| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/SimpleTimeZone.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/Set.html)   [**NEXT CLASS**](http://docs.google.com/java/util/SortedMap.html) | [**FRAMES**](http://docs.google.com/index.html?java/util/SimpleTimeZone.html)    [**NO FRAMES**](http://docs.google.com/SimpleTimeZone.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | [FIELD](#3znysh7) | [CONSTR](#tyjcwt) | [METHOD](#3dy6vkm) | DETAIL: [FIELD](#2s8eyo1) | [CONSTR](#lnxbz9) | [METHOD](#z337ya) |

## **java.util**

Class SimpleTimeZone

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

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

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

SimpleTimeZone is a concrete subclass of TimeZone that represents a time zone for use with a Gregorian calendar. The class holds an offset from GMT, called *raw offset*, and start and end rules for a daylight saving time schedule. Since it only holds single values for each, it cannot handle historical changes in the offset from GMT and the daylight saving schedule, except that the [setStartYear](http://docs.google.com/java/util/SimpleTimeZone.html#setStartYear(int)) method can specify the year when the daylight saving time schedule starts in effect.

To construct a SimpleTimeZone with a daylight saving time schedule, the schedule can be described with a set of rules, *start-rule* and *end-rule*. A day when daylight saving time starts or ends is specified by a combination of *month*, *day-of-month*, and *day-of-week* values. The *month* value is represented by a Calendar [MONTH](http://docs.google.com/java/util/Calendar.html#MONTH) field value, such as [Calendar.MARCH](http://docs.google.com/java/util/Calendar.html#MARCH). The *day-of-week* value is represented by a Calendar [DAY\_OF\_WEEK](http://docs.google.com/java/util/Calendar.html#DAY_OF_WEEK) value, such as [SUNDAY](http://docs.google.com/java/util/Calendar.html#SUNDAY). The meanings of value combinations are as follows.

* **Exact day of month**  
  To specify an exact day of month, set the *month* and *day-of-month* to an exact value, and *day-of-week* to zero. For example, to specify March 1, set the *month* to [MARCH](http://docs.google.com/java/util/Calendar.html#MARCH), *day-of-month* to 1, and *day-of-week* to 0.
* **Day of week on or after day of month**  
  To specify a day of week on or after an exact day of month, set the *month* to an exact month value, *day-of-month* to the day on or after which the rule is applied, and *day-of-week* to a negative [DAY\_OF\_WEEK](http://docs.google.com/java/util/Calendar.html#DAY_OF_WEEK) field value. For example, to specify the second Sunday of April, set *month* to [APRIL](http://docs.google.com/java/util/Calendar.html#APRIL), *day-of-month* to 8, and *day-of-week* to -[SUNDAY](http://docs.google.com/java/util/Calendar.html#SUNDAY).
* **Day of week on or before day of month**  
  To specify a day of the week on or before an exact day of the month, set *day-of-month* and *day-of-week* to a negative value. For example, to specify the last Wednesday on or before the 21st of March, set *month* to [MARCH](http://docs.google.com/java/util/Calendar.html#MARCH), *day-of-month* is -21 and *day-of-week* is -[WEDNESDAY](http://docs.google.com/java/util/Calendar.html#WEDNESDAY).
* **Last day-of-week of month**  
  To specify, the last day-of-week of the month, set *day-of-week* to a [DAY\_OF\_WEEK](http://docs.google.com/java/util/Calendar.html#DAY_OF_WEEK) value and *day-of-month* to -1. For example, to specify the last Sunday of October, set *month* to [OCTOBER](http://docs.google.com/java/util/Calendar.html#OCTOBER), *day-of-week* to [SUNDAY](http://docs.google.com/java/util/Calendar.html#SUNDAY) and *day-of-month* to -1.

The time of the day at which daylight saving time starts or ends is specified by a millisecond value within the day. There are three kinds of *mode*s to specify the time: [WALL\_TIME](http://docs.google.com/java/util/SimpleTimeZone.html#WALL_TIME), [STANDARD\_TIME](http://docs.google.com/java/util/SimpleTimeZone.html#STANDARD_TIME) and [UTC\_TIME](http://docs.google.com/java/util/SimpleTimeZone.html#UTC_TIME). For example, if daylight saving time ends at 2:00 am in the wall clock time, it can be specified by 7200000 milliseconds in the [WALL\_TIME](http://docs.google.com/java/util/SimpleTimeZone.html#WALL_TIME) mode. In this case, the wall clock time for an *end-rule* means the same thing as the daylight time.

