
Event Kit Framework Reference

Data Management



2010-03-25



Apple Inc.
© 2010 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, iCal, iPhone, and Objective-C are trademarks of Apple Inc., registered in the United States and other countries.

iOS is a trademark or registered trademark of Cisco in the U.S. and other countries and is used under license.

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

Introduction **Introduction** 5

Part I **Classes** 7

Chapter 1 **EKAlarm Class Reference** 9

Overview 9
Tasks 9
Properties 10
Class Methods 10

Chapter 2 **EKCalendar Class Reference** 13

Overview 13
Tasks 13
Properties 13
Constants 15

Chapter 3 **EKEvent Class Reference** 19

Overview 19
Tasks 19
Properties 21
Class Methods 25
Instance Methods 26
Constants 27

Chapter 4 **EKEventStore Class Reference** 29

Overview 29
Tasks 29
Properties 30
Instance Methods 31
Constants 34
Notifications 35

Chapter 5 **EKParticipant Class Reference** 37

Overview 37
Tasks 37
Properties 38

Instance Methods 39
Constants 40

Chapter 6 **EKRecurrenceDayOfWeek Class Reference 43**

Overview 43
Tasks 43
Properties 44
Class Methods 44

Chapter 7 **EKRecurrenceEnd Class Reference 47**

Overview 47
Tasks 47
Properties 48
Class Methods 48

Chapter 8 **EKRecurrenceRule Class Reference 51**

Overview 51
Tasks 51
Properties 52
Instance Methods 56
Constants 58

Part II **Constants 61**

Chapter 9 **Event Kit Constants Reference 63**

Overview 63
Constants 63

Document Revision History 67

Introduction

Framework	/System/Library/Frameworks/EventKit.framework
Header file directories	EventKit.framework/Headers
Companion guide	Event Kit Programming Guide
Declared in	EKAlarm.h EKCalendar.h EKError.h EKEvent.h EKEventStore.h EKParticipant.h EKRecurrenceRule.h

The Event Kit framework provides classes for accessing and modifying calendar event information.

Classes

EKAlarm Class Reference

Inherits from	NSObject
Conforms to	NSCopying NSObject (NSObject)
Framework	/System/Library/Frameworks/EventKit.framework
Availability	Available in iOS 4.0 and later.
Declared in	EKAlarm.h
Companion guide	Event Kit Programming Guide

Overview

An `EKAlarm` object represents an alarm in Event Kit. Use the `alarmWithAbsoluteDate:` (page 10) and `alarmWithRelativeOffset:` (page 11) class methods to create an alarm and use the properties to set information about an alarm.

Tasks

Creating an Alarm

- + `alarmWithAbsoluteDate:` (page 10)
Creates and returns an alarm with an absolute date.
- + `alarmWithRelativeOffset:` (page 11)
Creates and returns an alarm with a relative offset.

Accessing Alarm Properties

- `absoluteDate` (page 10) *property*
The absolute date for the alarm.
- `relativeOffset` (page 10) *property*
The offset from the start of an event, at which the alarm fires.

Properties

For more about Objective-C properties, see “Properties” in *The Objective-C Programming Language*.

absoluteDate

The absolute date for the alarm.

```
@property(copy) NSDate *absoluteDate
```

Discussion

If you set this property for a relative offset alarm, it loses the relative offset and becomes an absolute alarm.

Availability

Available in iOS 4.0 and later.

Declared In

EKAlarm.h

relativeOffset

The offset from the start of an event, at which the alarm fires.

```
@property NSTimeInterval relativeOffset
```

Discussion

If you set this value for an absolute alarm, it loses its absolute date and becomes a relative offset alarm.

Availability

Available in iOS 4.0 and later.

Declared In

EKAlarm.h

Class Methods

alarmWithAbsoluteDate:

Creates and returns an alarm with an absolute date.

```
+ (EKAlarm *)alarmWithAbsoluteDate:(NSDate *)date
```

Parameters

date

The date for the alarm.

Return Value

The created alarm.

Availability

Available in iOS 4.0 and later.

Declared In

EKAlarm.h

alarmWithRelativeOffset:

Creates and returns an alarm with a relative offset.

```
+ (EKAlarm *)alarmWithRelativeOffset:(NSTimeInterval)offset
```

Parameters

offset

The offset from start of an event, at which the alarm fires. Should be a negative value or 0.

Return Value

The created alarm.

Discussion

If *offset* is greater than zero, it is pinned to zero.

Availability

Available in iOS 4.0 and later.

Declared In

EKAlarm.h

EKCalendar Class Reference

Inherits from	NSObject
Conforms to	NSObject (NSObject)
Framework	/System/Library/Frameworks/EventKit.framework
Availability	Available in iOS 4.0 and later.
Declared in	EKCalendar.h

Overview

An `EKCalendar` object represents a calendar in Event Kit. Use the properties in this class to get attributes about a calendar, such as its title and type.

Tasks

Accessing Calendar Properties

- [allowsContentModifications](#) (page 14) *property*
Indicates whether you can add, remove, and modify items in the calendar. (read-only)
- [CGColor](#) (page 14) *property*
The calendar's color. (read-only)
- [supportedEventAvailabilities](#) (page 14) *property*
The event availability settings supported by this calendar, as indicated by a bitmask. (read-only)
- [title](#) (page 14) *property*
The calendar's title. (read-only)
- [type](#) (page 15) *property*
The calendar's type. (read-only)

Properties

For more about Objective-C properties, see “Properties” in *The Objective-C Programming Language*.

allowsContentModifications

Indicates whether you can add, remove, and modify items in the calendar. (read-only)

```
@property(nonatomic, readonly) BOOL allowsContentModifications
```

Availability

Available in iOS 4.0 and later.

Declared In

EKCalendar.h

CGColor

The calendar’s color. (read-only)

```
@property(nonatomic, readonly) CGColorRef CGColor
```

Availability

Available in iOS 4.0 and later.

Declared In

EKCalendar.h

supportedEventAvailabilities

The event availability settings supported by this calendar, as indicated by a bitmask. (read-only)

```
@property(nonatomic, readonly) EKCalendarEventAvailabilityMask
supportedEventAvailabilities
```

Discussion

If the calendar doesn’t support event availability settings, this value is [EKCalendarEventAvailabilityNone](#) (page 16).

Availability

Available in iOS 4.0 and later.

See Also

“[EKCalendarEventAvailability](#)” (page 16)

Declared In

EKCalendar.h

title

The calendar’s title. (read-only)

```
@property(nonatomic, readonly) NSString *title
```

Availability

Available in iOS 4.0 and later.

Declared In

EKCalendar.h

type

The calendar's type. (read-only)

`@property(nonatomic, readonly) EKCalendarType type`**Availability**

Available in iOS 4.0 and later.

See Also[“EKCalendarType”](#) (page 15)**Declared In**

EKCalendar.h

Constants

EKCalendarType

Represents possible calendar types.

```
typedef enum {
    EKCalendarTypeLocal,
    EKCalendarTypeCalDAV,
    EKCalendarTypeExchange,
    EKCalendarTypeSubscription,
    EKCalendarTypeBirthday,
} EKCalendarType;
```

Constants

EKCalendarTypeLocal

Represents a local calendar.

