NSTimeZone Class Cluster

Class Cluster Description

From the NSTimeZone class cluster, you can obtain immutable objects that represent time zones. Time zone are split at lines of longitude and are split latitudinally at political boundaries. For example, US/Pacific and Canada/Pacific are two time zone regions. They are roughly the same time zone, but they differ when Daylight Savings Time is in effect.

The abstract class, NSTimeZone, represents knowledge about a geopolitical time zone region (such as “US/Pacific” or “Canada/Pacific”). An abstract public subclass of NSTimeZone, NSTimeZoneDetail, provides the specific details about a time zone: its colloquial abbreviation (such as “PST”), its offset from Greenwich Mean Time (GMT), and whether it is a Daylight Savings Time time zone.

NSTimeZone objects are used in conjunction with NSDate objects. NSDate objects represent dates as *universal time*. Universal time treats a date and time value as identical in, for instance, Redwood City and New York City. There is no provision for locale and, consequently, for time zone. Provision for locale is critical for string representations and other expressions of conventional dates and times. NSTimeZone expands the temporal value of date objects so that they reflect locale information related to time zones.

NSTimeZone objects are referred to simply as *time zone objects*. Because of the nature of class clusters, time zone objects returned by this class are *not* instances of that abstract class (or of NSTimeZoneDetail) but of one of their private subclasses. Although a time zone object’s class is private, its interface is public, as declared by the abstract superclass, NSTimeZone. (See “Class Clusters” in the introduction to the Foundation Kit for more information on class clusters and creating subclasses within a cluster.)

You use the class methods **defaultTimeZone**, **localTimeZone**, **timeZoneWithName:**, **timeZoneWithAbbreviation:**, and **timeZoneForSecondsFromGMT:** to get suitable time zone objects. The instance method **timeZoneDetailForDate:** returns the time-zone detail object for a specific date.

NSTimeZoneDetail doesn’t itself provide any class methods for obtaining time zone objects. It exists mainly as a refinement of NSTimeZone. It does, however, provide accessor methods for obtaining abbreviations and GMT offsets and for determining whether the time zone is a Daylight Savings Time time zone.

Subclassing NSTimeZone or NSTimeZoneDetail is not recommended.

NSTimeZone

Inherits From:	NSObject
Conforms To:	NSCoding, NSCopying NSObject (NSObject)
Declared In:	Foundation/NSDate.h

Class Description

NSTimeZone is an abstract class that defines the behavior of time zone objects. Time zone objects represent geopolitical regions. Consequently, these objects have names for these regions. Time zone objects, in their detail form, also represent a temporal offset, either plus or minus, from Greenwich Mean Time (GMT) and an abbreviation (such as “PST”).

Note: By itself, the NSTimeZone class only *names* a time zone. It does not associate an abbreviation or a temporal offset with a time zone; that’s done by NSTimeZoneDetail. An instance of NSTimeZone, however, “knows” about the set of time-zone detail objects related to it and can provide the one associated with a given date specified in **timeZoneDetailForDate:**.

The system supplies many choices for time zones. The system stores time zone information in the directory **NextLibrary/Frameworks/Foundation.framework/Resources/TimeZoneInfo** on Windows platforms and **/etc/zoneinfo** on Mach platforms. The files in this directory store time zone information. The NSTimeZone class object can provide you with the information in this directory. The class method **timeZoneArray** lists the geopolitical time zone regions to choose from for each longitudinal area. The class method **abbreviationDictionary** selects a subset of geopolitical regions that provide a non-overlapping set of abbreviations from their NSTimeZoneDetail instances. For example, the geopolitical regions US/Pacific and Canada/Pacific are both in the same longitudinal area and both use the abbreviation “PST.” **abbreviationDictionary** limits the choices for an abbreviation so that there is one geopolitical region per abbreviation. Depending on where you are, the following message will return either “US/Pacific” or “Canada/Pacific”.

```
[[NSTimeZone abbreviationDictionary] objectForKey:@"PST"];
```

The **RegionsDictionary** file in the time zone information directory supplies these methods with the information necessary to make such choices.

NSTimeZone provides several class methods to get time zone objects, with or without detail: **timeZoneWithName:**, **timeZoneWithAbbreviation:**, and **timeZoneForSecondsFromGMT:**. The class also permits you to set the default time zone for your locale (**setDefaultTimeZone:**) You can access this default time zone at any time with the **defaultTimeZone** class method, and with the **localTimeZone** class

method, you can get a relative time zone object that decodes itself to become the default time zone for any locale in which it finds itself.

Some NSCalendarDate methods return date objects that are automatically bound to time-zone detail objects. These date objects use the functionality of NSTimeZone to adjust dates for the proper locale. Unless you specify otherwise, objects returned from NSCalendarDate are bound to the default time zone for the current locale. A useful instance method is **timeZoneDetailForDate:**, which returns a time-zone detail object associated with a specific date.

Adopted Protocols

NSCoding	– encodeWithCoder: – initWithCoder:
NSCopying	– copyWithZone:

Method Types

Getting time zones	+ timeZoneWithAbbreviation: + timeZoneWithName: + timeZoneForSecondsFromGMT: – timeZoneDetailForDate:
Getting the default time zone	+ localTimeZone + defaultTimeZone + setDefaultTimeZone:
Getting time zone information	+ timeZoneArray + abbreviationDictionary
Getting information about a specific time zone	– timeZoneDetailArray – timeZoneName

Class Methods

abbreviationDictionary

+ (NSDictionary *)**abbreviationDictionary**

Returns a dictionary holding the mappings of time zone region name to time zone abbreviation. Region names are typically file names in the directory **/NextLibrary/Frameworks/Foundation.framework/Resources/TimeZoneInfo** on Windows platforms

and **/etc/zoneinfo** on Mach platforms. For example, the “US/Pacific” region is mapped to the abbreviations “PST” and “PDT.” Because a region may have more than one abbreviation associated with it, the region is the value and the abbreviation is the key.

