

CalendarLib EC Suite

CalendarLib EC commands. Requires OS X 10.11 or higher.

fetch store *v* : Fetches the event store, prompting authorization if necessary.

fetch store

→ any : The event store. You need this before you can fetch calendars and events.

fetch calendar *v* : Fetches a calendar.

fetch calendar text : The name of the calendar.

cal type [cal local](#)/[cal cloud](#)/[cal exchange](#)/[cal subscription](#)/[cal birthday](#) : The type of calendar. One of [cal local](#)/[cal cloud](#)/[cal exchange](#)/[cal subscription](#)/[cal birthday](#).

event store any : The event store, as returned by the 'fetch store' command.

→ any : The designated calendar, or missing value if none found.

fetch calendars *v* : Fetches a list of calendars. Requires version 1.1 of CalendarLib.

fetch calendars list of text : A list of calendar names. Pass an empty list to include all names.

[**cal type list** list of [cal local](#)/[cal cloud](#)/[cal exchange](#)/[cal subscription](#)/[cal birthday](#)] : A list of calendar types, from [cal local](#)/[cal cloud](#)/[cal exchange](#)/[cal subscription](#)/[cal birthday](#). Pass an empty list for all types. Default is all types.

event store any : The event store, as returned by the 'fetch store' command.

→ list of any : A list of calendars.

cal types *enum* : Calendar types

cal local : Local.

cal cloud : iCloud/CalDAV.

cal exchange : Exchange.

cal subscription : Subscription.

cal birthday : Birthday.

fetch events by UID *v* : Fetch events matching the provided UID.

fetch events by UID text : The UID of the event.

event store any : The event store, as returned by the 'fetch store' command.

→ list of any : A list of matching events (there can be more than one match in some cases).

fetch events *v* : Fetch events that occur within a specified time span.

fetch events

starting date any : The start date. Can be an AppleScript date or Cocoa NSDate.

ending date any : The end date. Can be an AppleScript date or Cocoa NSDate.

searching cals list of any : List or array of calendars to search.

event store any : The event store, as returned by the 'fetch store' command.

→ list of any : A list of matching events, sorted by start date.

filter events *v* : Filter a list of events by matching the values with those of the following parameters.

filter events

event list list of any : The list or array of events to filter.

[**event summary** text] : The summary/title of the event to match. Omit or pass missing value to ignore it.

[**event location** any] : The location of the event. Omit or pass missing value to ignore it.

[**event description** text] : The event's description/notes. Omit or pass missing value to ignore it.

[**has attendees** boolean] : Whether the event has attendees. Omit or pass missing value to ignore it.

[**has recurrence** boolean] : Whether the event is recurring. Omit or pass missing value to ignore it.

[**runs all day** boolean] : Whether the event is an all-day event. Omit or pass missing value to ignore it.

→ list of any : Filtered list of events.

filter events by pattern *v* : Filter a list of events by matching the values with the patterns supplied in the following parameters.

filter events by pattern

event list list of any : The list or array of events to filter.

[**event summary** any] : The pattern for the summary/title of the event to match. Omit or pass missing value to ignore it.

[**event description** any] : The pattern for the description/notes of the event. Omit or pass missing value to ignore it.

[**event location** any] : The pattern for the location of the event. Omit or pass missing value to ignore it.

[**using regex** boolean] : Whether to use regular expression matching (true) or simple wildcard matching (where ? is a wildcard for single characters and * is a wildcard for zero or more characters). Default is true.

→ list of any : Filtered list of events.

remove event *v* : Delete an event.

remove event

event any : The event to remove.

event store any : The event store, as returned by the 'fetch store' command.

[**future events** boolean] : Whether to also remove any future occurrences if the event is recurring. Default is true.

create event *v* : Create a new event. Once created, you need to use 'store event' to save it.

create event

event store any : The event store, as returned by the 'fetch store' command.

destination calendar any : The calendar for the event.

event summary text : The summary/title for the event.

starting date any : The start date for the event. Can be AppleScript date or Cocoa NSDate.

ending date any : The end date for the event. Can be AppleScript date or Cocoa NSDate.

[**runs all day** boolean] : Whether it should be an all-day event. Default is false.
[**event location** text] : The location for the event. Omit or pass missing value if no location required.
[**event description** text] : The notes/description for the event. Omit or pass missing value if there should be no description.
→ any : The new, unsaved, event.

modify event v : Modify one or more properties of an event. You need to use 'store event' to save the changes.

modify event

event any : The event to modify.
destination calendar any : The calendar for the event. Use this to move an event to a new calendar.
[**event summary** any] : New summary/title for event.
[**starting date** any] : New start date for event. Can be AppleScript date or Cocoa NSDate.
[**ending date** any] : New end date for event. Can be AppleScript date or Cocoa NSDate.
[**runs all day** boolean] : Whether it should be an all-day event.
[**event location** text] : New location for the event.
[**event description** any] : New notes/description for event.
→ any : The event.

modify zone v : Modify the time zone of an event. You need to use 'store event' to save the changes.

modify zone

event any : The event to modify.
time zone any : The time zone can be missing value (for floating times), a valid time zone name, or a Cocoa NSTimeZone.
→ any : The event.

modify recurrence v : Modify the recurrence rule of an event. Specify either the 'repeats for' or the 'repeats until' parameter. You need to use 'store event' to save the changes.

modify recurrence

event any : The event to modify.
event frequency [do daily/do weekly/do monthly/do yearly](#) : Occurrence frequency. One of do daily/do weekly/do monthly/do yearly.
event interval integer : Occurrence interval.
[**repeats for** integer] : Numer of times to repeat.
[**repeats until** any] : Date to end repeating. Can be AppleScript date or Cocoa NSDate.
→ any : The event.

event frequency enum : Event frequency.

do daily : Daily.
do weekly : Weekly.
do monthly : Monthly.
do yearly : Yearly.

stop recurrence *v* : Turns off recurrence for event. You need to use 'store event' to save the changes.

stop recurrence

event any : The event to modify.

→ any : The event.

store event *v* : Saves an event after it has been created or modified.

store event

event any : The event to save.

event store any : The event store.

[future events] boolean : Whether changes should also be made to future occurrences of a recurrent event. Default is true.

event identifier for *v* : Get the UID of the event, as used by Calendar.app.

event identifier for

event any : The event.

→ text : The UID.

event info for *v* : Returns the properties of an event.

event info for

event any : The event.

→ record : A record containing the main properties of the event.

event attendees for *v* : Returns a list of the properties of attendees. Attendee details cannot be modified, alas.

event attendees for

event any : The event.

→ list of record : A list of records containing the properties of an attendees.

event recurrence for *v* : Returns details of the recurrence rule for the event.

event recurrence for

event any : The event.

→ record : A record describing the recurrence rule of the event, or missing value if the event is not recurring.