

Managing Your Schedule

If you've ever been late for an appointment, accidentally double-booked your time, or missed an important birthday or anniversary, you know you need to keep a schedule. Whether you carry a paper planner or encode your appointments on your phone, keep a personal calendar on your computer or share it with others on the Web, you may find it useful to keep your schedule information in Emacs.

You can set up Emacs to remind you before an appointment so that you don't get lost in your code. You can review your weekly agenda for upcoming events and milestones so that you can send a postcard in time for your best friend's birthday or plan your tasks to avoid that last-minute crunch. Here are some reasons why you might want to manage your schedule inside Emacs:

- You can spend less time and mental effort switching between applications when checking your schedule.
- You can customize your schedule manager to fit your quirks.
- You can copy, back up, and process the plain-text data using other programs and scripts.

Using Emacs to keep track of your schedule works best if you spend a lot of time working with Emacs. Check out the previous chapters for tips on moving more of your life into Emacs. If you spend most of your time with an Emacs window in the foreground, then you'll notice appointment reminders and you'll be encouraged to keep more information in it.

Emacs is still helpful even if you spend most of your time away from the computer. Write down the day's schedule in a notebook or on an index card and take that with you. As you get new tasks and events through the day, write those notes down. Review and update your schedule at night, and plan the next day as well. This helps you plan ahead. Using a computer to keep track of your schedule provides you with a searchable record and allows you to schedule things months or years in advance.

There are lots of ways to use Emacs to keep track of your schedule. A flat text file with dates and details works if you're using the One Big File approach to personal information management. Emacs has many modules to support more sophisticated time management. **Diary** and **Appointment** are built into Emacs and allow you to keep track of one-time and recurring

events such as birthdays, holidays, and taxes. **Org** and **Planner** offer more sophisticated ways to manage your tasks, schedule, and notes.

Choosing a Schedule Manager

To help you decide, here are some examples of what daily and weekly calendars look like in Diary, Org, and Planner:

Diary

Data / simple diary display

```
Nov 26, 2007 2pm Mentoring call
%%(diary-cyclic 14 10 5 2007) Payday (%d%s check, hooray!)
Nov 26, 2007 9:00-12:00 Client meeting
Nov 29, 2007 8:30am-9:00am Weekly update call with manager
```

Fancy diary display for 7 days, appointments sorted by time

```
Monday, November 26, 2007
=====
 9:00-12:00 Client meeting
 2pm Mentoring call

Thursday, November 29, 2007
=====
8:30am-9:00am Weekly update call with manager

Friday, November 30, 2007
=====
Payday (4th check, hooray!)
```

Diary and cal-desk-calendar.el

Monday, November 26, 2007

=====

8:00
:30
9:00 | Client meeting
:30 |
10:00 |
:30 |
11:00 |
:30 |
12:00
:30
13:00
:30
14:00 | Mentoring call
:30
15:00
:30
16:00
:30
17:00

Diary and weekly-view.el

	Sunday	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday	
8:00								
8:30					Weekly up			
9:00		Client me						
9:30		eting####						
10:00		#####						
10:30		#####						
11:00		#####						
11:30		#####						
12:00								
12:30								
13:00								
13:30								
14:00		Mentoring						
14:30		call####						
15:00								
15:30								
16:00								
16:30								
17:00								
17:30								

Org

Data

Org data can be stored in a Diary file or an Org agenda file similar to this:

* Mentoring call

<2007-11-26 Mon 2:00pm>

phone details, other notes go here

* Client meeting

<2007-11-26 Mon 9:00>--<2007-11-26 Mon 12:00>

* Weekly update call with manager

<2007-11-29 Thu 8:30>--<2007-11-26 Mon 9:00>

- agenda item 1

- agenda item 2

Day view

Day-agenda:

Monday 26 November 2007

8:00..... -----

Diary: 9:00-12:00 Client meeting

10:00..... -----

12:00..... -----

14:00..... -----

Diary: 14:00..... Mentoring call

16:00..... -----

18:00..... -----

organizer: Scheduled: **TODO** [#A] Plan my week

organizer: Scheduled: **TODO** [#A] 10min Plan my day

organizer: Scheduled: **TODO** [#C] Check the weather for the week

Week view

Week-agenda:

