| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/TimeZone.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/TimerTask.html)   [**NEXT CLASS**](http://docs.google.com/java/util/TooManyListenersException.html) | [**FRAMES**](http://docs.google.com/index.html?java/util/TimeZone.html)    [**NO FRAMES**](http://docs.google.com/TimeZone.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | [FIELD](#tyjcwt) | [CONSTR](#3dy6vkm) | [METHOD](#1t3h5sf) | DETAIL: [FIELD](#2s8eyo1) | [CONSTR](#26in1rg) | [METHOD](#35nkun2) |

## **java.util**

Class TimeZone

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

**All Implemented Interfaces:** [Serializable](http://docs.google.com/java/io/Serializable.html), [Cloneable](http://docs.google.com/java/lang/Cloneable.html) **Direct Known Subclasses:** [SimpleTimeZone](http://docs.google.com/java/util/SimpleTimeZone.html)

public abstract class **TimeZone**extends [Object](http://docs.google.com/java/lang/Object.html)implements [Serializable](http://docs.google.com/java/io/Serializable.html), [Cloneable](http://docs.google.com/java/lang/Cloneable.html)

TimeZone represents a time zone offset, and also figures out daylight savings.

Typically, you get a TimeZone using getDefault which creates a TimeZone based on the time zone where the program is running. For example, for a program running in Japan, getDefault creates a TimeZone object based on Japanese Standard Time.

You can also get a TimeZone using getTimeZone along with a time zone ID. For instance, the time zone ID for the U.S. Pacific Time zone is "America/Los\_Angeles". So, you can get a U.S. Pacific Time TimeZone object with:

TimeZone tz = TimeZone.getTimeZone("America/Los\_Angeles");

You can use the getAvailableIDs method to iterate through all the supported time zone IDs. You can then choose a supported ID to get a TimeZone. If the time zone you want is not represented by one of the supported IDs, then a custom time zone ID can be specified to produce a TimeZone. The syntax of a custom time zone ID is:

*CustomID:*  
 GMT *Sign* *Hours* : *Minutes*  
 GMT *Sign* *Hours* *Minutes*  
 GMT *Sign* *Hours*  
 *Sign:* one of  
 + -  
 *Hours:*  
 *Digit*  
 *Digit* *Digit*  
 *Minutes:*  
 *Digit* *Digit*  
 *Digit:* one of  
 0 1 2 3 4 5 6 7 8 9

*Hours* must be between 0 to 23 and *Minutes* must be between 00 to 59. For example, "GMT+10" and "GMT+0010" mean ten hours and ten minutes ahead of GMT, respectively.

The format is locale independent and digits must be taken from the Basic Latin block of the Unicode standard. No daylight saving time transition schedule can be specified with a custom time zone ID. If the specified string doesn't match the syntax, "GMT" is used.

When creating a TimeZone, the specified custom time zone ID is normalized in the following syntax:

*NormalizedCustomID:*  
 GMT *Sign* *TwoDigitHours* : *Minutes*  
 *Sign:* one of  
 + -  
 *TwoDigitHours:*  
 *Digit* *Digit*  
 *Minutes:*  
 *Digit* *Digit*  
 *Digit:* one of  
 0 1 2 3 4 5 6 7 8 9

For example, TimeZone.getTimeZone("GMT-8").getID() returns "GMT-08:00".

#### Three-letter time zone IDs

For compatibility with JDK 1.1.x, some other three-letter time zone IDs (such as "PST", "CTT", "AST") are also supported. However, **their use is deprecated** because the same abbreviation is often used for multiple time zones (for example, "CST" could be U.S. "Central Standard Time" and "China Standard Time"), and the Java platform can then only recognize one of them.

**Since:** JDK1.1 **See Also:**[Calendar](http://docs.google.com/java/util/Calendar.html), [GregorianCalendar](http://docs.google.com/java/util/GregorianCalendar.html), [SimpleTimeZone](http://docs.google.com/java/util/SimpleTimeZone.html), [Serialized Form](http://docs.google.com/serialized-form.html#java.util.TimeZone)

| **Field Summary** | |
| --- | --- |
| static int | [**LONG**](http://docs.google.com/java/util/TimeZone.html#LONG)            A style specifier for getDisplayName() indicating a long name, such as "Pacific Standard Time." |
| static int | [**SHORT**](http://docs.google.com/java/util/TimeZone.html#SHORT)            A style specifier for getDisplayName() indicating a short name, such as "PST." |

| **Constructor Summary** | |
| --- | --- |
| [**TimeZone**](http://docs.google.com/java/util/TimeZone.html#TimeZone())()            Sole constructor. |

