

Managing Your Tasks

Most time management books will tell you to think of your three- to five-year goals, come up with projects, and then make tasks based on those projects.

You know that life doesn't happen as neatly as that. Instead, tasks come at you from all sides: the bugs you discover while you read code, the e-mails you get from coworkers, the milk that you need to pick up on the way home. The task list just keeps getting bigger.

I want to help you set up Emacs so that it will:

- capture all those tasks so that you don't worry about forgetting anything,
- show you what you need to do so that you can work effectively,
- help you organize your day, your week, your life,
- show you what you've accomplished so that you have a feeling of progress,
- help you plan the things that you want to do, and
- fit the way you want to work, whether that's Franklin-Covey, GTD, or a home-brew system for keeping sane.

If you've been keeping your task list in your head, you'll find it immensely helpful to get it out and into your computer. If you swear by your day planner or index cards, you might find that Emacs is not only a good backup, but it can also give you an overview of where you're going and where you've been. If you need to bill for your time, you'll love how time tracking is integrated into your task list. And even if you've never written a line of Emacs Lisp before, you might find that customizing Emacs to fit the way you think will transform the way you think.

Give it a try. Use Emacs to manage your tasks for a month. Write everything down in it. Use it to run your life. At the end of the month, if it doesn't feel natural to you, at least you can say that Emacs really isn't just an editor, it's a way of life. It might not be your way of life, but you'll have learned something from it, guaranteed.

There are a number of ways to manage your tasks in Emacs. You might start out by keeping everything in a single text file. If so, then all you probably need is something like this in your `~/.`emacs:

```
(global-set-key (kbd "C-c t")
  (lambda () (interactive) (find-file-other-window "~/todo.txt")))
```

This binds the keyboard shortcut C-c t to open a *todo.txt* file in your home directory. Done!

But you can do so much more than that. This chapter will cover how to use the Emacs Org module for GTD or day planning, two popular ways of managing tasks. It will also briefly cover Planner, another widely-used personal information manager for Emacs. Which one should you start with?

Choosing between Org and Planner

The decision between Org and Planner is really a personal one. On one hand, Org has powerful commands for working with outline structures. On the other hand, Planner supports more free-form planning, and you can publish your tasks and notes to a website. If you're the kind of person who likes planning with paper because you can scribble in notes outside the lines, try Planner first. If you like getting different views of your tasks, such as daily, weekly, and project views, try Org first. If you're new to planning and aren't sure yet which way works best for you, read the sections on Org and try them out. Spend a month getting the hang of Org, then try Planner to see if that's a better fit.

You can also jump ahead to “A Day in the Life”, where I describe how I've used Org and Planner to plan my day. If either of those descriptions resonate with you, try the corresponding system first. Have fun!

Integrating Emacs with the Rest of Your Life

No matter which system you choose, you'll probably need to integrate it with the rest of your life. You could use Emacs to manage the tasks for just your Emacs-related work, such as software projects or webpages. You could also use Emacs to manage the rest of your life. Task management systems work best when your tasks are gathered in one trusted place instead of scattered among scraps of paper, scratch pads, personal digital assistants (PDA)s, and computers. In that case, you'll need to find some way to refer to your task list while you're away from your computer, and to capture new tasks so that you can put them into your system when you get back.

I find that using Org or Planner's day views gives me a nice, short task list that I can copy onto an index card when I need to move. When a task comes up, I write it down in a pocket notebook and enter it into Emacs at the first opportunity I get. Other people have written scripts to export some of their tasks into printable forms. If you frequently need paper copies of your task list, check out Org's export functionality, or just print your Planner day page to a text file.

Org

You'd like to use the Org Mode for Emacs to manage your tasks. In this section, we'll cover the minimum you need to get started. There are a million ways to plan, but we're going to focus on two.

The first approach is Getting Things Done (GTD), described by David Allen in the book of the same title. GTD focuses on next actions (the very next thing you can do) and uses context lists to keep things manageable. Popular ways to do GTD are with index cards, recycled business cards, or software programs. If most of your tasks are in your head or scattered on scraps of paper, GTD will probably give you the most organizational bang for the least effort.

The second approach is day planning. You plan your week based on your projects and priorities, write your tasks onto the pages for each day, and copy unfinished tasks over to the next day. If you've used one of those Filofax, Franklin-Covey or Dayrunner personal organizers, you're probably used to this way of planning.

As you learn more about Emacs and task management, you'll probably develop your own way of doing things. These two are good starting points.

If you use GTD, read on. Otherwise, read “Setup” and then skip ahead to “Org as a Day Planner.”

Set up

Org is part of Emacs 22. To make it even easier to collect tasks and notes, install a separate package called Remember.

First, download and unpack Remember. As of this writing, Remember is at version 1.9. You can get the TAR.GZ from <http://download.gna.org/remember-el/remember-1.9.tar.gz> or the ZIP archive from <http://download.gna.org/remember-el/remember-1.9.zip>. If these instructions are out of date, check <http://www.emacswiki.org/cgi-bin/wiki/RememberMode> to find out where to get Remember.

Then add this basic configuration for Org and Remember to your `~/.emacs`:

```
(add-to-list 'load-path ❶ "~/elisp/remember-1.9")
(require 'remember-autoloads)
(setq org-remember-templates
      '((?t "* TODO %?\n %i\n %a" ❷ "~/organizer.org")
        (?a "* Appointment: %?\n%^T\n%i\n %a" "~/organizer.org")))
(setq remember-annotation-functions '(org-remember-annotation))
(setq remember-handler-functions '(org-remember-handler))
(eval-after-load 'remember
  '(add-hook 'remember-mode-hook 'org-remember-apply-template))
(global-set-key (kbd ❸ "C-c r") 'remember)

(require 'org)
(add-to-list 'auto-mode-alist '(❹ "\\\\.org$" . org-mode))
(global-set-key (kbd ❺ "C-c a") 'org-agenda)
```

```
(setq org-todo-keywords ⑥ ("TODO" "NEXT" "STARTED" "WAITING" "DONE"))
(setq ⑦org-agenda-include-diary t)
```

①: Change the directory as necessary.

②: Although this section refers to *~/organizer.org* throughout, you can also use a different filename.

③: This is a keyboard shortcut.

④: This tells Emacs to open all .org files in org-mode.

⑤: This keyboard shortcut is used widely in this section. If you change it, you'll need to remember your new shortcut every time you see a reference to `CTRL-C A` (`org-agenda`).

This convention for keyboard shortcuts confuses me, and it will probably confuse other Emacs geeks. We're used to seeing this as C-c a, which is the convention for all the other Emacs documentation. What would you suggest?

⑥: These TODO keywords are particularly useful for GTD, although they're also useful for day planning.

⑦: This makes it easy to pull in holidays and other events (see Chapter 9).

After you evaluate that code by calling `META-X eval-buffer` or restarting Emacs, you're ready to create an Org file.

1. Open *~/organizer.org*.
2. Save it. This is probably the only time you'll have an empty TODO list.
3. Use `C-c |` (`org-agenda-file-to-front`) to add it to your org-agenda-files. You only need to do this once for this agenda file.

Read on to find out how to use your new Org file for GTD, or skip ahead to the section on Day Planning to find out how to plan by day.

Org and GTD

So you've read David Allen's book about Getting Things Done (or any of the countless summaries of it on the Net), and you'd like to get started with Emacs and Org mode. I'll show you the bare minimum you need to support the five phases in the GTD task workflow:

Phase	GTD	Org
Collect	Capture everything you need to	Put everything into your inbox.

	do.	
Process	Actionable? Yes: do, delegate, or defer; no: file, throw, or incubate	Put tasks on your list; track delegated tasks.
Organize	Next actions, projects, waiting for, someday/maybe	Set tasks as TODO or NEXT; Tag tasks as MAYBE; view tasks by tag.
Review	Daily, weekly, etc.	View your agenda
Do	Actually do the work!	No, Emacs won't do the work for you... (But it can brew coffee!)

Collect/Process

The first thing you need to do is get all the tasks out of your head, off scraps of paper, out of your e-mail, and so on. If this is the first time you're putting tasks into Org, you have a lot of tasks to collect. The best way to collect lots of tasks is to open your Org agenda file ([~/organizer.org](#)) and put this heading at the end of the file:

```
* Inbox
```

Now go to the end of the file, and type in **** TODO** and the first task you can think of, like this:

```
** TODO Buy milk
```

Press C-M-RET and keep typing other tasks. Keep going until you've gone through all the things in your head and all the scraps of paper lying around. Do not get distracted. Your goal is to write all the tasks down. If you are as easily distracted as I am, do not even open up a browser window or look at your e-mail. It can be a real struggle sometimes to focus long enough to get everything down, especially when you're writing down all these tasks that you can work on. Do *not* do a task unless it takes less than two minutes to do. In fact, if you are just starting out with GTD, you might find it better to resist all temptations to do tasks during this step. Get it all out.

Now that you've gotten your tasks out of your head and into your organizer.org file, breathe. There's less stress in your brain now, because you don't have to worry about forgetting things (as long as you remember to check your Org file, that is!).

Do not fill your brain back up with other things you need to do. The brain is a wonderful thing, but it's not good at remembering what you need to do. Whenever a task comes your way through e-mail, in conversation, in the shower, put it in your [~/organizer.org](#). Well, you probably don't want to drip all

over the computer, so sometimes you'll need to hang on to an idea, but get it out of your head and into your organizer as quickly as possible.

To collect tasks within Emacs as they come up, use Remember. With the basic configuration you set up in the previous section, you can use `C-c r t` (or `M-x remember` and `"t"` for the Tasks template) to pop up a buffer where you can type in the task description and some notes. It will look like this:

```
## Filing location: Select interactively, default, or last used:
##      C-u C-c C-c  to select file and header location interactively.
##      C-c C-c  "~/notebook/personal/organizer.org" -> "* Tasks"
## C-u C-u C-c C-c  "???" -> "* ???"
## To switch templates, use `M-x org-remember'.
```

* TODO

And if you're lucky, there will even be a hyperlink to the file or e-mail you were looking at when you called `C-c r t` (remember, tasks).