Monday 26 November 2007

Diary: 9:00-12:00 Client meeting

Diary: 14:00..... Mentoring call

organizer: Scheduled: **TODO** [#A] Plan my week

organizer: Scheduled: **TODO** [#A] 10min Plan my day

organizer: Scheduled: **TODO** [#C] Check the weather for the week

Tuesday 27 November 2007

organizer: Scheduled: **TODO** [#A] 10min Plan my day

Wednesday 28 November 2007

organizer: Scheduled: **TODO** [#A] 10min Plan my day

Thursday 29 November 2007

Diary: 8:30- 9:00 Weekly update call with manager

organizer: Scheduled: **TODO** [#A] 10min Plan my day

Friday 30 November 2007

Diary: Payday (4th check, hooray!)

organizer: 17:00-17:30 Scheduled: **TODO** 30min Report time

organizer: Scheduled: **TODO** [#A] 10min Plan my day

Saturday 1 December 2007

organizer: Scheduled: **TODO** [#A] 10min Plan my day

Sunday 2 December 2007

organizer: Scheduled: **TODO** [#A] 10min Plan my day

Planner (with planner-appt, schedule sorted by time)

*** Tasks**

#A _ Plan my week

#A _ 10min Plan my day

#C _ Check the weather for this week

*** Schedule**

@09:00 | 12:00 | Client meeting

@2:00pm | | Mentoring call

*** Notes**

Planner doesn't come with a built-in weekly view, but you can code one. In **Creating and Viewing Appointments**, you'll learn how to create a view like this:

*** Monday 2007.11.26**

@09:00 | 12:00 | Client meeting

@2:00pm | | Mentoring call

*** Thursday 2007.11.27**

@08:30 | 09:00 | Weekly update call with manager

Haven't decided based on the screenshots? Here's more help.

If you're just starting out, Diary can be a good fit for you. All it takes is one text file specifying the dates and descriptions of one-time or recurring events, and one line in your ~/.emacs file to display the diary entries for today whenever you start up. If you switch to either Org or Planner later on, you can configure your Emacs to include information from your old

diary file. If you want to import or export to other formats, use Diary for your appointments. More modules have been written to work with Diary data than with the other schedule managers.

If you want more integration with your task list and your notes so that you can see everything related to the day, then go with either Org or Planner. The choice between the two depends on how you want to plan your day.

If you've worked with a paper-based planner or a typical calendar application for a while, then Planner might be a good fit for the way you think. Planner has a strong day orientation. With Planner, you plan your schedule in terms of day pages, and optionally cross-reference the tasks with project pages. You can keep your schedule as plain text on your day page, or you can follow the format suggested by Planner so that your scheduled tasks and appointments will be added to Emacs' appointment reminder system. Planner is similar to the scheduling method of Franklin-Covey day planners and other paper organizers. I like Planner because it's easy to scribble notes onto my daily schedule, just like I would pencil comments into my paper planner.

Org focuses on projects instead of days. With Org, you plan your projects first, and then schedule specific tasks or appointments onto your calendar. Your daily view is then dynamically generated from the headlines of the outline nodes that are scheduled for the day. You need to use a special format, but built-in commands make it easy to create and edit timestamps. Org has great support for David Allen's Getting Things Done (GTD) productivity technique. If you think in terms of next actions or outlines, Org is a good fit for you.

Both modules are flexible. You can still think in terms of projects with Planner, but you'll need to remember to use the built-in commands for editing functions so that your changes are also reflected in the day pages. You don't need to organize your Org file by project. You can throw everything into one outline if you want. However, you'll still need to use the dynamically-generated summary view to see your daily or weekly schedule, and reading all the text associated with the day could mean some jumping around.

In short:

- If you don't need a combined view of your tasks and schedule, use Diary.
- If you need to synchronize with other calendar formats, use Diary.
- If you work with recurring events, use Diary.

- If you think in terms of projects or outlines, start with Org.
If you think in terms of day schedules, start with Planner.
- If you want to publish your tasks, schedules, or notes, start with Planner.

If you haven't figured out how your brain works yet, I recommend starting with Org. I think that the project-based approach will probably help more people out in the long run. I have to admit that I love Planner and I've been using it very happily for more than four years. I love being able to add all sorts of free-form notes while going through the day, and Planner makes it easy to publish that as my blog. (Automatically removing all my private notes, of course!) I've gotten used to the way Planner works, and because of my extensive modifications, Planner's gotten used to the way I work as well. If you're starting from scratch, you might like Org's better support for projects, outlines, and overviews. Give Org a try first. If you find yourself wanting more flexibility in your day view, try Planner next.