Available in iOS 4.0 and later.

Declared in EKCalendar.h.

EKCalendarTypeCalDAV

Represents a CalDAV calendar.

Available in iOS 4.0 and later.

Declared in EKCalendar.h.

EKCalendarTypeExchange

Represents an Exchange calendar.

Available in iOS 4.0 and later.

Declared in EKCalendar.h.

`EKCalendarTypeSubscription`
 Represents a subscribed calendar.
 Available in iOS 4.0 and later.
 Declared in `EKCalendar.h`.

`EKCalendarTypeBirthday`
 Represents a birthday calendar.
 Available in iOS 4.0 and later.
 Declared in `EKCalendar.h`.

EKCalendarEventAvailabilityMask

A bitmask indicating the event availability settings that the calendar can support.

```
typedef NSUInteger EKCalendarEventAvailabilityMask;
```

Availability
 Available in iOS 4.0 and later.

Declared In
`EKCalendar.h`

EKCalendarEventAvailability

Represents event availability settings that the calendar can support.

```
enum {
    EKCalendarEventAvailabilityNone           = 0,
    EKCalendarEventAvailabilityBusy          = (1 << 0),
    EKCalendarEventAvailabilityFree         = (1 << 1),
    EKCalendarEventAvailabilityTentative    = (1 << 2),
    EKCalendarEventAvailabilityUnavailable = (1 << 3),
};
```

Constants

`EKCalendarEventAvailabilityNone`
 The calendar does not support event availability settings.
 Available in iOS 4.0 and later.
 Declared in `EKCalendar.h`.

`EKCalendarEventAvailabilityBusy`
 The calendar supports the busy event availability setting.
 Available in iOS 4.0 and later.
 Declared in `EKCalendar.h`.

`EKCalendarEventAvailabilityFree`
 The calendar supports the free event availability setting.
 Available in iOS 4.0 and later.
 Declared in `EKCalendar.h`.

`EKCalendarEventAvailabilityTentative`

The calendar supports the tentative event availability setting.

Available in iOS 4.0 and later.

Declared in `EKCalendar.h`.

`EKCalendarEventAvailabilityUnavailable`

The calendar supports the unavailable event availability setting.

Available in iOS 4.0 and later.

Declared in `EKCalendar.h`.

EKEvent Class Reference

Inherits from	NSObject
Conforms to	NSObject (NSObject)
Framework	/System/Library/Frameworks/EventKit.framework
Availability	Available in iOS 4.0 and later.
Declared in	EKEvent.h
Companion guide	Event Kit Programming Guide

Overview

An `EKEvent` object represents an event added to a calendar in Event Kit.

Use the [eventWithEventStore:](#) (page 25) method to create a new event. Use the properties in the class to get and modify certain information about an event.

You can add and remove alarms from an event with the [addAlarm:](#) (page 26) and [removeAlarm:](#) (page 27) methods.

Tasks

Creating an Event

- + [eventWithEventStore:](#) (page 25)
Creates and returns a new event belonging to a specified event store.

Accessing Event Properties

- [alarms](#) (page 21) *property*
The alarms associated with the event, as an array of `EKAlarm` objects.
- [allDay](#) (page 21) *property*
Indicates whether the event is an all-day event.
- [attendees](#) (page 21) *property*
The attendees associated with the event, as an array of `EKParticipant` objects. (read-only)

- [availability](#) (page 21) *property*
The availability setting for the event.
- [calendar](#) (page 22) *property*
The calendar for the event.
- [endDate](#) (page 22) *property*
The end date for the event.
- [eventIdentifier](#) (page 22) *property*
A unique identifier for the event. (read-only)
- [isDetached](#) (page 23) *property*
Indicates whether an event is a detached instance of a repeating event. (read-only)
- [lastModifiedDate](#) (page 23) *property*
The date the event was last modified. (read-only)
- [location](#) (page 23) *property*
The location associated with the event.
- [notes](#) (page 24) *property*
The notes associated with the event.
- [organizer](#) (page 24) *property*
The organizer associated with the event. (read-only)
- [recurrenceRule](#) (page 24) *property*
The recurrence rule associated with the event.
- [startDate](#) (page 24) *property*
The start date for the event.
- [status](#) (page 25) *property*
The event's status. (read-only)
- [title](#) (page 25) *property*
The title for the event.

Adding and Removing Alarms

- [addAlarm:](#) (page 26)
Adds an alarm to the receiver.
- [removeAlarm:](#) (page 27)
Removes an alarm from the event.

Comparing Events

- [compareStartDateWithEvent:](#) (page 26)
Compares the start date of the receiving event with the start date of another event.

Refreshing Event Data

- [refresh](#) (page 26)
Updates the event's data with the current information in the Calendar database.

Properties

For more about Objective-C properties, see “Properties” in *The Objective-C Programming Language*.

alarms

The alarms associated with the event, as an array of `EKAlarm` objects.

```
@property(nonatomic, copy) NSArray *alarms
```

Discussion

This property is `nil` if the event has no alarms.

Availability

Available in iOS 4.0 and later.

Declared In

`EKEvent.h`

allDay

Indicates whether the event is an all-day event.

```
@property(nonatomic, getter=isAllDay) BOOL allDay
```

Availability

Available in iOS 4.0 and later.

Declared In

`EKEvent.h`

attendees

The attendees associated with the event, as an array of `EKParticipant` objects. (read-only)

```
@property(nonatomic, readonly) NSArray *attendees
```

Discussion

This property is `nil` if the event has no attendees.

Availability

Available in iOS 4.0 and later.

Declared In

`EKEvent.h`

availability

The availability setting for the event.

```
@property(n nonatomic) EKEventAvailability availability
```

Discussion

This setting is used by CalDAV and Exchange servers to indicate how the event should be treated for scheduling purposes.

If the event's calendar does not support availability settings, this property's value is [EKEventAvailabilityNotSupported](#) (page 27).

Availability

Available in iOS 4.0 and later.

See Also

“[EKEventAvailability](#)” (page 27)

Declared In

EKEvent.h

calendar

The calendar for the event.

```
@property(n nonatomic, retain) EKCalendar *calendar
```

Availability

Available in iOS 4.0 and later.

Declared In

EKEvent.h

endDate

The end date for the event.

```
@property(n nonatomic, copy) NSDate *endDate
```

Availability

Available in iOS 4.0 and later.

Declared In

EKEvent.h

eventIdentifier

A unique identifier for the event. (read-only)

```
@property(n nonatomic, readonly) NSString *eventIdentifier
```

Discussion

You can use this identifier to look up an event with the EKEventStore method [eventWithIdentifier:](#) (page 32).

If the calendar of an event changes, its identifier most likely changes as well.

Availability

Available in iOS 4.0 and later.

Declared In

EKEvent.h

isDetached

Indicates whether an event is a detached instance of a repeating event. (read-only)

```
@property(nonatomic, readonly) BOOL isDetached
```

Discussion

This value is YES if and only if the event is part of a repeating event and one or more of its attributes have been modified from the repeating event's default attributes.