The following are examples of parameters for constructing time zone objects.

// Base GMT offset: -8:00  
 // DST starts: at 2:00am in standard time  
 // on the first Sunday in April  
 // DST ends: at 2:00am in daylight time  
 // on the last Sunday in October  
 // Save: 1 hour  
 SimpleTimeZone(-28800000,  
 "America/Los\_Angeles",  
 Calendar.APRIL, 1, -Calendar.SUNDAY,  
 7200000,  
 Calendar.OCTOBER, -1, Calendar.SUNDAY,  
 7200000,  
 3600000)  
  
 // Base GMT offset: +1:00  
 // DST starts: at 1:00am in UTC time  
 // on the last Sunday in March  
 // DST ends: at 1:00am in UTC time  
 // on the last Sunday in October  
 // Save: 1 hour  
 SimpleTimeZone(3600000,  
 "Europe/Paris",  
 Calendar.MARCH, -1, Calendar.SUNDAY,  
 3600000, SimpleTimeZone.UTC\_TIME,  
 Calendar.OCTOBER, -1, Calendar.SUNDAY,  
 3600000, SimpleTimeZone.UTC\_TIME,  
 3600000)

These parameter rules are also applicable to the set rule methods, such as setStartRule.

**Since:** 1.1 **See Also:**[Calendar](http://docs.google.com/java/util/Calendar.html), [GregorianCalendar](http://docs.google.com/java/util/GregorianCalendar.html), [TimeZone](http://docs.google.com/java/util/TimeZone.html), [Serialized Form](http://docs.google.com/serialized-form.html#java.util.SimpleTimeZone)

| **Field Summary** | |
| --- | --- |
| static int | [**STANDARD\_TIME**](http://docs.google.com/java/util/SimpleTimeZone.html#STANDARD_TIME)            Constant for a mode of start or end time specified as standard time. |
| static int | [**UTC\_TIME**](http://docs.google.com/java/util/SimpleTimeZone.html#UTC_TIME)            Constant for a mode of start or end time specified as UTC. |
| static int | [**WALL\_TIME**](http://docs.google.com/java/util/SimpleTimeZone.html#WALL_TIME)            Constant for a mode of start or end time specified as wall clock time. |

| **Fields inherited from class java.util.**[**TimeZone**](http://docs.google.com/java/util/TimeZone.html) |
| --- |
| [LONG](http://docs.google.com/java/util/TimeZone.html#LONG), [SHORT](http://docs.google.com/java/util/TimeZone.html#SHORT) |

| **Constructor Summary** | |
| --- | --- |
| [**SimpleTimeZone**](http://docs.google.com/java/util/SimpleTimeZone.html#SimpleTimeZone(int,%20java.lang.String))(int rawOffset, [String](http://docs.google.com/java/lang/String.html) ID)            Constructs a SimpleTimeZone with the given base time zone offset from GMT and time zone ID with no daylight saving time schedule. |
| [**SimpleTimeZone**](http://docs.google.com/java/util/SimpleTimeZone.html#SimpleTimeZone(int,%20java.lang.String,%20int,%20int,%20int,%20int,%20int,%20int,%20int,%20int))(int rawOffset, [String](http://docs.google.com/java/lang/String.html) ID, int startMonth, int startDay, int startDayOfWeek, int startTime, int endMonth, int endDay, int endDayOfWeek, int endTime)            Constructs a SimpleTimeZone with the given base time zone offset from GMT, time zone ID, and rules for starting and ending the daylight time. |
| [**SimpleTimeZone**](http://docs.google.com/java/util/SimpleTimeZone.html#SimpleTimeZone(int,%20java.lang.String,%20int,%20int,%20int,%20int,%20int,%20int,%20int,%20int,%20int))(int rawOffset, [String](http://docs.google.com/java/lang/String.html) ID, int startMonth, int startDay, int startDayOfWeek, int startTime, int endMonth, int endDay, int endDayOfWeek, int endTime, int dstSavings)            Constructs a SimpleTimeZone with the given base time zone offset from GMT, time zone ID, and rules for starting and ending the daylight time. |
| [**SimpleTimeZone**](http://docs.google.com/java/util/SimpleTimeZone.html#SimpleTimeZone(int,%20java.lang.String,%20int,%20int,%20int,%20int,%20int,%20int,%20int,%20int,%20int,%20int,%20int))(int rawOffset, [String](http://docs.google.com/java/lang/String.html) ID, int startMonth, int startDay, int startDayOfWeek, int startTime, int startTimeMode, int endMonth, int endDay, int endDayOfWeek, int endTime, int endTimeMode, int dstSavings)            Constructs a SimpleTimeZone with the given base time zone offset from GMT, time zone ID, and rules for starting and ending the daylight time. |

