

Efficient Note-taking in Emacs with Howm

Andrei Sukhovskii

Efficient Note-taking in Emacs with Howm

2nd edition

Andrei Sukhovskii



Contents

| | | |
|-----------|-----------------------------------|-----------|
| 1 | Introduction | 4 |
| 2 | The Concept of Howm | 6 |
| 3 | Emacs Key Combinations | 8 |
| 4 | Installing Howm | 10 |
| 5 | Howm Main Menu | 12 |
| 6 | Creating Notes | 15 |
| 7 | Working with Lists | 17 |
| 8 | Sorting Notes in Lists | 21 |
| 9 | Linking Notes in Howm | 23 |
| 10 | Come-from Links (keywords) | 28 |
| 11 | Search Functions | 31 |
| 12 | Filter Functions | 33 |
| 13 | Deleting Notes | 36 |
| 14 | Stitching Notes | 37 |
| 15 | Handling Long Notes | 39 |
| 16 | Working with Buffers | 42 |
| 17 | Quick Notes | 43 |
| 18 | Duplication | 45 |
| 19 | Descriptive Names | 46 |
| 20 | Task Types in Howm | 48 |
| 21 | Making and Listing Plans | 53 |

| | |
|--|-----------|
| <i>CONTENTS</i> | 3 |
| 22 Fine-Tuning Plans | 57 |
| 23 Advanced Settings | 60 |
| 24 Questions and Concerns | 67 |
| 25 Howm Command List | 68 |

Chapter 1

Introduction

This is a tutorial on taking notes in Emacs using the Howm package. I assume that readers have already recognized the importance of “thinking by writing” and are familiar with the fundamental principles of knowledge management.

Essentially, you can count these principles on one hand:

1. Take notes on anything you find useful, important, interesting.
2. Limit the length of each note to one idea; this makes it easier to construct them later on in any manner.
3. Link notes when their respective ideas engage in a dialogue.
4. Use a reliable program that you can trust for your writing.

This tutorial is about such a program. I will try to show why functions from the Howm arsenal are useful for organizing intellectual work. However, I do not want to impose Howm on the reader. There are other good solutions (even within Emacs). I only hope that you will appreciate the simplicity and reliability of this package.

This text is tailored for absolute beginners in Emacs who are just learning the basics. Throughout the narrative, I will explain some universal techniques that are applicable not only in Howm. Naturally, experienced users may get bored with all this. They are free to choose the sections of the tutorial that best suit their needs.

I did not intend to create an exhaustive technical manual, so I have omitted some specifics. It’s also important to mention that this is my subjective perception of the tool. This book doesn’t prescribe how to use Howm, but rather explores its potential applications.

While writing this text, I repeatedly pestered the creator of the Howm package—Kazuyuki Hiraoka—with questions. I would like to express my gratitude to him for his patience and willingness to help. Domo arigato!

**

The first version of the tutorial in English appeared online in December 2023. Since then, Howm has gained several useful features, along with a more polished design.

Initially, I intended only to review the latest enhancements in Howm and refresh the text appearance using L^AT_EX. However, this required minor reorganization of the content, which led to major refinements and, eventually, extensive editing.

As a result, I ended up with a completely rewritten book. I hope this new edition helps readers integrate Howm into their workflow and develop a strong note-taking habit.

Andrei Sukhovskii
August 6, 2025

Chapter 2

The Concept of Howm

The name “Howm” is an acronym for the Japanese phrase “Hitori Otegaru Wiki Modoki” (一人お手軽 Wiki もどき), which can be roughly translated as “Handy Own Wiki Mimic.”

The Howm package surprises with its compactness and elegant solutions. It makes sense that the program is from Japan, the land of “Wabi-sabi.” The author, Kazuyuki Hiraoka, is a professor at Asia University (Japan), Faculty of Business Administration, Department of Data Science. The project is over 20 years old, as Hiraoka started it in 2002.

What’s the appeal of Howm? What’s so special about it? In a nutshell, it was invented for unorganized people.