General Tips

Here are some general productivity tips for getting the most out of Emacs scheduling:

- **WARNING:** Emacs will happily let you create appointments that overlap. Check your schedule before committing to anything.
- Make it easy to add to your schedule. Bind your appointment-creation functions to shortcut keys and practice using them until you've memorized where they are.
- Make it easy to review your schedule for the day by binding the appropriate command to a shortcut like C-c s.
- Set up an automatic reminder system such as Appt. If you can trust that your reminder system will interrupt you even if you're focusing on something else, you'll worry less about missing appointments. You'll find it easier to stay focused because you won't need to check your schedule as often.
- If you need to be reminded earlier than normal, create another, earlier appointment. For example, if the commute to a meeting takes half an hour, you don't want to be reminded of the meeting only 10 minutes before it starts. Create another appointment that starts at least half an hour before so that you'll be on your way. It might be a good idea to start even earlier.

- Planning ahead helps you save time and avoid stress. Plan your day the night before, or in the morning if necessary. During the weekend, review the upcoming week, making sure to schedule in time for your important tasks.

Creating and Viewing Appointments

Diary

Before you create any diary entries, decide whether you're going to use the American calendar system (month, day, year), or the European calendar system (day, month, year). By default, Emacs uses the American calendar system. To switch to the European calendar system, use `M-x customize` to set `european-calendar-style` to `t`. You only need to do this once, but do this before you create diary entries, or your diary file will get hopelessly confused.

Diary reads appointment information from your diary-file, which is `~/diary` by default. You can create a diary entry from the Emacs Calendar, or edit the diary file directly. The Emacs Calendar tends to be easier and less error-prone. Here's how:

1. Display the calendar with `M-x calendar`, or jump to a specific month and year with `C-u M-x calendar`.
2. Move the cursor to the date for your appointment with `g d` (`calendar-goto-date`), the arrow keys, or these shortcuts:

<i>Movement</i>	<i>Backward</i>	<i>Forward</i>
Day	C-b	C-f
Week	C-n	C-p
Month	M- {	M- }
Year	C-x [C-x]

3. Create an appointment by typing `i d`, an optional time for the appointment, and the appointment description. The date will be filled in from the calendar. After you finish, save the diary file with `C-x C-s` (`save-buffer`). For example:

Nov 4 2007 3pm Meet Jane Doe regarding FooBar

You can also edit the diary file directly. Be careful! Some diary lines may be invisible, and deleting a line may delete invisible entries as well. To be safe, use `M-x diary-show-all-entries` before editing the diary file, or type `s` (`diary-show-all-entries`) in the calendar view (`M-x calendar`).

I like the fancy diary display much than the default diary display. You should use it if you're planning to use advanced Diary features, such as calculated displays or included Diary files. You can enable it with:

```
(add-hook 'diary-display-hook 'fancy-diary-display)
(add-hook 'list-diary-entries-hook 'sort-diary-entries t)
(add-hook 'list-diary-entries-hook 'include-other-diary-files)
(add-hook 'mark-diary-entries-hook 'mark-included-diary-files)
```

To view your appointments, use `M-x diary`. You can use `M-x calendar` to browse different dates, using `d` (`diary-view-entries`) to see the diary entries for the selected day. To see the next seven days of appointments whenever you start up, add the following line to the end of your `~/.emacs`, after all your Diary-related code:

```
(diary 7)
```

You can also view your schedule by typing `C-u 7 M-x diary`. This shows the diary entries for the next seven days.

Diary, cal-desk-calendar.el, and weekly-view.el

For an even fancier bar-graph views, you can use the following modules:

- <http://wickedcolemacs.nsp.com/lisp/cal-desk-calendar.el>
- <http://wickedcooleamcs.nsp.com/lisp/weekly-view.el> (requires cal-desk-calendar)

Add these files to a directory in your load-path. To set `M-x diary` to show the cal-desk-calendar bar graph for the day, use the following code:

```
(require 'cal-desk-calendar)
(add-hook 'diary-display-hook 'sort-diary-entries)
(add-hook 'diary-display-hook 'fancy-schedule-display-desk-calendar t)
```

To view the weekly bar graph, load the weekly-view library with:

```
(require 'weekly-view)
```

and then use `w` (`week-graph-view-diary-entries`) in the Calendar view (`M-x calendar`) to see the bar graph for the week.