| **Method Summary** | |
| --- | --- |
| [Object](http://docs.google.com/java/lang/Object.html) | [**clone**](http://docs.google.com/java/util/SimpleTimeZone.html#clone())()            Returns a clone of this SimpleTimeZone instance. |
| boolean | [**equals**](http://docs.google.com/java/util/SimpleTimeZone.html#equals(java.lang.Object))([Object](http://docs.google.com/java/lang/Object.html) obj)            Compares the equality of two SimpleTimeZone objects. |
| int | [**getDSTSavings**](http://docs.google.com/java/util/SimpleTimeZone.html#getDSTSavings())()            Returns the amount of time in milliseconds that the clock is advanced during daylight saving time. |
| int | [**getOffset**](http://docs.google.com/java/util/SimpleTimeZone.html#getOffset(int,%20int,%20int,%20int,%20int,%20int))(int era, int year, int month, int day, int dayOfWeek, int millis)            Returns the difference in milliseconds between local time and UTC, taking into account both the raw offset and the effect of daylight saving, for the specified date and time. |
| int | [**getOffset**](http://docs.google.com/java/util/SimpleTimeZone.html#getOffset(long))(long date)            Returns the offset of this time zone from UTC at the given time. |
| int | [**getRawOffset**](http://docs.google.com/java/util/SimpleTimeZone.html#getRawOffset())()            Gets the GMT offset for this time zone. |
| int | [**hashCode**](http://docs.google.com/java/util/SimpleTimeZone.html#hashCode())()            Generates the hash code for the SimpleDateFormat object. |
| boolean | [**hasSameRules**](http://docs.google.com/java/util/SimpleTimeZone.html#hasSameRules(java.util.TimeZone))([TimeZone](http://docs.google.com/java/util/TimeZone.html) other)            Returns true if this zone has the same rules and offset as another zone. |
| boolean | [**inDaylightTime**](http://docs.google.com/java/util/SimpleTimeZone.html#inDaylightTime(java.util.Date))([Date](http://docs.google.com/java/util/Date.html) date)            Queries if the given date is in daylight saving time. |
| void | [**setDSTSavings**](http://docs.google.com/java/util/SimpleTimeZone.html#setDSTSavings(int))(int millisSavedDuringDST)            Sets the amount of time in milliseconds that the clock is advanced during daylight saving time. |
| void | [**setEndRule**](http://docs.google.com/java/util/SimpleTimeZone.html#setEndRule(int,%20int,%20int))(int endMonth, int endDay, int endTime)            Sets the daylight saving time end rule to a fixed date within a month. |
| void | [**setEndRule**](http://docs.google.com/java/util/SimpleTimeZone.html#setEndRule(int,%20int,%20int,%20int))(int endMonth, int endDay, int endDayOfWeek, int endTime)            Sets the daylight saving time end rule. |
| void | [**setEndRule**](http://docs.google.com/java/util/SimpleTimeZone.html#setEndRule(int,%20int,%20int,%20int,%20boolean))(int endMonth, int endDay, int endDayOfWeek, int endTime, boolean after)            Sets the daylight saving time end rule to a weekday before or after the given date within a month, e.g., the first Monday on or after the 8th. |
| void | [**setRawOffset**](http://docs.google.com/java/util/SimpleTimeZone.html#setRawOffset(int))(int offsetMillis)            Sets the base time zone offset to GMT. |
| void | [**setStartRule**](http://docs.google.com/java/util/SimpleTimeZone.html#setStartRule(int,%20int,%20int))(int startMonth, int startDay, int startTime)            Sets the daylight saving time start rule to a fixed date within a month. |
| void | [**setStartRule**](http://docs.google.com/java/util/SimpleTimeZone.html#setStartRule(int,%20int,%20int,%20int))(int startMonth, int startDay, int startDayOfWeek, int startTime)            Sets the daylight saving time start rule. |
| void | [**setStartRule**](http://docs.google.com/java/util/SimpleTimeZone.html#setStartRule(int,%20int,%20int,%20int,%20boolean))(int startMonth, int startDay, int startDayOfWeek, int startTime, boolean after)            Sets the daylight saving time start rule to a weekday before or after the given date within a month, e.g., the first Monday on or after the 8th. |
| void | [**setStartYear**](http://docs.google.com/java/util/SimpleTimeZone.html#setStartYear(int))(int year)            Sets the daylight saving time starting year. |
| [String](http://docs.google.com/java/lang/String.html) | [**toString**](http://docs.google.com/java/util/SimpleTimeZone.html#toString())()            Returns a string representation of this time zone. |
| boolean | [**useDaylightTime**](http://docs.google.com/java/util/SimpleTimeZone.html#useDaylightTime())()            Queries if this time zone uses daylight saving time. |