Availability

Available in iOS 4.0 and later.

Declared In

EKEvent.h

lastModifiedDate

The date the event was last modified. (read-only)

```
@property(nonatomic, readonly) NSDate *lastModifiedDate
```

Availability

Available in iOS 4.0 and later.

Declared In

EKEvent.h

location

The location associated with the event.

```
@property(nonatomic, copy) NSString *location
```

Discussion

This property is nil if the event has no location.

Availability

Available in iOS 4.0 and later.

Declared In

EKEvent.h

notes

The notes associated with the event.

```
@property(n nonatomic, copy) NSString *notes
```

Availability

Available in iOS 4.0 and later.

Declared In

EKEvent.h

organizer

The organizer associated with the event. (read-only)

```
@property(n nonatomic, readonly) EKParticipant *organizer
```

Discussion

This property is `nil` if the event has no organizer.

Availability

Available in iOS 4.0 and later.

Declared In

EKEvent.h

recurrenceRule

The recurrence rule associated with the event.

```
@property(n nonatomic, retain) EKRecurrenceRule *recurrenceRule
```

Availability

Available in iOS 4.0 and later.

Declared In

EKEvent.h

startDate

The start date for the event.

```
@property(n nonatomic, copy) NSDate *startDate
```

Availability

Available in iOS 4.0 and later.

Declared In

EKEvent.h

status

The event’s status. (read-only)

```
@property(nonatomic, readonly) EKEventStatus status
```

Discussion

You should act based on an event’s status only if the status is [EKEventStatusCanceled](#) (page 28), which indicates that the event has been canceled. Other statuses should be considered informational.

Availability

Available in iOS 4.0 and later.

See Also

“EKEventStatus” (page 28)

Declared In

EKEvent.h

title

The title for the event.

```
@property(nonatomic, copy) NSString *title
```

Availability

Available in iOS 4.0 and later.

Declared In

EKEvent.h

Class Methods

eventWithEventStore:

Creates and returns a new event belonging to a specified event store.

```
+ (EKEvent *)eventWithEventStore:(EKEventStore *)eventStore
```

Parameters

eventStore

The event store the event belongs to.

Return Value

The created event.

Availability

Available in iOS 4.0 and later.

Declared In

EKEvent.h

Instance Methods

addAlarm:

Adds an alarm to the receiver.

```
- (void)addAlarm:(EKAlarm *)alarm
```

Parameters

alarm

The alarm to add.

Availability

Available in iOS 4.0 and later.

Declared In

EKEvent.h

compareStartDateWithEvent:

Compares the start date of the receiving event with the start date of another event.

```
- (NSComparisonResult)compareStartDateWithEvent:(EKEvent *)other
```

Parameters

other

The event to compare the receiver with.

Return Value

`NSOrderedAscending` if the start date of the receiver precedes the start date of *other*, `NSOrderedSame` if the start dates of the two events are identical, and `NSOrderedDescending` if the start date of the receiver comes after the start date of *other*.

Discussion

You can pass the selector for this method to the `NSArray` method `sortedArrayUsingSelector:` to create an array of events sorted by start date.

Availability

Available in iOS 4.0 and later.

Declared In

EKEvent.h

refresh

Updates the event's data with the current information in the Calendar database.

```
- (BOOL)refresh
```

Return Value

`YES` if the event was successfully refreshed; otherwise, `NO`

Discussion

You should call this method only on events that your application is editing, and only when your application receives the [EKEventStoreChangedNotification](#) (page 35) notification. If this method returns NO, the event has been deleted or otherwise invalidated, and you should not continue to use it.

This method does not replace the values of any properties that you have modified.

Availability

Available in iOS 4.0 and later.

Declared In

EKEvent.h

removeAlarm:

Removes an alarm from the event.

```
- (void)removeAlarm:(EKAlarm *)alarm
```

Parameters

alarm

The alarm to remove.

Availability

Available in iOS 4.0 and later.

Declared In

EKEvent.h

Constants

EKEventAvailability

Represents the event's availability setting for scheduling purposes.

```
typedef enum {
    EKEventAvailabilityNotSupported = -1,
    EKEventAvailabilityBusy = 0,
    EKEventAvailabilityFree,
    EKEventAvailabilityTentative,
    EKEventAvailabilityUnavailable
} EKEventAvailability;
```

Constants

EKEventAvailabilityNotSupported

Availability settings are not supported by the event's calendar.

Available in iOS 4.0 and later.

Declared in EKEvent.h.

EKEventAvailabilityBusy

The event has a busy availability setting.

Available in iOS 4.0 and later.

Declared in `EKEvent.h`.

EKEventAvailabilityFree

The event has a free availability setting.

Available in iOS 4.0 and later.

Declared in `EKEvent.h`.

EKEventAvailabilityTentative

The event has a tentative availability setting.

Available in iOS 4.0 and later.

Declared in `EKEvent.h`.

EKEventAvailabilityUnavailable

The event has an unavailable availability setting.

Available in iOS 4.0 and later.

Declared in `EKEvent.h`.

EKEventStatus

Represents the event's status.

```
typedef enum {
    EKEventStatusNone = 0,
    EKEventStatusConfirmed,
    EKEventStatusTentative,
    EKEventStatusCanceled,
} EKEventStatus;
```

Constants

EKEventStatusNone

The event has no status.

Available in iOS 4.0 and later.

Declared in `EKEvent.h`.

EKEventStatusConfirmed

The event is confirmed.

Available in iOS 4.0 and later.

Declared in `EKEvent.h`.

EKEventStatusTentative

The event is tentative.

Available in iOS 4.0 and later.

Declared in `EKEvent.h`.

EKEventStatusCanceled

The event is canceled.

Available in iOS 4.0 and later.

Declared in `EKEvent.h`.

EKEventStore Class Reference

Inherits from	NSObject
Conforms to	NSObject (NSObject)
Framework	/System/Library/Frameworks/EventKit.framework
Availability	Available in iOS 4.0 and later.
Declared in	EKEventStore.h
Companion guide	Event Kit Programming Guide

Overview

The `EKEventStore` object represents the Calendar database. It is an application's point of contact for accessing calendar data.

Initialize an event store object with the `init` method. Create a predicate, or a search query for the Calendar database, with the `predicateForEventsWithStartDate:endDate:calendars:` (page 32) method. Fetch and process events that match a given predicate with the `eventsMatchingPredicate:` (page 31) and `enumerateEventsMatchingPredicate:usingBlock:` (page 31) methods. Save and delete events from the event store with the `saveEvent:span:error:` (page 33) and `removeEvent:span:error:` (page 33) methods.

Tasks

Finding Events

- `eventWithIdentifier:` (page 32)
Returns the event with a given identifier.
- `eventsMatchingPredicate:` (page 31)
Returns all events that match a given predicate.
- `enumerateEventsMatchingPredicate:usingBlock:` (page 31)
Finds all events that match a given predicate and calls a given callback for each event found.
- `predicateForEventsWithStartDate:endDate:calendars:` (page 32)
Creates and returns a predicate for finding events in the event store that fall within a given date range.