Org

You can store your appointments in Diary and include them in your Org agenda, or you can store your appointments in your Org file. If you use Org for your appointments, then you can edit and organize them more easily. But if you have a number of appointments in Diary or you use Diary to import or export data to other calendaring applications, just use `M-x customize` to set `org-agenda-include-diary` to `t`, and create new appointments using the instructions for Diary. The diary entries will be included in your agenda view.

On the other hand, storing your appointments in Org instead of Diary makes it easier to keep related notes, appointments, and tasks together. **To keep your appointments in Org**, set up an Org agenda file and some convenient shortcuts. I like using `~/organizer.org` as my organizer agenda file, but you can use any file. Add the following lines to your `~/emacs` and evaluate them:

```
(add-to-list 'auto-mode-alist '(("\\.org" . Org-mode))
(global-set-key "\C-cl" 'org-store-link)
(global-set-key "\C-ca" 'org-agenda)
```

Open your Org agenda file. I like using `~/organizer.org`. Type `C-c [` (`org-agenda-file-to-front`), which adds the file to your agenda files list and saves the list. If you want to remove the current file, type `C-c]` (`org-remove-file`).

Now that you've created an agenda file, you can create appointments by typing in outlines and adding timestamps. Use `C-c .` (`org-time-stamp`) to insert a timestamp at point, or type the timespan in manually. For example, your agenda file could look like this:

```
* Meetings
** Meet Jane Doe regarding FooBar
<2007-11-05 Mon 15:00-16:00>
- Agenda item 1
- Agenda item 2
```

**** Meet J. Random Hacker regarding Bazqux**

<2007-11-05 Mon 10:00>

- Agenda item 1

A more convenient way to create appointments is to use Remember, which you can download from <https://gna.org/projects/remember-el>. As of this writing, Remember is at version 1.9. You can use Remember to create an appointment entry in your Org file from anywhere in Emacs. If you were editing a file, viewing a webpage, reading mail, or doing anything that Org knows about, Org will even automatically insert a hyperlink for you. This is tremendously useful when creating appointments based on e-mail messages, because you can jump back to the e-mail message when you need to review the details.

Unpack Remember and add it to your load path. For example, if the Remember files are in `~/elisp/remember`, use the following Emacs Lisp code:

```
;; Set up Remember
(add-to-list 'load-path "~/elisp/remember")
(require 'remember-autoloads)
(global-set-key "\C-cr" 'remember) ; Bind C-c r to remember
;; Set up Remember for Org
(setq remember-annotation-functions '(org-remember-annotation))
(setq remember-handler-functions '(org-remember-handler))
;; Create a remember template for appointments
(setq org-remember-templates
      '(("Tasks" ?t "* TODO %?\n %i\n %a" "~/organizer.org")
        ("Appointments" ?a "* Appointment: %?\n%^T--%^T\n%i\n %a"
         "~/organizer.org")))
(eval-after-load 'remember
  '(add-hook 'remember-mode-hook 'org-remember-apply-template))
```

When you type `C-c r a` ([remember](#), (a)ppointment), Emacs will prompt you for the start date/time and the end date/time. You can then fill in the rest of the appointment details and type `C-c C-c` to save it to the Org file defined in [org-remember-templates](#), or type `C-u C-c C-c` to select the Org file interactively.

To view your agenda for the week, type `C-c a a` ([org-agenda](#), [org-agenda-list](#)). To change to a daily view, type `d` from the weekly agenda view. If you don't see your agenda items,

make sure that the Org file is included in `org-agenda-files` by opening the file and using C-c [`(org-agenda-file-to-front)` to add it.

Planner

If you haven't already installed Planner, download the latest version of Planner from <http://www.gna.org/projects/planner-el>. As of this writing, Planner is at version 3.4.1. Unpack Planner and add the `lisp` directory to your `load-path`. For example, if the Planner files are in `~/elisp/planner-el`, add the following lines to your `~/.emacs` and evaluate them:

```
(add-to-list 'load-path "~/elisp/planner-el/lisp")
(require 'planner-appt)
(require 'planner-autoloads)
```

One way to keep track of your appointments is to add text to the corresponding day pages. If you format this information in a certain way, `planner-appt` can automatically pick up the information and remind you of upcoming appointments.