| **Methods inherited from class java.util.**[**TimeZone**](http://docs.google.com/java/util/TimeZone.html) |
| --- |
| [getAvailableIDs](http://docs.google.com/java/util/TimeZone.html#getAvailableIDs()), [getAvailableIDs](http://docs.google.com/java/util/TimeZone.html#getAvailableIDs(int)), [getDefault](http://docs.google.com/java/util/TimeZone.html#getDefault()), [getDisplayName](http://docs.google.com/java/util/TimeZone.html#getDisplayName()), [getDisplayName](http://docs.google.com/java/util/TimeZone.html#getDisplayName(boolean,%20int)), [getDisplayName](http://docs.google.com/java/util/TimeZone.html#getDisplayName(boolean,%20int,%20java.util.Locale)), [getDisplayName](http://docs.google.com/java/util/TimeZone.html#getDisplayName(java.util.Locale)), [getID](http://docs.google.com/java/util/TimeZone.html#getID()), [getTimeZone](http://docs.google.com/java/util/TimeZone.html#getTimeZone(java.lang.String)), [setDefault](http://docs.google.com/java/util/TimeZone.html#setDefault(java.util.TimeZone)), [setID](http://docs.google.com/java/util/TimeZone.html#setID(java.lang.String)) |

| **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)) |

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

### WALL\_TIME

public static final int **WALL\_TIME**

Constant for a mode of start or end time specified as wall clock time. Wall clock time is standard time for the onset rule, and daylight time for the end rule.

**Since:** 1.4 **See Also:**[Constant Field Values](http://docs.google.com/constant-values.html#java.util.SimpleTimeZone.WALL_TIME)

### STANDARD\_TIME

public static final int **STANDARD\_TIME**

Constant for a mode of start or end time specified as standard time.

**Since:** 1.4 **See Also:**[Constant Field Values](http://docs.google.com/constant-values.html#java.util.SimpleTimeZone.STANDARD_TIME)

### UTC\_TIME

public static final int **UTC\_TIME**

Constant for a mode of start or end time specified as UTC. European Union rules are specified as UTC time, for example.

**Since:** 1.4 **See Also:**[Constant Field Values](http://docs.google.com/constant-values.html#java.util.SimpleTimeZone.UTC_TIME)

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

### SimpleTimeZone

public **SimpleTimeZone**(int rawOffset,  
 [String](http://docs.google.com/java/lang/String.html) ID)

Constructs a SimpleTimeZone with the given base time zone offset from GMT and time zone ID with no daylight saving time schedule.

**Parameters:**rawOffset - The base time zone offset in milliseconds to GMT.ID - The time zone name that is given to this instance.

### SimpleTimeZone

public **SimpleTimeZone**(int rawOffset,  
 [String](http://docs.google.com/java/lang/String.html) ID,  
 int startMonth,  
 int startDay,  
 int startDayOfWeek,  
 int startTime,  
 int endMonth,  
 int endDay,  
 int endDayOfWeek,  
 int endTime)

Constructs a SimpleTimeZone with the given base time zone offset from GMT, time zone ID, and rules for starting and ending the daylight time. Both startTime and endTime are specified to be represented in the wall clock time. The amount of daylight saving is assumed to be 3600000 milliseconds (i.e., one hour). This constructor is equivalent to:

SimpleTimeZone(rawOffset,  
 ID,  
 startMonth,  
 startDay,  
 startDayOfWeek,  
 startTime,  
 SimpleTimeZone.[WALL\_TIME](http://docs.google.com/java/util/SimpleTimeZone.html#WALL_TIME),  
 endMonth,  
 endDay,  
 endDayOfWeek,  
 endTime,  
 SimpleTimeZone.[WALL\_TIME](http://docs.google.com/java/util/SimpleTimeZone.html#WALL_TIME),  
 3600000)

