| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/UUID.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/util/UnknownFormatFlagsException.html)   [**NEXT CLASS**](http://docs.google.com/java/util/Vector.html) | [**FRAMES**](http://docs.google.com/index.html?java/util/UUID.html)    [**NO FRAMES**](http://docs.google.com/UUID.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | FIELD | [CONSTR](#3znysh7) | [METHOD](#2et92p0) | DETAIL: FIELD | [CONSTR](#3dy6vkm) | [METHOD](#4d34og8) |

## **java.util**

Class UUID

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

**All Implemented Interfaces:** [Serializable](http://docs.google.com/java/io/Serializable.html), [Comparable](http://docs.google.com/java/lang/Comparable.html)<[UUID](http://docs.google.com/java/util/UUID.html)>

public final class **UUID**extends [Object](http://docs.google.com/java/lang/Object.html)implements [Serializable](http://docs.google.com/java/io/Serializable.html), [Comparable](http://docs.google.com/java/lang/Comparable.html)<[UUID](http://docs.google.com/java/util/UUID.html)>

A class that represents an immutable universally unique identifier (UUID). A UUID represents a 128-bit value.

There exist different variants of these global identifiers. The methods of this class are for manipulating the Leach-Salz variant, although the constructors allow the creation of any variant of UUID (described below).

The layout of a variant 2 (Leach-Salz) UUID is as follows: The most significant long consists of the following unsigned fields:

0xFFFFFFFF00000000 time\_low  
 0x00000000FFFF0000 time\_mid  
 0x000000000000F000 version  
 0x0000000000000FFF time\_hi

The least significant long consists of the following unsigned fields:

0xC000000000000000 variant  
 0x3FFF000000000000 clock\_seq  
 0x0000FFFFFFFFFFFF node

The variant field contains a value which identifies the layout of the UUID. The bit layout described above is valid only for a UUID with a variant value of 2, which indicates the Leach-Salz variant.

The version field holds a value that describes the type of this UUID. There are four different basic types of UUIDs: time-based, DCE security, name-based, and randomly generated UUIDs. These types have a version value of 1, 2, 3 and 4, respectively.

For more information including algorithms used to create UUIDs, see  [*RFC 4122: A Universally Unique IDentifier (UUID) URN Namespace*](http://www.ietf.org/rfc/rfc4122.txt), section 4.2 "Algorithms for Creating a Time-Based UUID".

**Since:** 1.5 **See Also:**[Serialized Form](http://docs.google.com/serialized-form.html#java.util.UUID)

| **Constructor Summary** | |
| --- | --- |
| [**UUID**](http://docs.google.com/java/util/UUID.html#UUID(long,%20long))(long mostSigBits, long leastSigBits)            Constructs a new UUID using the specified data. |

