IV. The first 2\*(wordSize/8) bytes of the buffer are copied to protect against subsequent modification. **Throws:** [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if iv is null or (iv.length < 2 \* (wordSize / 8))

### RC5ParameterSpec

public **RC5ParameterSpec**(int version,  
 int rounds,  
 int wordSize,  
 byte[] iv,  
 int offset)

Constructs a parameter set for RC5 from the given version, number of rounds, word size (in bits), and IV.

The IV is taken from iv, starting at offset inclusive. Note that the size of the IV (block size), starting at offset inclusive, must be twice the word size. The bytes that constitute the IV are those between iv[offset] and iv[offset+2\*(wordSize/8)-1] inclusive.

**Parameters:**version - the version.rounds - the number of rounds.wordSize - the word size in bits.iv - the buffer with the IV. The first 2\*(wordSize/8) bytes of the buffer beginning at offset inclusive are copied to protect against subsequent modification.offset - the offset in iv where the IV starts. **Throws:** [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if iv is null or (iv.length - offset < 2 \* (wordSize / 8))

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

### getVersion

public int **getVersion**()

Returns the version.

**Returns:**the version.

### getRounds

public int **getRounds**()

Returns the number of rounds.

**Returns:**the number of rounds.

### getWordSize

public int **getWordSize**()

Returns the word size in bits.

**Returns:**the word size in bits.

### getIV

public byte[] **getIV**()

Returns the IV or null if this parameter set does not contain an IV.

**Returns:**the IV or null if this parameter set does not contain an IV. Returns a new array each time this method is called.

### equals

public boolean **equals**([Object](http://docs.google.com/java/lang/Object.html) obj)

Tests for equality between the specified object and this object. Two RC5ParameterSpec objects are considered equal if their version numbers, number of rounds, word sizes, and IVs are equal. (Two IV references are considered equal if both are null.)

**Overrides:**[equals](http://docs.google.com/java/lang/Object.html#equals(java.lang.Object)) in class [Object](http://docs.google.com/java/lang/Object.html) **Parameters:**obj - the object to test for equality with this object. **Returns:**true if the objects are considered equal, false if obj is null or otherwise.**See Also:**[Object.hashCode()](http://docs.google.com/java/lang/Object.html#hashCode()), [Hashtable](http://docs.google.com/java/util/Hashtable.html)

### hashCode

public int **hashCode**()

Calculates a hash code value for the object. Objects that are equal will also have the same hashcode.

**Overrides:**[hashCode](http://docs.google.com/java/lang/Object.html#hashCode()) in class [Object](http://docs.google.com/java/lang/Object.html) **Returns:**a hash code value for this object.**See Also:**[Object.equals(java.lang.Object)](http://docs.google.com/java/lang/Object.html#equals(java.lang.Object)), [Hashtable](http://docs.google.com/java/util/Hashtable.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/RC5ParameterSpec.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/RC2ParameterSpec.html)   [**NEXT CLASS**](http://docs.google.com/javax/crypto/spec/SecretKeySpec.html) | [**FRAMES**](http://docs.google.com/index.html?javax/crypto/spec/RC5ParameterSpec.html)    [**NO FRAMES**](http://docs.google.com/RC5ParameterSpec.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).