Hiraoka made a good point about the predicament of note-taking:

If you don’t organize notes, reading feels like a hassle.

If you try to organize notes, writing feels like a hassle.

This trade-off is a dilemma.

“Hassle” is an important keyword. There is a big difference between “I can do it” and “I can do it easily.”

And then he writes:

The point of compromise may vary from person to person, but I prioritize the ease of writing. I have designed a memo tool to create an environment where one can freely jot down notes without feeling pressure to “organize” them and still maintain coherence.

Hiraoka claims that rigid structure and categorization quickly become a hindrance. Howm’s motto—“Write fragmentarily and read collectively”—in my opinion, captures the essence of a good note-taking system.

Howm is a minor mode in Emacs that plays well with many other modes. Whether your notes are in .txt, .org, or .md format, they can be part of your wiki.

The Howm syntax predates both Org-mode and Markdown. It’s distinctive yet highly intuitive and easy to use. Naturally, you can customize Howm to your liking—after all, it’s an Emacs package! We’ll cover some tinkering tricks in the *Advanced Settings* chapter.

Each new note in Howm is a new file that gets a unique number based on the date and time it was created. There are no tags or even links in the traditional

sense. Instead of this, Kazuyuki Hiraoka employed a clever trick. In the Howm, a link serves as a trigger for a search query. The outcome is a list of notes that contain the linked word.¹

The functionality of Howm was inspired by the file organization system of Japanese economist Yukio Noguchi.

In a nutshell, Noguchi's system suggests replacing thematic information distribution with a frequency-based distribution, ensuring that the most important documents are always at hand.

Howm follows a similar principle. It does away with categories or sections,² opting for a general list where notes are organized based on the date of the last edit. This algorithm of working with information turns out to be more efficient than striving for perfect order.

Howm's architecture drew inspiration from various applications, including ChangeLogMemo, Q-Pocket, HyperCard, HashedWiki, and more.

We will delve into a discussion of Howm features later, but before that, I need to make a couple of introductory remarks about using Emacs, installing new packages, and so on. Experienced Emacs users can skip straight to the *Howm Main Menu* chapter.

¹Similarly, links handling was organized in Notational Velocity, an application developed by Zachary Schneirov in 2002.

²This approach is also common among Zettelkästen adherents.

Chapter 3

Emacs Key Combinations

I'm assuming you already have Emacs installed on your computer. Congratulations! You can work with documents, create and edit notes, track various tasks, and do many other things.

In the next section, we'll discuss the installation process for Howm. Before diving into that, I should cover the basic keyboard commands in Emacs.

If you are new to Emacs, many things may seem unfamiliar. The best advice is to go through the built-in Emacs Tutorial, which is readily available on the start screen right after launching the program (It will be the first hyperlink).

Emacs has a very long history and its own, formed over decades, terms, ways of solving problems, etc.

One of the key features of Emacs is its “keyboard-centric” design. Virtually any action can be executed without lifting your fingers from the keyboard. It may initially slow you down, but once you become accustomed to it, every action you take will be two to three times faster.

This approach also aids in avoiding distractions caused by routine operations as they become ingrained in your muscle memory. Of course, this principle isn't exclusive to Emacs. Even in Microsoft Word, it's more efficient to copy text with **Ctrl-c** and paste it with **Ctrl-v** than to navigate with the cursor to the corresponding icons.

In addition to the general Emacs hotkeys, there are specific commands for different packages. In the upcoming chapters, we will explore the commands utilized in Howm. The list of these commands can be found at the end of this tutorial (see page 68). The purpose of this section is to demonstrate how to decipher Emacs command notation.



The command `C-x C-s` denotes the following sequence of actions:

1. Press and hold the `Ctrl` key, then press the `x` key.
2. Press and hold the `Ctrl` key, then press the `s` key.

This command can be simplified, as both parts of it start with the `Ctrl` key. Just press and hold the `Ctrl` key, then press the `x` key followed by the `s` key.

Commands are case-sensitive. If a keyboard shortcut specifies a capital letter, that is the letter you should use.

