

**NAME**

**icalBuddy** — query the OS X calendar database for items

**SYNOPSIS**

**icalBuddy** [**-V****ufnd**] [**--strEncoding** *encoding*] [**-sc**] [**-sd**] [**-nc**] [**-nrd**] [**-nnpn**] [**-n**] [**-sed**] [**-uid**] [**-eed**] [**-li** *num*] [**-std**] [**-stda**] [**-tf** *format*] [**-df** *format*] [**-po** *value*] [**-ps** *value*] [**-b** *value*] [**-ab** *value*] [**-ss** *value*] [**-ic** *value*] [**-ec** *value*] [**-iep** *value*] [**-eep** *value*] [**-itp** *value*] [**-etp** *value*] [**-cf** *path*] [**-lf** *path*] [**-nnr** *value*] *command*

**DESCRIPTION**

**icalBuddy** queries the OS X calendar database for items and outputs them.

The options are as follows:

*command* Specifies the general action to take. Possible values are:

<b>eventsToday</b>	Print events occurring today
<b>eventsToday+NUM</b>	Print events occurring between today and NUM days into the future, where NUM is an integer
<b>eventsNow</b>	Print events occurring at present time
<b>eventsFrom:START to:END</b>	Print events occurring between the two specified dates. The dates (START and END) may be specified in a natural language form (such as <code>tomorrow at noon</code> or <code>june 10 at 6 pm</code> ) or as relative dates (such as <code>today+3</code> or <code>yesterday-2</code> ) but the safest format is <code>YYYY-MM-DD HH:MM:SS +HHMM</code> (the <code>+HHMM</code> part at the end specifies the timezone offset in hours and minutes from GMT, and it may have a plus or minus prefix)
<b>uncompletedTasks</b>	Print uncompleted tasks
<b>tasksDueBefore:DATE</b>	Print uncompleted tasks that are due before the given date. DATE may be specified in a natural language form (such as <code>tomorrow at noon</code> or <code>june 10 at 6 pm</code> ) or as a relative date (such as <code>today+3</code> or <code>yesterday-2</code> ) but the safest format is <code>YYYY-MM-DD HH:MM:SS +HHMM</code> (the <code>+HHMM</code> part at the end specifies the timezone offset in hours and minutes from GMT, and it may have a plus or minus prefix)
<b>calendars</b>	Print a list of all the calendars
<b>strEncodings</b>	Print all the possible string encodings
<b>editConfig</b>	Open the configuration file for editing in a GUI editor application
<b>editConfigCLI</b>	Open the configuration file for editing in a command-line editor

**-V, --version**

Print version number and exit.

**-u, --checkForUpdates**

Check for updates to self. Also allows for automatic updating of self (with minimal user interaction, straight from the command line.)

**-d, --debug**

Print diagnostic information into the standard error stream (for example, how input dates are interpreted).



- sc, --separateByCalendar**  
Separate events/tasks in the output by calendar.
- sd, --separateByDate**  
Separate events/tasks in the output by date.
- f, --formatOutput**  
Use ANSI escape sequences to format the output and hopefully make it more readable.
- nc, --noCalendarNames**  
Omit calendar names from the output. Does not apply if the **-sc** argument is used.
- nrd, --noRelativeDates**  
Never use “natural language relative dates” (e.g. “today”, “yesterday” or “day after tomorrow”) instead of the default date format presentation (see the **-df** argument).
- nnp, --noPropertyNames**  
Omit property names from the output (i.e. only show property values — e.g. “My house” instead of “location: My house”).
- n, --includeOnlyEventsFromNowOn**  
If the *command* value `eventsToday` is used, only output events that occur between the current time and the end of the day (as opposed to events that occur between the start and end of the day, as by default.)
- sed, --showEmptyDates**  
Display sections for empty dates (i.e. dates that have no items) that fall within the specified range as well. Applies only if the **-sd** argument is used.
- uid, --showUIDs**  
Display event/task UIDs (unique identifiers).
- eed, --excludeEndDates**  
Do not output the dates/times of when events end.
- li, --limitItems *num***  
Limit the number of printed items (events/tasks) in the output to a maximum of *num* items.
- std, --sortTasksByDate**  
Sort tasks by their due dates, in descending order.
- stda, --sortTasksByDateAscending**  
Sort tasks by their due dates, in ascending order.
- strEncoding *encoding***  
Use the specified string encoding for the output. Run the app once with the ‘strEncodings’ *command* value to see all the possible values you can use here.
- tf, --timeFormat *format***  
Format times included in the output based on the *format* formatting string. See <http://tinyurl.com/b9mgyp> [apple.com] for documentation on the syntax to use.
- df, --dateFormat *format***  
Format dates included in the output based on the *format* formatting string. See <http://tinyurl.com/b9mgyp> [apple.com] for documentation on the syntax to use. (note: in addition to the standard datetime element specifiers, you can also use the %RW specifier, which stands for “relative week” (e.g. “next week”) and the %RD specifier which stands for “relative day” (e.g. “5 days from now”).
- po, --propertyOrder *value***  
Print properties of events/tasks in the order specified by *value*, which must be a comma-separated list of property names. The allowed property names are: title, location, notes, url, datetime, priority.