If you brain-dump your tasks and use `C-c r t` to collect tasks as they come up, you can free up your brain for other things, such as contemplating the meaning of life.

Process

Now that you've collected all those tasks into your inbox, you can process them. Open your Org agenda file and go to your inbox.

For every item there, decide if it's something that you need to act on. Is it really just a note? If so, take out the TODO keyword and organize it like you would store other notes. If it's a true-blue task, decide if it's something you can do within the next two minutes, delegate to someone else, or leave on your task list. Go through your list systematically, delegating and eliminating whenever possible.

If you delegate the task, change it to WAITING by moving your cursor to the headline with the TODO keyword and typing `S-r` (`org-shiftright`) until it changes to WAITING. To keep track of who you delegated it to, just edit the task description to reflect it. Your organizer file will look like this:

```
** WAITING Buy milk - WJY
```

Organize

After you've processed all of these, you should now have a list of tasks that you need to act on. If you've braindumped everything that people have asked you to do and that you've thought of doing, this

is probably a very long list. Intimidatingly long. The next step in restoring sanity to your life is to organize your list into next actions, projects, things you're waiting for, and someday/maybe tasks.

Review that task list. For each task, decide if it's something you can do immediately. Is it something you can do in one sitting, and do you have everything you need in order to do it? If so, great! It's a next action. Leave it on your task list.

If you can't immediately work on a task, it may be a project in disguise, and it needs to be broken down into smaller, concrete next actions. For example, the task:

```
** TODO Write a book about Emacs
```

would probably result in me getting complete writer's block. If you're faced with a big task like this, move it out of your inbox and make it a project. Then you can think of the very next action you need to do. My ~/organizer.org looks something like this:

```
* Projects
** Write a book
*** TODO Write about basic Org and GTD
** Other projects...
* Inbox
** TODO ... lots of other things go here ...
** TODO ... lots of other things go here ...
** TODO ... lots of other things go here ...
```

A task might also be stuck because you need to wait for someone else. For example, I'm currently working on renewing my visa, but I need to wait for the embassy. Mark those stuck tasks as WAITING with S-right (org-shiftright).

Someday/maybe tasks are nice to think about once in a while, but you don't want to clutter your day-to-day tasks with them. A basic way to deal with this is to move those tasks into a separate Organizer file such as `~/someday.org`. Another is to use tags, which we'll cover in the section on intermediate Org. For now, just move them to another file.

Review

You've gone from a whole bunch of tasks in your brain and on pieces of paper to one text file containing everything you need to do, with an easy way to get to just the things you can do right now. To view all your tasks, type C-c a t (org-agenda, tasks). You'll get something that looks like this:

```
Global list of TODO items of type: ALL
Available with `N r': (0)ALL (1)TODO (2)STARTED (3)WAITING (4)DONE
```

```
TODO Write about basic Org and GTD
TODO Blog
TODO Answer my mail
TODO Alter slacks
...
```

Type "l r" to show only the active tasks, and review what you're waiting for with "3 r". Review this WAITING list every so often to make sure that nothing falls through the cracks.

Type "f" to start follow mode, which displays the relevant lines from your Org agenda file as you move around. This is helpful for quickly reviewing your task list.

Do

All of the above should take you less than fifteen minutes of planning each day. The rest of the time, you can focus on doing the work, undistracted by shiny new tasks that pop up because you can get them out of your way with C-c r t.

To work, review your task list with either C-c a t (org-agenda, tasks) or C-a a (org-agenda, agenda). From the agenda view, type "t" (org-agenda-todo) to change the task status. I find it helpful to mark a task as STARTED because it helps me remember what I was working on in case I get distracted by something urgent, but you can also use C-u t to jump to a status without cycling through the ones in between (say, marking a task as DONE). You can also press ENTER to jump to the task headline and edit it directly.

Going back to reviews: As you mark tasks done, you'll also want to do daily and weekly reviews. You can see those with C-c a a (org-agenda, org-agenda-list), which opens an Org agenda view. To see completed tasks in the Org agenda view, type l (org-agenda-log-mode). To switch to the day view, type d (org-agenda-day-view). To switch to the week view, type w (org-agenda-week-view). The basic configuration I've suggested here will automatically include unfinished tasks at the beginning of the agenda. Scroll up to review your tasks, and press ENTER on a line to jump to it.

Org as a Day Planner

If you want to use Org as day planner, read this section to learn how to use Org to manage your tasks day by day. I assume that you've set up Org and Remember according to the basic configuration suggested in "Setup". If you haven't done that yet, please review "Setup", then return here.

Here's what you'll learn how to do:

1. Collect your tasks
2. Schedule the tasks for specific days

3. View your daily or weekly agenda
4. Mark tasks as done
5. Reschedule a task
6. Review your accomplishments

Collecting Tasks

If you're adding many tasks, you may find it easier to edit your Organizer file directly. Open [~/organizer.org](#) in Emacs and go to the end of the file. Add headlines like this:

```
* Inbox
** TODO your task description here
** TODO another task...
```

Instead of typing `** TODO` again and again, you can use `C-M-RET` to create another `TODO` heading at the same level as the previous one. Think of all the things you need to do over the next few days and add them to your Org agenda file.

More tasks will come up as you work on things. Instead of switching to your Org agenda file each time you need to add a task, you can use `C-c r t` (remember, Tasks template) to remember the task quickly. Try it now by typing `C-c r t`. Type in the task description and press `C-c C-c` (`org-ctrl-c-ctrl-c`) to add the task to the end of the [~/organizer.org](#) file.

Now you have plenty of tasks on your list, but no idea when you need to do them. Here's where scheduling and deadlines come in.

Scheduling Tasks

To schedule a task, move your cursor to the `TODO` headline and press `C-c C-s` (`org-schedule`). Org will prompt you for a date. It understands full, partial, and relative dates. For example, if today is December 29, 2007, then it understands any of the following:

Input	Result	Explanation
(blank)	2007-12-30	Today
10:30	2007-12-30 10:30	Time today
3:30pm	2007-12-30 15:30	Time today
31	2007-12-31	Day in the current month
12-31	2007-12-31	Month and day in the current year
2008-01-01	2008-01-01	Date
2008-01-01 12:30am	2008-01-01 00:30	Date and time (also works with partial dates)
+2	2008-12-31	Two days from now

-3	2007-12-26	Three days ago
Fri	2008-01-04	The nearestFriday (on or after today)
+2w	2008-01-12	Two weeks from today

To set a deadline for a task, type C-c C-d (org-deadline). It accepts the same kinds of date that org-schedule does.

Try this out by scheduling all of your tasks over the next few days, adding deadlines where necessary.

Now that you've added date information to your tasks, you probably want to see those tasks organized by date instead of in the random way you entered them. Agenda views are going to become your new best friend.

Viewing Your Agenda

Type C-c a a (org-agenda, org-agenda-list) to view your agenda. By default, Org shows a weekly view of your scheduled tasks and appointments. This is your Org agenda view.

Here are some useful navigational keys:

- Switch to a daily view with d (org-agenda-day-view)
- Switch to a weekly view with w (org-agenda-week-view)
- View earlier or later days/weeks with your left and right arrow keys (org-agenda-earlier, org-agenda-later)
- Jump to a specific day with j (org-agenda-goto-date)

Get into the habit of typing C-c a a (org-agenda, org-agenda-list) to check your task list. It may also help to add this to the bottom of your *~/.emacs*:

```
(org-agenda-list)
```

This opens your Org agenda view when you start up Emacs. Start your Emacs day with your Org agenda, check it every time you finish a task, and review it before you end the day. This will help you make sure that nothing falls through the cracks.

Marking Tasks

The easiest way to mark a task as done is to go to its line in your Org agenda view. Type C-c a a (org-agenda, org-agenda-list) to view your tasks for today, move your cursor to the task, and type t (org-agenda-todo) to cycle the task status until it's marked DONE. You can also type C-u t (org-agenda-todo

with a prefix argument) to jump to a specific task status. For example, you could type C-u t DONE to mark a task as done.

You can also mark tasks done from your `~/organizer.org` file. Open the file and move your cursor to the item. Type C-c C-t (org-todo) to change the task status. Again, you can type C-u C-c C-t (org-todo) to jump to a specific task status.

I find it helpful to mark tasks as STARTED when I start working on them, WAITING if I need something else in order to continue working on the task, and DONE when I'm finished with it. That way, I can quickly see which task I was supposed to be working on before I got distracted by something bright and shiny, and I can also see what I'm waiting for. Get into the habit of doing that, and you'll find it easier to get back on track after distractions.

Unfortunately, Org does not come with a M-x org-zap-distractions command. There will be days when you can't do everything on your task list, and that's when you may need to reschedule tasks.

Rescheduling Tasks

You don't have to reschedule your tasks. Org will remind you of unfinished, scheduled tasks every single day. It will even helpfully tell you how many days you've procrastinated on that task. If you use C-c a a (org-agenda, org-agenda-list) when you have unfinished tasks on previous days, you'll see task reminders like this:

```
Saturday 29 December 2007
  organizer: Scheduled:  TODO Respond to mail
  organizer: Sched. 6x:  TODO Write notes from mentoring conversation
  organizer: Sched. 2x:  WAITING Report time
```

You could let your unfinished tasks snowball on you in a big mass of procrastination. If you let your task list grow to an intimidating size, though, you may start stressing out about the things you aren't doing. Let me show you how to procrastinate--I mean, *reschedule* your tasks effectively--so that you can work with a more manageable task list.

