

[UIKit](#) / [ELError](#)

## Structure

# ELError

An [UIKit](#) error.

[iOS](#) | [iPadOS](#) | [Mac Catalyst](#) | [macOS](#) | [visionOS](#) | [watchOS](#)

```
struct ELError
```

## Topics

### Error Codes

```
static var eventNotMutable: ELError.Code
```

The event isn't mutable and you can't save or delete it.

```
static var noCalendar: ELError.Code
```

The event isn't associated with a calendar.

```
static var noEndDate: ELError.Code
```

The event has no start date set.

```
static var noStartDate: ELError.Code
```

The event has no end date set.

```
static var datesInverted: ELError.Code
```

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

```
static var internalFailure: ELError.Code
```

An internal error occurred.

`static var calendarReadOnly: EKError.Code`

The calendar is read-only and you can't add events to it.

`static var durationGreaterThanRecurrence: EKError.Code`

The duration of an event is greater than its recurrence interval.

`static var alarmGreaterThanRecurrence: EKError.Code`

The alarm interval is greater than the recurrence interval.

`static var startDateTooFarInFuture: EKError.Code`

The start date is further into the future than the calendar can display.

`static var startDateCollidesWithOtherOccurrence: EKError.Code`

The event's start date collides with another occurrence of the event.

`static var objectBelongsToDifferentStore: EKError.Code`

The object belongs to a different calendar store.

`static var invitesCannotBeMoved: EKError.Code`

You can't move the event because it's an invitation.

`static var invalidSpan: EKError.Code`

The system encountered an invalid span during a save or deletion.

`static var calendarHasNoSource: EKError.Code`

You can't save the calendar without setting a source first.

`static var calendarSourceCannotBeModified: EKError.Code`

You can't move the calendar to another source.

`static var calendarIsImmutable: EKError.Code`

The calendar is immutable and you can't modify or delete it.

`static var sourceDoesNotAllowCalendarAddDelete: EKError.Code`

The source doesn't allow you to add or delete calendars.

`static var recurringReminderRequiresDueDate: EKError.Code`

The recurring reminder requires a due date.

`static var structuredLocationsNotSupported: EKError.Code`

The source to which this calendar belongs doesn't support structured locations.

`static var reminderLocationsNotSupported: EKError.Code`

The source doesn't support locations on reminders.

`static var alarmProximityNotSupported: EKError.Code`

The source doesn't allow geofences on alarms.

`static var calendarDoesNotAllowEvents: EKError.Code`

The calendar doesn't allow you to add events.

`static var calendarDoesNotAllowReminders: EKError.Code`

The calendar doesn't allow you to add reminders.

`static var sourceDoesNotAllowReminders: EKError.Code`

The source doesn't allow calendars supporting reminder entity types.

`static var sourceDoesNotAllowEvents: EKError.Code`

The source doesn't allow calendars supporting event entity types.

`static var priorityIsInvalid: EKError.Code`

The priority number for the reminder is invalid.

`static var invalidEntityType: EKError.Code`

The entity type is invalid.

`static var procedureAlarmsNotMutable: EKError.Code`

You can't create or modify procedure alarms.

`static var eventStoreNotAuthorized: EKError.Code`

The user hasn't authorized your app to access events or reminders.

`static var osNotSupported: EKError.Code`

The action isn't supported on the current operating system.

`static var invalidInviteReplyCalendar: EKError.Code`

The calendar is invalid or nil.

`static var notificationsCollectionFlagNotSet: EKError.Code`

The notification collection doesn't have the notifications collection flag.

`static var sourceMismatch: EKError.Code`

The object's source doesn't match its container's source.

`static var notificationCollectionMismatch: EKError.Code`

The notification collection that contains this notification doesn't match the collection the system is trying to save.

`static var notificationSavedWithoutCollection: EKError.Code`

The notification can't save because you haven't added it to a notification collection and saved the collection first.

`static var last: EKError.Code`

This error is for internal use.

## Error Domain

`let EKErrorDomain: String`

A string that identifies the EventKit error domain.

## Type Properties

`static var errorDomain: String`

`static var reminderAlarmContainsEmailOrUrl: EKError.Code`

---

# Relationships

## Conforms To

CustomNSError  
Equatable  
Error  
Hashable  
Sendable  
SendableMetatype

---

## See Also

# Errors

`enum Code`

Error codes for EventKit errors.

`let EKErrorDomain: String`

A string that identifies the EventKit error domain.