

etm cheat sheet

Data

Data items begin with a data type character and continue on one or more lines either until the end of the file is reached or another line is found that begins with a type character. The beginning type character for each item is followed by the item summary and then, perhaps, by one or more **@key value** options.

action	Begins with ~ (tilde). A record of time or money spent.
event	Begins with * (asterick). Happens on a particular day and time.
occasion	Begins with ^ (caret sign). Marks a date such as a holiday, anniversary or birthday.
note	Begins with ! (exclamation point). A record of some useful information.
task	Begins with - (minus sign). Something that needs to be done.
delegated task	Begins with % (percent sign). Something assigned to someone else to be done.
task group	Begins with + (plus sign). Related tasks, some may be prerequisites for others.
inbasket	Begins with \$ (dollar sign). Quick entry to be edited later when time permits.
someday maybe	Begins with ? (question mark).
hidden	Begins with # (hash mark). Hidden from all etm views except folder view.
defaults	Begins with = (equal sign). Sets default options for subsequent entries.

@key value options

@a	alert (see alerts below).
@b	beginby. An integer number of days before the starting datetime to begin displaying an upcoming notification.
@c	context. E.g., errands, home, office, phone.
@d	description. An elaboration of the details of the item.
@e	extent. A time period (see fuzzy datetimes and time periods below).
@f	done;due. Fuzzy datetimes specifying when a task was finished and when it was due.
@h	history of done;due completions. Task groups only.
@g	goto. A file path or url to be opened using the system default application when the user presses <i>Ctrl-G</i> in the details view of the item.
@j	job. Task groups only. A component task.
@k	keyword. A heirarchical classifier for an item using a format such as client:project .
@l	location. The location at which, for example, an event will take place.
@m	memo. Further details about the item not included in the summary or the description.
@o	overdue. Repeating tasks only (see repetition rules below).
@p	priority. Either 0 (no priority) or an integer between 1 (highest priority) and 9 (lowest priority).

@r	repetition rule. The specification of how an item is to repeat (see repetition rules).
@s	starting datetime. When an action is started, an event begins or a task is due.
@t	tags. A tag or a comma separated list of tags.
@u	user. A user or a comma separated list of users.
@v	value. A key from values in etm.cfg used in actions to specify the billing rate to be applied to the time spent.
@w	weight. A key from weights in etm.cfg used in actions to specify the markup rate to be applied to @x expenses.
@x	expense. A currency amount. Used in conjunction with @w markup.
@z	time zone. A time zone such as US/Eastern .
@+	include. Repeated items only. A datetime or list of datetimes to be added to the repetitons generated by the repetition rule. If an explicit times is not entered, 12:00am will be the assumed time.
@-	exclude. Repeated items only. A datetime or list of datetimes to be removed from the repetitons generated by the repetition rule. If an explicit time is not entered, 12:00am will be the assumed time.

alerts

Examples:

@a 10m,5m	Trigger the default alert ten minutes and five minutes before the starting datetime of the item.
@a 1h:s	Trigger a sound alert one hour before the starting datetime.
@a 2d:e;who@what.com;filepath	Send an email to the listed recipient exactly 2 days (48 hours) before the starting time of the item.

The format for each of these:

@a <trigger times>[:action[;arguments]]

Possible values for **action**:

d	display. Execute alert_displaycmd in etm.cfg .
e	email. :e;recipients[;attachments] . Send an email to recipients optionally attaching attachments .
m	message. Display an internal etm message box.
s	sound. Execute alert_soundcmd in etm.cfg .
t	text message. :t[;phonenumber] . Send a text message to phonenumber using the sms settings from etm.cfg . If no numbers are given, then the setting for sms.phone will be used.
v	voice. Execute alert_voicecmd in etm.cfg
p	process. :p;process . Execute process .

Actions **e** and **p** can be combined with other actions in a single alert but not with one another.

fuzzy datetimes and time periods

Suppose, for example, that it is currently 8:30am on Wednesday, November 14, 2012. Then, in any **@key** calling for a datetime, **value** would expand as follows:

mon 2p	2:00pm Monday, November 19
fri	12:00am Friday, November 16.
9a -1/1	9:00am Monday, October 1.
+2/15	12:00am Tuesday, January 15 2013.
8p +7	8:00pm Monday, November 26.
-14	12:00am Monday, November 5.
now	8:30am Wednesday, November 14.

In any **@key** calling for a time period, **value** would expand as follows:

2h30m	2 hours and thirty minutes.
7d	7 days.
45	45 minutes.

repetition rules

The specification of how an item is to repeat. Repeating items must have an **@s** entry as well as one or more **@r** entries. Generated datetimes are those satisfying any of the **@r** entries and falling *on or after* the datetime given in **@s**.

A repetition rule begins with **@r frequency** where **frequency** is one of the following characters:

y	yearly.
m	monthly.
w	weekly.
d	daily.
l	list (a list of datetimes will be provided using @+).