**Parameters:**rawOffset - The given base time zone offset from GMT.ID - The time zone ID which is given to this object.startMonth - The daylight saving time starting month. Month is a [MONTH](http://docs.google.com/java/util/Calendar.html#MONTH) field value (0-based. e.g., 0 for January).startDay - The day of the month on which the daylight saving time starts. See the class description for the special cases of this parameter.startDayOfWeek - The daylight saving time starting day-of-week. See the class description for the special cases of this parameter.startTime - The daylight saving time starting time in local wall clock time (in milliseconds within the day), which is local standard time in this case.endMonth - The daylight saving time ending month. Month is a [MONTH](http://docs.google.com/java/util/Calendar.html#MONTH) field value (0-based. e.g., 9 for October).endDay - The day of the month on which the daylight saving time ends. See the class description for the special cases of this parameter.endDayOfWeek - The daylight saving time ending day-of-week. See the class description for the special cases of this parameter.endTime - The daylight saving ending time in local wall clock time, (in milliseconds within the day) which is local daylight time in this case. **Throws:** [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if the month, day, dayOfWeek, or time parameters are out of range for the start or end rule

### SimpleTimeZone

public **SimpleTimeZone**(int rawOffset,  
 [String](http://docs.google.com/java/lang/String.html) ID,  
 int startMonth,  
 int startDay,  
 int startDayOfWeek,  
 int startTime,  
 int endMonth,  
 int endDay,  
 int endDayOfWeek,  
 int endTime,  
 int dstSavings)

Constructs a SimpleTimeZone with the given base time zone offset from GMT, time zone ID, and rules for starting and ending the daylight time. Both startTime and endTime are assumed to be represented in the wall clock time. This constructor is equivalent to:

SimpleTimeZone(rawOffset,  
 ID,  
 startMonth,  
 startDay,  
 startDayOfWeek,  
 startTime,  
 SimpleTimeZone.[WALL\_TIME](http://docs.google.com/java/util/SimpleTimeZone.html#WALL_TIME),  
 endMonth,  
 endDay,  
 endDayOfWeek,  
 endTime,  
 SimpleTimeZone.[WALL\_TIME](http://docs.google.com/java/util/SimpleTimeZone.html#WALL_TIME),  
 dstSavings)

**Parameters:**rawOffset - The given base time zone offset from GMT.ID - The time zone ID which is given to this object.startMonth - The daylight saving time starting month. Month is a [MONTH](http://docs.google.com/java/util/Calendar.html#MONTH) field value (0-based. e.g., 0 for January).startDay - The day of the month on which the daylight saving time starts. See the class description for the special cases of this parameter.startDayOfWeek - The daylight saving time starting day-of-week. See the class description for the special cases of this parameter.startTime - The daylight saving time starting time in local wall clock time, which is local standard time in this case.endMonth - The daylight saving time ending month. Month is a [MONTH](http://docs.google.com/java/util/Calendar.html#MONTH) field value (0-based. e.g., 9 for October).endDay - The day of the month on which the daylight saving time ends. See the class description for the special cases of this parameter.endDayOfWeek - The daylight saving time ending day-of-week. See the class description for the special cases of this parameter.endTime - The daylight saving ending time in local wall clock time, which is local daylight time in this case.dstSavings - The amount of time in milliseconds saved during daylight saving time. **Throws:** [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if the month, day, dayOfWeek, or time parameters are out of range for the start or end rule**Since:** 1.2

### SimpleTimeZone

public **SimpleTimeZone**(int rawOffset,  
 [String](http://docs.google.com/java/lang/String.html) ID,  
 int startMonth,  
 int startDay,  
 int startDayOfWeek,  
 int startTime,  
 int startTimeMode,  
 int endMonth,  
 int endDay,  
 int endDayOfWeek,  
 int endTime,  
 int endTimeMode,  
 int dstSavings)

Constructs a SimpleTimeZone with the given base time zone offset from GMT, time zone ID, and rules for starting and ending the daylight time. This constructor takes the full set of the start and end rules parameters, including modes of startTime and endTime. The mode specifies either [wall time](http://docs.google.com/java/util/SimpleTimeZone.html#WALL_TIME) or [standard time](http://docs.google.com/java/util/SimpleTimeZone.html#STANDARD_TIME) or [UTC time](http://docs.google.com/java/util/SimpleTimeZone.html#UTC_TIME).

