



# TimeZone Class

Represents a time zone. Contains methods for creating a new time zone and obtaining time zone properties, such as the time zone ID, offset, and display name.

## Namespace

System

## Usage

You can use the methods in this class to get properties of a time zone, such as the properties of the time zone returned by `UserInfo.getTimeZone()`, or the time zone returned by `getTimeZone` of this class.

## Example

This example shows how to get properties of the current user's time zone and display them to the debug log. The output of the sample varies based on the user's time zone.

```
TimeZone tz = UserInfo.getTimeZone();
System.debug('Display name: ' + tz.getDisplayName());
System.debug('ID: ' + tz.getID());
// During daylight saving time for the America/Los_Angeles time zone
System.debug('Offset: ' + tz.getOffset(DateTime.newInstance(2012,10,23,12,0,0)));
// Not during daylight saving time for the America/Los_Angeles time zone
System.debug('Offset: ' + tz.getOffset(DateTime.newInstance(2012,11,23,12,0,0)));
System.debug('String format: ' + tz.toString());
```

This second example shows how to create a time zone for the New York time zone and get the offset of this time zone to the GMT time zone. The example uses two dates to get the offset from. One date is before DST (Daylight Saving Time), and one is after DST. In 2000, DST ended on Sunday, October 29 for the New York time zone. Because the date occurs after DST ends, the offset on the first date is -5 hours to GMT. In 2012, DST ended on Sunday, November 4. Because the date is within DST, the offset on the second date is -4 hours.

```
// Get the New York time zone
Timezone tz = Timezone.getTimeZone('America/New_York');

// Create a date before the 2007 shift of DST into November
DateTime dtpre = DateTime.newInstanceGMT(2000, 11, 1, 0, 0, 0);
system.debug(tz.getOffset(dtpre));    //-18000000 (= -5 hours = EST)

// Create a date after the 2007 shift of DST into November
DateTime dtpost = DateTime.newInstanceGMT(2012, 11, 1, 0, 0, 0);
system.debug(tz.getOffset(dtpost));    //-14400000 (= -4 hours = EDT)
```

This next example is similar to the previous one except that it gets the offset around the boundary of DST. In 2014, DST ended on Sunday, November 2 at 2:00 AM local time for the New York time zone. The first offset is obtained right before DST ends, and the second offset is obtained right after DST ends. The dates are created by using the `DateTime.newInstanceGMT` method. This method expects the passed-in date values to be based on the GMT time zone.



```
// Before DST ends
DateTime dtpre = DateTime.newInstanceGMT(2014, 11, 2, 5, 59, 59); //1:59:59AM local EDT
system.debug(tz.getOffset(dtpre)); // -14400000 (= -4 hours = still on DST)

// After DST ends
DateTime dtpost = DateTime.newInstanceGMT(2014, 11, 2, 6, 0, 0); //1:00:00AM local EST
system.debug(tz.getOffset(dtpost)); // -18000000 (= -5 hours = back one hour)
```

## TimeZone Methods

The following are methods for `TimeZone`.

- [`getDisplayName\(\)`](#)  
Returns this time zone's display name.
- [`getID\(\)`](#)  
Returns this time zone's ID.
- [`getOffset\(date\)`](#)  
Returns the time zone offset, in milliseconds, of the specified date to the GMT time zone.
- [`getTimeZone\(timeZoneIdString\)`](#)  
Returns the time zone corresponding to the specified time zone ID.
- [`toString\(\)`](#)  
Returns the string representation of this time zone.

### `getDisplayName()`

Returns this time zone's display name.

#### Signature

```
public String getDisplayName()
```

#### Return Value

Type: [`String`](#)

#### Versioned Behavior Changes

In API version 45.0 and later, `getDisplayName` displays Daylight Savings Time appropriately when daylight savings are in effect. For example, British Summer Time is displayed for Europe/London and Pacific Daylight Time for America/Los\_Angeles.

### `getID()`

Returns this time zone's ID.

#### Signature

```
public String getID()
```

#### Return Value

Type: [`String`](#)

### `getOffset(date)`

Returns the time zone offset, in milliseconds, of the specified date to the GMT time zone.

#### Signature

```
public Integer getOffset(DateTime date)
```



The *date* argument is the date and time to evaluate.

### Return Value

Type: [Integer](#)

### Usage

#### Note

The returned offset is adjusted for daylight saving time if the *date* argument falls within daylight saving time for this time zone.

## getTimeZone(timeZoneIdString)

Returns the time zone corresponding to the specified time zone ID.

### Signature

```
public static TimeZone getTimeZone(String timeZoneIdString)
```

### Parameters

#### *timeZoneIdString*

Type: [String](#)

The time zone values you can use for the *Id* argument are any valid time zone values that the [Java TimeZone class](#) supports.

### Return Value

Type: [TimeZone](#)

### Example

```
TimeZone tz = TimeZone.getTimeZone('America/Los_Angeles');
String tzName = tz.getDisplayName();
System.assert(tzName.equals('GMT-08:00 Pacific Standard Time (America/Los_Angeles)') ||
    tzName.equals('GMT-07:00 Pacific Daylight Time (America/Los_Angeles)'));
```

## toString()

Returns the string representation of this time zone.

### Signature

```
public String toString()
```

### Return Value

Type: [String](#)

### DID THIS ARTICLE SOLVE YOUR ISSUE?

Let us know so we can improve!

[Share your feedback](#)



MuleSoft  
Tableau  
Commerce Cloud  
Lightning Design System  
Einstein  
Quip

Component Library  
APIs  
Trailhead  
Sample Apps  
Podcasts  
AppExchange

Events and Calendar  
Partner Community  
Blog  
Salesforce Admins  
Salesforce Architects

© Copyright 2025 Salesforce, Inc. [All rights reserved.](#) Various trademarks held by their respective owners. Salesforce, Inc.  
Salesforce Tower, 415 Mission Street, 3rd Floor, San Francisco, CA 94105, United States

[Privacy Information](#) [Terms of Service](#) [Legal](#) [Use of Cookies](#) [Trust](#) [Cookie Preferences](#)

[Your Privacy Choices](#) [Responsible Disclosure](#) [Contact](#)