Consider, for example, the command `C-u C-c , K`.

This means you should press and hold the `Ctrl` key, then press the `u` key followed by the `c` key. After that, press `,`. And `Shift ↑ + K`. Don't worry, as most commands in Emacs are only 2-4 characters long.

The command `M-x` means: Press and hold the `Alt` key, then press the `x` key.

The letter "M" is short for Meta. There used to be a Meta key on some keyboards. Today, the `Alt` key is most often used instead, but the tradition of labeling it with the letter "M" has survived.

You can also press the `Esc` key once and then press `x` to achieve the same result. While less convenient for typing, this method is useful when other programs have already claimed Alt-key combinations. In such cases, `Esc` helps you "escape"—literally and figuratively!

The command `M-x`, by the way, invokes a special dialog—the minibuffer at the bottom of the screen.

Strictly speaking, any keyboard shortcut is a way to avoid typing lengthy commands. For instance, the keyboard shortcut `C-c , c` is responsible for creating a new note in Howm. In this case, Emacs actually executes the "howm-create" command. If you invoke the minibuffer by `M-x` and type `howm-create`, it will achieve the same result.

In most cases, shortcuts do not need to be intentionally memorized. Through regular use, they tend to "stick" to your fingers naturally. This is especially true for common, everyday commands. Muscle memory develops quite quickly. It's akin to learning fingering on the piano.

But of course, Emacs offers both icons and menus for those not ready to change habits. No pressure!

Chapter 4

Installing Howm

Even without any enhancements, Emacs stands out as an excellent text editor. Numerous intricate features are available immediately after installation. However, Emacs features can be adapted to your needs thanks to additional packages.

Howm can be installed on Emacs just like many other extensions. So let's talk about the general techniques.

Before proceeding, let's first check which packages are already installed.

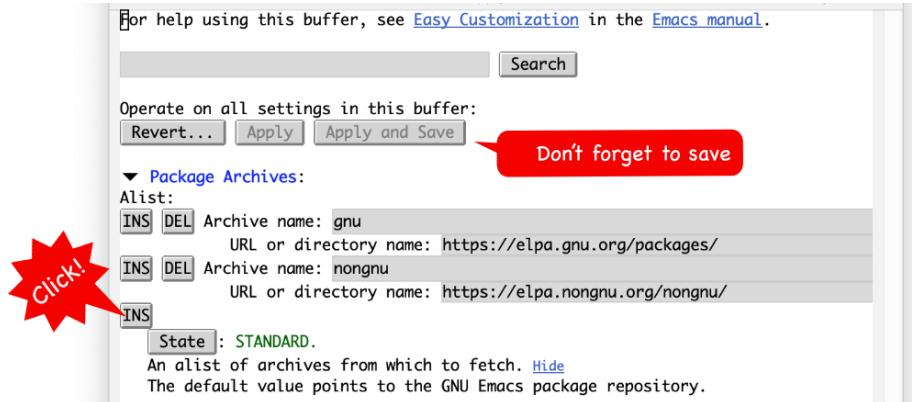
When you start Emacs, it opens a welcome screen, with a logo and introductory information. On the top, there is a menu bar containing various choices. To view a list of packages, navigate to the *Options* tab and choose “Manage Emacs Packages.”

After pressing **RET** on “Manage Emacs Packages,” a new window will appear, displaying an extensive list. The built-in packages will be at the very end of it. Feel free to install additional packages according to your needs, but do so thoughtfully to avoid cluttering Emacs with unnecessary features.

The ELPA repository (Emacs Lisp Package Archive) is accessible by default, offering numerous useful packages. However, some developers prefer the more democratic MELPA (Milkypostman’s Emacs Lisp Package Archive). We need to combine both lists within our Emacs setup since the Howm package is hosted on MELPA.

Fortunately, there is a simple way to accomplish this.

1. Press **M-x**.
2. Type `customize-variable` in the dialog box that opens below and press **RET**.
3. Type `package-archives` and press **RET**. You will be directed to the Package Archive page.
4. Click on **INS** button.
5. In the open form add Archive name *Melpa* and directory name:
`https://melpa.org/packages/`
6. On the top of the page, press **Apply and Save** button.