Saving and Removing Events

- `saveEvent:span:error:` (page 33)
Saves changes to an event permanently.
- `removeEvent:span:error:` (page 33)
Removes an event from the event store.

Accessing Event Store Properties

- `calendars` (page 30) *property*
The calendars associated with the event store, as an array of `EKCalendar` objects. (read-only)
- `defaultCalendarForNewEvents` (page 30) *property*
The calendar that events are added to by default, as specified by user settings. (read-only)
- `eventStoreIdentifier` (page 31) *property*
The unique identifier for the event store. (read-only)

New Methods

Properties

For more about Objective-C properties, see “Properties” in *The Objective-C Programming Language*.

calendars

The calendars associated with the event store, as an array of `EKCalendar` objects. (read-only)

```
@property(nonatomic, readonly) NSArray *calendars
```

Availability

Available in iOS 4.0 and later.

Declared In

`EKEventStore.h`

defaultCalendarForNewEvents

The calendar that events are added to by default, as specified by user settings. (read-only)

```
@property(nonatomic, readonly) EKCalendar *defaultCalendarForNewEvents
```

Availability

Available in iOS 4.0 and later.

Declared In

`EKEventStore.h`

eventStoreIdentifier

The unique identifier for the event store. (read-only)

```
@property(nonatomic, readonly) NSString *eventStoreIdentifier
```

Availability

Available in iOS 4.0 and later.

Declared In

EKEventStore.h

Instance Methods

enumerateEventsMatchingPredicate:usingBlock:

Finds all events that match a given predicate and calls a given callback for each event found.

```
- (void)enumerateEventsMatchingPredicate:(NSPredicate *)predicate
    usingBlock:(EKEventSearchCallback)block
```

Parameters

predicate

The search predicate. Must be created with the [predicateForEventsWithStartDate:endDate:calendars:](#) (page 32) method.

block

The block callback to call for each event. The callback must match the signature defined by [EKEventSearchCallback](#) (page 35).

Discussion

This method is synchronous. For asynchronous behavior, run it on another thread with `dispatch_async` or `NSOperation`.

Availability

Available in iOS 4.0 and later.

See Also

[EKEventSearchCallback](#) (page 35)

- [predicateForEventsWithStartDate:endDate:calendars:](#) (page 32)

Declared In

EKEventStore.h

eventsMatchingPredicate:

Returns all events that match a given predicate.

```
- (NSArray *)eventsMatchingPredicate:(NSPredicate *)predicate
```

Parameters*predicate*

The search predicate. Must be created with the [predicateForEventsWithStartDate:endDate:calendars:](#) (page 32) method.

Return Value

All events that match *predicate*, as an array of `EKEvent` objects.

Discussion

This method is synchronous. For asynchronous behavior, run it on another thread with `dispatch_async` or `NSOperation`.

Availability

Available in iOS 4.0 and later.

See Also

- [predicateForEventsWithStartDate:endDate:calendars:](#) (page 32)

Declared In

`EKEventStore.h`

eventWithIdentifier:

Returns the event with a given identifier.

```
- (EKEvent *)eventWithIdentifier:(NSString *)identifier
```

Parameters*identifier*

The identifier of the event.

Return Value

The event corresponding to *identifier*, or `nil` if no event is found.

Availability

Available in iOS 4.0 and later.

Declared In

`EKEventStore.h`

predicateForEventsWithStartDate:endDate:calendars:

Creates and returns a predicate for finding events in the event store that fall within a given date range.

```
- (NSPredicate *)predicateForEventsWithStartDate:(NSDate *)startDateendDate:(NSDate *)endDatecalendars:(NSArray *)calendars
```

Parameters*startDate*

The start date of the range of events fetched.

endDate

The end date of the range of events fetched.

calendars

The calendars to search, as an array of `EKCalendar` objects. Passing `nil` indicates to search all calendars.

Return Value

The created predicate.

Availability

Available in iOS 4.0 and later.

Declared In

`EKEventStore.h`

removeEvent:span:error:

Removes an event from the event store.

```
- (BOOL)removeEvent:(EKEvent *)event span:(EKSpan)span error:(NSError **)error
```

Parameters*event*

The event to remove.

span

The span to use. Indicates whether to remove future instances of the event in the case of a recurring event.

error

The error if one occurred; otherwise, `nil`.

Return Value

YES if the event is successfully removed; otherwise, NO.

Discussion

This method returns NO if `event` is not in the event store to begin with.

This method raises an exception if it is passed an event from another event store.

Availability

Available in iOS 4.0 and later.

Declared In

`EKEventStore.h`

saveEvent:span:error:

Saves changes to an event permanently.

```
- (BOOL)saveEvent:(EKEvent *)event span:(EKSpan)span error:(NSError **)error
```

Parameters*event*

The event to save.

span

The span to use. Indicates whether the save affects future instances of the event in the case of a recurring event.

error

The error that occurred, if any did. Otherwise, `nil`.

Return Value

YES if the event is successfully saved; otherwise, NO.

Discussion

This method returns NO if `event` does not need to be saved because it has not been modified.

This method raises an exception if it is passed an event from another event store.

When an event is saved, it is updated in the Calendar database. Any fields you did not modify are updated to reflect the most recent value in the database. If the event has been deleted from the database, it is recreated as a new event.

Availability

Available in iOS 4.0 and later.

Declared In

`EKEventStore.h`

Constants

EKSpan

Indicates whether modifications to an instance of a recurring event should affect future instances of the event.

```
typedef enum {
    EKSpanThisEvent,
    EKSpanFutureEvents
} EKSpan;
```

Constants

`EKSpanThisEvent`

Modifications to this event instance should affect only this instance.

Available in iOS 4.0 and later.

Declared in `EKEventStore.h`.

`EKSpanFutureEvents`

Modifications to this event instance should also affect future instances of this event.

Available in iOS 4.0 and later.

Declared in `EKEventStore.h`.

EKEventSearchCallback

Defines the signature for a block object used for operating on events with the [enumerateEventsMatchingPredicate:usingBlock:](#) (page 31) method.

```
typedef void (^EKEventSearchCallback)(EKEvent *event, BOOL *stop);
```

Discussion

The `event` argument is the event to operate on. You can set the value of the `stop` argument to `YES` to tell the `enumerateEventsMatchingPredicate:usingBlock:` method to stop processing events when this block returns.

Availability

Available in iOS 4.0 and later.

Declared In

`EKEventStore.h`

Notifications

EKEventStoreChangedNotification

Posted whenever changes are made to the Calendar database, including adding, removing, and changing events. Individual changes are not described. When you receive this notification, you should release and refetch all `EKEvent` objects you have retained. If you are actively editing an event and do not wish to refetch it unless it is absolutely necessary to do so, you can call the `refresh` method on it. If the method returns `YES`, you do not need to refetch the event.

Availability

Available in iOS 4.0 and later.

Declared In

`EKEventStore.h`

EKParticipant Class Reference

Inherits from	NSObject
Conforms to	NSCopying NSObject (NSObject)
Framework	/System/Library/Frameworks/EventKit.framework
Availability	Available in iOS 4.0 and later.
Declared in	EKParticipant.h
Companion guide	Event Kit Programming Guide

Overview

An `EKParticipant` object represents a participant in a calendar event. A participant can be a person, group, room, or other resource.

