9 Capture - Refile - Archive

An important part of any organization system is the ability to quickly capture new ideas and tasks, and to associate reference material with them. Org uses the '`remember.el`' package to create tasks, and stores files related to a task (*attachments*) in a special directory. Once in the system, tasks and projects need to be moved around. Moving completed project trees to an archive file keeps the system compact and fast.

9.1 Remember

The Remember package by John Wiegley lets you store quick notes with little interruption of your work flow. It is an excellent way to add new notes and tasks to Org files. The `remember.el` package is part of Emacs 23, not Emacs 22. See http://www.emacswiki.org/cgi-bin/wiki/RememberMode for more information.

Org significantly expands the possibilities of Remember: you may define templates for different note types, and associate target files and headlines with specific templates. It also allows you to select the location where a note should be stored interactively, on the fly.

9.1.1 Setting up Remember for Org

The following customization will tell Remember to use Org files as target, and to create annotations compatible with Org links.

```
(org-remember-insinuate)
(setq org-directory "~/path/to/my/orgfiles/")
(setq org-default-notes-file (concat org-directory "/notes.org"))
(define-key global-map "\C-cr" 'org-remember)
```

The last line binds the command `org-remember` to a global key[1]. `org-remember` basically just calls Remember, but it makes a few things easier: if there is an active region, it will automatically copy the region into the Remember buffer. It also allows to jump to the buffer and location where Remember notes are being stored: just call `org-remember` with a prefix argument. If you use two prefix arguments, Org jumps to the location where the last remember note was stored.

The Remember buffer will actually use `org-mode` as its major mode, so that all editing features of Org mode are available. In addition to this, a minor mode `org-remember-mode` is turned on, for the single purpose that you can use its keymap `org-remember-mode-map` to override some of Org mode's key bindings.

You can also call `org-remember` in a special way from the agenda, using the *`k r`* key combination. With this access, any timestamps inserted by the selected Remember template (see below) will default to the cursor date in the agenda, rather than to the current date.

[1] Please select your own key, *`C-c r`* is only a suggestion.

9.1.2 Remember templates

In combination with Org, you can use templates to generate different types of Remember notes. For example, if you would like to use one template to create general TODO entries, another one for journal entries, and a third one for collecting random ideas, you could use:

```
(setq org-remember-templates
 '(("Todo" ?t "* TODO %?\n  %i\n  %a" "~/org/TODO.org" "Tasks")
   ("Journal" ?j "* %U %?\n\n  %i\n  %a" "~/org/JOURNAL.org")
   ("Idea" ?i "* %^{Title}\n  %i\n  %a" "~/org/JOURNAL.org" "New Ideas")))
```

In these entries, the first string is just a name, and the character specifies how to select the template. It is useful if the character is also the first letter of the name. The next string specifies the template. Two more (optional) strings give the file in which, and the headline under which, the new note should be stored. The file (if not present or `nil`) defaults to `org-default-notes-file`, the heading to `org-remember-default-headline`. If the file name is not an absolute path, it will be interpreted relative to `org-directory`.

The heading can also be the symbols `top` or `bottom` to send notes as level 1 entries to the beginning or end of the file, respectively. It may also be the symbol `date-tree`. Then, a tree with year on level 1, month on level 2 and day on level three will be built in the file, and the entry will be filed into the tree under the current date[2]

An optional sixth element specifies the contexts in which the user can select the template. This element can be a list of major modes or a function. `org-remember` will first check whether the function returns `t` or if we are in any of the listed major modes, and exclude templates for which this condition is not fulfilled. Templates that do not specify this element at all, or that use `nil` or `t` as a value will always be selectable.

So for example:

```
(setq org-remember-templates
 '(("Bug" ?b "* BUG %?\n  %i\n  %a" "~/org/BUGS.org" "Bugs" (emacs-lisp-mode))
   ("Journal" ?j "* %U %?\n\n  %i\n  %a" "~/org/JOURNAL.org" "X" my-check)
   ("Idea" ?i "* %^{Title}\n  %i\n  %a" "~/org/JOURNAL.org" "New Ideas")))
```

The first template will only be available when invoking `org-remember` from a buffer in `emacs-lisp-mode`. The second template will only be available when the function `my-check` returns `t`. The third template will be proposed in any context.

When you call *M-x org-remember* (or *M-x remember*) to remember something, Org will prompt for a key to select the template (if you have more than one template) and then prepare the buffer like

```
* TODO
  [[file:link to where you called remember]]
```

During expansion of the template, special `%`-escapes[3] allow dynamic insertion of content:

`%^{prompt}`	prompt the user for a string and replace this sequence with it. You may specify a default value and a completion table with %^{prompt|default|completion2|completion3...}

[2] If the file contains an entry with a `DATE_TREE` property, the entire date tree will be built under that entry.