**-ps, --propertySeparators *value***

Use the strings specified in *value* as the separators between item properties in the output. *value* must be a list of strings where each component is separated by an arbitrary character that is present also as the first and the last character in the value (that means you can use any character you'd like as the separator, as long as you also use it at the beginning and the end.) The components specified in *value* will be used in order, for each item, starting from the first printed property. If the number of printed properties for an item is bigger than the number of components in *value*, the last component is used as the separator for the remaining properties. For example, a value of `|, | - |` would specify `,` as the separator between the first and the second property and `-` as the separator between all the rest of the properties. The default is `|\n |` (i.e. use a newline and four spaces as the separator for all properties.)

**-b, --bullet *value***

Sets *value* as the string to be used as the bullet point in the output. The default is `* .`

**-ab, --alertBullet *value***

Sets *value* as the string to be used as the alert bullet point in the output. The default is `! .`

**-ss, --sectionSeparator *value***

Sets *value* as the string to be used as the section separator in the output. The default is a newline followed by a bunch of hyphens (`-`).

**-ic, --includeCals *value***

Includes only items from specific calendars into the output. *value* must be a comma-separated list of calendar titles or UIDs (you can see UIDs (unique identifiers) for your calendars by using the 'calendars' *command* parameter.) The **-ic** and **-ec** parameters will be handled in the order of first include, then exclude.

**-ec, --excludeCals *value***

Excludes items from specific calendars from the output. *value* must be a comma-separated list of calendar titles or UIDs (you can see UIDs (unique identifiers) for your calendars by using the 'calendars' *command* parameter.) The **-ic** and **-ec** parameters will be handled in the order of first include, then exclude.

**-iep, --includeEventProps *value***

Includes only event properties specified by *value* into the output. *value* must be a comma-separated list of property names. Possible property names are: location, url, notes, datetime, title.

**-eep, --excludeEventProps *value***

Excludes event properties specified by *value* from the output. *value* must be a comma-separated list of property names. See **-iep** for a list of possible property names. A value of `*` will exclude all properties and make sure just the title is printed.

**-itp, --includeTaskProps *value***

Includes only task properties specified by *value* into the output. *value* must be a comma-separated list of property names. Possible property names are: url, notes, datetime and priority.

**-etp, --excludeTaskProps *value***

Excludes task properties specified by *value* from the output. *value* must be a comma-separated list of property names. See **-itp** for a list of possible property names. A value of `*` will exclude all properties and make sure just the title is printed.

**-cf, --configFile *path***

Use the configuration file located at *path* instead of the default one (`~/icalBuddyConfig.plist`). Pass in an empty string (`""`) to make icalBuddy ignore the configuration file completely.

**-lf, --localizationFile *path***

Use the localization file located at *path* instead of the default one (`~/icalBuddyLocalization.plist`).



**-nnr, --notesNewlineReplacement** *value*

Replaces newlines in values of the notes property with *value*.

## EXAMPLES

```
$ icalBuddy -sc uncompletedTasks
```

Outputs all uncompleted tasks, separated by calendar

```
$ icalBuddy -sd -ic Work,Home -ec WeekNumbers eventsToday
```

Outputs all events occurring today from all calendars where the title or UID matches 'Work' or 'Home' and the title or UID does not match 'WeekNumbers', separated by date

```
$ icalBuddy eventsFrom:"2009-01-01 00:00:00 +02:00" to:"2009-01-31 23:59:"
```

Outputs all events occurring between the first of january, 2009 and the 31st of january, 2009 (using the timezone GMT+02:00)

## SEE ALSO

*icalBuddyConfig*(1), *icalBuddyLocalization*(1)

## AUTHORS

This man page has been written by Ali Rantakari (<http://hasseg.org>)