| **Method Summary** | |
| --- | --- |
| [Object](http://docs.google.com/java/lang/Object.html) | [**clone**](http://docs.google.com/java/util/TimeZone.html#clone())()            Creates a copy of this TimeZone. |
| static [String](http://docs.google.com/java/lang/String.html)[] | [**getAvailableIDs**](http://docs.google.com/java/util/TimeZone.html#getAvailableIDs())()            Gets all the available IDs supported. |
| static [String](http://docs.google.com/java/lang/String.html)[] | [**getAvailableIDs**](http://docs.google.com/java/util/TimeZone.html#getAvailableIDs(int))(int rawOffset)            Gets the available IDs according to the given time zone offset in milliseconds. |
| static [TimeZone](http://docs.google.com/java/util/TimeZone.html) | [**getDefault**](http://docs.google.com/java/util/TimeZone.html#getDefault())()            Gets the default TimeZone for this host. |
| [String](http://docs.google.com/java/lang/String.html) | [**getDisplayName**](http://docs.google.com/java/util/TimeZone.html#getDisplayName())()            Returns a name of this time zone suitable for presentation to the user in the default locale. |
| [String](http://docs.google.com/java/lang/String.html) | [**getDisplayName**](http://docs.google.com/java/util/TimeZone.html#getDisplayName(boolean,%20int))(boolean daylight, int style)            Returns a name of this time zone suitable for presentation to the user in the default locale. |
| [String](http://docs.google.com/java/lang/String.html) | [**getDisplayName**](http://docs.google.com/java/util/TimeZone.html#getDisplayName(boolean,%20int,%20java.util.Locale))(boolean daylight, int style, [Locale](http://docs.google.com/java/util/Locale.html) locale)            Returns a name of this time zone suitable for presentation to the user in the specified locale. |
| [String](http://docs.google.com/java/lang/String.html) | [**getDisplayName**](http://docs.google.com/java/util/TimeZone.html#getDisplayName(java.util.Locale))([Locale](http://docs.google.com/java/util/Locale.html) locale)            Returns a name of this time zone suitable for presentation to the user in the specified locale. |
| int | [**getDSTSavings**](http://docs.google.com/java/util/TimeZone.html#getDSTSavings())()            Returns the amount of time to be added to local standard time to get local wall clock time. |
| [String](http://docs.google.com/java/lang/String.html) | [**getID**](http://docs.google.com/java/util/TimeZone.html#getID())()            Gets the ID of this time zone. |
| abstract  int | [**getOffset**](http://docs.google.com/java/util/TimeZone.html#getOffset(int,%20int,%20int,%20int,%20int,%20int))(int era, int year, int month, int day, int dayOfWeek, int milliseconds)            Gets the time zone offset, for current date, modified in case of daylight savings. |
| int | [**getOffset**](http://docs.google.com/java/util/TimeZone.html#getOffset(long))(long date)            Returns the offset of this time zone from UTC at the specified date. |
| abstract  int | [**getRawOffset**](http://docs.google.com/java/util/TimeZone.html#getRawOffset())()            Returns the amount of time in milliseconds to add to UTC to get standard time in this time zone. |
| static [TimeZone](http://docs.google.com/java/util/TimeZone.html) | [**getTimeZone**](http://docs.google.com/java/util/TimeZone.html#getTimeZone(java.lang.String))([String](http://docs.google.com/java/lang/String.html) ID)            Gets the TimeZone for the given ID. |
| boolean | [**hasSameRules**](http://docs.google.com/java/util/TimeZone.html#hasSameRules(java.util.TimeZone))([TimeZone](http://docs.google.com/java/util/TimeZone.html) other)            Returns true if this zone has the same rule and offset as another zone. |
| abstract  boolean | [**inDaylightTime**](http://docs.google.com/java/util/TimeZone.html#inDaylightTime(java.util.Date))([Date](http://docs.google.com/java/util/Date.html) date)            Queries if the given date is in daylight savings time in this time zone. |
| static void | [**setDefault**](http://docs.google.com/java/util/TimeZone.html#setDefault(java.util.TimeZone))([TimeZone](http://docs.google.com/java/util/TimeZone.html) zone)            Sets the TimeZone that is returned by the getDefault method. |
| void | [**setID**](http://docs.google.com/java/util/TimeZone.html#setID(java.lang.String))([String](http://docs.google.com/java/lang/String.html) ID)            Sets the time zone ID. |
| abstract  void | [**setRawOffset**](http://docs.google.com/java/util/TimeZone.html#setRawOffset(int))(int offsetMillis)            Sets the base time zone offset to GMT. |
| abstract  boolean | [**useDaylightTime**](http://docs.google.com/java/util/TimeZone.html#useDaylightTime())()            Queries if this time zone uses daylight savings time. |

| **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)) |

| **Field Detail** |
| --- |

### SHORT

public static final int **SHORT**

A style specifier for getDisplayName() indicating a short name, such as "PST."