If tasks are starting to accumulate, it's a good sign that you need to review those tasks. Do you really need to do them? If not, delete them by moving to the line in your Org agenda view and pressing C-k (org-agenda-kill). You can also edit your `~/organizer.org` file and delete them, but org-agenda-kill is more convenient.

If you really need to do the tasks, but there's no point in seeing it in today's task list because you can't do it today anyway, use C-c C-s (org-agenda-schedule) to reschedule the task. If you're only

moving it a couple of days ahead, use S-right (org-agenda-later) to move it forward, and S-left (org-agenda-earlier) if you overshoot.

Some tasks show up again and again on your task list, and you *know* you need to do them, but you don't know where to getting started. "TODO Write a book" is not a good task, because it's just too big to do in one sitting and it doesn't tell you what to do right now. Big tasks are often projects in disguise. Break it down into smaller tasks, and schedule *those* instead. If you're in the Org agenda view, press RET (org-agenda-switch-to) to jump to the task in your `~/organizer.org` file. Break it down into smaller tasks by adding sub-headings and more TODOs, like this:

```
** Write a book
*** TODO Make an outline of what to write
*** TODO Read sample query letters
*** TODO Write a query letter
```

... and so on.

Then you can use C-c C-s (org-schedule) to schedule those tasks.

Use these commands to keep your task list manageable. That way, you get the warm and fuzzy feeling of accomplishment when you finish what's on your list and you look at everything you've done today.

Reviewing Accomplishments

If you've kept your tasks in your `~/organizer.org` file, working with your Org agenda view, and marking tasks as DONE when you finish them, you'll find it easy (and satisfying!) to review your accomplishments. Just open your daily or weekly Org agenda view with C-c a a (org-agenda, org-agenda-list). Type l (org-agenda-log-mode) to show completed tasks. Pat yourself on the back, then plan yourself another wonderful day tomorrow!

Org and Timeclocking

Many professionals bill clients for their time. Even if you don't, keeping track of the time you actually spend on tasks can help you improve your time estimates and check if you're spending enough time on the things that are important to you. For example, keeping track of the time you spend on tasks might show you that you spend two and a half hours each day just responding to e-mail. If you can identify problem areas like that, then you can look for more effective ways to perform the tasks that take up a lot of your time.

I love Org's timeclocking support, and I think you will too. Because it's integrated with your task list, you don't have to switch to separate application or reenter data. You can get more detailed time reports, too. All you have to do is remember to clock in before you start a task and clock out when you finish it.

Starting and Stopping the Clock

You can clock in by moving your cursor to the task headline in either your organizer.org file or the org agenda view, and then pressing C-c C-x C-i (org-agenda-clock-in or org-clock-in, depending on context). This adds the time stamp to the task. If you are already clocked into another task in that organizer file, you'll be clocked out of it to prevent you from accidentally double-billing.

To clock out of a task, type C-c C-x C-o (org-agenda-clock-out) from the task headline. Marking a task as done will also automatically stop the clock, if that was the task with the active clock.

Here's some code to make this even easier. The following code clocks in whenever you market task is started, and clocks out when you market a task as WAITING. It also automatically market task is started if you clock in. This takes advantage of the Org configuration suggested in “Setup”. Add this to your *~/.emacs* and evaluate it:

```
(defun wicked/org-clock-in-if-starting ()
  "Clock in when the task is marked STARTED."
  (when (and (string= state "STARTED")
             (not (string= last-state state))))
  (org-clock-in))
(add-hook 'org-after-todo-state-change-hook
  'wicked/org-clock-in-if-starting)
(defun wicked/org-clock-out-if-waiting ()
  "Clock out when the task is marked WAITING."
  (when (and (string= state "WAITING")
             (equal (marker-buffer org-clock-marker)
                    (current-buffer))
             (< (point) org-clock-marker)
             (> (save-excursion (outline-next-heading) (point))
                 org-clock-marker)
             (not (string= last-state state))))
  (org-clock-out))
(add-hook 'org-after-todo-state-change-hook
  'wicked/org-clock-out-if-waiting)
```

What if you forgot to clock into a task when you started? No problem. Simply clock in and out of it, then edit the starting timestamp for the task in your `~/organizer.org` file. To find a starting timestamp, move your cursor to the task headline. If the task has been collapsed to a single line, press TAB to expand it. Look for a line that starts with CLOCK:, or a collapsed segment that starts with :CLOCK:. If you see a collapsed segment, he expanded by moving a cursor to it and pressing tab. Find the clock entry you want to change, and if the timestamp, and press C-c C-y (org-evaluate-time-range) to update the time total.

Reporting Time By Project

To see how much time you've spent on a project or task, open your `~/organizer.org` file and press C-c C-x C-d (org-clock-display). Total times will be added to each headline, summarizing the times for each subtree.

You can also use one of Org's dynamic blocks. Open your `~/organizer.org` file, move your cursor to where you want the report inserted, and type C-c C-x C-r (org-clock-report). This inserts a dynamic block that starts out like this:

```
#+BEGIN: clocktable :maxlevel 2 :emphasize nil
#+END
```

If you change a clock entry and would like to update the clocktable, move your point inside the clocktable and type C-c C-x C-u (org-dblock-update).

By default, the reports will include all the second-level headings for all the days. What if you want to limit the report to just the time you clocked last week?

By Time Period

To summarize it for a span of days, change the starting line from:

```
#+BEGIN: clocktable :maxlevel 2 :emphasize nil
```

to something like:

```
#+BEGIN: clocktable :maxlevel 2 :emphasize nil
:tstart ❶ "<2007-12-25 Sun>" :tend ❷ "<2007-12-31 Mon>"
```

This is broken into two lines to fit this page, but should be on one line in your `~/organizer.org` file. where ❶ is the starting time/date and ❷ is the ending time/date. You can add the timestamps either manually or with C-c C-. (org-time-stamp). After you change the block definition, update the clock table by typing C-c C-x C-u (org-dblock-update).

You can also use a definition like:

```
#+BEGIN: clocktable :maxlevel 2 :emphasize nil :block today
```

to see today's entries. Other block keywords are [yesterday](#), [thisweek](#), [lastweek](#), [thismonth](#), [lastmonth](#), [thisyear](#), or [lastyear](#).

If you need more levels of headings, change the value of maxlevel. For example, to see a detailed clock table with up to 10 levels of headings, use

```
#+BEGIN: clocktable :maxlevel 10 :emphasize nil :block today
```

The clocktable report summarizes the reported time for the period. What if you want the time broken down by day?

Reporting Time by Day

The following code creates a custom dynamic block that breaks the reported time by date. Add the following code to your [~/emacs](#):

```
(defun org-dblock-write:rangereport (params)
  "Display day-by-day time reports."
  (let* ((ts (plist-get params :tstart))
         (te (plist-get params :tend))
         (start (time-to-seconds
                 (apply 'encode-time (org-parse-time-string ts))))
         (end (time-to-seconds
              (apply 'encode-time (org-parse-time-string te))))
         day-numbers)
    (setq params (plist-put params :tstart nil))
    (setq params (plist-put params :end nil))
    (while (<= start end)
      (save-excursion
        (insert "\n\n"
                (format-time-string (car org-time-stamp-formats)
                                     (seconds-to-time start))
                "-----\n")
        (org-dblock-write:clocktable
         (plist-put
          (plist-put
           params
            :tstart
```

```
(format-time-string (car org-time-stamp-formats)
                   (seconds-to-time start))

:tend
(format-time-string (car org-time-stamp-formats)
                   (seconds-to-time end)))

(setq start (+ 86400 start))))))
```

After you load that code, you'll be able to use a dynamic block of the form

```
#+BEGIN: rangereport :maxlevel 2 :tstart "<2007-12-25 Tue>"
❶:tend "<2007-12-30 Sun>"
...
#+END:
```

to see your time reported by date. Again, this line has been broken to fit, and ❶ should be on the previous line. Add this to your [~/organizer.org](https://www.gnu.org/software/organizer/) and update it by moving your cursor within the block and typing C-c C-x C-u (org-dblock-update).

Org makes it easy to capture timeclock information by integrating the timeclock into your task list so that you don't even have to think about it, and it can report this time by project or by date. You can use this information to bill clients, improve your time estimates, or reflect on the way you do things. All you have to do is clock in by marking a task as STARTED, and clock out by marking a task as WAITING or DONE. Don't get discouraged if the time clock shows you do only a few hours of productive work each day. Use that to help you figure out how to do things better!

Org and Projects

Organizing your tasks into projects can help you plan ahead, track your progress, and stay motivated. Working from a project list allows you to plan your day or week instead of just reacting to other people's requests. Keeping your projects and tasks in Org makes it easier for you to review your completed tasks and plan the next step. If you include some text describing why you want to do the project and what your intended outcome is, this can help you stay motivated throughout a long project.

Projects can take a single day or several years. They can be large projects involving lots of other people and resources, or small projects that you do on your own. Projects may involve a handful of separate steps or a hundred things you need to do in order to achieve your goal. The important thing is that there is more than one step. If you organize your task list so that related tasks are together, then you'll find it easier to get a sense of where you are, where you're going, and what you need to do next.

In this section, you will learn how to:

- Create projects,
- Organize your tasks into projects,
- Review your ongoing projects, and
- Mark projects as finished.

I'll assume that you're using Emacs 22, and that you've set up Org using an [~/organizer.org](http://www.orgmode.org/~organizer.org) agenda file and the basic configuration suggested in either "Org and GTD" or "Org as a Day Planner." I'll also assume that you're familiar with switching between the Org agenda view and the Org organizer file, and that you're comfortable navigating around Emacs.

The examples I'll use focus on yearly goals. You might also have short-term projects or long-term plans. Feel free to adapt the examples as needed.