You do not create `EKParticipant` objects directly. Send `attendees` to an `EKEvent` object to get an array of `EKParticipant` objects.

Use the properties in this class to get information about a participant.

Tasks

Accessing Participant Properties

- [name](#) (page 38) *property*
The participant's name. (read-only)
- [participantRole](#) (page 38) *property*
The participant's role in the event. (read-only)
- [participantStatus](#) (page 38) *property*
The participant's attendance status. (read-only)
- [participantType](#) (page 39) *property*
The participant's type. (read-only)
- [URL](#) (page 39) *property*
The URL representing this participant. (read-only)

Finding Participant Address Book Records

- [ABRecordWithAddressBook](#): (page 39)

Returns the address book record that represents the participant.

Properties

For more about Objective-C properties, see “Properties” in *The Objective-C Programming Language*.

name

The participant’s name. (read-only)

```
@property(readonly) NSString *name
```

Availability

Available in iOS 4.0 and later.

Declared In

EKParticipant.h

participantRole

The participant’s role in the event. (read-only)

```
@property(readonly) EKParticipantRole participantRole
```

Availability

Available in iOS 4.0 and later.

See Also

[“Participant Roles”](#) (page 40)

Declared In

EKParticipant.h

participantStatus

The participant’s attendance status. (read-only)

```
@property(readonly) EKParticipantStatus participantStatus
```

Availability

Available in iOS 4.0 and later.

See Also

[“Participant Statuses”](#) (page 40)

Declared In

EKParticipant.h

participantType

The participant's type. (read-only)

@property(readonly) EKParticipantType participantType

Availability

Available in iOS 4.0 and later.

See Also[“Participant Types”](#) (page 42)**Declared In**

EKParticipant.h

URL

The URL representing this participant. (read-only)

@property(readonly) NSURL *URL

Availability

Available in iOS 4.0 and later.

Declared In

EKParticipant.h

Instance Methods

ABRecordWithAddressBook:

Returns the address book record that represents the participant.

```
- (ABRecordRef)ABRecordWithAddressBook:(ABAddressBookRef)addressBook
```

Parameters*addressBook*

The address book to search.

Return ValueThe address book record for the participant, or `nil` if the record is not found.**Discussion**

This method searches for a record match based on the participant's email address.

Availability

Available in iOS 4.0 and later.

Declared In

EKParticipant.h

Constants

Participant Roles

Represents a participant's role for an event.

```
typedef enum {  
    EKParticipantRoleUnknown,  
    EKParticipantRoleRequired,  
    EKParticipantRoleOptional,  
    EKParticipantRoleChair,  
    EKParticipantRoleNonParticipant  
} EKParticipantRole;
```

Constants

EKParticipantRoleUnknown

The participant's role is unknown.

Available in iOS 4.0 and later.

Declared in EKParticipant.h.

EKParticipantRoleRequired

The participant's attendance is required.

Available in iOS 4.0 and later.

Declared in EKParticipant.h.

EKParticipantRoleOptional

The participant's attendance is optional.

Available in iOS 4.0 and later.

Declared in EKParticipant.h.

EKParticipantRoleChair

The participant is the chair of the event.

Available in iOS 4.0 and later.

Declared in EKParticipant.h.

EKParticipantRoleNonParticipant

The participant does not have an active role in the event.

Available in iOS 4.0 and later.

Declared in EKParticipant.h.

Participant Statuses

Represents a participant's attendance status for an event.


```
typedef enum {
    EKParticipantStatusUnknown,
    EKParticipantStatusPending,
    EKParticipantStatusAccepted,
    EKParticipantStatusDeclined,
    EKParticipantStatusTentative,
    EKParticipantStatusDelegated,
    EKParticipantStatusCompleted,
    EKParticipantStatusInProgress
} EKParticipantStatus;
```

Constants

- EKParticipantStatusUnknown**
 The participant's attendance status is unknown.
 Available in iOS 4.0 and later.
 Declared in `EKParticipant.h`.
- EKParticipantStatusPending**
 The participant has yet to respond to the event.
 Available in iOS 4.0 and later.
 Declared in `EKParticipant.h`.
- EKParticipantStatusAccepted**
 The participant has accepted the event.
 Available in iOS 4.0 and later.
 Declared in `EKParticipant.h`.
- EKParticipantStatusDeclined**
 The participant has declined the event.
 Available in iOS 4.0 and later.
 Declared in `EKParticipant.h`.
- EKParticipantStatusTentative**
 The participant's attendance status is tentative.
 Available in iOS 4.0 and later.
 Declared in `EKParticipant.h`.
- EKParticipantStatusDelegated**
 The participant has delegated attendance to another participant.
 Available in iOS 4.0 and later.
 Declared in `EKParticipant.h`.
- EKParticipantStatusCompleted**
 The participant's event has completed.
 Available in iOS 4.0 and later.
 Declared in `EKParticipant.h`.
- EKParticipantStatusInProgress**
 The participant's event is currently in process.
 Available in iOS 4.0 and later.
 Declared in `EKParticipant.h`.

Participant Types

Represent's a participant's type.

```
typedef enum {
    EKParticipantTypeUnknown,
    EKParticipantTypePerson,
    EKParticipantTypeRoom,
    EKParticipantTypeResource,
    EKParticipantTypeGroup
} EKParticipantType;
```

Constants

`EKParticipantTypeUnknown`
The participant's type is unknown.
Available in iOS 4.0 and later.
Declared in `EKParticipant.h`.

`EKParticipantTypePerson`
The participant is a person.
Available in iOS 4.0 and later.
Declared in `EKParticipant.h`.

`EKParticipantTypeRoom`
The participant is a room.
Available in iOS 4.0 and later.
Declared in `EKParticipant.h`.

`EKParticipantTypeResource`
The participant is a resource.
Available in iOS 4.0 and later.
Declared in `EKParticipant.h`.

`EKParticipantTypeGroup`
The participant is a group.
Available in iOS 4.0 and later.
Declared in `EKParticipant.h`.

EKRecurrenceDayOfWeek Class Reference

Inherits from	NSObject
Conforms to	NSCopying NSObject (NSObject)
Framework	/System/Library/Frameworks/EventKit.framework
Availability	Available in iOS 4.0 and later.
Declared in	EKRecurrenceRule.h
Companion guide	Event Kit Programming Guide

Overview

The `EKRecurrenceDayOfWeek` class represents a day of the week for use with an `EKRecurrenceRule` object. A day of the week can optionally have a week number, indicating a specific day in the recurrence rule's frequency. For example, a day of the week with a day value of `Tuesday` and a week number of `2` would represent the second Tuesday of every month in a monthly recurrence rule, and the second Tuesday of every year in a yearly recurrence rule. A day of the week with a week number of `0` ignores its week number.

Tasks

Creating a Day of the Week

- + `dayOfWeek:` (page 44)
Initializes and returns a day of the week with a given day.
- + `dayOfWeek:weekNumber:` (page 45)
Initializes and returns an autoreleased day of the week with a given day and week number.