| **Method Summary** | |
| --- | --- |
| int | [**clockSequence**](http://docs.google.com/java/util/UUID.html#clockSequence())()            The clock sequence value associated with this UUID. |
| int | [**compareTo**](http://docs.google.com/java/util/UUID.html#compareTo(java.util.UUID))([UUID](http://docs.google.com/java/util/UUID.html) val)            Compares this UUID with the specified UUID. |
| boolean | [**equals**](http://docs.google.com/java/util/UUID.html#equals(java.lang.Object))([Object](http://docs.google.com/java/lang/Object.html) obj)            Compares this object to the specified object. |
| static [UUID](http://docs.google.com/java/util/UUID.html) | [**fromString**](http://docs.google.com/java/util/UUID.html#fromString(java.lang.String))([String](http://docs.google.com/java/lang/String.html) name)            Creates a UUID from the string standard representation as described in the [toString()](http://docs.google.com/java/util/UUID.html#toString()) method. |
| long | [**getLeastSignificantBits**](http://docs.google.com/java/util/UUID.html#getLeastSignificantBits())()            Returns the least significant 64 bits of this UUID's 128 bit value. |
| long | [**getMostSignificantBits**](http://docs.google.com/java/util/UUID.html#getMostSignificantBits())()            Returns the most significant 64 bits of this UUID's 128 bit value. |
| int | [**hashCode**](http://docs.google.com/java/util/UUID.html#hashCode())()            Returns a hash code for this UUID. |
| static [UUID](http://docs.google.com/java/util/UUID.html) | [**nameUUIDFromBytes**](http://docs.google.com/java/util/UUID.html#nameUUIDFromBytes(byte%5B%5D))(byte[] name)            Static factory to retrieve a type 3 (name based) UUID based on the specified byte array. |
| long | [**node**](http://docs.google.com/java/util/UUID.html#node())()            The node value associated with this UUID. |
| static [UUID](http://docs.google.com/java/util/UUID.html) | [**randomUUID**](http://docs.google.com/java/util/UUID.html#randomUUID())()            Static factory to retrieve a type 4 (pseudo randomly generated) UUID. |
| long | [**timestamp**](http://docs.google.com/java/util/UUID.html#timestamp())()            The timestamp value associated with this UUID. |
| [String](http://docs.google.com/java/lang/String.html) | [**toString**](http://docs.google.com/java/util/UUID.html#toString())()            Returns a String object representing this UUID. |
| int | [**variant**](http://docs.google.com/java/util/UUID.html#variant())()            The variant number associated with this UUID. |
| int | [**version**](http://docs.google.com/java/util/UUID.html#version())()            The version number associated with this UUID. |

| **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()), [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** |
| --- |

### UUID

public **UUID**(long mostSigBits,  
 long leastSigBits)

Constructs a new UUID using the specified data. mostSigBits is used for the most significant 64 bits of the UUID and leastSigBits becomes the least significant 64 bits of the UUID.

**Parameters:**mostSigBits - leastSigBits -

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

### randomUUID

public static [UUID](http://docs.google.com/java/util/UUID.html) **randomUUID**()

Static factory to retrieve a type 4 (pseudo randomly generated) UUID. The UUID is generated using a cryptographically strong pseudo random number generator.

**Returns:**a randomly generated UUID.

### nameUUIDFromBytes

public static [UUID](http://docs.google.com/java/util/UUID.html) **nameUUIDFromBytes**(byte[] name)

Static factory to retrieve a type 3 (name based) UUID based on the specified byte array.

**Parameters:**name - a byte array to be used to construct a UUID. **Returns:**a UUID generated from the specified array.

### fromString

public static [UUID](http://docs.google.com/java/util/UUID.html) **fromString**([String](http://docs.google.com/java/lang/String.html) name)

Creates a UUID from the string standard representation as described in the [toString()](http://docs.google.com/java/util/UUID.html#toString()) method.

**Parameters:**name - a string that specifies a UUID. **Returns:**a UUID with the specified value. **Throws:** [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if name does not conform to the string representation as described in [toString()](http://docs.google.com/java/util/UUID.html#toString()).

### getLeastSignificantBits

public long **getLeastSignificantBits**()

Returns the least significant 64 bits of this UUID's 128 bit value.

**Returns:**the least significant 64 bits of this UUID's 128 bit value.

### getMostSignificantBits

public long **getMostSignificantBits**()

Returns the most significant 64 bits of this UUID's 128 bit value.

**Returns:**the most significant 64 bits of this UUID's 128 bit value.

### version

public int **version**()

The version number associated with this UUID. The version number describes how this UUID was generated. The version number has the following meaning:

* 1 Time-based UUID
* 2 DCE security UUID
* 3 Name-based UUID
* 4 Randomly generated UUID

**Returns:**the version number of this UUID.

### variant

public int **variant**()

The variant number associated with this UUID. The variant number describes the layout of the UUID. The variant number has the following meaning:

* 0 Reserved for NCS backward compatibility
* 2 The Leach-Salz variant (used by this class)
* 6 Reserved, Microsoft Corporation backward compatibility
* 7 Reserved for future definition

**Returns:**the variant number of this UUID.

### timestamp

public long **timestamp**()

The timestamp value associated with this UUID.

The 60 bit timestamp value is constructed from the time\_low, time\_mid, and time\_hi fields of this UUID. The resulting timestamp is measured in 100-nanosecond units since midnight, October 15, 1582 UTC.