Open your [~/organizer.org](http://www.orgmode.org/~organizer.org) file. If you've collected your tasks as suggested in "Org as a Day Planner" or "Org and GTD", your [~/organizer.org](http://www.orgmode.org/~organizer.org) file might look something like this:

```
* Inbox
** TODO Read Emacs Lisp intro
** TODO Write yearly review
** TODO Exercise
** TODO Browse the Emacs Wiki
```

Create new top-level headings for this year's goals or the projects that you're working on. You can create a top-level heading by typing `*` and the heading, like this:

```
* Learn more about Emacs
* Go on vacation
...
* Inbox
** TODO Read Emacs manual
...
```

It's a good idea to add the projects to the beginning of the file (before your Inbox) because `M-x remember` adds new tasks or notes to the end of the file. If the last major heading is `* Inbox`, then the tasks and notes are automatically added to it. If the last major heading is a project, the tasks and notes may get misfiled.

If you're a top-down planner, you'll find it easy to list your projects. In fact, you might have a ten- or even twenty-year plan already written down. You'll find this section straightforward, because you're already used to planning in terms of projects. Go ahead and adapt the examples to your long-term plans.

If you're a bottom-up planner, you might be giving me a weird look right now. "Yearly goals? I'm lucky if I can figure out how to get through the next day!" This section will also show you how to find the recurring themes in your task list and organize them into projects. Give project-based planning a try for a month. If this way of thinking doesn't work for you, Org will work just fine without projects.

You probably have projects, even if you can't think of any right now. Review your <~/organizer.org> file. If you haven't written down everything you needed to do yet, go through the section on basic configuration for your planning style (GTD or day planning). Once you have a list of things to do, you can then review it for big tasks, related tasks, and other project clues.

Read your tasks and ask yourself the following questions:

- Can I do this in one sitting? Big tasks such as "Write a book" are often projects in disguise. Use projects so that you can break them down into smaller, doable tasks.
- Is this related to other tasks? Related tasks such as "Book a flight" and "Plan my itinerary" are often clues to a project like "Go on vacation". Use projects so that you can review related tasks together.
- Why am I doing this? When you think about the reason why you're doing something, you'll often find a bigger project. For example, if one of your tasks is "Set up an automatic retirement savings plan", then the question "Why am I doing this?" may lead you to the project "Plan for retirement". Use projects to help you think of other ways to move towards that goal.

Big tasks need to be broken down into smaller tasks anyway, and organizing them into projects will help you make them more manageable. You may not want to organize all of your other tasks into projects. If you can pick some major themes to focus on, though, then you'll be able to see how the different things you do are related to each other, and you'll be able to think of other ways to work on those projects. If you're starting out with project-based thinking, maybe you can pick three to five projects and try to do a little work on each of them every day.

If you still don't identify any projects, that's okay. You can use Org as a straightforward task list. Jump ahead to the section on "Tags", as you'll probably find that useful.

On the other hand, if this step turns up plenty of projects, resist the temptation to over-correct and end up with hundreds of projects. I find that more than 7 active projects gets hard to manage. Pick a few main themes that you'd like to work on, and make everything else something you plan to do someday.

Creating Tasks

Now that you have project headings, think of the next thing you need to do in order to move those projects forward. If you've already written down those tasks, move them under the appropriate project heading. If not, type them in.

In order for a task to belong to a project, it needs to be under the heading and at a lower level. For example, if your project heading has one star, like this:

```
* Learn Emacs
```

then your TODO headings should have two stars, like this:

```
* Learn Emacs
** TODO Read the Emacs manual
** TODO Read the Emacs Lisp Intro manual (eintr)
** TODO Install the Emacs source code
```

If your tasks are not at the right level, you can add the star manually by editing the heading. You can also use M-right and M-left (org-metaright and org-metaleft) while on a heading in order to promote or demote it, and you can use M-S-right and M-S-left (org-shift-metaright and org-shift-metaleft) to promote or demote entire subtrees.

To move tasks up and down within the project, you can copy and paste the text. You can use M-Up and M-Down (org-metaup and org-metadown) to move subtrees.

Think of tasks you can do within the next week in order to move each of your projects forward. Add next actions to all of your active projects. Creating next actions for each of your projects makes it easier to remember to keep moving forward.

Organizing Tasks

If you have many tasks in a project, you may want to organize them into sub-projects. For example, you might divide a software project into components. If you're starting from scratch, you can create the project structure by typing in more stars for sub-project headings. For example:

```
* Learn Emacs
** Read mail
*** TODO Choose a mail client
*** TODO Install and configure the mail client
*** TODO Send a message
...
** Browse the Web
*** Read through the w3m documentation
```

...

You can also demote an existing project into a subproject. Use M-S-right (org-shift-metaright) on the current project headline in order to demote it to a sub-project. This will also demote the tasks within the project. For example, demoting this:

```
* Learn Emacs
** TODO Choose a mail client
** TODO Install and configure the mail client
** TODO Send a message
```

will result in this:

```
** Learn Emacs
*** TODO Choose a mail client
*** TODO Install and configure the mail client
*** TODO Send a message
```

Then you can change the heading and add another heading above it, like this:

```
* Learn Emacs
** Read mail
*** TODO Choose a mail client
*** TODO Install and configure the mail client
*** TODO Send a message
```

This kind of organization is optional, but it can help you get an idea of the overall structure of your project. Using different levels allows you to hide and show groups of headings by pressing TAB on the heading.

Now that you've created your project tasks and organized them the way you want, it's time to actually do the work.

Working on Tasks

If you use Org as a day planner, you may also want to schedule the tasks onto specific days with C-c C-s (org-schedule). You can review your daily or weekly agenda with C-c a a (org-agenda, org-agenda-list), switching between daily and weekly views with d and w (org-agenda-day-view and org-agenda-week-view).

You can work with the next actions in the same way you work with other tasks, rescheduling them or marking them as `STARTED`, `WAITING` or `DONE` with the keyboard shortcuts introduced in the previous section on Org and GTD or Org as a Day Planner.

When you finish a project task, think of the next action you can do in order to move that project forward. If you use Org as a day planner, schedule the next action onto your calendar as well.

Reviewing Projects

You can review your projects by opening your `~/organizer.org` and browsing through the headings. `S-tab` (`org-shifttab`) changes the visibility of headings, so you can see just the top-level headings or all the details. You can use `TAB` (`org-cycle`) on a headline to show or hide subtrees.

If you have many projects, you'll want a shorter view of just your active projects. To make it easier to review projects, add a `PROJECT` tag to all your active project headlines. You can add a tag by editing your `~/organizer.org` and moving your cursor to the headline and typing `C-c C-c` (`org-ctrl-c-ctrl-c`), followed by the name of the tag (`PROJECT`). You can also manually type `:TAGNAME:` at the end of the headings, like this:

```
* Learn more about Emacs           :PROJECT:
** TODO Read the Emacs manual
** TODO Read the Emacs Lisp Intro manual (eintr)
...
* Go on vacation                   :PROJECT:
...
* Inbox
...
```

You might classify some of your projects as `someday/maybe` - things that are nice to think about, but which you aren't acting on right now. Tag your inactive or `someday/maybe` projects with `PROJECT` and `MAYBE`. If you're editing the `~/organizer.org` file, just add `:PROJECT:MAYBE:` to the heading. If you're tagging it with `C-c C-c` (`org-ctrl-c-ctrl-c`), specify `PROJECT:MAYBE` as the tag.

```
* Learn more about Emacs           :PROJECT:
* Go on vacation                   :PROJECT:MAYBE:
...
* Inbox
** TODO Read Emacs manual
...
```

Now that you've tagged your projects, you can view just your project headlines with a custom agenda command. Custom agenda views are a terrific feature in Org, and you can do a lot with them if you know a little Emacs Lisp. Here's what you need to add to your `~/.emacs` in order to get a list of your active projects and your someday/maybe projects:

```
(setq org-agenda-custom-commands
  '(❶("p" tags "PROJECT-MAYBE-DONE" nil)
    ❷("m" tags "PROJECT&MAYBE" nil)
    ;; ... put your other custom commands here
  ))
```

- **❶** makes C-c a p (org-agenda, p) show your active projects.
- **❷** makes C-c a m (org-agenda, m) show your "maybe" projects.

With these two commands, you can quickly review your active and inactive projects. To jump to a project from the agenda view, move your cursor to the heading and press RET (org-agenda-switch-to). If you want to scan through the projects quickly, use f (org-agenda-follow-mode) in the agenda view to turn on follow mode, then move to different headlines. Another window will show the headline at point.

If you review your projects at least once a week, you'll find it easier to make regular progress. If you want to combine your weekly/daily review with your project list, you can do that with org-agenda-custom-commands as well. Here's what you'd put in your `~/.emacs`:

```
(setq org-agenda-custom-commands
  (('("p" tags "PROJECT-MAYBE-DONE" nil)
    ("m" tags "PROJECT&MAYBE" nil)
    (❶"a" "My agenda"
      (❷(org-agenda-list)
        (❸(tags "PROJECT-MAYBE-DONE"))))
    ;; ... put your other custom commands here
  ))
```

This configures C-c a a (org-agenda, "My agenda") to display your agenda and a list of your project headings. The first item in **❶** is the shortcut key, the second is a name for the agenda view. This is composed of two parts. **❷** displays your daily or weekly agenda. The d and w (org-agenda-day-view and org-agenda-week-view) shortcuts work if the point is within this section. **❸** displays a list of your active projects. After you evaluate the code, try C-c a a (org-agenda, "My agenda") to see your custom view. You can press RET (org-agenda-switch-to) to jump to a project from its agenda line.

Reviewing Your Stuck Projects

You might have forgotten to create next actions for some of your active projects. Org can help you find projects which don't have next actions. You can then decide if the project is complete or if it needs further action.