Accessing Properties of a Day of the Week

- `dayOfTheWeek` (page 44) *property*
The day of the week. Values are from 1 to 7, with Sunday being 1. (read-only)
- `weekNumber` (page 44) *property*
The week number of the day of the week. (read-only)

Properties

For more about Objective-C properties, see “Properties” in *The Objective-C Programming Language*.

dayOfTheWeek

The day of the week. Values are from 1 to 7, with Sunday being 1. (read-only)

```
@property(readonly) NSInteger dayOfTheWeek
```

Availability

Available in iOS 4.0 and later.

Declared In

EKRecurrenceRule.h

weekNumber

The week number of the day of the week. (read-only)

```
@property(readonly) NSInteger weekNumber
```

Availability

Available in iOS 4.0 and later.

Declared In

EKRecurrenceRule.h

Class Methods

dayOfWeek:

Initializes and returns a day of the week with a given day.

```
+ (id)dayOfWeek:(NSInteger)dayOfTheWeek
```

Parameters

dayOfTheWeek

The day of the week. Values range from 1 to 7, with Sunday being 1.

Return Value

The initialized day of the week.

Discussion

The week number of the returned day of the week is 0.

Availability

Available in iOS 4.0 and later.

Declared In

EKRecurrenceRule.h

dayOfWeek:weekNumber:

Initializes and returns an autoreleased day of the week with a given day and week number.

```
+ (id)dayOfWeek:(NSInteger)dayOfTheWeek weekNumber:(NSInteger)weekNumber
```

Parameters*dayOfTheWeek*

The day of the week.

weekNumber

The week number.

Return Value

The initialized day of the week.

Availability

Available in iOS 4.0 and later.

Declared In

EKRecurrenceRule.h

EKRecurrenceEnd Class Reference

Inherits from	NSObject
Conforms to	NSCopying NSObject (NSObject)
Framework	/System/Library/Frameworks/EventKit.framework
Availability	Available in iOS 4.0 and later.
Declared in	EKRecurrenceRule.h
Companion guide	Event Kit Programming Guide

Overview

The `EKRecurrenceEnd` class defines the end of a recurrence rule defined by an `EKRecurrenceRule` object. The recurrence end can be specified by a date (date-based) or by a maximum count of occurrences (count-based).

Tasks

Creating a Recurrence End

- + `recurrenceEndWithEndDate:` (page 48)
Initializes and returns a date-based recurrence end with a given end date.
- + `recurrenceEndWithOccurrenceCount:` (page 49)
Initializes and returns a count-based recurrence end with a given maximum occurrence count.

Accessing Recurrence End Properties

- `endDate` (page 48) *property*
The end date of the recurrence end, or `nil` if the recurrence end is count-based. (read-only)
- `occurrenceCount` (page 48) *property*
The occurrence count of the recurrence end, or `0` if the recurrence end is date-based. (read-only)

Properties

For more about Objective-C properties, see “Properties” in *The Objective-C Programming Language*.

endDate

The end date of the recurrence end, or `nil` if the recurrence end is count-based. (read-only)

```
@property(readonly) NSDate *endDate
```

Availability

Available in iOS 4.0 and later.

Declared In

EKRecurrenceRule.h

occurrenceCount

The occurrence count of the recurrence end, or 0 if the recurrence end is date-based. (read-only)

```
@property(readonly) NSUInteger occurrenceCount
```

Availability

Available in iOS 4.0 and later.

Declared In

EKRecurrenceRule.h

Class Methods

recurrenceEndWithEndDate:

Initializes and returns a date-based recurrence end with a given end date.

```
+ (id)recurrenceEndWithEndDate:(NSDate *)endDate
```

Parameters

endDate

The end date.

Return Value

The initialized recurrence end.

Availability

Available in iOS 4.0 and later.

Declared In

EKRecurrenceRule.h

recurrenceEndWithOccurrenceCount:

Initializes and returns a count-based recurrence end with a given maximum occurrence count.

```
+ (id)recurrenceEndWithOccurrenceCount:(NSUInteger)occurrenceCount
```

Parameters

occurrenceCount

The maximum occurrence count.

Return Value

The initialized recurrence end.

Availability

Available in iOS 4.0 and later.

Declared In

EKRecurrenceRule.h

EKRecurrenceRule Class Reference

Inherits from	NSObject
Conforms to	NSObject (NSObject)
Framework	/System/Library/Frameworks/EventKit.framework
Availability	Available in iOS 4.0 and later.
Declared in	EKRecurrenceRule.h
Companion guide	Event Kit Programming Guide

Overview

The `EKRecurrenceRule` class is used to describe the recurrence pattern for a recurring event.

After you create a recurrence rule, assign it to an event with the `setRecurrenceRule:` method of `EKEvent`.

Recurrence rules can have an end, represented by an `EKRecurrenceEnd` object. The end can be based on a specific date or a maximum number of occurrences.

Tasks

Creating a Recurrence Rule

- `initWithFrequency:interval:daysOfTheWeek:daysOfTheMonth:monthsOfTheYear:weeksOfTheYear:daysOfTheYear:setPositions:end:` (page 56)
Initializes and returns a recurrence rule with a given frequency and additional scheduling information.
- `initWithFrequency:interval:end:` (page 57)
Initializes and returns a simple recurrence rule with a given frequency, interval, and end.

Accessing Recurrence Rule Properties

- `calendarIdentifier` (page 53) *property*
The identifier for the recurrence rule's calendar. (read-only)

daysOfTheMonth (page 53) *property*

The days of the month associated with the recurrence rule, as an array of `NSNumber` objects. Values can be from 1 to 31 and from -1 to -31. This property value is valid only for recurrence rules that were initialized with specific days of the month and a frequency type of `EKMonthlyRecurrence`. (read-only)

daysOfTheWeek (page 53) *property*

The days of the week associated with the recurrence rule, as an array of `EKRecurrenceDayOfWeek` objects. This property value is valid only for recurrence rules that were initialized with specific days of the week and a frequency type of `EKWeeklyRecurrence`, `EKMonthlyRecurrence`, or `EKYearlyRecurrence`. (read-only)

daysOfTheYear (page 54) *property*

The days of the year associated with the recurrence rule, as an array of `NSNumber` objects. Values can be from 1 to 366 and from -1 to -366. This property value is valid only for recurrence rules initialized with a frequency type of `EKYearlyRecurrence`. (read-only)

firstDayOfTheWeek (page 54) *property*

Indicates which day of the week the recurrence rule treats as the first day of the week. Values of 1 to 7 correspond to Sunday through Saturday. A value of 0 indicates that this property is not set for the recurrence rule. (read-only)

frequency (page 54) *property*

The frequency of the recurrence rule. (read-only)

interval (page 55) *property*

Specifies how often the recurrence rule repeats over the unit of time indicated by its frequency. For example, a recurrence rule with a frequency type of `EKWeeklyRecurrence` and an interval of 2 repeats every two weeks. (read-only)

monthsOfTheYear (page 55) *property*

The months of the year associated with the recurrence rule, as an array of `NSNumber` objects. Values can be from 1 to 12. This property value is valid only for recurrence rules initialized with specific months of the year and a frequency type of `EKYearlyRecurrence`. (read-only)

recurrenceEnd (page 55) *property*

Indicates when the recurrence rule ends. This can be represented by an end date or a number of occurrences.