**Since:** 1.2 **See Also:**[LONG](http://docs.google.com/java/util/TimeZone.html#LONG), [Constant Field Values](http://docs.google.com/constant-values.html#java.util.TimeZone.SHORT)

### LONG

public static final int **LONG**

A style specifier for getDisplayName() indicating a long name, such as "Pacific Standard Time."

**Since:** 1.2 **See Also:**[SHORT](http://docs.google.com/java/util/TimeZone.html#SHORT), [Constant Field Values](http://docs.google.com/constant-values.html#java.util.TimeZone.LONG)

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

### TimeZone

public **TimeZone**()

Sole constructor. (For invocation by subclass constructors, typically implicit.)

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

### getOffset

public abstract int **getOffset**(int era,  
 int year,  
 int month,  
 int day,  
 int dayOfWeek,  
 int milliseconds)

Gets the time zone offset, for current date, modified in case of daylight savings. This is the offset to add to UTC to get local time.

This method returns a historically correct offset if an underlying TimeZone implementation subclass supports historical Daylight Saving Time schedule and GMT offset changes.

**Parameters:**era - the era of the given date.year - the year in the given date.month - the month in the given date. Month is 0-based. e.g., 0 for January.day - the day-in-month of the given date.dayOfWeek - the day-of-week of the given date.milliseconds - the milliseconds in day in *standard* local time. **Returns:**the offset in milliseconds to add to GMT to get local time.**See Also:**[Calendar.ZONE\_OFFSET](http://docs.google.com/java/util/Calendar.html#ZONE_OFFSET), [Calendar.DST\_OFFSET](http://docs.google.com/java/util/Calendar.html#DST_OFFSET)

### getOffset

public int **getOffset**(long date)

Returns the offset of this time zone from UTC at the specified date. If Daylight Saving Time is in effect at the specified date, the offset value is adjusted with the amount of daylight saving.

This method returns a historically correct offset value if an underlying TimeZone implementation subclass supports historical Daylight Saving Time schedule and GMT offset changes.

**Parameters:**date - the date represented in milliseconds since January 1, 1970 00:00:00 GMT **Returns:**the amount of time in milliseconds to add to UTC to get local time.**Since:** 1.4 **See Also:**[Calendar.ZONE\_OFFSET](http://docs.google.com/java/util/Calendar.html#ZONE_OFFSET), [Calendar.DST\_OFFSET](http://docs.google.com/java/util/Calendar.html#DST_OFFSET)

### setRawOffset

public abstract void **setRawOffset**(int offsetMillis)

Sets the base time zone offset to GMT. This is the offset to add to UTC to get local time.

If an underlying TimeZone implementation subclass supports historical GMT offset changes, the specified GMT offset is set as the latest GMT offset and the difference from the known latest GMT offset value is used to adjust all historical GMT offset values.

**Parameters:**offsetMillis - the given base time zone offset to GMT.

### getRawOffset

public abstract int **getRawOffset**()

Returns the amount of time in milliseconds to add to UTC to get standard time in this time zone. Because this value is not affected by daylight saving time, it is called *raw offset*.

If an underlying TimeZone implementation subclass supports historical GMT offset changes, the method returns the raw offset value of the current date. In Honolulu, for example, its raw offset changed from GMT-10:30 to GMT-10:00 in 1947, and this method always returns -36000000 milliseconds (i.e., -10 hours).

**Returns:**the amount of raw offset time in milliseconds to add to UTC.**See Also:**[Calendar.ZONE\_OFFSET](http://docs.google.com/java/util/Calendar.html#ZONE_OFFSET)

### getID

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

Gets the ID of this time zone.

**Returns:**the ID of this time zone.

### setID

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

Sets the time zone ID. This does not change any other data in the time zone object.

**Parameters:**ID - the new time zone ID.

### getDisplayName

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

Returns a name of this time zone suitable for presentation to the user in the default locale. This method returns the long name, not including daylight savings. If the display name is not available for the locale, then this method returns a string in the [normalized custom ID format](#2et92p0).

**Returns:**the human-readable name of this time zone in the default locale.**Since:** 1.2

### getDisplayName

public final [String](http://docs.google.com/java/lang/String.html) **getDisplayName**([Locale](http://docs.google.com/java/util/Locale.html) locale)

Returns a name of this time zone suitable for presentation to the user in the specified locale. This method returns the long name, not including daylight savings. If the display name is not available for the locale, then this method returns a string in the [normalized custom ID format](#2et92p0).

**Parameters:**locale - the locale in which to supply the display name. **Returns:**the human-readable name of this time zone in the given locale.**Since:** 1.2

### getDisplayName

public final [String](http://docs.google.com/java/lang/String.html) **getDisplayName**(boolean daylight,  
 int style)

Returns a name of this time zone suitable for presentation to the user in the default locale. If the display name is not available for the locale, then this method returns a string in the [normalized custom ID format](#2et92p0).