To list stuck projects, you first need to tell Org what a stuck project is. Add this to your `~/.emacs` and evaluate it:

```
(setq org-stuck-projects
  '( ❶ "PROJECT-MAYBE-DONE"
      ❷ ("TODO" "STARTED")
      ❸ nil
      ❹ "\\<IGNORE\\>"))
```

That defines a stuck project as one that meets the following conditions:

- ❶ - It is an active project (tagged “PROJECT” but not “MAYBE” or “DONE”)
- ❷ - It doesn't have a TODO or STARTED action,
- ❹ - The body of the project doesn't contain "<IGNORE>".

If you have projects that match these three criteria but are not supposed to be considered stuck, set ❸ to a tag query that matches those projects.

Once you've configured the criteria Org uses to find stuck projects, you can use M-x `org-agenda-list-stuck-projects` or C-a a # (`org-agenda`, `org-agenda-list-stuck-projects`) to show only the stuck projects. Review this list and jump to the appropriate headlines.

Want to add that to your custom agenda view? Modify the `org-agenda-custom-commands` value in your `~/.emacs` to be like this:

```
(setq org-agenda-custom-commands
  (('("p" tags "PROJECT-MAYBE-DONE" nil)
   ("m" tags "PROJECT&MAYBE" nil)
   ("a" "My agenda"
    ((org-agenda-list)
     ❶(org-agenda-list-stuck-projects)
     (tags "PROJECT-MAYBE-DONE"))
    ;; ... put your other custom commands here
   )))
```

- ❶ lists your stuck projects before your regular projects so you can see what needs your attention.

What about finished projects? You might want to keep them in your Org file, but they shouldn't show up in your active and inactive project lists. Org can keep track of those projects too.

Marking Projects as Done

If you look at the custom commands above, you'll notice the "-DONE" specifier. "DONE" is the tag we'll use to indicate done projects. To tag a project as done, move the point to the project heading and type C-c C-c (org-ctrl-c-ctrl-c). The tag prompt will default to the current tags. Just add "DONE" and press Enter. With the custom commands we've set up, projects tagged DONE will not show in your active, inactive, or stuck project lists.

You can also add the tag manually. For example, if the project heading is

```
* Learn Emacs      :PROJECT:
```

and you're happy with your level of Emacs proficiency, then you can mark it done by changing it to

```
* Learn Emacs      :PROJECT:DONE:
```

If you have plenty of completed projects, your Org file might be quite large. You can mark a subtree for archiving by typing C-c C-x C-a (org-toggle-archive-tag). This hides it from most Org commands. You can also archive a tree into a different file with C-c C-x C-s (org-advertized-archive-subtree).

Wrapping Up

Now you can create projects, manage your project tasks, and review your active, inactive, and stuck projects in Org. You know how to mark projects as completed and how to archive them. You've also started using tags to dynamically generate reports from your Org file.

Tags can do a lot more. To find out what else you can do with tags, read on.

Org and Tags

The section on projects introduced tags as a way to differentiate active and inactive projects. In this section, you'll learn more about tags and how you can use them to filter your task list.

What's a tag, anyway? In Org, tags are keywords at the end of headlines. Each tag can contain letters, numbers, and the symbols '_' and '@'. Tags begin and end with colons, and a single colon separates multiple tags. For example, you could have headlines like this:

```
* Personal          :PERSONAL:
** TODO Buy milk    :@ERRANDS:
** TODO Call Mom    :@PHONE:
```

```
** TODO Send letters           :@ERRANDS:
* Work                         :WORK:
** TODO Call John about report :@PHONE:JOHN:
** TODO Prepare for presentation on Monday
** TODO Call Mary about the presentation :@PHONE:URGENT:MARY:
```

One way to use tags is to filter your task list by priority. For example, you may want to focus on your urgent tasks first without getting distracted by other items on your task list. Another way to use tags is to keep track of the context of your tasks as suggested in GTD. By doing similar tasks together, you might be able to work more efficiently. For example, if you're on the phone at the office, it may be a good idea to do all of your work-related phone calls. If you're going to go to the post office, you might want to drop by the supermarket on your way back. You can use tags to categorize your headlines any way you want.

Tags can take advantage of the outline structure. For example, although the tasks "Buy milk", "Call Mom", and "Send letters" have one tag each, they also inherit the "PERSONAL" tag from the parent headline. A tag search for "PERSONAL" would display all three tasks. To customize this behavior, look at the documentation for the variables `org-use-tag-inheritance` and `org-tags-match-list-sublevels`.

Tags can help you organize and filter your task list. In this section, you'll learn how to add tags to your headlines, view tagged items in your Org file and in your agenda, and define custom agenda views. You'll also learn about custom tag searches and other interesting things you can do once you've tagged your headlines.

Adding Tags

You can edit your `~/organizer.org` file and add tags manually by typing in `:tagname:` at the end of the headline. You can also add tags by typing `C-c C-c` (`org-ctrl-c-ctrl-c`) when the point is on a headline. Use `M-TAB` (`complete-symbol`) to complete a tag based on all the tags used in the current file. If `Alt-TAB` is not processed by Emacs, you can use `ESC-TAB` instead.

Separate multiple tags with a single colon, like this: `(:@PHONE:URGENT:)`. The beginning and ending colons are optional when using `C-c C-c` (`org-ctrl-c-ctrl-c`), because the function will automatically add them.

If you add certain tags frequently, you can set up single-key shortcuts. For example, if you frequently tag tasks as "URGENT", you may want to define a shortcut (at least until your life gets under control). You can assign shortcuts globally by adding this code to your `~/emacs`:

```
(setq org-tag-alist '(("URGENT" . ?u)
```

```
( "@PHONE" . ?p)
( "@ERRANDS" . ?e) )
```

You can also set this on a per-file basis by adding the following line to the beginning of your file:

```
#+TAGS: URGENT(u) @PHONE(p) @ERRANDS(e)
```

You can then use these C-c C-c (org-ctrl-c-ctrl-c) to enter these single-key shortcuts, ending it with RET. If you are assigning a single tag, type C-c C-c C-c (org-ctrl-c-ctrl-c, next change exits) to make it even faster by skipping the RET.

If you use single-key shortcuts, you'll need another way to enter tags that start with the shortcut key. You can type them in manually, or you can use C-c C-c (org-ctrl-c-ctrl-c) and type TAB to enter in any tag.

To remove a tag, you could use C-c C-c (org-ctrl-c-ctrl-c) again, or delete it manually. To remove all tags, use C-c C-c (org-ctrl-c-ctrl-c) and press SPC.

Tags in Your Agenda

To view tagged headlines by themselves, use C-c a m (org-agenda, org-tags-view) and specify the search tag. For example, you can view your urgent tasks by specifying "URGENT". Note that this command displays the top headlines matching that tag, whether they're tasks or not. For example, if you searched for "WORK", you would just get the "* Work" headline. To view tagged tasks, use C-c a M (org-agenda, org-tags-view with a prefix argument). This shows only the tasks that have that tag.

To search for a combination of tags, you can combine tags like this:

- WORK&@PHONE - only your work phone calls
- PERSONAL-@ERRANDS - personal tasks, but without errands
- JOHN|MARY - anything tagged with "JOHN" or "MARY"

If you check certain lists often, you might want to create a custom agenda command for them. In the section on Projects, you configured custom agenda commands for active and inactive projects by adding the following code in your *~/.emacs*:

```
(setq org-agenda-custom-commands
  '(("p" tags "PROJECT-MAYBE-DONE" nil)
    ("m" tags "PROJECT&MAYBE" nil)
    ("a" "My agenda"
      ((org-agenda-list)
       (tags "PROJECT-MAYBE-DONE"))))
```

```
;; ... put your other custom commands here
))
```

You can use the same idea to create quick custom views for your other tagged tasks. For example, to create custom views for your urgent work tasks and your phone calls, modify the `org-agenda-custom-commands` setting in your `~/.emacs` to be like this:

```
(setq org-agenda-custom-commands
 '(
  ❶("u" todo "WORK&URGENT" nil)           ;; (1)
  ❷("c" todo "WORK&@PHONE" nil)         ;; (2)
  ❸("h" todo "PERSONAL-@ERRANDS" nil)   ;; (3)
  ❹("p" tags "PROJECT-MAYBE-DONE" nil)  ;; (4)
  ("m" tags "PROJECT&MAYBE" nil)
  ("a" "My agenda"
   ((org-agenda-list)
    ❺(tags-todo "URGENT")                 ;; (5)
    ❻(tags "PROJECT-MAYBE-DONE")))      ;; (6)
  ;; ... put your other custom commands here
 ))
```

- ❶ "u" is for "urgent", "todo" specifies that TODO headlines are to be shown, "WORK&URGENT" is the query string, and the last item means that there aren't any options
- ❷ "c" is for "call"
- ❸ "h" is for "home"
- ❹ The second item here is "tags" instead of "todo", which means that the highest-level matching headlines should be shown whether or not they're tasks.
- ❺ This is how to add a tag search for tasks into a custom agenda command.
- ❻ This is how to add a tag search for headlines into a custom agenda command.

Sometimes you'll want to see more context instead of just a list of headlines. You can jump from your Org agenda to the corresponding headline by pressing RET (`org-agenda-switch-to`) on the entry. You can also quickly browse through the headlines in another window by pressing f (`org-agenda-follow-mode`) while in the Org agenda view, then moving your point to the different lines. These commands work with the summary in the Org agenda view.

If you want to show only matching headlines in your `~/organizer.org` file, you can use Org's sparse tree search commands.

Tags in Your File

A sparse tree shows only the matching headlines in the context of the headlines above them. This is useful when you want to see your tasks within your outline structure. All other headlines are collapsed so that they're easy to skip. To do a sparse tree search, type C-c \ (org-tags-sparse-tree). You can then expand and collapse subtrees with the TAB (org-cycle) command. To limit the search to only task headlines, type C-u C-c \ (org-tags-sparse-tree with a prefix).