setPositions (page 55) *property*

An array of ordinal numbers that filters which recurrences to include in the recurrence rule's frequency. For example, a yearly recurrence rule that has a `daysOfTheWeek` value that specifies Monday through Friday, and a `setPositions` array containing 2 and -1, occurs only on the second weekday and last weekday of every year. (read-only)

weeksOfTheYear (page 56) *property*

The weeks of the year associated with the recurrence rule, as an array of `NSNumber` objects. Values can be from 1 to 53 and from -1 to -53. This property value is valid only for recurrence rules initialized with specific weeks of the year and a frequency type of `EKYearlyRecurrence`. (read-only)

Properties

For more about Objective-C properties, see “Properties” in *The Objective-C Programming Language*.

calendarIdentifier

The identifier for the recurrence rule's calendar. (read-only)

```
@property(readonly) NSString *calendarIdentifier
```

Availability

Available in iOS 4.0 and later.

Declared In

EKRecurrenceRule.h

daysOfTheMonth

The days of the month associated with the recurrence rule, as an array of `NSNumber` objects. Values can be from 1 to 31 and from -1 to -31. This property value is valid only for recurrence rules that were initialized with specific days of the month and a frequency type of `EKMonthlyRecurrence`. (read-only)

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

Discussion

This property corresponds to `BYMONTHDAY` in the iCalendar specification.

Negative values indicate counting backwards from the end of the month.

Availability

Available in iOS 4.0 and later.

Declared In

EKRecurrenceRule.h

daysOfTheWeek

The days of the week associated with the recurrence rule, as an array of `EKRecurrenceDayOfWeek` objects. This property value is valid only for recurrence rules that were initialized with specific days of the week and a frequency type of `EKWeeklyRecurrence`, `EKMonthlyRecurrence`, or `EKYearlyRecurrence`. (read-only)

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

Discussion

This property corresponds to `BYDAY` in the iCalendar specification.

Availability

Available in iOS 4.0 and later.

Declared In

EKRecurrenceRule.h

daysOfTheYear

The days of the year associated with the recurrence rule, as an array of `NSNumber` objects. Values can be from 1 to 366 and from -1 to -366. This property value is valid only for recurrence rules initialized with a frequency type of `EKYearlyRecurrence`. (read-only)

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

Discussion

This property corresponds to `BYEARDAY` in the iCalendar specification.

Negative values indicate counting backwards from the end of the year.

Availability

Available in iOS 4.0 and later.

Declared In

`EKRecurrenceRule.h`

firstDayOfTheWeek

Indicates which day of the week the recurrence rule treats as the first day of the week. Values of 1 to 7 correspond to Sunday through Saturday. A value of 0 indicates that this property is not set for the recurrence rule. (read-only)

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

Availability

Available in iOS 4.0 and later.

Declared In

`EKRecurrenceRule.h`

frequency

The frequency of the recurrence rule. (read-only)

```
@property(readonly) EKRecurrenceFrequency frequency
```

Availability

Available in iOS 4.0 and later.

See Also

[“EKRecurrenceFrequency”](#) (page 58)

Declared In

`EKRecurrenceRule.h`

interval

Specifies how often the recurrence rule repeats over the unit of time indicated by its frequency. For example, a recurrence rule with a frequency type of `EKWeeklyRecurrence` and an interval of 2 repeats every two weeks. (read-only)

```
@property(readonly) NSInteger interval
```

Availability

Available in iOS 4.0 and later.

Declared In

`EKRecurrenceRule.h`

monthsOfTheYear

The months of the year associated with the recurrence rule, as an array of `NSNumber` objects. Values can be from 1 to 12. This property value is valid only for recurrence rules initialized with specific months of the year and a frequency type of `EKYearlyRecurrence`. (read-only)

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

Discussion

This property corresponds to `BYMONTH` in the iCalendar specification.

Availability

Available in iOS 4.0 and later.

Declared In

`EKRecurrenceRule.h`

recurrenceEnd

Indicates when the recurrence rule ends. This can be represented by an end date or a number of occurrences.

```
@property(copy) EKRecurrenceEnd *recurrenceEnd
```

Availability

Available in iOS 4.0 and later.

Declared In

`EKRecurrenceRule.h`

setPositions

An array of ordinal numbers that filters which recurrences to include in the recurrence rule's frequency. For example, a yearly recurrence rule that has a `daysOfTheWeek` value that specifies Monday through Friday, and a `setPositions` array containing 2 and -1, occurs only on the second weekday and last weekday of every year. (read-only)

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

Discussion

This property corresponds to BYSETPOS in the iCalendar specification.

Values can be from 1 to 366 and from -1 to -366.

Negative values indicate counting backwards from the end of the recurrence rule's frequency (week, month, or year).

Availability

Available in iOS 4.0 and later.

Declared In

EKRecurrenceRule.h

weeksOfTheYear

The weeks of the year associated with the recurrence rule, as an array of `NSNumber` objects. Values can be from 1 to 53 and from -1 to -53. This property value is valid only for recurrence rules initialized with specific weeks of the year and a frequency type of `EKYearlyRecurrence`. (read-only)

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

Discussion

This property corresponds to BYWEEK in the iCalendar specification.

Negative values indicate counting backwards from the end of the year.

Availability

Available in iOS 4.0 and later.

Declared In

EKRecurrenceRule.h

Instance Methods

initRecurrenceWithFrequency:interval:daysOfTheWeek:daysOfTheMonth:monthsOfTheYear:weeksOfTheYear:daysOfTheYear:setPositions:end:

Initializes and returns a recurrence rule with a given frequency and additional scheduling information.

```
- (id)initRecurrenceWithFrequency:(EKRecurrenceFrequency)type
    interval:(NSInteger)interval daysOfTheWeek:(NSArray *)days
    daysOfTheMonth:(NSArray *)monthDays monthsOfTheYear:(NSArray *)months
    weeksOfTheYear:(NSArray *)weeksOfTheYear daysOfTheYear:(NSArray *)daysOfTheYear
    setPositions:(NSArray *)setPositions end:(EKRecurrenceEnd *)end
```

Parameters

type

The frequency of the recurrence rule. Can be daily, weekly, monthly, or yearly.

interval

The interval between instances of this recurrence. For example, a weekly recurrence rule with an interval of 2 occurs every other week. Must be greater than 0.

days

The days of the week that the event occurs, as an array of `EKRecurrenceDayOfWeek` objects.

monthDays

The days of the month that the event occurs, as an array of `NSNumber` objects. Values can be from 1 to 31 and from -1 to -31. This parameter is only valid for recurrence rules of type `EKMonthlyRecurrence`.

months

The months of the year that the event occurs, as an array of `NSNumber` objects. Values can be from 1 to 12. This parameter is only valid for recurrence rules of type `EKYearlyOccurrence`.

weeksOfTheYear

The weeks of the year that the event occurs, as an array of `NSNumber` objects. Values can be from 1 to 53 and from -1 to -53. This parameter is only valid for recurrence rules of type `EKYearlyOccurrence`.

daysOfTheYear

The days of the year that the event occurs, as an array of `NSNumber` objects. Values can be from 1 to 366 and from -1 to -366. This parameter is only valid for recurrence rules of type `EKYearlyOccurrence`.

setPositions

An array of ordinal numbers that filters which recurrences to include in the recurrence rule's frequency. See [setPositions](#) (page 55) for more information.

end

The end of the recurrence rule.