The **@r frequency** entry can, optionally, be followed by one or more **&key value** pairs:

&i	interval (positive integer, default = 1) E.g, with frequency w , interval 3 would repeat every three weeks.
&t	total (positive integer) Include no more than this total number of repetitions.
&s	bysetpos (integer). When multiple dates satisfy the rule, take the date from this position in the list, e.g, &s 1 would choose the first element and &s -1 the last.
&u	until (datetime). Only include repetitions falling <i>before</i> (not including) this datetime.
&M	bymonth (1, 2, ..., 12)
&m	bymonthday (1, 2, ..., 31)
&W	byweekno (1, 2, ..., 53)
&w	byweekday (English weekday abbreviation SU ... SA). Use, e.g., 3WE for the third Wednesday or -1FR for the last Friday in each month.
&h	byhour (0 ... 23)
&n	byminute (0 ... 59)

examples

@r d &h 10, 14 18, 22:	Daily at 10am, 2pm, 6pm and 10pm.
@r y &i 4 &M 11 &m range(2,9) &w TU:	The first Tuesday after a Monday in November every four years (presidential election day).
@r m &w MO, TU, WE, TH, FR &m -1, -2, -3 &s -1:	The last weekday of each month. (The &s -1 entry extracts the last date which is both a weekday and falls within the last three days of the month.)

overdue

A repeating *task* may optionally also include an **@o <k|s|r>** entry (default: **k**):

- @o **k** Keep the current due date if it becomes overdue and use the next due date from the recurrence rule if it is finished early.
- @o **r** Restart the repetitions based on the last completion date.
- @o **s** Skip overdue due dates and set the due date for the next repetition to the first due date from the recurrence rule on or after the current date.

reports

A *report specification* is created by entering a report type character followed by a groupby setting and, perhaps, by one or more report options. Together, the type character, groupby setting and options determine which items will appear in the report and how they will be organized and displayed.

There are two possible report type characters, *a* and *c*:

- a**: actions only with totals.
- c**: any item types without totals.

groupby

A semicolon separated list of elements that determine how items will be grouped and sorted. Possible elements include *date specifications* and elements from

- c** context.
- f** file path.
- k** keyword.
- t** tag.
- u** user.

A *date specification* is either **w** which groups by week using a 4-digit year, week number and date range as the header, e.g., 2014 Week 18: Apr 28 - May 4, or a combination of one or more of the following:

- yy** 2-digit year, e.g., 13.
- yyyy** 4-digit year, e.g., 2013.
- M** month, 1 - 12.
- MM** month, 01 - 12.
- MMM** locale specific abbreviated month name, e.g., Jan.
- MMMM** locale specific month name, e.g., January.
- dd** month day, 01 - 31.
- ddd** locale specific abbreviated week day, e.g, Mon.
- dddd** locale specific week day, e.g., Monday.

For example, suppose that keywords have the format `client:project`. Then `c MMM yyyy; k[0]; k[1] ...` would group by year and month, then client and finally project:

```
Apr 2011
  client a
    project 1
      items for client a, project 1 in April
    project 2
      items for client a, project 2 in April
  client b
    project i
      items for client b, project i in April
  ...
```

Items that are missing an element specified in `groupby` will be omitted from the output, e.g., items without `keywords` will be omitted if `k` is included. Similarly, undated items will be omitted when a date specification is included.

omit

Show/hide a)ctions, d)elegated tasks, e)vents, g)roup tasks, n)otes, o)ccasions and/or other t)asks. E.g. use `-o on` to omit

occasions and notes and `-o !on` to show only occasions and notes.

options

Report options are listed below. Report type `c` supports all options except `-d`. Report type `a` supports all options except `-o` and `-h`.

- b** begin (datetime). Limit the display of dated items to those with datestimes falling *on or after* this datetime.
- c** context (regular expression).
- d** depth (integer). CLI only. The default, `-d 0`, includes all outline levels. Use `-d 1` to include only level 1, `-d 2` to include levels 1 and 2 and so forth.
- e** end (datetime). Limit the display of dated items to those with datetimes falling *before* this datetime.
- f** file (relative file path).
- h** hue (0, 1 or 2). `-h 2`, uses all possible colors for leaf fonts, `-h 1` uses red for past due items and black for everything else and `-h 0` uses black for everything.
- k** keyword (regular expression).
- l** location (regular expression).
- o** omit (see omit below).
- s** summary (regular expression).
- t** tags (comma separated list of regular expressions).
- u** user (regular expression).
- w** width1 (integer number of characters for column 1).
- W** width2 (integer number of characters for column 2).

With regular expressions, you can use `!` (exclamation point) as a prefix to negate the result. E.g., `-t tag1, !tag2` would select items with one or more tags that match `tag1` but none that match `tag2`.