Other cool things you can do with tags

And if you ever want to know how much time you spent on urgent tasks, you can call the following function from your organizer.org file with M-x wicked/org-calculate-tag-time RET URGENT RET to see something like this:

```
Time: 98:44 (98 hours and 44 minutes)
```

You can call it with a prefix in order to be prompted for a start time (inclusive) and end time (exclusive). Here's the code to add to your *~/emacs*:

```
(defun wicked/org-calculate-tag-time (matcher &optional ts te)
  "Return the total minutes clocked in headlines matching MATCHER.
MATCHER is a string or a Lisp form to be evaluated, testing if a
given set of tags qualifies a headline for inclusion. TS and TE
are time start (inclusive) and time end (exclusive). Call with a
prefix to be prompted for TS and TE."
```

For example, to see how much time you spent on tasks tagged as URGENT, call M-x wicked/org-calculate-tag-time RET URGENT RET. To see how much time you spent on tasks tagged as URGENT today, call C-u M-x wicked/org-calculate-tag-time RET URGENT RET . RET +1 RET."

```
(interactive (list
              (read-string "Tag query: ")
              (if current-prefix-arg (org-read-date))
              (if current-prefix-arg (org-read-date))))
;; Convert strings to proper arguments
(if (stringp matcher)
    (setq matcher (cdr (org-make-tags-matcher matcher))))
(if (stringp ts)
    (setq ts (time-to-seconds
              (apply 'encode-time (org-parse-time-string ts)))))
(if (stringp te)
    (setq te (time-to-seconds
```

```

        (apply 'encode-time (org-parse-time-string te))))))
(let*
  ((re
   (concat "[\n\r]" outline-regexp " *\\(\\<\\( "
    (mapconcat 'regexp-quote org-todo-keywords-1 "\\|")
    (org-re
     (concat "\\>\\)\\)?"
      " *\\(\\..*?\\)"
      "\\(:[[:alnum:]_@:]+:\\)?[ \t]*$" ))))
   (case-fold-search nil)
   lpos
   tags tags-list tags-alist (llast 0) rtn level category i txt p
   marker entry priority (total 0))
  (save-excursion
   (org-clock-sum ts te)
   (goto-char (point-min))
   (while (re-search-forward re nil t)
    (catch :skip
     (setq tags (if (match-end 4) (match-string 4)))
     (goto-char (setq lpos (1+ (match-beginning 0))))
     (setq level (org-reduced-level (funcall outline-level))
           category (org-get-category))
     (setq i llast llast level)
     ;; remove tag lists from same and sublevels
     (while (>= i level)
      (when (setq entry (assoc i tags-alist))
       (setq tags-alist (delete entry tags-alist))
       (setq i (1- i)))
      ;; add the next tags
      (when tags
       (setq tags (mapcar 'downcase (org-split-string tags ":"))
             tags-alist
             (cons (cons level tags) tags-alist)))
      ;; compile tags for current headline
      (setq tags-list
            (if org-use-tag-inheritance
                (apply 'append (mapcar 'cdr tags-alist))
                tags))
      (when (and (eval matcher)
                 (or (not org-agenda-skip-archived-trees)

```

```

                (not (member org-archive-tag tags-list))))
;; Get the time for the headline at point
(goto-char (line-beginning-position))
(setq total (+ total (or
                    (get-text-property
                     (1+ (point))
                     :org-clock-minutes)
                    0)))
;; if we are to skip sublevels, jump to end of subtree
(org-end-of-subtree t))))
(if (interactive-p)
    (let* ((h (/ total 60))
           (m (- total (* 60 h))))
          (message "Time: %d:%02d (%d hours and %d minutes)" h m h m))
      total))

```

Now you can slice and dice your timeclock records any way you want, thanks to tags!

Org and Time Estimates

If you want to see how your time estimates fit into your workday with a load estimate like this:

15.8% load: 90 minutes to be scheduled, 570 minutes free, 480 minutes gap

add estimated number of minutes to your tasks like this:

```

** TODO 60 Browse through a book
** TODO 15 Scan my RSS feeds

```

and use the following code in your *~/.emacs*:

```

(defun wicked/org-agenda-load (&rest ignore)
  "Estimated load.
Can be included in `org-agenda-custom-commands'."
  (let ((inhibit-read-only t)
        (time (wicked/org-calculate-free-time
                ;; today
                (calendar-gregorian-from-absolute org-starting-day)
                ;; now if today, else start of day
                (if (= org-starting-day
                      (time-to-days (current-time)))
                    (let* ((now (decode-time))

```

```

        (cur-hour (nth 2 now))
        (cur-min (nth 1 now))
        (+ (* cur-hour 60) cur-min))
    (let ((start (car (elt org-agenda-time-grid 2))))
        (+ (* (/ start 100) 60) (% start 100)))
        ;; until the last time in my time grid
    (let ((last (car (last (elt org-agenda-time-grid 2)))))
        (+ (* (/ last 100) 60) (% last 100))))))
(goto-char (point-max))
(insert (format
        (concat "%.1f%% load: %d minutes scheduled, "
                "%d minutes to be scheduled, %d minutes free, "
                "%d minutes gap - %.1f total work hours planned\n"
                (/ (elt time 1) (* .01 (elt time 2)))
                (elt time 0)
                (elt time 1)
                (elt time 2)
                (- (elt time 2) (elt time 1))
                (/ (+ (elt time 0) (elt time 1)) 60)
                )))

```

```
(defun wicked/org-calculate-free-time (date start-time end-of-day)
```

"Return the minutes scheduled, unscheduled, and free for DATE.

Scheduled - all the minutes in appointments and tasks with times.

Unscheduled - all the estimated minutes in tasks without specific time.

Gap - free time between scheduled appointments and tasks.

In an ideal world, unscheduled would be less than gap.

DATE is a list of the form (MONTH DAY YEAR). START-TIME and

END-OF-DAY are the number of minutes past midnight."

```
(save-window-excursion
```

```

  (let ((files org-agenda-files)
        (total-unscheduled 0)
        (total-gap 0)
        file
        rtn
        rtnall
        entry
        (total-scheduled 0)
        (last-timestamp start-time)
        scheduled-entries)

```

```

(while (setq file (car files))
(catch 'nextfile
  (org-check-agenda-file file)
  (setq rtn
    (org-agenda-get-day-entries file date
      :scheduled :timestamp))
  (setq rtnall (append rtnall rtn)))
(setq files (cdr files)))
;; For each item on the list
(while (setq entry (car rtnall))
  (let ((time (get-text-property 1 'time entry)))
    (cond
      ((and time (string-match "\\([^-]+\\)-\\([^-]+\\)" time))
        (setq scheduled-entries
          (cons
            (cons
              (save-match-data
                (appt-convert-time (match-string 1 time)))
              (save-match-data
                (appt-convert-time (match-string 2 time))))
            scheduled-entries)))
      ((and time
        (string-match "\\([^-]+\\)\\.+" time)
        (string-match "^[A-Z]+ \\(\\([#[A-Z]\\) \\)?\\([0-9]+\\)"
          (get-text-property 1 'txt entry))))
        (setq scheduled-entries
          (let ((start (and (string-match "\\([^-]+\\)\\.+" time)
            (appt-convert-time
              (match-string 1 time))))
            (cons
              (cons
                start
              (and (string-match
                "^[A-Z]+ \\(\\([#[A-Z]\\) \\)?\\([0-9]+\\)"
                (get-text-property 1 'txt entry))
                (+ start
                  (string-to-number
                    (match-string
                      2
                    (get-text-property 1 'txt entry))))))
            scheduled-entries))))))

```

```

        scheduled-entries))))
((string-match "[A-Z]+ \\(\\#[A-Z]\\) \\)?\\([0-9]+\\)"
  (get-text-property 1 'txt entry))
(setq total-unscheduled
  (+ (string-to-number
      (match-string
        2
        (get-text-property 1 'txt entry)))
    total-unscheduled))))
(setq rtnall (cdr rtnall))
;; Sort the scheduled entries by time
(setq scheduled-entries
  (sort scheduled-entries (lambda (a b) (< (car a) (car b)))))

(while scheduled-entries
  (let ((start (car (car scheduled-entries)))
        (end (cdr (car scheduled-entries))))
    (cond
      ;; are we in the middle of this timeslot?
      ((and (>= last-timestamp start)
            (<= last-timestamp end))
       ;; move timestamp later, no change to time
       (setq total-scheduled
         (+ total-scheduled (- end last-timestamp)))
       (setq last-timestamp end))
      ;; are we completely before this timeslot?
      ((< last-timestamp start)
       ;; add gap to total, skip to the end
       (setq total-gap (+ (- start last-timestamp) total-gap))
       (setq total-scheduled (+ total-scheduled (- end start)))
       (setq last-timestamp end)))
    (setq scheduled-entries (cdr scheduled-entries))))
(if (< last-timestamp end-of-day)
  (setq total-gap (+ (- end-of-day last-timestamp) total-gap)))
(list total-scheduled total-unscheduled total-gap)))

```

This creates a new command that you can use in org-agenda-custom-commands in your ~/.emacs like this:

```
(setq org-agenda-custom-commands
```

```
'(("a" "My agenda"
  (org-agenda-list)
  (wicked/org-agenda-load)))
```

When you have that in your `~/.emacs` and estimated minutes in task descriptions like `** TODO 60 Read a book`, then you can use `C-c a a (org-agenda, "My agenda")` to view your load for the day.

Other Features of Org

You've learned how to set up Org for GTD or day planning. You've also learned how to keep track of your time, work with projects, and organize your task list using tags. Org has many other features to help you work with repeating tasks, export your Org file or agenda, repeating tasks, exporting your Org file or your agenda, and archive your finished projects. Check out the Org manual for more information, and join the mailing list at <http://lists.gnu.org/mailman/listinfo/emacs-orgmode> to keep up with new releases. Have fun!