Return Value

The initialized recurrence rule, or `nil` if invalid values are provided.

Discussion

Negative values indicate counting backwards from the end of the recurrence rule's frequency.

Availability

Available in iOS 4.0 and later.

Declared In

`EKRecurrenceRule.h`

initWithFrequency:interval:end:

Initializes and returns a simple recurrence rule with a given frequency, interval, and end.

```
- (id) initWithFrequency:(EKRecurrenceFrequency) type
    interval:(NSUInteger) interval end:(EKRecurrenceEnd *) end
```

Parameters*type*

The frequency of the recurrence rule. Can be daily, weekly, monthly, or yearly.

interval

The interval between instances of this recurrence. For example, a weekly recurrence rule with an interval of 2 occurs every other week. Must be greater than 0.

end

The end of the recurrence rule.

Return Value

The initialized recurrence rule, or `nil` if invalid values are provided.

Availability

Available in iOS 4.0 and later.

Declared In

EKRecurrenceRule.h

Constants

EKRecurrenceFrequency

Defines frequencies for recurrence rules.

```
typedef enum {
    EKRecurrenceFrequencyDaily,
    EKRecurrenceFrequencyWeekly,
    EKRecurrenceFrequencyMonthly,
    EKRecurrenceFrequencyYearly
} EKRecurrenceFrequency;
```

Constants

`EKRecurrenceFrequencyDaily`
Indicates a daily recurrence rule.
 Available in iOS 4.0 and later.
 Declared in EKRecurrenceRule.h.

`EKRecurrenceFrequencyWeekly`
Indicates a weekly recurrence rule.
 Available in iOS 4.0 and later.
 Declared in EKRecurrenceRule.h.

`EKRecurrenceFrequencyMonthly`
Indicates a monthly recurrence rule.
 Available in iOS 4.0 and later.
 Declared in EKRecurrenceRule.h.

`EKRecurrenceFrequencyYearly`
Indicates a yearly recurrence rule.
 Available in iOS 4.0 and later.
 Declared in EKRecurrenceRule.h.

Days of the Week

Defines values for the days of the week.

```
enum {
    EKSunday = 1,
    EKMonday,
    EKTuesday,
    EKWednesday,
    EKThursday,
    EKFriday,
    EKSaturday
};
```

Constants

EKSunday

The value for Sunday.

Available in iOS 4.0 and later.

Declared in `EKRecurrenceRule.h`.

EKMonday

The value for Monday.

Available in iOS 4.0 and later.

Declared in `EKRecurrenceRule.h`.

EKTuesday

The value for Tuesday.

Available in iOS 4.0 and later.

Declared in `EKRecurrenceRule.h`.

EKWednesday

The value for Wednesday.

Available in iOS 4.0 and later.

Declared in `EKRecurrenceRule.h`.

EKThursday

The value for Thursday.

Available in iOS 4.0 and later.

Declared in `EKRecurrenceRule.h`.

EKFriday

The value for Friday.

Available in iOS 4.0 and later.

Declared in `EKRecurrenceRule.h`.

EKSaturday

The value for Saturday.

Available in iOS 4.0 and later.

Declared in `EKRecurrenceRule.h`.

Constants

Event Kit Constants Reference

Framework:	/System/Library/Frameworks/EventKit.framework
Declared in	EventKit/EKError.h
Companion guide	Event Kit Programming Guide

Overview

This document defines constants in the Event Kit framework that are not associated with a particular class.

Constants

Event Kit Error Domains

Values that are returned as the error domain property of an `NSError` object.

```
NSString *const EKErrorDomain;
```

Constants

`EKErrorDomain`

Event Kit errors.

Available in iOS 4.0 and later.

Declared in `EKError.h`.

Event Kit Error Codes

Values that are returned as the error code property of an `NSError` object.

```
typedef enum EKErrorCode {
    EKErrorEventNotMutable,
    EKErrorNoCalendar,
    EKErrorNoStartDate,
    EKErrorNoEndDate,
    EKErrorDatesInverted,
    EKErrorInternalFailure,
    EKErrorCalendarReadOnly,
    EKErrorDurationGreaterThanRecurrence,
    EKErrorAlarmGreaterThanRecurrence,
    EKErrorStartDateTooFarInFuture,
    EKErrorStartDateCollidesWithOtherOccurrence,
    EKErrorObjectBelongsToDifferentStore,
    EKErrorInvitesCannotBeMoved,
};
```

Constants

`EKErrorEventNotMutable`

The event is not mutable and cannot be saved or deleted.

Available in iOS 4.0 and later.

Declared in `EKError.h`.

`EKErrorNoCalendar`

The event is not associated with a calendar.

Available in iOS 4.0 and later.

Declared in `EKError.h`.

`EKErrorNoStartDate`

The event has no start date set.

Available in iOS 4.0 and later.

Declared in `EKError.h`.

`EKErrorNoEndDate`

The event has no end date set.

Available in iOS 4.0 and later.

Declared in `EKError.h`.

`EKErrorDatesInverted`

The event's end date is before its start date.

Available in iOS 4.0 and later.

Declared in `EKError.h`.

`EKErrorInternalFailure`

An internal error occurred.

Available in iOS 4.0 and later.

Declared in `EKError.h`.

`EKErrorCalendarReadOnly`

The calendar is read-only and cannot have events added to it.

Available in iOS 4.0 and later.

Declared in `EKError.h`.

- `EKErrorDurationGreaterThanRecurrence`
The duration of an event is greater than its recurrence interval.
Available in iOS 4.0 and later.
Declared in `EKError.h`.
- `EKErrorAlarmGreaterThanRecurrence`
The alarm interval is greater than the recurrence interval.
Available in iOS 4.0 and later.
Declared in `EKError.h`.
- `EKErrorStartDateTooFarInFuture`
The start date is further into the future than the calendar will display.
Available in iOS 4.0 and later.
Declared in `EKError.h`.
- `EKErrorStartDateCollidesWithOtherOccurrence`
The event's start date collides with another occurrence of the event.
Available in iOS 4.0 and later.
Declared in `EKError.h`.
- `EKErrorObjectBelongsToDifferentStore`
The object belongs to a different calendar store.
Available in iOS 4.0 and later.
Declared in `EKError.h`.
- `EKErrorInvitesCannotBeMoved`
The event cannot be moved because it is an invite.
Available in iOS 4.0 and later.
Declared in `EKError.h`.

Document Revision History

This table describes the changes to *Event Kit Framework Reference*.

Date	Notes
2010-03-25	First revision of this document.

REVISION HISTORY

Document Revision History