| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/Timestamp.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/sql/Time.html)   [**NEXT CLASS**](http://docs.google.com/java/sql/Types.html) | [**FRAMES**](http://docs.google.com/index.html?java/sql/Timestamp.html)    [**NO FRAMES**](http://docs.google.com/Timestamp.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | FIELD | [CONSTR](#3znysh7) | [METHOD](#2et92p0) | DETAIL: FIELD | [CONSTR](#1t3h5sf) | [METHOD](#17dp8vu) |

## **java.sql**

Class Timestamp

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

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

public class **Timestamp**extends [Date](http://docs.google.com/java/util/Date.html)

A thin wrapper around java.util.Date that allows the JDBC API to identify this as an SQL TIMESTAMP value. It adds the ability to hold the SQL TIMESTAMP fractional seconds value, by allowing the specification of fractional seconds to a precision of nanoseconds. A Timestamp also provides formatting and parsing operations to support the JDBC escape syntax for timestamp values.

The precision of a Timestamp object is calculated to be either:

* 19 , which is the number of characters in yyyy-mm-dd hh:mm:ss
* 20 + s , which is the number of characters in the yyyy-mm-dd hh:mm:ss.[fff...] and s represents the scale of the given Timestamp, its fractional seconds precision.

**Note:** This type is a composite of a java.util.Date and a separate nanoseconds value. Only integral seconds are stored in the java.util.Date component. The fractional seconds - the nanos - are separate. The Timestamp.equals(Object) method never returns true when passed an object that isn't an instance of java.sql.Timestamp, because the nanos component of a date is unknown. As a result, the Timestamp.equals(Object) method is not symmetric with respect to the java.util.Date.equals(Object) method. Also, the hashcode method uses the underlying java.util.Date implementation and therefore does not include nanos in its computation.

Due to the differences between the Timestamp class and the java.util.Date class mentioned above, it is recommended that code not view Timestamp values generically as an instance of java.util.Date. The inheritance relationship between Timestamp and java.util.Date really denotes implementation inheritance, and not type inheritance.

**See Also:**[Serialized Form](http://docs.google.com/serialized-form.html#java.sql.Timestamp)

| **Constructor Summary** | |
| --- | --- |
| [**Timestamp**](http://docs.google.com/java/sql/Timestamp.html#Timestamp(int,%20int,%20int,%20int,%20int,%20int,%20int))(int year, int month, int date, int hour, int minute, int second, int nano)  **Deprecated.** *instead use the constructor Timestamp(long millis)* |
| [**Timestamp**](http://docs.google.com/java/sql/Timestamp.html#Timestamp(long))(long time)            Constructs a Timestamp object using a milliseconds time value. |

| **Method Summary** | |
| --- | --- |
| boolean | [**after**](http://docs.google.com/java/sql/Timestamp.html#after(java.sql.Timestamp))([Timestamp](http://docs.google.com/java/sql/Timestamp.html) ts)            Indicates whether this Timestamp object is later than the given Timestamp object. |
| boolean | [**before**](http://docs.google.com/java/sql/Timestamp.html#before(java.sql.Timestamp))([Timestamp](http://docs.google.com/java/sql/Timestamp.html) ts)            Indicates whether this Timestamp object is earlier than the given Timestamp object. |
| int | [**compareTo**](http://docs.google.com/java/sql/Timestamp.html#compareTo(java.util.Date))([Date](http://docs.google.com/java/util/Date.html) o)            Compares this Timestamp object to the given Date, which must be a Timestamp object. |
| int | [**compareTo**](http://docs.google.com/java/sql/Timestamp.html#compareTo(java.sql.Timestamp))([Timestamp](http://docs.google.com/java/sql/Timestamp.html) ts)            Compares this Timestamp object to the given Timestamp object. |
| boolean | [**equals**](http://docs.google.com/java/sql/Timestamp.html#equals(java.lang.Object))([Object](http://docs.google.com/java/lang/Object.html) ts)            Tests to see if this Timestamp object is equal to the given object. |
| boolean | [**equals**](http://docs.google.com/java/sql/Timestamp.html#equals(java.sql.Timestamp))([Timestamp](http://docs.google.com/java/sql/Timestamp.html) ts)            Tests to see if this Timestamp object is equal to the given Timestamp object. |
| int | [**getNanos**](http://docs.google.com/java/sql/Timestamp.html#getNanos())()            Gets this Timestamp object's nanos value. |
| long | [**getTime**](http://docs.google.com/java/sql/Timestamp.html#getTime())()            Returns the number of milliseconds since January 1, 1970, 00:00:00 GMT represented by this Timestamp object. |
| void | [**setNanos**](http://docs.google.com/java/sql/Timestamp.html#setNanos(int))(int n)            Sets this Timestamp object's nanos field to the given value. |
| void | [**setTime**](http://docs.google.com/java/sql/Timestamp.html#setTime(long))(long time)            Sets this Timestamp object to represent a point in time that is time milliseconds after January 1, 1970 00:00:00 GMT. |
| [String](http://docs.google.com/java/lang/String.html) | [**toString**](http://docs.google.com/java/sql/Timestamp.html#toString())()            Formats a timestamp in JDBC timestamp escape format. |
| static [Timestamp](http://docs.google.com/java/sql/Timestamp.html) | [**valueOf**](http://docs.google.com/java/sql/Timestamp.html#valueOf(java.lang.String))([String](http://docs.google.com/java/lang/String.html) s)            Converts a String object in JDBC timestamp escape format to a Timestamp value. |