Now that you have connected the MELPA archive with additional packages to your Emacs, examine the expanded list using the command: **M-x list-packages**. Installing and removing packages in Emacs is straightforward:

1. Press **M-x**.
2. Type `package-install` and press **RET**.
3. Type the name of the package—`howm` in our case—and press **RET**.

Now that the Howm package is installed, we need to inform Emacs by configuring its activation.

Use command **C-x C-f**.

Type `.emacs` and press **RET**.

In open document, add the following line below the others:

```
(require 'howm)
```

Please note that the mark before the word `howm` must be a straight typewriter apostrophe. A Punctuation apostrophe ‘ won’t do.

Use command **C-x C-s** to save result.

The `.emacs` file stores all the system modifications you have made in Emacs. Emacs will return to its default settings if you delete this file.

Following these steps should be enough for the package to work properly. However, you can customize Howm to better fit your needs and preferences (we will discuss some options in the *Advanced Settings* chapter).

Chapter 5

Howm Main Menu

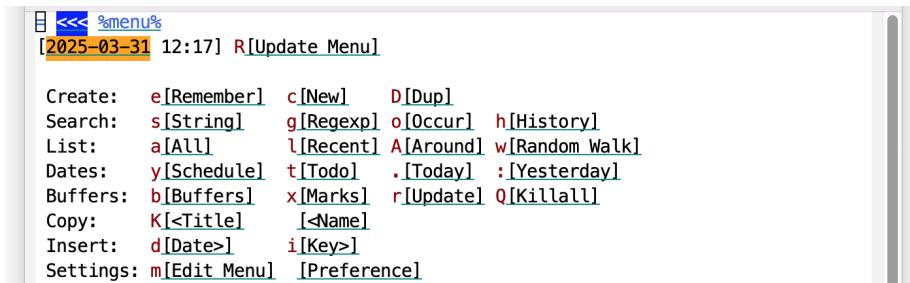
You can become familiar with the basic functions of Howm in a matter of hours.

The first and most important thing to remember is the launch command for Howm: `C-c , ,`.

This takes you to the main menu, which functions much like the captain's bridge. From here, you can easily navigate, issue commands, and monitor various compartments of your "ship."

The menu has different segments of information. We will review each segment separately.

At the top is a list of the basic commands grouped by type.



The screenshot shows a terminal window with the title bar "%menu%". The date and time are displayed as [2025-03-31] 12:17. A red box highlights the command `R[Update Menu]`. Below the title bar, there is a list of commands grouped by category:

| | | | | |
|-----------|-----------------------------|---------------------------|------------------------|-----------------------------|
| Create: | <code>e[Remember]</code> | <code>c[New]</code> | <code>D[Dup]</code> | |
| Search: | <code>s[String]</code> | <code>g[Regexp]</code> | <code>o[Occur]</code> | <code>h[History]</code> |
| List: | <code>a[All]</code> | <code>l[Recent]</code> | <code>A[Around]</code> | <code>w[Random Walk]</code> |
| Dates: | <code>y[Schedule]</code> | <code>t[Todo]</code> | <code>.[Today]</code> | <code>:[Yesterday]</code> |
| Buffers: | <code>b[Buffers]</code> | <code>x[Marks]</code> | <code>r[Update]</code> | <code>Q[Killall]</code> |
| Copy: | <code>K[<Title]</code> | <code>[<Name]</code> | | |
| Insert: | <code>d[Date]</code> | <code>i[Key]</code> | | |
| Settings: | <code>m>Edit Menu</code> | <code>[Preference]</code> | | |

The underlined items are active. They serve as "buttons." Placing the cursor on them and pressing `RET` will execute a command or follow a link.

You can jump between "buttons" using the Tab key to jump forward and `Tab` + `Alt` to jump backward.

All basic Emacs keystrokes for moving the cursor are available as well.