Open the planner page for the appropriate day by using M-x `planner-goto` and add your schedule information to a section called "Schedule", which you can create if necessary. Here's what a sample day page looks like:

*** Schedule**

```
@06:00 |           | Wake up
@08:00 | 17:00 | Go to work
@17:30 | 19:00 | Dinner, errands
@19:30 | 21:30 | Trapeze class
```

*** Tasks**

*** Notes**

If `planner-appt` cannot find a Schedule section, it will create one at the beginning of the file (or at the end, depending on `planner-create-section-function`). If you want to control where your schedule information will be, set `planner-day-page-template` so that new pages will have a schedule section. For example, to put your schedule after your tasks, use the following Emacs Lisp code:

```
(setq planner-day-page-template "* Tasks\n\n\n* Schedule\n\n\n* Notes")
```

If you want to use a different name for your schedule, you can set or customize `planner-appt-schedule-section`. For example, to use "Agenda" as the name of the section, use the following Emacs Lisp code:

```
(setq planner-appt-schedule-section "Agenda")  
(setq planner-day-page-template "* Agenda\n\n\n* Tasks\n\n\n* Notes")
```

How is this different from using tasks? Tasks are carried over when you call M-x `plan`, while text that you add to the schedule section is not carried over to new days.

To view your appointments for a particular day, open the page using M-x `planner-goto`. You can bind M-x `planner-goto` to a handy shortcut key like C-c p by adding the following line to your `~/.emacs` and evaluating it:

```
(global-set-key "C-cp" 'planner-goto)
```

To automatically sort the schedule whenever the file is saved, use the following Emacs Lisp code:

```
(require 'planner-appt)  
(setq planner-appt-sort-schedule-on-update-flag t)
```

Planner 3.41 doesn't have a built-in weekly preview, but you can use the following code to create a buffer showing you the next few days of appointments.

```
(defun wicked/planner-show-schedule-preview (start end)  
  "Show the schedule preview from START to END.  
  If called interactively, show the schedule preview for the next 7 days,  
  or the number specified by the prefix argument."  
  (interactive  
    (list (planner-today)  
          (planner-calculate-date-from-day-offset  
            (planner-today)  
            (or current-prefix-arg 7))))  
  ;; Collect the schedule sections  
  (let ((pages (planner-get-day-pages start end))  
        (result ""))
```

```

(save-window-excursion
  (with-planner-update-setup
    (while pages
      (planner-goto (caar pages))
      (planner-appt-update-appt-section)
      (save-restriction
        (widen)
        (planner-narrow-to-section planner-appt-task-appointments-section)
        (goto-char (point-min))
        (forward-line 1) ; skip the heading
        (skip-chars-forward " \t\n")
        (if (not (eobp))
            (setq result
              (concat
                "* " (caar pages) " "
                (calendar-day-name
                 (planner-filename-to-calendar-date (caar pages)))
                "\n"
                (buffer-substring (point) (point-max))
                result))))
          (setq pages (cdr pages))))))
(with-current-buffer (get-buffer-create "*Planner Schedule Preview*")
  (setq muse-current-project (muse-project planner-project))
  (erase-buffer)
  (setq buffer-read-only nil)
  (insert result)
  (planner-mode)
  (pop-to-buffer (current-buffer))))

```

This assembles a temporary buffer with the schedule entries from the specified day pages.

Setting up Appointment Reminders

Appointment reminders in Emacs are handled by the Appt module, which you can load by adding `(require 'appt)` to your `~/.emacs`. Activate Appt for the scheduling module you've chosen, and then configure it to fit your needs. Here's what you need to add to `~/.emacs` and evaluate for the different modules:

Diary

If you store all of your appointment data in your diary file, creating appointment reminders is as easy as adding the following line to your `~/.emacs` and evaluating it:

```
(appt-activate)
```

The appointment list is automatically refreshed at midnight. If you change your diary file, remember to call `M-x appt-activate` afterwards to update the appointment list.