Planner

Org isn't the only personal information manager available for Emacs. The EmacsWiki lists at least ten task managers at <http://www.emacswiki.org/cgi-bin/wiki/Category/TODO>. I can't do all of them justice, but I'd like to tell you about Planner, which also has a following among the Emacs crowd.

If you're the kind of person who likes scribbling free-form tasks and notes in your day planner, then Planner might be a good fit for you. In this section, I'll show you how to use Planner to organize the things you need to do by the day you need to do them, check the things you need to do, mark tasks complete, and review what you've finished. I assume that you've already got GNU Emacs 22 installed and that you're comfortable with using Emacs as a text editor, although it might not yet be your way of life.

Setup

Here's the bare minimum you need in order to use Planner to manage your tasks day by day. You'll need Planner, which is a separate package that you can get from <http://www.gna.org/projects/planner-el>. As of this writing, Planner is at version 3.41. You'll also need Muse, the markup engine that Planner is based on. You can get Muse from <https://gna.org/projects/muse-el>. As of this writing, Muse is at version 3.11.

Download the latest versions of Planner and Muse, and unpack them. If you don't know where to unpack them, I suggest creating an `~/elisp` directory and extracting the archives to that directory. You should end up with two new directories: `~/elisp/planner-3.41` and `~/elisp/muse-3.11`.

Create a directory such as *~/Plans*. This is where your Planner pages will be. Then add the following to your *~/.emacs*:

```
;; Load paths - change as necessary
(add-to-list 'load-path "~/elisp/planner-3.41")
(add-to-list 'load-path ❶ "~/elisp/muse-3.11/lisp")

;; Basic configuration
(require 'muse-project)
(require 'planner-autoloads)
(setq planner-project "WikiPlanner")

;; Adjust this if you already have other Muse projects
(setq muse-project-alist
      (('("WikiPlanner"
         (❷ "~/Plans"
          :default ❸ "TaskPool"
          :major-mode planner-mode
          :visit-link planner-visit-link)))

;; Some handy keyboard shortcuts
❹(global-set-key (kbd "C-c d") 'planner-goto)
(global-set-key (kbd "C-c t") 'planner-create-task-from-buffer)
❺(global-set-key (kbd "<f9> d") 'planner-goto)
(global-set-key (kbd "<f9> t") 'planner-create-task-from-buffer)
❻(global-set-key (kbd "<f9> <f9>") 'planner-goto-today)
❼(global-set-key (kbd "<f9> <f8>") 'planner-goto-yesterday)
(global-set-key (kbd "<f9> <f10>") 'planner-goto-tomorrow)

❸(plan)
```

- ❶ The lisp/ subdirectory is added to the load-path, not the base directory.
- ❷ Change this if you put your planner projects somewhere else.
- ❸ Muse will open the default page if you open a project.
- ❹ You'll use these two commands often, so it helps to keep them close by. You can remember them as C-c d(ate), which jumps to a Planner date, and C-c t(ask), which creates a Planner task.

- ⑤ I like dedicating one of my function keys to different Planner shortcuts. "F9 d(ate)" jumps to a Planner date and F9 t(ask) creates a task.
- ⑥ You'll check today's page frequently, so make it an easy-to-hit shortcut. I like F9 F9 because I can just tap the key twice to see my tasks and schedule.
- ⑦ This pair of keyboard shortcuts make it easy to navigate through pages.
- ⑧ If this line is one of the last few in your `~/.emacs`, then your Planner page for the day will display every time you start Emacs. It's a good way to remember that it exists. ;)

Evaluate the code with M-x eval-buffer, or restart Emacs. Then you're ready to plan!

Planner and day pages

The first thing you need to learn is how to get to day pages quickly. Planner needs to be at least as fast as opening a paper-based day planner and finding the right page. Here are the two keybindings from the previous section on setting up Planner:

F9 d(ate) or C-c d	plannergoto	Jumps to any day's page
F9 F9	plannergototoday	Shows today's page

F9 d (planner-goto) is smart. You can click on a date in the calendar that pops up, or navigate to a date and press RET. Typing in the date is much faster. The date format is yyyy.mm.dd, and it understands partial dates (mm.dd, or just dd).

For example, if the date today is December 28, 2007, here's what it understands:

Example	Result	Explanation
.	2007.12.28	Today
30	2007.12.30	Which day in the current month
1.30	2007.01.30	Which day and month in the current year
2008.01.30	2008.01.30	The full date

planner-goto also understands relative dates, and this is where things get more interesting. If you are looking at a day page, dates are calculated based on the day you're currently looking at, or today if you're not looking at a day page. This allows you to use "+2fri" to jump to successive paydays or use -1 to see the day before the one you're reading. Here are some examples that will show you what you can do:

1. Jump to the day page for December 28, 2007 with F9 d or C-c d, specifying 2007.12.28 for the date.
2. Use F9 d +5 to jump five days ahead. You should now see the page for 2008.01.02.

3. Use F9 d -3 to jump three days back. (2007.12.30)
4. Use F9 d +tue to jump to the next Tuesday. (2008.01.01)
5. Use F9 d -2fri to jump to two Fridays ago. (2007.12.21)
6. Use F9 d +2wed2007.12.01 to jump to the second Wednesday after Dec 1, 2007. (2007.12.12). +2wed01 would have worked, too.

Practice jumping around to different dates using F9 d or C-c d, the two shortcuts we set up earlier. If one of those shortcuts feels more natural to you, go ahead and use it. (Or bind it to something else, if you want.) While you're opening different day pages, write a few reminders to yourself.

The basic configuration I suggested also has some handy shortcuts for going to the previous and next days. Press F9 F8 (planner-goto-yesterday) to go to the day before the one that's currently displayed, and F9 F10 (planner-goto-tomorrow) to go to the day after the one that's displayed. You'll find this handy when doing your weekly planning. The standard shortcuts are C-c C-j C-y and C-c C-j C-t. You may find those easier to type, but they were like a game of Twister on my tiny keyboard.

So now you know how to open different day pages. You can stop here and already have a decent, minimalist day planner, using it like a collection of text files that just happen to have useful navigational commands. However, with a little more structure and some handy shortcuts, you can be even more effective at managing your tasks.

Creating tasks

Two of the keyboard shortcuts in the sample configuration are C-c t and F9 t, both bound to planner-create-task-from-buffer. This is an incredibly useful function, and it gets even better as you set up more parts of Planner. The key idea behind planner-create-task-from-buffer is that you should be able to quickly jot down a task and *get back to work right away*. No need to fiddle around with other files or dig your planner out of your backpack. No switching to another application (at least, if you do most of your work within Emacs). And if you set it up, you even get hyperlinks back to whatever you were looking at, saving you time in searching for the file you wanted to work on or the e-mail you wanted to answer.

Try it for yourself. Use C-c t, F9 t, or M-x planner-create-task-from-buffer to create a task. Type in the task description. For now, accept the default date and plan page. The task will be created on today's page. The task will also be copied to the TaskPool page. To view today's page, type F9 F9. You can use TAB (muse-next-reference) to move the cursor to the next hyperlink, and RET to visit the link.

planner-create-task-from-buffer understands all the date shortcuts that planner-goto does, so you can easily schedule a task for this Saturday (+sat) or three days from now (+3). If you create a task

that's scheduled for some other day, you can either open the day page with F9 d (planner-goto), or review it on the TaskPool.

Okay. You've got day pages. You've got tasks. You probably want to find out how to mark tasks as done before your growing TODO list turns into a monster and eats you.

Marking tasks as done, pending, or cancelled

When you finish a task, go to the day page or the plan page it's on and use C-c C-x (planner-task-done) to mark it as finished. Think of it as marking completed tasks with a big X. In addition to the satisfaction of seeing completed tasks grayed and crossed out, you'll also see the completed tasks drop to the bottom of your task list when you save the file. This makes it easy to see what else you need to do. Just pick the next item off your list and keep working.

Not quite done? You can mark it as pending with C-c C-p (planner-task-pending). You can think of it as Pending or Postponed or Procrastinated.

Realized that you didn't need to do it after all? Either delete the task with M-x planner-delete-task, or mark it as cancelled with C-c M-C-x (planner-task-cancelled). Think of C-c M-C-x as similar to C-c C-x (planner-task-done), but even better—you've gotten away without doing something. C-c M-C-x doesn't work on all terminals, so if your computer gets confused and marks the task as done, call M-x planner-task-cancelled instead.

NOTE: If your task is on both a day page and a plan page, make sure you use these Planner commands and M-x planner-edit-task-description in order to change the task status or description, and M-x planner-delete-task to delete the task. These commands update the linked page as well. If not, your tasks could get out of sync.

Even with your newfound powers of Planner task management, you'll probably still be left with unfinished business at the end of the day. Unfortunately, Planner does not have a M-x planner-dilate-time command, so you'll just have to reschedule the tasks for another day.

Rescheduling tasks

If you wrote your tasks into your calendar using a paper-based planner, you'd have to copy unfinished tasks to the next day one by one. This is a powerful incentive to trim your task list and keep it short. Planner can automatically copy unfinished tasks from the previous days onto today's page, saving you a lot of scribbling. Do not let this tempt you into procrastination.

If you go back to the basic configuration, you'll notice that it ends with one command:

(plan)

This reviews the past few days of pages for unfinished tasks, carrying them over to today's page. By default, the past 3 days are checked, which should be enough to get you through a blissful no-computer weekend. If you're in the habit of going for long spans of time without opening Emacs (like 4 days!), you may want to change the line in your `~/emacs` to something like

```
(plan 5)          ;; Check the last 5 days
```

or even

```
(plan t)          ;; Check all days. Can be slow!
```

You can also call this interactively with something like `C-u 5 M-x plan`, which checks the last 5 days.