| **Methods inherited from class java.util.**[**Date**](http://docs.google.com/java/util/Date.html) |
| --- |
| [after](http://docs.google.com/java/util/Date.html#after(java.util.Date)), [before](http://docs.google.com/java/util/Date.html#before(java.util.Date)), [clone](http://docs.google.com/java/util/Date.html#clone()), [getDate](http://docs.google.com/java/util/Date.html#getDate()), [getDay](http://docs.google.com/java/util/Date.html#getDay()), [getHours](http://docs.google.com/java/util/Date.html#getHours()), [getMinutes](http://docs.google.com/java/util/Date.html#getMinutes()), [getMonth](http://docs.google.com/java/util/Date.html#getMonth()), [getSeconds](http://docs.google.com/java/util/Date.html#getSeconds()), [getTimezoneOffset](http://docs.google.com/java/util/Date.html#getTimezoneOffset()), [getYear](http://docs.google.com/java/util/Date.html#getYear()), [hashCode](http://docs.google.com/java/util/Date.html#hashCode()), [parse](http://docs.google.com/java/util/Date.html#parse(java.lang.String)), [setDate](http://docs.google.com/java/util/Date.html#setDate(int)), [setHours](http://docs.google.com/java/util/Date.html#setHours(int)), [setMinutes](http://docs.google.com/java/util/Date.html#setMinutes(int)), [setMonth](http://docs.google.com/java/util/Date.html#setMonth(int)), [setSeconds](http://docs.google.com/java/util/Date.html#setSeconds(int)), [setYear](http://docs.google.com/java/util/Date.html#setYear(int)), [toGMTString](http://docs.google.com/java/util/Date.html#toGMTString()), [toLocaleString](http://docs.google.com/java/util/Date.html#toLocaleString()), [UTC](http://docs.google.com/java/util/Date.html#UTC(int,%20int,%20int,%20int,%20int,%20int)) |

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

### Timestamp

[@Deprecated](http://docs.google.com/java/lang/Deprecated.html)  
public **Timestamp**(int year,  
 int month,  
 int date,  
 int hour,  
 int minute,  
 int second,  
 int nano)

**Deprecated.** *instead use the constructor Timestamp(long millis)*

Constructs a Timestamp object initialized with the given values.

**Parameters:**year - the year minus 1900month - 0 to 11date - 1 to 31hour - 0 to 23minute - 0 to 59second - 0 to 59nano - 0 to 999,999,999 **Throws:** [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if the nano argument is out of bounds

### Timestamp

public **Timestamp**(long time)

Constructs a Timestamp object using a milliseconds time value. The integral seconds are stored in the underlying date value; the fractional seconds are stored in the nanos field of the Timestamp object.

**Parameters:**time - milliseconds since January 1, 1970, 00:00:00 GMT. A negative number is the number of milliseconds before January 1, 1970, 00:00:00 GMT.**See Also:**[Calendar](http://docs.google.com/java/util/Calendar.html)

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

### setTime

public void **setTime**(long time)

Sets this Timestamp object to represent a point in time that is time milliseconds after January 1, 1970 00:00:00 GMT.

**Overrides:**[setTime](http://docs.google.com/java/util/Date.html#setTime(long)) in class [Date](http://docs.google.com/java/util/Date.html) **Parameters:**time - the number of milliseconds.**See Also:**[getTime()](http://docs.google.com/java/sql/Timestamp.html#getTime()), [Timestamp(long time)](http://docs.google.com/java/sql/Timestamp.html#Timestamp(long)), [Calendar](http://docs.google.com/java/util/Calendar.html)

### getTime

public long **getTime**()

Returns the number of milliseconds since January 1, 1970, 00:00:00 GMT represented by this Timestamp object.