[3] If you need one of these sequences literally, escape the `%` with a backslash.

	The arrow keys access a prompt-specific history.
`%a`	annotation, normally the link created with `org-store-link`
`%A`	like `%a`, but prompt for the description part
`%i`	initial content, the region when remember is called with C-u. The entire text will be indented like `%i` itself.
`%t`	timestamp, date only
`%T`	timestamp with date and time
`%u, %U`	like the above, but inactive timestamps
`%^t`	like `%t`, but prompt for date. Similarly `%^T`, `%^u`, `%^U` You may define a prompt like `%^{Birthday}t`
`%n`	user name (taken from `user-full-name`)
`%c`	Current kill ring head.
`%x`	Content of the X clipboard.
`%^C`	Interactive selection of which kill or clip to use.
`%^L`	Like `%^C`, but insert as link.
`%k`	title of the currently clocked task
`%K`	link to the currently clocked task
`%^g`	prompt for tags, with completion on tags in target file.
`%^G`	prompt for tags, with completion all tags in all agenda files.
`%^{`*prop*`}p`	Prompt the user for a value for property *prop*
`%:keyword`	specific information for certain link types, see below
`%[`*file*`]`	insert the contents of the file given by *file*
`%(`*sexp*`)`	evaluate Elisp *sexp* and replace with the result
`%!`	immediately store note after completing the template (skipping the `C-c C-c` that normally triggers storing)
`%&`	jump to target location immediately after storing note

For specific link types, the following keywords will be defined[4]:

```
Link type          |  Available keywords
-------------------+----------------------------------------------
bbdb               |  %:name %:company
bbdb               |  %::server %:port %:nick
vm, wl, mh, rmail  |  %:type %:subject %:message-id
                   |  %:from %:fromname %:fromaddress
                   |  %:to   %:toname   %:toaddress
                   |  %:fromto (either "to NAME" or "from NAME")[5]
gnus               |  %:group, for messages also all email fields
w3, w3m            |  %:url
info               |  %:file %:node
calendar           |  %:date"
```

To place the cursor after template expansion use:

`%?`	After completing the template, position cursor here.

[4] If you define your own link types (see Section A.3 [Adding hyperlink types], page 151), any property you store with `org-store-link-props` can be accessed in remember templates in a similar way.

[5] This will always be the other, not the user. See the variable `org-from-is-user-regexp`.

If you change your mind about which template to use, call `org-remember` in the remember buffer. You may then select a new template that will be filled with the previous context information.

9.1.3 Storing notes

When you are finished preparing a note with Remember, you have to press *C-c C-c* to file the note away. If you have started the clock in the Remember buffer, you will first be asked if you want to clock out now[6]. If you answer *n*, the clock will continue to run after the note was filed away.

The handler will then store the note in the file and under the headline specified in the template, or it will use the default file and headline. The window configuration will be restored, sending you back to the working context before the call to Remember. To re-use the location found during the last call to Remember, exit the Remember buffer with *C-0 C-c C-c*, i.e. specify a zero prefix argument to *C-c C-c*. Another special case is *C-2 C-c C-c* which files the note as a child of the currently clocked item, and *C-3 C-c C-c* files as a sibling of the currently clocked item.

If you want to store the note directly to a different place, use *C-1 C-c C-c* instead to exit Remember[7]. The handler will then first prompt for a target file—if you press ⟨RET⟩, the value specified for the template is used. Then the command offers the headings tree of the selected file, with the cursor position at the default headline (if you specified one in the template). You can either immediately press ⟨RET⟩ to get the note placed there. Or you can use the following keys to find a different location:

⟨TAB⟩	Cycle visibility.
⟨down⟩ / ⟨up⟩	Next/previous visible headline.
`n / p`	Next/previous visible headline.
`f / b`	Next/previous headline same level.
`u`	One level up.

Pressing ⟨RET⟩ or ⟨left⟩ or ⟨right⟩ then leads to the following result.

Cursor position	**Key**	**Note gets inserted**
on headline	⟨RET⟩	as sublevel of the heading at cursor, first or last depending on `org-reverse-note-order`.
	⟨left⟩/⟨right⟩	as same level, before/after current heading
buffer-start	⟨RET⟩	as level 2 heading at end of file or level 1 at beginning depending on `org-reverse-note-order`.
not on headline	⟨RET⟩	at cursor position, level taken from context.

Before inserting the text into a tree, the function ensures that the text has a headline, i.e. a first line that starts with a '`*`'. If not, a headline is constructed from the current date. If you have indented the text of the note below the headline, the indentation will be adapted if inserting the note into the tree requires demotion from level 1.

---

[6] To avoid this query, configure the variable `org-remember-clock-out-on-exit`.

[7] Configure the variable `org-remember-store-without-prompt` to make this behavior the default.