

---

# CalRecurrenceRule Class Reference

[Cocoa](#) > [Apple Applications](#)



2009-03-04



Apple Inc.  
© 2009 Apple Inc.  
All rights reserved.

No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, mechanical, electronic, photocopying, recording, or otherwise, without prior written permission of Apple Inc., with the following exceptions: Any person is hereby authorized to store documentation on a single computer for personal use only and to print copies of documentation for personal use provided that the documentation contains Apple's copyright notice.

The Apple logo is a trademark of Apple Inc.

Use of the "keyboard" Apple logo (Option-Shift-K) for commercial purposes without the prior written consent of Apple may constitute trademark infringement and unfair competition in violation of federal and state laws.

No licenses, express or implied, are granted with respect to any of the technology described in this document. Apple retains all intellectual property rights associated with the technology described in this document. This document is intended to assist application developers to develop applications only for Apple-labeled computers.

Every effort has been made to ensure that the information in this document is accurate. Apple is not responsible for typographical errors.

Apple Inc.  
1 Infinite Loop  
Cupertino, CA 95014  
408-996-1010

Apple, the Apple logo, Cocoa, iCal, Mac, Mac OS, and Objective-C are trademarks of Apple Inc., registered in the United States and other countries.

Simultaneously published in the United States and Canada.

**Even though Apple has reviewed this document, APPLE MAKES NO WARRANTY OR REPRESENTATION, EITHER EXPRESS OR IMPLIED, WITH RESPECT TO THIS DOCUMENT, ITS QUALITY, ACCURACY, MERCHANTABILITY, OR FITNESS FOR A PARTICULAR PURPOSE. AS A RESULT, THIS DOCUMENT IS PROVIDED "AS IS," AND YOU, THE READER, ARE ASSUMING THE ENTIRE RISK AS TO ITS QUALITY AND ACCURACY.**

**IN NO EVENT WILL APPLE BE LIABLE FOR DIRECT, INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES RESULTING FROM ANY**

**DEFECT OR INACCURACY IN THIS DOCUMENT, even if advised of the possibility of such damages.**

**THE WARRANTY AND REMEDIES SET FORTH ABOVE ARE EXCLUSIVE AND IN LIEU OF ALL OTHERS, ORAL OR WRITTEN, EXPRESS OR IMPLIED. No Apple dealer, agent, or employee is authorized to make any modification, extension, or addition to this warranty.**

**Some states do not allow the exclusion or limitation of implied warranties or liability for incidental or consequential damages, so the above limitation or exclusion may not apply to you. This warranty gives you specific legal rights, and you may also have other rights which vary from state to state.**

# Contents

---

## CalRecurrenceRule Class Reference 5

---

Overview	5
Tasks	6
Initializing Recurrence Rules	6
Getting Recurrence Properties	6
Properties	7
daysOfTheMonth	7
daysOfTheWeek	7
firstDayOfTheWeek	8
monthsOfTheYear	8
nthWeekDaysOfTheMonth	8
recurrenceEnd	8
recurrenceInterval	9
recurrenceType	9
Instance Methods	9
initDailyRecurrenceWithInterval:end:	9
initMonthlyRecurrenceWithInterval:end:	10
initMonthlyRecurrenceWithInterval:forDayOfTheWeek:forWeekOfTheMonth:end:	10
initMonthlyRecurrenceWithInterval:forDaysOfTheMonth:end:	11
initWeeklyRecurrenceWithInterval:end:	12
initWeeklyRecurrenceWithInterval:forDaysOfTheWeek:end:	12
initYearlyRecurrenceWithInterval:end:	13
initYearlyRecurrenceWithInterval:forDayOfTheWeek:forWeekOfTheMonth:forMonthsOfTheYear:end:	14
initYearlyRecurrenceWithInterval:forMonthsOfTheYear:end:	14
Constants	15
Recurrence Rule Constants	15
CalRecurrenceType	15

---

## Document Revision History 17

---

## Index 19

---



# CalRecurrenceRule Class Reference

---

<b>Inherits from</b>	NSObject
<b>Conforms to</b>	NSCopying NSObject (NSObject)
<b>Framework</b>	/System/Library/Frameworks/CalendarStore.framework
<b>Availability</b>	Available in Mac OS X v10.5 and later.
<b>Companion guide</b>	Calendar Store Programming Guide
<b>Declared in</b>	CalRecurrenceRule.h

## Overview

The `CalRecurrenceRule` class is used to describe the recurrence pattern for a recurring event. The recurrence rules that you can create are restricted to the recurrence patterns that you have access to in iCal. It is not possible to directly modify a `CalRecurrenceRule` object or any of its properties. Properties defined in this class are read-only.