**Parameters:**rawOffset - The given base time zone offset from GMT.ID - The time zone ID which is given to this object.startMonth - The daylight saving time starting month. Month is a [MONTH](http://docs.google.com/java/util/Calendar.html#MONTH) field value (0-based. e.g., 0 for January).startDay - The day of the month on which the daylight saving time starts. See the class description for the special cases of this parameter.startDayOfWeek - The daylight saving time starting day-of-week. See the class description for the special cases of this parameter.startTime - The daylight saving time starting time in the time mode specified by startTimeMode.startTimeMode - The mode of the start time specified by startTime.endMonth - The daylight saving time ending month. Month is a [MONTH](http://docs.google.com/java/util/Calendar.html#MONTH) field value (0-based. e.g., 9 for October).endDay - The day of the month on which the daylight saving time ends. See the class description for the special cases of this parameter.endDayOfWeek - The daylight saving time ending day-of-week. See the class description for the special cases of this parameter.endTime - The daylight saving ending time in time time mode specified by endTimeMode.endTimeMode - The mode of the end time specified by endTimedstSavings - The amount of time in milliseconds saved during daylight saving time. **Throws:** [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if the month, day, dayOfWeek, time more, or time parameters are out of range for the start or end rule, or if a time mode value is invalid.**Since:** 1.4 **See Also:**[WALL\_TIME](http://docs.google.com/java/util/SimpleTimeZone.html#WALL_TIME), [STANDARD\_TIME](http://docs.google.com/java/util/SimpleTimeZone.html#STANDARD_TIME), [UTC\_TIME](http://docs.google.com/java/util/SimpleTimeZone.html#UTC_TIME)

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

### setStartYear

public void **setStartYear**(int year)

Sets the daylight saving time starting year.

**Parameters:**year - The daylight saving starting year.

### setStartRule

public void **setStartRule**(int startMonth,  
 int startDay,  
 int startDayOfWeek,  
 int startTime)

Sets the daylight saving time start rule. For example, if daylight saving time starts on the first Sunday in April at 2 am in local wall clock time, you can set the start rule by calling:

setStartRule(Calendar.APRIL, 1, Calendar.SUNDAY, 2\*60\*60\*1000);

**Parameters:**startMonth - The daylight saving time starting month. Month is a [MONTH](http://docs.google.com/java/util/Calendar.html#MONTH) field value (0-based. e.g., 0 for January).startDay - The day of the month on which the daylight saving time starts. See the class description for the special cases of this parameter.startDayOfWeek - The daylight saving time starting day-of-week. See the class description for the special cases of this parameter.startTime - The daylight saving time starting time in local wall clock time, which is local standard time in this case. **Throws:** [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if the startMonth, startDay, startDayOfWeek, or startTime parameters are out of range

### setStartRule

public void **setStartRule**(int startMonth,  
 int startDay,  
 int startTime)

Sets the daylight saving time start rule to a fixed date within a month. This method is equivalent to:

setStartRule(startMonth, startDay, 0, startTime)

**Parameters:**startMonth - The daylight saving time starting month. Month is a [MONTH](http://docs.google.com/java/util/Calendar.html#MONTH) field value (0-based. e.g., 0 for January).startDay - The day of the month on which the daylight saving time starts.startTime - The daylight saving time starting time in local wall clock time, which is local standard time in this case. See the class description for the special cases of this parameter. **Throws:** [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if the startMonth, startDayOfMonth, or startTime parameters are out of range**Since:** 1.2

### setStartRule

public void **setStartRule**(int startMonth,  
 int startDay,  
 int startDayOfWeek,  
 int startTime,  
 boolean after)

Sets the daylight saving time start rule to a weekday before or after the given date within a month, e.g., the first Monday on or after the 8th.

**Parameters:**startMonth - The daylight saving time starting month. Month is a [MONTH](http://docs.google.com/java/util/Calendar.html#MONTH) field value (0-based. e.g., 0 for January).startDay - The day of the month on which the daylight saving time starts.startDayOfWeek - The daylight saving time starting day-of-week.startTime - The daylight saving time starting time in local wall clock time, which is local standard time in this case.after - If true, this rule selects the first dayOfWeek on or *after* dayOfMonth. If false, this rule selects the last dayOfWeek on or *before* dayOfMonth. **Throws:** [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if the startMonth, startDay, startDayOfWeek, or startTime parameters are out of range**Since:** 1.2

### setEndRule

public void **setEndRule**(int endMonth,  
 int endDay,  
 int endDayOfWeek,  
 int endTime)