The `plan` command carries unfinished tasks from previous days to today. What if you want to trim today's task list to a manageable size by proactively and intentionally procrastinating things that you don't need to do today? That's where `planner-copy-or-move-task` comes in.

To reschedule a task, move your cursor to the task on the day or plan page. Type `C-c C-c` (`planner-copy-or-move-task`) and specify the date. Again, `planner-copy-or-move-task` understands all the Planner date shortcuts. If you reschedule a task from a day page, remember that relative dates will be calculated based on the day page. For example, if you're on 2008.08.12 and you want to reschedule a task, `+1` means 2008.08.13. If you reschedule a task from a plan page, dates are relative to today.

If you want to reschedule many tasks, you might find it more convenient to use `M-x planner-copy-or-move-region`. Move to the beginning of the first task you want to move, press `C-SPC` to mark the beginning of the region, move to the end of the last task you want to move, and call `M-x planner-copy-or-move-region`.

Ruthlessly reschedule until your task list for today looks manageable. A large task list can be overwhelming. It feels better to complete everything on your task list and then add some more, than to end each day with many unfinished tasks.

Review

You've got your day pages. You've added and scheduled tasks. You've checked them off. At the end of the week, you're wondering where all the time went. Just hit `F9 F9` (`planner-goto-today`) to jump to today's page, and then use `F9 F8` (`planner-goto-yesterday`) and `F9 F10` (`planner-goto-tomorrow`) to navigate around. (See, those keybindings were there for a reason!)

To learn more about timeclocking, working with plan pages, integrating with mail, and other Planner goodies, and read the Planner manual. In fact, why don't you go and make that one of your tasks for today?

Day in the Life

In this section, I'll show you how I use Emacs to manage my day. I currently use Org to manage my tasks, but I also used Planner extensively when I was maintaining it. I hope that these stories will help you get an idea of how things fit together.

I spend most of my day working on or near a computer: writing, replying to e-mail, making phone calls, and so on. I use Emacs to keep track of what I need to do because I find Emacs more useful than commercial personal information managers. Here's how I work.

Org

The first thing I do each morning is quickly go through my e-mail looking for action items. Instead of leaving them in my mail inbox, I create tasks for each of the items in my Org to-do list. I add notes so that I don't have to find the e-mails again. I also add other tasks that come to mind. I add time estimates whenever I can.

This step is also a good time to review my personal objectives and proactively set myself some tasks so that I can move towards them. For example, if I want to learn more about career growth, I can't wait for e-mail from someone else telling me what to do. I have to think of something to do, and build time into my day in order to do it. I put all those tasks into my task inbox, which is just a header at the bottom of my task list.

Now that I've captured all these tasks, I review my day. What are my appointments? What items are due today? I double-check this against my official calendar to make sure I haven't missed any appointments. After I make sure that the important tasks and appointments are there, I start filling in gaps, scheduling some of my important-but-not-urgent tasks onto today.

After I've decided what to do that day, I review the rest of the week. Does my workload look reasonable? Are there upcoming deadlines? I try to make sure that all of my inbox items have scheduled dates. I also review my list of waiting tasks to see if I need to follow up with anyone.

After I've gotten the tasks all down and scheduled, it's easy to organize them under the appropriate headings. Billable tasks are filed under the projects they belong to, and unbilled tasks are categorized by how I need to report them and how I organize them into personal projects. Organizing tasks into categories *after* I schedule them means less jumping around looking for unscheduled tasks.

This process (e-mail scan and daily and weekly overview) takes me around 15 minutes, and helps keep me sane.

I try to keep my mornings free for creative brainstorming and heavy lifting. Once I've reviewed my plans for the day and the week, I pick a task and work on it for an hour or so. When I mark the task as STARTED ("t" or "C-u t" from the Org agenda view), the clock automatically starts ticking. When I mark the task as WAITING or DONE, the clock automatically stops. Org timeclocking makes it easier for me to report my hours at the end of each week. I can also compare the actual times against my estimates, helping me improve my accuracy.

This process is my morning ritual: putting tasks into my Org agenda, organizing them, reviewing and planning from my calendar, and then getting started by working on something useful *before* I spend time catching up with the blogosphere or responding to my mail. It's complemented by my afternoon ritual of reviewing my completed tasks with the Org agenda logbook and possibly blogging about some lessons learned.

Planner

The daily rituals I described for Org can also be done with Planner. In fact, I spent about four years managing my tasks using Planner before I tried out Org. My process then was similar to my Org process now in that I wrote the tasks down before I tried to organize or do them. I also built in some time for a review. The difference was that I didn't organize most of my tasks into separate projects (or plan pages, as Planner calls them). Instead, I tended to organize them according to day. This was helpful when publishing my blog, as I could post my task list along with it. I also liked the greater control I had over my daily task list, and I often used blank lines or extra notes to keep things organized.

The core of the process was the same, though:

1. Get the tasks out of your head.
2. Schedule the tasks.
3. Do the tasks.
4. Review the tasks.

Get the Tasks out of Your Head

The first thing I did each day was to put all of my tasks into my Planner. I briefly scanned through my mail looking for action items, and whenever I found something I needed to do that would take more than two minutes, I created a task for it using `planner-create-task-from-buffer`. This automatically

created a hyperlink back to the e-mail or file I was reading, and included useful information such as the e-mail author.

I treated appointments just like tasks, although I added a timestamp like @13:00 to make it easier to see and sort my appointments.

planner-create-task-from-buffer helped me keep track of reactive tasks, or things that I needed to do because someone else asked me to do them. However, I also made a point of reviewing my goals and making up my own tasks so that I could work on my personal projects. For example, I had a plan page where I put all my writing ideas, and I added those tasks to today's page once in a while.

I also scanned the last few day pages to see if I'd missed anything, and the next few days to see if I needed to plan for anything. Although the M-x plan command automatically brings tasks forward, I didn't use it to start my day. Instead, I reviewed the last three days or so, marking tasks that I'd completed but not updated, cancelling tasks I no longer wanted or needed to do, and rescheduling other tasks forward using C-c C-c (planner-copy-or-move-task). I also reviewed the next three days or so. C-c C-j C-y (planner-goto-yesterday) and C-c C-j C-t (planner-goto-tomorrow) were handy for flipping through pages. To make the shortcuts shorter and more natural, I bound them to F9 F8 and F9 F10 with:

```
(global-set-key (kbd " ") 'planner-goto-yesterday)
(global-set-key (kbd " ") 'planner-goto-tomorrow)
```

At the end of this step, I had all the things I needed to do scheduled for today.

Organize the Tasks

Getting the tasks out of my head often resulted in a long list of tasks, not all of which I needed to do or could even do that day. Then it was time to ruthlessly use C-c C-c (planner-copy-or-move-task) and M-x planner-copy-or-move-region to trim my task list down to a manageable size. I moved tasks that didn't need to be done on a certain date to the [TaskPool](#), which I checked whenever I had some free time.

If I wanted to assign a task to a specific time, I just added a timestamp such as @14:30 to the task. I had some code which automatically sorted the tasks by time, so it would go to the right place.

I also used M-down and M-up (planner-lower-task and planner-raise-task) to move tasks around. To visually group tasks, I added blank lines and explanatory text. For example, I put errands together and I moved the tasks up and down into a logical order for doing things.

If there were a lot of small items in my day, I also separated the "must-be-dones" from the "nice-to-dos". This wasn't related to the importance of the task, just the urgency. Urgent items went in the first group, and non-urgent items went in the second group of tasks.

At the end of this step, I had a daily plan page which showed me my tasks and appointments for the day, in the rough order in which I planned to do them. I also had some tasks on future days, and some tasks in my TaskPool.

Do the Tasks

Now that all my tasks were on the page, it was easy to go through them. To start working on a task, I marked it as in-progress with C-c C-i (planner-task-in-progress). If I needed to postpone it, I used C-c C-p (planner-task-pending). To mark it finished, I used C-c C-x (planner-task-done). To mark it cancelled, I used C-c C-S-x (planner-task-cancelled; does not work in some terminals).

Because I had loaded planner-timeclock.el using code like this:

```
(require 'planner-timeclock)
```

the clock automatically started when I marked tasks as in-progress and stopped when I marked tasks as pending or done. I could also clock out of a task manually by using C-c C-o (timeclock-out).

Review the Tasks

To review what I did that day, all I had to do was go to the day page using planner-goto-today, which I bound to F9 F9 with:

```
(global-set-key (kbd "<f9> <f9>") 'planner-goto-today)
```

I checked my timeclock reports with M-x planner-timeclock-summary-show, which gave me reports that looked like this:

```
Timeclock summary report for 2007.11.25
```

Project	Time	Ratio	Task
Not Planned	0:28:38	38.6%	Do weekly review
.	0:45:37	61.4%	Figure out blog on Wordpress
Total:	1:14:15	100.0%	.

```
Day began: 21:01:20, Day ended: 22:16:06
```

```
Time elapsed: 1:14:46, Time clocked: 1:14:15
```

```
Time clocked ratio: 99.3%
```

It's easy to configure `planner-timeclock-summary` to add that report to day pages automatically, but I rarely checked my time usage, so I didn't set that up. I also didn't usually need to see my time summary for a particular project, but you could get that with `planner-timeclock-summary-proj.el`.

So at the end of this step, I had a warm and fuzzy feeling from seeing many checked-off tasks on my task list. I published my task list to the Net, too, which made it easy for other people to keep up to date with what I was doing. Good stuff.

Planner helped me keep track of the different things I wanted to do. Working with the other Planner geeks was also an incredible experience. I've switched to Org for my task management because Org does timeclocking better, but I still miss being able to easily organize, publish, and tweak my daily task list. If you're new to Emacs planning, I suggest giving both Org and Planner a try.

Wrapping Up

In this chapter, you learned how to manage your tasks in Emacs using either Org or Planner. Emacs can help you manage not only your tasks, but your schedule, too. Find out more in Chapter 9!