Use one of the `init...` methods to create the desired recurrence rule. These `init...` methods set the receiver's properties. For example, if you invoke one of the `initMonthly...` methods, the `recurrenceType` property is set to `CalRecurrenceMonthly` (page 16). All recurrence rules have a value set for the `recurrenceType` and `recurrenceInterval` properties. If the `recurrenceEnd` property is `nil`, an event repeats forever. The rest of the properties defined in this class have values depending on the type of recurrence rule and initializer method used to create the recurring event pattern. If a property is not needed to describe a recurrence pattern, then its value is `nil`.

Every initializer method has an interval and a `CalRecurrenceEnd` object as arguments. The interval is a value greater than 0 that is the number of units between occurrences of an event. For example, if the `recurrenceType` property is `CalRecurrenceMonthly` (page 16), then the unit of measurement is one month. If the interval is 1, the event occurs every month, but if it is 2, it occurs every other month. The `CalRecurrenceEnd` object specifies when a recurring event ends. This can be specified using a counter or an end date (see *CalRecurrenceEnd Class Reference* for details). Pass `nil` as the recurrence end argument if the event never ends.

In addition, there are some initializers for more sophisticated recurrence patterns. However, these initializers are restricted to patterns that can be represented in iCal and include custom repeating patterns. For example, you can specify a pattern where an event occurs on the Friday of the 2nd week of every month, or the Monday of the 3rd week of every fourth month of a year. You can create custom patterns for weekly, monthly, and yearly recurrence rules.

After you create a recurrence rule, use the `recurrenceRule` `CalEvent` property to set the recurrence rule for an event. Use the `saveEvent:span:error:` method to save your changes to the iCal database.

## Tasks

### Initializing Recurrence Rules

- [initDailyRecurrenceWithInterval:end:](#) (page 9)  
Initializes and returns a daily recurrence rule with the specified interval and ending rule.
- [initWeeklyRecurrenceWithInterval:end:](#) (page 12)  
Initializes and returns a weekly recurrence rule with the specified interval and ending rule.
- [initWeeklyRecurrenceWithInterval:forDaysOfTheWeek:end:](#) (page 12)  
Initializes and returns a weekly recurrence rule with the specified interval, ending rule, and specific days of the week.
- [initMonthlyRecurrenceWithInterval:end:](#) (page 10)  
Initializes and returns a monthly recurrence rule with the specified interval and end rule.
- [initMonthlyRecurrenceWithInterval:forDaysOfTheMonth:end:](#) (page 11)  
Initializes and returns a monthly recurrence rule that represents an event that occurs more than once a month in a monthly pattern. The pattern repeats at the specified interval.
- [initMonthlyRecurrenceWithInterval:forDayOfTheWeek:forWeekOfTheMonth:end:](#) (page 10)  
Initializes and returns a monthly recurrence rule that represents an event that occurs on a specific day of the week and week of the month pattern. The pattern repeats at the specified monthly interval.
- [initYearlyRecurrenceWithInterval:end:](#) (page 13)  
Initializes and returns a yearly recurrence rule with the specified yearly interval and end rule.
- [initYearlyRecurrenceWithInterval:forMonthsOfTheYear:end:](#) (page 14)  
Initializes and returns a yearly recurrence rule representing an event that occurs multiple months within a year at a specified yearly interval.
- [initYearlyRecurrenceWithInterval:forDayOfTheWeek:forWeekOfTheMonth:forMonthsOfTheYear:end:](#) (page 14)  
Initializes and returns a yearly recurrence rule that represents an event that has a weekly and monthly pattern that repeats at the specified yearly interval.

### Getting Recurrence Properties

[firstDayOfTheWeek](#) (page 8) *property*

An integer value of 0 or 1 to 7 where 0 indicates no value is set, and 1 to 7 indicates the first day of the week where Sunday is represented by 1. (read-only)

[recurrenceEnd](#) (page 8) *property*

An object that describes how a recurring event ends by specifying an end date or a counter. (read-only)

[recurrenceType](#) (page 9) *property*

The unit of time between intervals. See [CalRecurrenceType](#) (page 15) for possible values. (read-only)



**Declared In**

CalRecurrenceRule.h

**firstDayOfTheWeek**

An integer value of 0 or 1 to 7 where 0 indicates no value is set, and 1 to 7 indicates the first day of the week where Sunday is represented by 1. (read-only)

```
@property(readonly) NSInteger firstDayOfTheWeek
```

**Discussion**

This property only affects the way the recurrence is expanded for weekly recurrence rules with an interval greater than 1. For those types of recurrence rules, Calendar Store sets this property to 2 (Monday). For all other recurrence rules, this property defaults to 0.