Sets the daylight saving time end rule. For example, if daylight saving time ends on the last Sunday in October at 2 am in wall clock time, you can set the end rule by calling: setEndRule(Calendar.OCTOBER, -1, Calendar.SUNDAY, 2\*60\*60\*1000);

**Parameters:**endMonth - The daylight saving time ending month. Month is a [MONTH](http://docs.google.com/java/util/Calendar.html#MONTH) field value (0-based. e.g., 9 for October).endDay - The day of the month on which the daylight saving time ends. See the class description for the special cases of this parameter.endDayOfWeek - The daylight saving time ending day-of-week. See the class description for the special cases of this parameter.endTime - The daylight saving ending time in local wall clock time, (in milliseconds within the day) which is local daylight time in this case. **Throws:** [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if the endMonth, endDay, endDayOfWeek, or endTime parameters are out of range

### setEndRule

public void **setEndRule**(int endMonth,  
 int endDay,  
 int endTime)

Sets the daylight saving time end rule to a fixed date within a month. This method is equivalent to:

setEndRule(endMonth, endDay, 0, endTime)

**Parameters:**endMonth - The daylight saving time ending month. Month is a [MONTH](http://docs.google.com/java/util/Calendar.html#MONTH) field value (0-based. e.g., 9 for October).endDay - The day of the month on which the daylight saving time ends.endTime - The daylight saving ending time in local wall clock time, (in milliseconds within the day) which is local daylight time in this case. **Throws:** [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - the endMonth, endDay, or endTime parameters are out of range**Since:** 1.2

### setEndRule

public void **setEndRule**(int endMonth,  
 int endDay,  
 int endDayOfWeek,  
 int endTime,  
 boolean after)

Sets the daylight saving time end rule to a weekday before or after the given date within a month, e.g., the first Monday on or after the 8th.

**Parameters:**endMonth - The daylight saving time ending month. Month is a [MONTH](http://docs.google.com/java/util/Calendar.html#MONTH) field value (0-based. e.g., 9 for October).endDay - The day of the month on which the daylight saving time ends.endDayOfWeek - The daylight saving time ending day-of-week.endTime - The daylight saving ending time in local wall clock time, (in milliseconds within the day) which is local daylight time in this case.after - If true, this rule selects the first endDayOfWeek on or *after* endDay. If false, this rule selects the last endDayOfWeek on or before endDay of the month. **Throws:** [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - the endMonth, endDay, endDayOfWeek, or endTime parameters are out of range**Since:** 1.2

### getOffset

public int **getOffset**(long date)

Returns the offset of this time zone from UTC at the given time. If daylight saving time is in effect at the given time, the offset value is adjusted with the amount of daylight saving.

**Overrides:**[getOffset](http://docs.google.com/java/util/TimeZone.html#getOffset(long)) in class [TimeZone](http://docs.google.com/java/util/TimeZone.html) **Parameters:**date - the time at which the time zone offset is found **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)

### getOffset

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

Returns the difference in milliseconds between local time and UTC, taking into account both the raw offset and the effect of daylight saving, for the specified date and time. This method assumes that the start and end month are distinct. It also uses a default [GregorianCalendar](http://docs.google.com/java/util/GregorianCalendar.html) object as its underlying calendar, such as for determining leap years. Do not use the result of this method with a calendar other than a default GregorianCalendar.

*Note: In general, clients should use Calendar.get(ZONE\_OFFSET) + Calendar.get(DST\_OFFSET) instead of calling this method.*

**Specified by:**[getOffset](http://docs.google.com/java/util/TimeZone.html#getOffset(int,%20int,%20int,%20int,%20int,%20int)) in class [TimeZone](http://docs.google.com/java/util/TimeZone.html) **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.millis - The milliseconds in day in *standard* local time. **Returns:**The milliseconds to add to UTC to get local time. **Throws:** [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - the era, month, day, dayOfWeek, or millis parameters are out of range**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)

### getRawOffset

public int **getRawOffset**()

Gets the GMT offset for this time zone.