Org

To load today's appointments from your Org agenda files, add the following command to your `~/.emacs`, *after* `org-agenda-files` has been initialized:

```
(org-agenda-to-appt)
```

This loads the appointments for today. If you make any changes or you want to refresh the list, call it again with `M-x org-agenda-to-appt`. To automatically load each day's appointments at midnight, use the following code:

```
(add-hook 'diary-hook 'org-agenda-to-appt)
```

Note that `org-agenda-to-appt` also replaces any appointments you may have defined with `M-x appt-add`. If you want to keep those appointments, add the following code to your `~/.emacs`:

```
(defadvice org-agenda-to-appt (around wickedcool activate)
  "Retain currently set appointments."
  (let ((my/appt-list (copy-sequence appt-time-msg-list)))
    ad-do-it
    (setq appt-time-msg-list
          (appt-sort-list
           (nconc appt-time-msg-list my/appt-list))))))
```

Planner

The following code initializes Planner's appointment support:

```
(planner-appt-use-tasks-and-schedule)
(planner-appt-insinuate)
```

```
(setq planner-appt-task-use-appointments-section-flag t)
(setq planner-appt-update-appts-on-save-flag t)
(setq planner-appt-sort-schedule-on-update-flag t)
```

Customizing Appt

There are a number of options for customizing appointment reminders. To change those options, use M-x `customize-group RET appt RET`. For example, if you want to always see the number of minutes to the next appointment in your modeline, set `appt-message-warning-time` to a large number. To turn off the beeps, set `appt-audible` to `nil`.

Creating Recurring Events

You can set up yearly events, monthly events, daily events, and more complex recurrences. Recurring appointments are best stored in Diary and imported into Org and Planner. This is easy to do in Org, but more complicated in Planner because Planner keeps multiple copies of scheduled events. Read the section on Diary in order to set up your recurring appointments, then read the section specific to your scheduling module to learn more.

Diary

An easy way to create regularly-repeating events in your diary is to create them from the calendar. Use M-x `calendar` to bring up a text calendar, then navigate to the day of week, month, or year for which you want to create a recurring event. Use `i w` (`insert-weekly-diary-entry`), `i m` (`insert-monthly-diary-entry`), or `i y` (`insert-yearly-diary-entry`) to create the entry.

You can also create these events manually by editing your diary file (`~/diary`). For example, my diary file includes:

```
Aug 12 My birthday
* 1, * Pay rent
```

Thursday 6:00 PM Toastmasters

To keep better track of birthdays and anniversaries, use Diary's anniversary feature. You can create yearly anniversaries by bringing up the calendar with M-x `calendar`, navigating to the start date (for example, August 12, 1983) and creating the event with C-u `i a` (`insert-anniversary-`

diary-entry). Type in the event description and save the diary file. You can also type this in manually if you want. The anniversary feature allows you to use the escape sequence %d to show the number of years since the start date. For example:

```
&%(diary-anniversary 8 12 1983) My birthday (%d years old)
```

will result in this fancy diary display for August 12, 2008:

Tuesday, August 12, 2008

=====

My birthday (25 years)

The C-u prefix to i a (insert-anniversary-diary-entry) creates a non-marking event that won't be highlighted in the calendar but will be displayed in the day's diary entries. This makes processing more efficient, as marking events like this in the calendar would mean evaluating the anniversary expression for each day visible. To create a marked event that would be highlighted in the calendar, omit the C-u prefix and use i a (insert-anniversary-diary-entry) in the calendar.

Diary can also help you keep track of conferences, vacations, and other events that span more than one day. This is great because you don't have to create multiple appointments. To create a block appointment:

1. Open the calendar with M-x calendar
2. Mark the start date by navigating to it and pressing C-SPC (set-mark-command).
3. Navigate to the end date.
4. Create a non-marking block entry with C-u i d (insert-block-diary-entry) and fill in the details.

For example, if the diary file included:

```
&%(diary-block 8 9 2008 8 16 2008) Trip to the Philippines
```

then the fancy diary display for August 12, 2008 would be

Tuesday, August 12, 2008

=====

My birthday (25 years)

Trip to the Philippines

Paydays and other bi-weekly events don't happen on a fixed day of the month. To create an entry for a 15-day cycle, use `C-u i c` (insert-cyclic-diary-entry) and specify the number of days. For example, this sets up payday every two weeks starting October 5, 2007 and also tells me how many checks I've received so far:

```
%%(diary-cyclic 14 10 5 2007) Payday (%d%s check, hooray!)
```

Because Emacs starts counting cyclic diary occurrences from 0, I created it *before* my official start date on October 15, 2007. That way, October 19, 2007 is properly reported as my first paycheck:

```
Friday, October 19, 2007
=====
Payday (1st check, hooray!)
```

Diary can handle even more complex expressions. Let's say that there's a tea party on the last Sunday of every month, and barbecues during the first Saturdays in summer:

```
&%%(diary-float t 0 -1) Tea party
&%%(diary-float '(6 7 8 9) 6 1) Barbecue
```

The first argument specifies the month (an integer, list of integers, or `t` for every month). The second argument specifies the day of week. The third argument specifies which in that month (1: first, -1: last).