**Availability**

Available in Mac OS X v10.5 and later.

**Declared In**

CalRecurrenceRule.h

**monthsOfTheYear**

If the recurrenceType property is CalRecurrenceYearly, an array of the monthsâinteger values ranging from 1 to 12 representing the month of a yearâin which the event occurs; otherwise, nil. (read-only)

```
@property(readonly) NSArray *monthsOfTheYear
```

**Availability**

Available in Mac OS X v10.5 and later.

**Declared In**

CalRecurrenceRule.h

**nthWeekDaysOfTheMonth**

If the recurrenceType property is CalRecurrenceMonthly or CalRecurrenceYearly, an array of CalNthWeekDay objects representing the days within the weeks of a month; otherwise, nil. (read-only)

```
@property(readonly) NSArray *nthWeekDaysOfTheMonth
```

**Availability**

Available in Mac OS X v10.5 and later.

**Declared In**

CalRecurrenceRule.h

**recurrenceEnd**

An object that describes how a recurring event ends by specifying an end date or a counter. (read-only)



@property(readonly) CalRecurrenceEnd \*recurrenceEnd

#### Discussion

If `nil`, the event never ends.

#### Availability

Available in Mac OS X v10.5 and later.

#### Declared In

CalRecurrenceRule.h

## recurrenceInterval

The number of intervals between the specified pattern of a recurring event. The actual time between a pattern depends on the value of the `recurrenceType` property. (read-only)

@property(readonly) NSUInteger recurrenceInterval

#### Discussion

For example, if the `recurrenceType` property is `CalRecurrenceMonthly` and the `recurrenceInterval` property is 1, then the event occurs every month. If the `recurrenceInterval` property is 2, then the event occurs every other month.

#### Availability

Available in Mac OS X v10.5 and later.

#### Declared In

CalRecurrenceRule.h

## recurrenceType

The unit of time between intervals. See [CalRecurrenceType](#) (page 15) for possible values. (read-only)

@property(readonly) CalRecurrenceType recurrenceType

#### Availability

Available in Mac OS X v10.5 and later.

#### Declared In

CalRecurrenceRule.h

## Instance Methods

### initWithDailyRecurrenceWithInterval:end:

Initializes and returns a daily recurrence rule with the specified interval and ending rule.

```
- (id)initWithDailyRecurrenceWithInterval:(NSUInteger)interval end:(CalRecurrenceEnd *)end
```

**Parameters***interval*

The interval in days between occurrences of a recurring event. Must be a value greater than 0.

*end*

Describes how a recurring event ends. Pass `nil` if the event never ends.

**Return Value**

An initialized daily recurrence rule objectâ with `recurrenceType` property set to `CalRecurrenceDaily`. Returns `nil` if *interval* is 0 or a negative number.

**Availability**

Available in Mac OS X v10.5 and later.

**Declared In**

`CalRecurrenceRule.h`

**initMonthlyRecurrenceWithInterval:end:**

Initializes and returns a monthly recurrence rule with the specified interval and end rule.

```
- (id)initMonthlyRecurrenceWithInterval:(NSUInteger)interval end:(CalRecurrenceEnd*)end
```

**Parameters***interval*

The interval in months between occurrences of a recurring event. Must be a value greater than 0.

*end*

Describes how a recurring event ends. Pass `nil` if the event never ends.

**Return Value**

A newly initialized monthly recurrence rule objectâ with `recurrenceType` property set to `CalRecurrenceMonthly`. Returns `nil` if *interval* is 0 or a negative number.

**Discussion**

The returned recurrence rule defaults to the first day of the month.

**Availability**

Available in Mac OS X v10.5 and later.