**Specified by:**[getRawOffset](http://docs.google.com/java/util/TimeZone.html#getRawOffset()) in class [TimeZone](http://docs.google.com/java/util/TimeZone.html) **Returns:**the GMT offset value in milliseconds**See Also:**[setRawOffset(int)](http://docs.google.com/java/util/SimpleTimeZone.html#setRawOffset(int))

### setRawOffset

public void **setRawOffset**(int offsetMillis)

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

**Specified by:**[setRawOffset](http://docs.google.com/java/util/TimeZone.html#setRawOffset(int)) in class [TimeZone](http://docs.google.com/java/util/TimeZone.html) **Parameters:**offsetMillis - the given base time zone offset to GMT.**See Also:**[getRawOffset()](http://docs.google.com/java/util/SimpleTimeZone.html#getRawOffset())

### setDSTSavings

public void **setDSTSavings**(int millisSavedDuringDST)

Sets the amount of time in milliseconds that the clock is advanced during daylight saving time.

**Parameters:**millisSavedDuringDST - the number of milliseconds the time is advanced with respect to standard time when the daylight saving time rules are in effect. A positive number, typically one hour (3600000).**Since:** 1.2 **See Also:**[getDSTSavings()](http://docs.google.com/java/util/SimpleTimeZone.html#getDSTSavings())

### getDSTSavings

public int **getDSTSavings**()

Returns the amount of time in milliseconds that the clock is advanced during daylight saving time.

**Overrides:**[getDSTSavings](http://docs.google.com/java/util/TimeZone.html#getDSTSavings()) in class [TimeZone](http://docs.google.com/java/util/TimeZone.html) **Returns:**the number of milliseconds the time is advanced with respect to standard time when the daylight saving rules are in effect, or 0 (zero) if this time zone doesn't observe daylight saving time.**Since:** 1.2 **See Also:**[setDSTSavings(int)](http://docs.google.com/java/util/SimpleTimeZone.html#setDSTSavings(int))

### useDaylightTime

public boolean **useDaylightTime**()

Queries if this time zone uses daylight saving time.

**Specified by:**[useDaylightTime](http://docs.google.com/java/util/TimeZone.html#useDaylightTime()) in class [TimeZone](http://docs.google.com/java/util/TimeZone.html) **Returns:**true if this time zone uses daylight saving time; false otherwise.

### inDaylightTime

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

Queries if the given date is in daylight saving time.

**Specified by:**[inDaylightTime](http://docs.google.com/java/util/TimeZone.html#inDaylightTime(java.util.Date)) in class [TimeZone](http://docs.google.com/java/util/TimeZone.html) **Parameters:**date - the given Date. **Returns:**true if daylight saving time is in effective at the given date; false otherwise.

### clone

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

Returns a clone of this SimpleTimeZone instance.

**Overrides:**[clone](http://docs.google.com/java/util/TimeZone.html#clone()) in class [TimeZone](http://docs.google.com/java/util/TimeZone.html) **Returns:**a clone of this instance.**See Also:**[Cloneable](http://docs.google.com/java/lang/Cloneable.html)

### hashCode

public int **hashCode**()

Generates the hash code for the SimpleDateFormat object.

**Overrides:**[hashCode](http://docs.google.com/java/lang/Object.html#hashCode()) in class [Object](http://docs.google.com/java/lang/Object.html) **Returns:**the hash code 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)

### equals

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

Compares the equality of two SimpleTimeZone objects.

**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 SimpleTimeZone object to be compared with. **Returns:**True if the given obj is the same as this SimpleTimeZone object; false otherwise.**See Also:**[Object.hashCode()](http://docs.google.com/java/lang/Object.html#hashCode()), [Hashtable](http://docs.google.com/java/util/Hashtable.html)

### hasSameRules

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

Returns true if this zone has the same rules and offset as another zone.

**Overrides:**[hasSameRules](http://docs.google.com/java/util/TimeZone.html#hasSameRules(java.util.TimeZone)) in class [TimeZone](http://docs.google.com/java/util/TimeZone.html) **Parameters:**other - the TimeZone object to be compared with **Returns:**true if the given zone is a SimpleTimeZone and has the same rules and offset as this one**Since:** 1.2

### toString

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

Returns a string representation of this time zone.

**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 time zone.

| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/SimpleTimeZone.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/Set.html)   [**NEXT CLASS**](http://docs.google.com/java/util/SortedMap.html) | [**FRAMES**](http://docs.google.com/index.html?java/util/SimpleTimeZone.html)    [**NO FRAMES**](http://docs.google.com/SimpleTimeZone.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | [FIELD](#3znysh7) | [CONSTR](#tyjcwt) | [METHOD](#3dy6vkm) | DETAIL: [FIELD](#2s8eyo1) | [CONSTR](#lnxbz9) | [METHOD](#z337ya) |

[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).