**Parameters:**daylight - if true, return the daylight savings name.style - either LONG or SHORT **Returns:**the human-readable name of this time zone in the default locale.**Since:** 1.2

### getDisplayName

public [String](http://docs.google.com/java/lang/String.html) **getDisplayName**(boolean daylight,  
 int style,  
 [Locale](http://docs.google.com/java/util/Locale.html) locale)

Returns a name of this time zone suitable for presentation to the user in the specified locale. If the display name is not available for the locale, then this method returns a string in the [normalized custom ID format](#2et92p0).

**Parameters:**daylight - if true, return the daylight savings name.style - either LONG or SHORTlocale - the locale in which to supply the display name. **Returns:**the human-readable name of this time zone in the given locale. **Throws:** [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - style is invalid.**Since:** 1.2

### getDSTSavings

public int **getDSTSavings**()

Returns the amount of time to be added to local standard time to get local wall clock time.

The default implementation always returns 3600000 milliseconds (i.e., one hour) if this time zone observes Daylight Saving Time. Otherwise, 0 (zero) is returned.

If an underlying TimeZone implementation subclass supports historical Daylight Saving Time changes, this method returns the known latest daylight saving value.

**Returns:**the amount of saving time in milliseconds**Since:** 1.4

### useDaylightTime

public abstract boolean **useDaylightTime**()

Queries if this time zone uses daylight savings time.

If an underlying TimeZone implementation subclass supports historical Daylight Saving Time schedule changes, the method refers to the latest Daylight Saving Time schedule information.

**Returns:**true if this time zone uses daylight savings time, false, otherwise.

### inDaylightTime

public abstract boolean **inDaylightTime**([Date](http://docs.google.com/java/util/Date.html) date)

Queries if the given date is in daylight savings time in this time zone.

**Parameters:**date - the given Date. **Returns:**true if the given date is in daylight savings time, false, otherwise.

### getTimeZone

public static [TimeZone](http://docs.google.com/java/util/TimeZone.html) **getTimeZone**([String](http://docs.google.com/java/lang/String.html) ID)

Gets the TimeZone for the given ID.

**Parameters:**ID - the ID for a TimeZone, either an abbreviation such as "PST", a full name such as "America/Los\_Angeles", or a custom ID such as "GMT-8:00". Note that the support of abbreviations is for JDK 1.1.x compatibility only and full names should be used. **Returns:**the specified TimeZone, or the GMT zone if the given ID cannot be understood.

### getAvailableIDs

public static [String](http://docs.google.com/java/lang/String.html)[] **getAvailableIDs**(int rawOffset)

Gets the available IDs according to the given time zone offset in milliseconds.

**Parameters:**rawOffset - the given time zone GMT offset in milliseconds. **Returns:**an array of IDs, where the time zone for that ID has the specified GMT offset. For example, "America/Phoenix" and "America/Denver" both have GMT-07:00, but differ in daylight savings behavior.**See Also:**[getRawOffset()](http://docs.google.com/java/util/TimeZone.html#getRawOffset())

### getAvailableIDs

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

Gets all the available IDs supported.

**Returns:**an array of IDs.

### getDefault

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

Gets the default TimeZone for this host. The source of the default TimeZone may vary with implementation.

**Returns:**a default TimeZone.**See Also:**[setDefault(java.util.TimeZone)](http://docs.google.com/java/util/TimeZone.html#setDefault(java.util.TimeZone))

### setDefault

public static void **setDefault**([TimeZone](http://docs.google.com/java/util/TimeZone.html) zone)

Sets the TimeZone that is returned by the getDefault method. If zone is null, reset the default to the value it had originally when the VM first started.

**Parameters:**zone - the new default time zone**See Also:**[getDefault()](http://docs.google.com/java/util/TimeZone.html#getDefault())

### hasSameRules

public boolean **hasSameRules**([TimeZone](http://docs.google.com/java/util/TimeZone.html) other)

Returns true if this zone has the same rule and offset as another zone. That is, if this zone differs only in ID, if at all. Returns false if the other zone is null.

**Parameters:**other - the TimeZone object to be compared with **Returns:**true if the other zone is not null and is the same as this one, with the possible exception of the ID**Since:** 1.2

### clone

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

Creates a copy of this TimeZone.

**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 of this TimeZone**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/TimeZone.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/TimerTask.html)   [**NEXT CLASS**](http://docs.google.com/java/util/TooManyListenersException.html) | [**FRAMES**](http://docs.google.com/index.html?java/util/TimeZone.html)    [**NO FRAMES**](http://docs.google.com/TimeZone.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | [FIELD](#tyjcwt) | [CONSTR](#3dy6vkm) | [METHOD](#1t3h5sf) | DETAIL: [FIELD](#2s8eyo1) | [CONSTR](#26in1rg) | [METHOD](#35nkun2) |

[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).