**See Also**

- [initMonthlyRecurrenceWithInterval:forDaysOfTheMonth:end:](#) (page 11)
- [initMonthlyRecurrenceWithInterval:forDayOfWeek:forWeekOfMonth:end:](#) (page 10)

**Declared In**

`CalRecurrenceRule.h`

**initMonthlyRecurrenceWithInterval:forDayOfTheWeek:forWeekOfMonth:end:**

Initializes and returns a monthly recurrence rule that represents an event that occurs on a specific day of the week and week of the month patternâ pattern repeats at the specified monthly interval.

```
- (id)initMonthlyRecurrenceWithInterval:(NSUInteger)interval
  forDayOfTheWeek:(NSUInteger)weekDay forWeekOfTheMonth:(NSUInteger)monthWeek
  end:(CalRecurrenceEnd *)end
```

**Parameters***interval*

The interval in months between occurrences of a recurring event. Must be a value greater than 0.

*weekDay*

The day of the week that the event occurs. An integer value ranging from 1 to 7 representing the day of the week where Sunday is equal to 1.

*monthWeek*

The week of the month that the event occurs. An integer value that is 1, 2, 3, 4, or -1 representing the week of a month where -1 is the last week of the month.

*end*

Describes how a recurring event ends. Pass `nil` if the event never ends.

**Return Value**

A newly initialized monthly recurrence rule object with `recurrenceType` property set to `CalRecurrenceMonthly`. Returns `nil` if *interval* is 0 or a negative number.

**Discussion**

For example, use this method to create a recurrence rule that represents an event that occurs on the first Monday of every month.

**Availability**

Available in Mac OS X v10.5 and later.

**See Also**

- [initMonthlyRecurrenceWithInterval:end:](#) (page 10)
- [initMonthlyRecurrenceWithInterval:forDaysOfTheMonth:end:](#) (page 11)

**Declared In**

CalRecurrenceRule.h

**initMonthlyRecurrenceWithInterval:forDaysOfTheMonth:end:**

Initializes and returns a monthly recurrence rule that represents an event that occurs more than once a month in a monthly pattern that repeats at the specified interval.

```
- (id)initMonthlyRecurrenceWithInterval:(NSUInteger)interval
  forDaysOfTheMonth:(NSArray *)monthDays end:(CalRecurrenceEnd *)end
```

**Parameters***interval*

The interval in months between occurrences of a recurring event. Must be a value greater than 0.

*monthDays*

An array of numbers specifying the days of the month pattern that repeats. The integer values can range from 1 to 31 representing the days of the month.

*end*

Describes how a recurring event ends. Pass `nil` if the event never ends.

**Return Value**

A newly initialized monthly recurrence rule objectâ `withRecurrenceType` property set to `CalRecurrenceMonthly`. Returns `nil` if `interval` is 0 or a negative number.

**Availability**

Available in Mac OS X v10.5 and later.

**See Also**

- [initMonthlyRecurrenceWithInterval:end:](#) (page 10)
- [initMonthlyRecurrenceWithInterval:forDayOfWeek:forWeekOfMonth:end:](#) (page 10)

**Declared In**

`CalRecurrenceRule.h`

**initWeeklyRecurrenceWithInterval:end:**

Initializes and returns a weekly recurrence rule with the specified interval and ending rule.

```
- (id)initWeeklyRecurrenceWithInterval:(NSUInteger)interval end:(CalRecurrenceEnd *)end
```

**Parameters**

*interval*

The interval in weeks between occurrences of a recurring event. Must be a value greater than 0.

*end*

Describes how a recurring event ends. Pass `nil` if the event never ends.

**Return Value**

An initialized weekly recurrence rule objectâ `withRecurrenceType` property set to `CalRecurrenceWeekly`. Returns `nil` if `interval` is 0 or a negative number.

**Availability**

Available in Mac OS X v10.5 and later.

**See Also**

- [initWeeklyRecurrenceWithInterval:forDaysOfTheWeek:end:](#) (page 12)

**Declared In**

`CalRecurrenceRule.h`

**initWeeklyRecurrenceWithInterval:forDaysOfTheWeek:end:**

Initializes and returns a weekly recurrence rule with the specified interval, ending rule, and specific days of the week.

```
- (id)initWeeklyRecurrenceWithInterval:(NSUInteger)interval forDaysOfTheWeek:(NSArray *)days end:(CalRecurrenceEnd *)end
```

**Parameters**

*interval*

The interval in weeks between occurrences of a recurring event. Must be a value greater than 0.

*days*

An array of numbers specifying the weekly pattern that repeats. The integer values can range from 1-7 where Sunday is equal to 1.

*end*

Describes how a recurring event ends. Pass `nil` if the event never ends.

#### Return Value

A newly initialized weekly recurrence rule objectâ with `recurrenceType` property set to `CalRecurrenceWeekly`. Returns `nil` if *interval* is 0 or a negative number, or any of the other arguments are invalid.