More than one region may use the same abbreviation. For example, US/Pacific and Canada/Pacific both use the abbreviation “PST.” **abbreviationDictionary** limits the choices for an abbreviation so that there is one geopolitical region per abbreviation.

See also: **+ timeZoneArray, – timeZoneDetailArray**

defaultTimeZone

+ (NSTimeZoneDetail *)**defaultTimeZone**

Returns the default time zone, set for the current locale.

See also: **+ localTimeZone**, **+ setDefaultTimeZone:**

localTimeZone

+ (NSTimeZone *)**localTimeZone**

Returns the default time zone in the current locale. During decoding, the object replaces itself with the current default time zone. This behavior is particularly useful for NSCalendarDate objects that are archived or sent as Distributed Objects and interpreted in different locales.

See also: **+ defaultTimeZone**, **+ setDefaultTimeZone:**

setDefaultTimeZone:

+ (void)**setDefaultTimeZone***:*(NSTimeZone *)*aTimeZone*

Sets the time zone appropriate for the current locale. There can be only one default time zone, so by setting a new default time zone, you lose the previous one.

See also: **+ defaultTimeZone**, **+ localTimeZone**

timeZoneArray

+ (NSArray *)**timeZoneArray**

Returns an array containing arrays of strings showing all current region names for each longitudinal area. Each array is grouped per latitudinal region (US/Pacific, US/Pacific-New, Canada/Pacific, for instance).

See also: + **abbreviationDictionary**, – **timeZoneDetailArray**

timeZoneForSecondsFromGMT:

+ (NSTimeZone *)**timeZoneForSecondsFromGMT**:(int)*seconds*

Returns the time zone object with *seconds* offset from Greenwich Mean Time. If there is no object matching the offset, this method creates and returns a new time zone object bearing the value *seconds* as a name.

See also: + **timeZoneWithAbbreviation:**, + **timeZoneWithName:**

timeZoneWithAbbreviation:

+ (NSTimeZoneDetail *)**timeZoneWithAbbreviation:**(NSString *)*abbreviation*

Returns the time zone object identified by *abbreviation*. Returns **nil** if there is no match for *abbreviation*.

See also: + **abbreviationDictionary**, + **timeZoneForSecondsFromGMT:**, + **timeZoneWithName:**

timeZoneWithName:

+ (NSTimeZone *)**timeZoneWithName**:(NSString *)*aTimeZoneName*

Returns the time zone object identified by the geopolitical region with the name *aTimeZoneName*. It searches both the **RegionsDictionary** and the time zone information directory for matching names. Returns **nil** if there is no match on the name.

See also: + **timeZoneForSecondsFromGMT:**, + **timeZoneWithAbbreviation:**

Instance Methods

timeZoneDetailArray

– (NSArray *)**timeZoneDetailArray**

Returns an array object that contains NSTimeZoneDetail objects associated with the current time zone object. For example, the US/Pacific time zone has two time-zone detail objects associated with it—one for Pacific Standard Time and one for Pacific Daylight Time.

See also: + **timeZoneArray**, – **timeZoneName**

timeZoneDetailForDate:

– (NSTimeZoneDetail *)**timeZoneDetailForDate:**(NSDate *)*date*

Returns the time-zone detail object that is associated with the *date*. Invoke this method when a region's time zone (that is, its offset value from GMT) varies over the year, as happens between Standard Time and Daylight Savings Time.

See also: – **timeZoneDetailArray**

timeZoneName

– (NSString *)**timeZoneName**

Returns the geopolitical region name that identifies the time zone.

See also: – **timeZoneDetailArray**

NSTimeZoneDetail

Inherits From:	NSTimeZone : NSObject
Conforms To:	NSCoding, NSCopying (NSTimeZone) NSObject (NSObject)
Declared In:	Foundation/NSDate.h

Class Description

NSTimeZone is an abstract class that refines the behavior provided by NSTimeZone. NSTimeZone identifies a geopolitical region with a name (such as “US/Pacific”). NSTimeZoneDetail augments this region name with more specific information appropriate for a particular date within its geopolitical region: an abbreviation, an offset (in seconds) from Greenwich Mean Time (GMT), and an indication of whether this is a Daylight Savings Time time zone. The specificity afforded through NSTimeZoneDetail helps to resolve conflicts between abbreviations and offsets that can arise within regions.

Even though it is an abstract subclass of NSTimeZone, NSTimeZoneDetail does *not* have class methods that create and return time zone objects. See the specification of NSTimeZone for creation methods.

However, NSTimeZoneDetail does have accessor methods that allow you to get the abbreviation and temporal offset of a time zone object, as well as determine whether this is a Daylight Savings Time time zone.

Method Types

Getting time zone information	– isDaylightSavingTimeZone – timeZoneAbbreviation – timeZoneSecondsFromGMT

Instance Methods

isDaylightSavingTimeZone

– (BOOL)**isDaylightSavingTimeZone**

Returns YES if the time-zone detail object is used in the representation of dates during Daylight Savings Time.

timeZoneAbbreviation

– (NSString *)**timeZoneAbbreviation**

Returns the abbreviation for the time zone object, such as “EDT” (Eastern Daylight Time).

timeZoneSecondsFromGMT

– (int)**timeZoneSecondsFromGMT**

Returns the difference in seconds between the receiver and GMT. This offset can be a positive or negative value.