The timestamp value is only meaningful in a time-based UUID, which has version type 1. If this UUID is not a time-based UUID then this method throws UnsupportedOperationException.

**Throws:** [UnsupportedOperationException](http://docs.google.com/java/lang/UnsupportedOperationException.html) - if this UUID is not a version 1 UUID.

### clockSequence

public int **clockSequence**()

The clock sequence value associated with this UUID.

The 14 bit clock sequence value is constructed from the clock sequence field of this UUID. The clock sequence field is used to guarantee temporal uniqueness in a time-based UUID.

The clockSequence value is only meaningful in a time-based UUID, which has version type 1. If this UUID is not a time-based UUID then this method throws UnsupportedOperationException.

**Returns:**the clock sequence of this UUID. **Throws:** [UnsupportedOperationException](http://docs.google.com/java/lang/UnsupportedOperationException.html) - if this UUID is not a version 1 UUID.

### node

public long **node**()

The node value associated with this UUID.

The 48 bit node value is constructed from the node field of this UUID. This field is intended to hold the IEEE 802 address of the machine that generated this UUID to guarantee spatial uniqueness.

The node value is only meaningful in a time-based UUID, which has version type 1. If this UUID is not a time-based UUID then this method throws UnsupportedOperationException.

**Returns:**the node value of this UUID. **Throws:** [UnsupportedOperationException](http://docs.google.com/java/lang/UnsupportedOperationException.html) - if this UUID is not a version 1 UUID.

### toString

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

Returns a String object representing this UUID.

The UUID string representation is as described by this BNF :

UUID = <time\_low> "-" <time\_mid> "-"  
 <time\_high\_and\_version> "-"  
 <variant\_and\_sequence> "-"  
 <node>  
 time\_low = 4\*<hexOctet>  
 time\_mid = 2\*<hexOctet>  
 time\_high\_and\_version = 2\*<hexOctet>  
 variant\_and\_sequence = 2\*<hexOctet>  
 node = 6\*<hexOctet>  
 hexOctet = <hexDigit><hexDigit>  
 hexDigit =  
 "0" | "1" | "2" | "3" | "4" | "5" | "6" | "7" | "8" | "9"  
 | "a" | "b" | "c" | "d" | "e" | "f"  
 | "A" | "B" | "C" | "D" | "E" | "F"

**Overrides:**[toString](http://docs.google.com/java/lang/Object.html#toString()) in class [Object](http://docs.google.com/java/lang/Object.html) **Returns:**a string representation of this UUID.

### hashCode

public int **hashCode**()

Returns a hash code for this UUID.

**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 UUID.**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)

### equals

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

Compares this object to the specified object. The result is true if and only if the argument is not null, is a UUID object, has the same variant, and contains the same value, bit for bit, as this UUID.

**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 compare with. **Returns:**true if the objects are the same; false otherwise.**See Also:**[Object.hashCode()](http://docs.google.com/java/lang/Object.html#hashCode()), [Hashtable](http://docs.google.com/java/util/Hashtable.html)

### compareTo

public int **compareTo**([UUID](http://docs.google.com/java/util/UUID.html) val)

Compares this UUID with the specified UUID.

The first of two UUIDs follows the second if the most significant field in which the UUIDs differ is greater for the first UUID.

**Specified by:**[compareTo](http://docs.google.com/java/lang/Comparable.html#compareTo(T)) in interface [Comparable](http://docs.google.com/java/lang/Comparable.html)<[UUID](http://docs.google.com/java/util/UUID.html)> **Parameters:**val - UUID to which this UUID is to be compared. **Returns:**-1, 0 or 1 as this UUID is less than, equal to, or greater than val.

| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/UUID.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/util/UnknownFormatFlagsException.html)   [**NEXT CLASS**](http://docs.google.com/java/util/Vector.html) | [**FRAMES**](http://docs.google.com/index.html?java/util/UUID.html)    [**NO FRAMES**](http://docs.google.com/UUID.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).