- `C-n` → next line
- `C-p` → previous line
- `C-f` → forward
- `C-b` → backward

You can, of course, move the cursor with the mouse or touchpad if you are more accustomed to it at this stage. However, the Howm menu provides a much simpler and faster method for performing actions.

Each command has an assigned key (a red letter shown before the square brackets). Pressing `c` starts a new note, pressing `A` lists all notes in the order they were created, and so on.

Naturally, there's no need to open the menu every time you use Howm—it primarily serves as a help system. Most of the commands found in the menu are also accessible via hotkeys. You can trigger them immediately after launching Emacs or while working in other documents outside of Howm.¹ Just use the keyboard shortcut [C-c ,] followed by the character shown in the menu.

We won't go through the entire "alphabet" of commands here, as we'll cover them in detail in later chapters. However, some commands are worth considering right away because they're specific to Howm's menu.

The date and time you opened the Howm's menu are shown at the top of the screen. All information in the menu is relevant to this time.

Howm's menu doesn't update dynamically. So if it's open in another window, tab, or frame, any changes you make to your notes won't be reflected. You'll need to reopen the menu or manually refresh it by pressing the [R] key. This will refill the sections with your latest plans and notes.

By the way, the menu is highly customizable—you can change it to your liking. It's just a .txt file in the Howm folder. Press the [m] key and it will open menu settings. You can regroup items, add new elements, and more.

From the menu, you can also access the Howm settings and change the behavior of the package. Use the [Preference] button for that (check out the *Advanced Settings* chapter for more info). However, I suggest keeping everything as it is until you feel more confident.

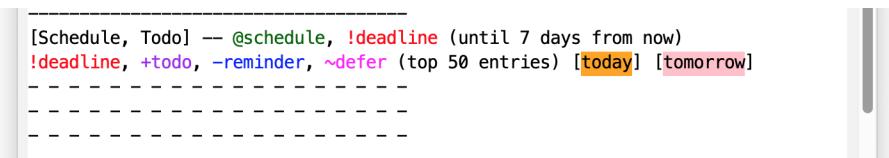
Keep in mind that some commands are intended for editing notes. When called from the menu (which is read-only), they are applied to the previously opened file. You can insert a date [d] or keyword [i] into the text—they will appear at the position where the cursor was in that document (so be careful not to insert a date in the middle of a word).

You can also apply the text search [o] to the file you were working on before accessing the menu.

Without leaving the menu, you can copy the title [K] or numeric name² of a previous note, and create a duplicate [D].

And of course, you can easily return from the menu to the note you were working on by pressing the [r] key.

The next segment of Howm's menu is dedicated to the time management.



```
[Schedule, Todo] -- @schedule, !deadline (until 7 days from now)
!deadline, +todo, -reminder, ~defer (top 50 entries) [today] [tomorrow]
```

Howm incorporates a rather flexible system of organization and planning. Empty lines will be filled with different plans once you write them down in your notes. We will return to this topic in dedicated chapters later.

Under the block with plans, you'll find lists of recent and random notes—a chance to stumble across something you've forgotten. By default, each section

¹ Howm responds to commands from major modes. If you are running some minor mode, you need to activate Howm: [M-x] howm-mode [RET]. In general, if a command doesn't work, check if Howm is activated.

² As you can see, there's no dedicated key for this command. You'll need to use the "button" from the menu: [<Name>].

displays up to 20 notes.

```
-----
Recent
> 0000-00-00-000000 | = <<< %menu%
> 0000-00-00-000000 | = <<< %Editing Menu%

Random -- If you do not like this, [Edit Menu] to erase it.
> 0000-00-00-000000 | = <<< %menu%
-----
```

Since you've just started, there's nothing here yet except the Howm system files. These will disappear from view once you've created enough notes.