#### Discussion

This initializer allows you to specify a weekly pattern that repeats. For example, if *days* contains the numbers 2 and 4, and *interval* is 2, then an event occurs biweekly on Mondays and Wednesdays.

#### Availability

Available in Mac OS X v10.5 and later.

#### See Also

- [initWithWeeklyRecurrenceWithInterval:end:](#) (page 12)

#### Declared In

`CalRecurrenceRule.h`

## **initWithYearlyRecurrenceWithInterval:end:**

Initializes and returns a yearly recurrence rule with the specified yearly interval and end rule.

```
- (id)initWithYearlyRecurrenceWithInterval:(NSUInteger)interval end:(CalRecurrenceEnd*)end
```

#### Parameters

*interval*

The interval in years between occurrences of a recurring event. Must be a value greater than 0.

*end*

Describes how a recurring event ends. Pass `nil` if the event never ends.

#### Return Value

A newly initialized yearly recurrence rule objectâ with `recurrenceType` property set to `CalRecurrenceYearly`. Returns `nil` if *interval* is 0 or a negative number.

#### Availability

Available in Mac OS X v10.5 and later.

#### See Also

- [initWithYearlyRecurrenceWithInterval:forMonthsOfTheYear:end:](#) (page 14)

- [initWithYearlyRecurrenceWithInterval:forDayOfWeek:forWeekOfMonth:forMonthsOfTheYear:end:](#) (page 14)

#### Declared In

`CalRecurrenceRule.h`

## initWithYearlyRecurrenceWithInterval:forDayOfTheWeek:forWeekOfTheMonth:forMonthsOfTheYear:end:

Initializes and returns a yearly recurrence rule that represents an event that has a weekly and monthly pattern that repeats at the specified yearly interval.

```
- (id)initWithYearlyRecurrenceWithInterval:(NSUInteger)interval
    forDayOfTheWeek:(NSUInteger)weekDay forWeekOfTheMonth:(NSInteger)monthWeek
    forMonthsOfTheYear:(NSArray *)months end:(CalRecurrenceEnd *)end
```

### Parameters

*interval*

The interval in years between this pattern of a recurring event. Must be a value greater than 0.

*weekDay*

The day of the week that the event occurs. An integer value ranging from 1-7 representing the day of the week where Sunday is equal to 1.

*monthWeek*

The week of the month that the event occurs. An integer value that is either 1, 2, 3, 4, or -1 representing the week of a month where -1 is the last week of the month.

*months*

The months of the year that the event occurs. An array of numbers where each number is an integer value ranging from 1-12 representing the month of a year.

*end*

Describes how a recurring event ends. Pass `nil` if the event never ends.

### Return Value

A newly initialized yearly recurrence rule object. The `recurrenceType` property is set to `CalRecurrenceYearly`. Returns `nil` if *interval* is 0 or a negative number.

### Discussion

This method allows you to create a pattern for an entire year that repeats. The event occurs on the same day of the week, in the same week of a month, and possibly more than one month of a year. For example, use this method to represent an event that occurs every year on the first Friday of the sixth and twelfth months.

### Availability

Available in Mac OS X v10.5 and later.

### See Also

- [initWithYearlyRecurrenceWithInterval:end:](#) (page 13)
- [initWithYearlyRecurrenceWithInterval:forMonthsOfTheYear:end:](#) (page 14)

### Declared In

CalRecurrenceRule.h

## initWithYearlyRecurrenceWithInterval:forMonthsOfTheYear:end:

Initializes and returns a yearly recurrence rule representing an event that occurs multiple months within a year at a specified yearly interval.

```
- (id)initWithYearlyRecurrenceWithInterval:(NSUInteger)interval
    forMonthsOfTheYear:(NSArray *)months end:(CalRecurrenceEnd *)end
```

**Parameters***interval*

The interval in years between occurrences of a recurring event. Must be a value greater than 0.

*end*

Describes how a recurring event ends. Pass `nil` if the event never ends.

**Return Value**

A newly initialized yearly recurrence rule object `CR` with `recurrenceType` property set to `CalRecurrenceYearly`. Returns `nil` if *interval* is 0 or a negative number.

**Discussion**

For example, use this method if you want to represent an event that occurs every year on the first day of the first and seventh month of the year. If you need to specify the day of the month that the event occurs, use the [initYearlyRecurrenceWithInterval:forDayOfWeek:forWeekOfMonth:forMonthsOfTheYear:end:](#) (page 14) method.