**Overrides:**[getTime](http://docs.google.com/java/util/Date.html#getTime()) in class [Date](http://docs.google.com/java/util/Date.html) **Returns:**the number of milliseconds since January 1, 1970, 00:00:00 GMT represented by this date.**See Also:**[setTime(long)](http://docs.google.com/java/sql/Timestamp.html#setTime(long))

### valueOf

public static [Timestamp](http://docs.google.com/java/sql/Timestamp.html) **valueOf**([String](http://docs.google.com/java/lang/String.html) s)

Converts a String object in JDBC timestamp escape format to a Timestamp value.

**Parameters:**s - timestamp in format yyyy-mm-dd hh:mm:ss[.f...]. The fractional seconds may be omitted. **Returns:**corresponding Timestamp value **Throws:** [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if the given argument does not have the format yyyy-mm-dd hh:mm:ss[.f...]

### toString

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

Formats a timestamp in JDBC timestamp escape format. yyyy-mm-dd hh:mm:ss.fffffffff, where ffffffffff indicates nanoseconds.

**Overrides:**[toString](http://docs.google.com/java/util/Date.html#toString()) in class [Date](http://docs.google.com/java/util/Date.html) **Returns:**a String object in yyyy-mm-dd hh:mm:ss.fffffffff format**See Also:**[Date.toLocaleString()](http://docs.google.com/java/util/Date.html#toLocaleString()), [Date.toGMTString()](http://docs.google.com/java/util/Date.html#toGMTString())

### getNanos

public int **getNanos**()

Gets this Timestamp object's nanos value.

**Returns:**this Timestamp object's fractional seconds component**See Also:**[setNanos(int)](http://docs.google.com/java/sql/Timestamp.html#setNanos(int))

### setNanos

public void **setNanos**(int n)

Sets this Timestamp object's nanos field to the given value.

**Parameters:**n - the new fractional seconds component **Throws:** [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if the given argument is greater than 999999999 or less than 0**See Also:**[getNanos()](http://docs.google.com/java/sql/Timestamp.html#getNanos())

### equals

public boolean **equals**([Timestamp](http://docs.google.com/java/sql/Timestamp.html) ts)

Tests to see if this Timestamp object is equal to the given Timestamp object.

**Parameters:**ts - the Timestamp value to compare with **Returns:**true if the given Timestamp object is equal to this Timestamp object; false otherwise

### equals

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

Tests to see if this Timestamp object is equal to the given object. This version of the method equals has been added to fix the incorrect signature of Timestamp.equals(Timestamp) and to preserve backward compatibility with existing class files. Note: This method is not symmetric with respect to the equals(Object) method in the base class.

**Overrides:**[equals](http://docs.google.com/java/util/Date.html#equals(java.lang.Object)) in class [Date](http://docs.google.com/java/util/Date.html) **Parameters:**ts - the Object value to compare with **Returns:**true if the given Object is an instance of a Timestamp that is equal to this Timestamp object; false otherwise**See Also:**[Date.getTime()](http://docs.google.com/java/util/Date.html#getTime())

### before

public boolean **before**([Timestamp](http://docs.google.com/java/sql/Timestamp.html) ts)

Indicates whether this Timestamp object is earlier than the given Timestamp object.

**Parameters:**ts - the Timestamp value to compare with **Returns:**true if this Timestamp object is earlier; false otherwise

### after

public boolean **after**([Timestamp](http://docs.google.com/java/sql/Timestamp.html) ts)

Indicates whether this Timestamp object is later than the given Timestamp object.

**Parameters:**ts - the Timestamp value to compare with **Returns:**true if this Timestamp object is later; false otherwise

### compareTo

public int **compareTo**([Timestamp](http://docs.google.com/java/sql/Timestamp.html) ts)

Compares this Timestamp object to the given Timestamp object.

**Parameters:**ts - the Timestamp object to be compared to this Timestamp object **Returns:**the value 0 if the two Timestamp objects are equal; a value less than 0 if this Timestamp object is before the given argument; and a value greater than 0 if this Timestamp object is after the given argument.**Since:** 1.4

### compareTo

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

Compares this Timestamp object to the given Date, which must be a Timestamp object. If the argument is not a Timestamp object, this method throws a ClassCastException object. (Timestamp objects are comparable only to other Timestamp objects.)

**Specified by:**[compareTo](http://docs.google.com/java/lang/Comparable.html#compareTo(T)) in interface [Comparable](http://docs.google.com/java/lang/Comparable.html)<[Date](http://docs.google.com/java/util/Date.html)>**Overrides:**[compareTo](http://docs.google.com/java/util/Date.html#compareTo(java.util.Date)) in class [Date](http://docs.google.com/java/util/Date.html) **Parameters:**o - the Date to be compared, which must be a Timestamp object **Returns:**the value 0 if this Timestamp object and the given object are equal; a value less than 0 if this Timestamp object is before the given argument; and a value greater than 0 if this Timestamp object is after the given argument.**Since:** 1.5

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