For more information about other recurring events, see the Emacs info manual (Emacs, Calendar/Diary, Advanced Calendar/Diary Usage).

Org

All you need to do is define your repeating appointments in Diary and include them in your organizer file. The following code includes diary entries in the agenda view:

```
(setq org-agenda-include-diary t)
```

Planner

This is where things get a little complicated. It isn't easy to have recurring appointments in Planner because the Planner way of doing things is to copy the data into multiple locations. The

best way is to use `planner-diary.el` to rewrite a Diary section in your Planner files. The following code sets this up:

```
(require 'planner-diary)
(add-hook 'diary-display-hook 'fancy-diary-display)
(setq planner-diary-use-diary t)
(planner-diary-insinuate)
```

You should also add a Diary section to your `planner-day-page` template:

```
(setq planner-day-page-template
  "* Tasks\n\n\n* Schedule\n\n\n* Diary\n\n\n\n* Notes")
```

If you have appointments defined in your Diary file, they will be included in the `* Diary` section when you go to the corresponding day page. NOTE: The Diary section will be completely erased and rewritten each time you open the file, so don't type anything else into this section or you will lose them when the diary entries are refreshed.

Working with Other Calendars

Import Appointments from iCalendar (Google Calendar, etc.)

iCalendar is a standard for exchanging calendar data. It is supported by Google Calendar, Facebook, Mozilla Calendar, Lotus Notes, and other desktop and web-based calendar programs. Import data from an iCalendar file (typically ending in `.ics`) by calling `M-x icalendar-import-file`. Emacs will prompt you for the name of the file, whether the event will be highlighted in the calendar displayed by `M-x calendar`, and the name of your diary file. Import a buffer containing iCalendar data by calling `M-x icalendar-import-buffer`. The entry will automatically be added to your diary file.

Export Appointments to iCalendar

You can also export appointments to iCalendar using `icalendar.el`. Call `M-x icalendar-export-file`, specify your diary file (`~/diary` by default) and an iCalendar file for output.

Import Appointments from Microsoft Outlook

Import an appointment from an e-mail sent by Microsoft Outlook by calling M-x diary-from-outlook. This works for the Gnus and Rmail mail readers inside Emacs. To import a collection of appointments from Microsoft Outlook, export the appointments from Microsoft Outlook in the comma-separated value format (CSV). Save these two libraries to somewhere in your load-path:

- <http://wickedcoolemacs.nsp.com/lisp/csv.el>
- <http://wickedcoolemacs.nsp.com/lisp/lookout.el>

and add the following to your ~/.emacs:

```
(require 'lookout)
```

Lookout overwrites the target diary file, so do not choose a ~/diary file with data. Instead, use M-x lookout-create-diary to create another diary file such as ~/diary.outlook. Add the following line to your ~/diary to include the imported data:

```
#include "diary.outlook"
```

and add the following lines to your ~/.emacs:

```
(add-hook 'list-diary-entries-hook 'include-other-diary-files)
(add-hook 'mark-diary-entries-hook 'mark-included-diary-files)
```

Summary

If you keep your schedule in Emacs, your schedule will always be at your fingertips, and you can configure the display and the reminders to fit your needs. To learn more about calendars and appointments in Emacs, read through the Info manuals for Emacs (Calendar/Diary), Org, and Planner, and browse through the CategoryCalendar pages on the EmacsWiki (<http://www.emacswiki.org/cgi-bin/wiki/CategoryCalendar>). Emacs scheduling is much more powerful than what I've described here. If you want to find out when to light Sabbath candles, observe Chinese birthdays, or convert Mayan dates, Emacs has just the right function for you. Let Emacs manage your day, and you'll be another step closer to the Emacs way of life.