**Availability**

Available in Mac OS X v10.5 and later.

**See Also**

- [initYearlyRecurrenceWithInterval:end:](#) (page 13)
- [initYearlyRecurrenceWithInterval:forDayOfWeek:forWeekOfMonth:forMonthsOfTheYear:end:](#) (page 14)

**Declared In**

`CalRecurrenceRule.h`

## Constants

### Recurrence Rule Constants

Constants used by this class.

```
extern NSInteger const CalDefaultRecurrenceInterval;
```

**Constants**

`CalDefaultRecurrenceInterval`

The default recurrence interval. The default value of 1 indicates that the event repeats daily, weekly, monthly, or yearly depending on the recurrence type.

Available in Mac OS X v10.5 and later.

Declared in `CalRecurrenceRule.h`.

**Declared In**

`CalendarStore/CalRecurrenceRule.h`

### CalRecurrenceType

The unit of measurement between occurrences of a recurring event.

```
typedef enum {  
    CalRecurrenceDaily,  
    CalRecurrenceWeekly,  
    CalRecurrenceMonthly,  
    CalRecurrenceYearly  
} CalRecurrenceType;
```

### Constants

CalRecurrenceDaily

Indicates a daily unit of measurement.

Available in Mac OS X v10.5 and later.

Declared in CalRecurrenceRule.h.

CalRecurrenceWeekly

Indicates a weekly unit of measurement.

Available in Mac OS X v10.5 and later.

Declared in CalRecurrenceRule.h.

CalRecurrenceMonthly

Indicates a monthly unit of measurement.

Available in Mac OS X v10.5 and later.

Declared in CalRecurrenceRule.h.

CalRecurrenceYearly

Indicates a yearly unit of measurement.

Available in Mac OS X v10.5 and later.

Declared in CalRecurrenceRule.h.

### Discussion

The four types of units are days, weeks, months, and years which correspond to the types of recurrence rules. For example, if the recurrence rule is CalRecurrenceWeekly, then an interval of 1 indicates the event repeats weekly. If the interval is 2, the event repeats biweekly.

### Declared In

CalendarStore/CalRecurrenceRule.h



# Document Revision History

---

This table describes the changes to *CalRecurrenceRule Class Reference*.

Date	Notes
2009-03-04	Minor edits throughout.
2007-07-08	New document that describes the class used to initialize and get recurrence rules and properties.

## REVISION HISTORY

### Document Revision History

# Index

---

## C

---

`CalDefaultRecurrenceInterval` **constant** [15](#)  
`CalRecurrenceDaily` **constant** [16](#)  
`CalRecurrenceMonthly` **constant** [16](#)  
`CalRecurrenceType` **15**  
`CalRecurrenceWeekly` **constant** [16](#)  
`CalRecurrenceYearly` **constant** [16](#)

## D

---

`daysOfTheMonth` **instance property** [7](#)  
`daysOfTheWeek` **instance property** [7](#)

## F

---

`firstDayOfTheWeek` **instance property** [8](#)

## I

---

`initDailyRecurrenceWithInterval:end:` **instance method** [9](#)  
`initMonthlyRecurrenceWithInterval:end:` **instance method** [10](#)  
`initMonthlyRecurrenceWithInterval:forDayOfTheWeek:forWeekOfTheMonth:end:` **instance method** [10](#)  
`initMonthlyRecurrenceWithInterval:forDaysOfTheMonth:end:` **instance method** [11](#)  
`initWeeklyRecurrenceWithInterval:end:` **instance method** [12](#)  
`initWeeklyRecurrenceWithInterval:forDaysOfTheWeek:end:` **instance method** [12](#)  
`initYearlyRecurrenceWithInterval:end:` **instance method** [13](#)

`initYearlyRecurrenceWithInterval:forDayOfTheWeek:forWeekOfTheMonth:forMonthsOfTheYear:end:` **instance method** [14](#)  
`initYearlyRecurrenceWithInterval:forMonthsOfTheYear:end:` **instance method** [14](#)

## M

---

`monthsOfTheYear` **instance property** [8](#)

## N

---

`nthWeekDaysOfTheMonth` **instance property** [8](#)

## R

---

**Recurrence Rule Constants** [15](#)  
`recurrenceEnd` **instance property** [8](#)  
`recurrenceInterval` **instance property** [9](#)  
`recurrenceType` **instance property** [9](#)