All the information doesn't fit in the window. You can swipe down by pressing the space bar or swipe up by pressing the backspace key.³

The rest of the menu serves as a reference section, where you'll find mnemonic hints on planning syntax

```
Format of schedule and todo (Please replace {} with []):
{2002-10-21}@1 schedule -- (shown in schedule part. @3 = "3 days schedule to 10@2
@-23")
{2002-10-21}+7 todo -- float up slowly from the date in 7 days
{2002-10-21}!7 deadline -- float up fast from 7 days before the date
{2002-10-21}-1 reminder -- float at the date and sink slowly one unit per 1 day
{2002-10-21}~30 defer -- float at the date and repeat sinking and floating with a
@30 days period
{2002-10-21}. done -- sink forever
(Numbers after marks are default values.)

How to remember:
- Schedule at(@) the date
- Reminder sinks down(-).
- Todo floats up(+).
- Deadline needs attention!
- Defer waves(~) up and down.
- Done is the end(.) .
```

and a side note that the menu functions as a “modal dialog.”

```
-----
Think of this menu as a "modal dialog" rather than a
workspace. Typically, the commands listed in the menu apply to the
buffer BEHIND the menu, not the menu buffer itself. In other words,
they affect the buffer that was active before the menu appeared.

You can edit this menu itself.
>>> %Editing Menu%
-- 
[Update Menu] [Edit Menu]
```

To exit the Howm menu, simply press the `q` key. This will return you to the previously opened buffer.

³Standard Emacs commands `C-v` and `M-v` perform the same action.

Chapter 6

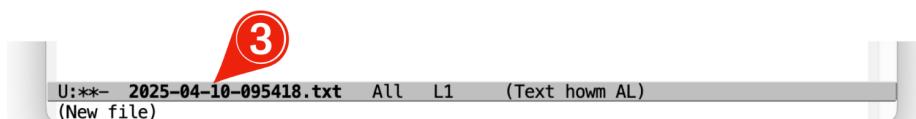
Creating Notes

Let's create a first note in Howm. If you are in the main menu, it is enough to just press the **c** key. It naturally corresponds to the initial letter of the word "create."

You can perform the same action, bypassing the main menu, with the **C-c , c** command. After starting Emacs, type this keyboard shortcut, and Howm will open a new note for you. What do we have here?



- ① By default in Howm, headings are labeled with an equals sign. This field can be left blank, although headings are helpful for navigating your records.
- ② The date and time the document was created. Today's date will be highlighted in orange.



- ③ The unique identifier of a note is its numeric name, which is based on a timestamp. You can see it on the mode line (the gray ribbon at the bottom of the screen).

That's the full "anatomy" of a note. You can type a title, or just move your cursor below the date and start writing. It's as simple as that.

Howm saves records in **.txt** format by default, so styling text—like bold or underline—is not possible. You can change the file extension to enrich text

design. Another option is to place an .org or .md file in the Howm folder, and their specific formatting will still work.

After you have written your first note, save it with the standard Emacs command: **C-x C-s**.

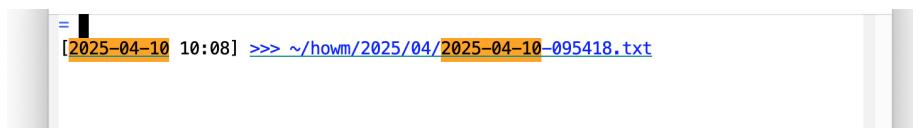
Checking the main menu, you'll notice that it appears in the Recent list.



To visit a note from the main menu, move the cursor to the corresponding line in the “Recent” section and press **RET**.

As you write your first note, you may be visited by ideas that further develop the topic. It is tempting to write them immediately, turning the text into an endless scroll. It is better, however, to keep notes modular.

While you are still in the first note, create another one using the same command **C-c , c**.



Notice that the new note template is slightly different. A link to the previous note appears next to the date. It's a handy way to signal the continuation of a thought.

If you don't need this link, use the undo command: **C-/**.

If you change your mind about creating a note at all, use undo again.

Speaking of saved notes, they are stored in the Howm folder within your home directory. Howm creates a subfolder for each year, and within each year, a subfolder for each month. The full file path looks like this:

```
/Users/UserName/howm/Year/Month/
```

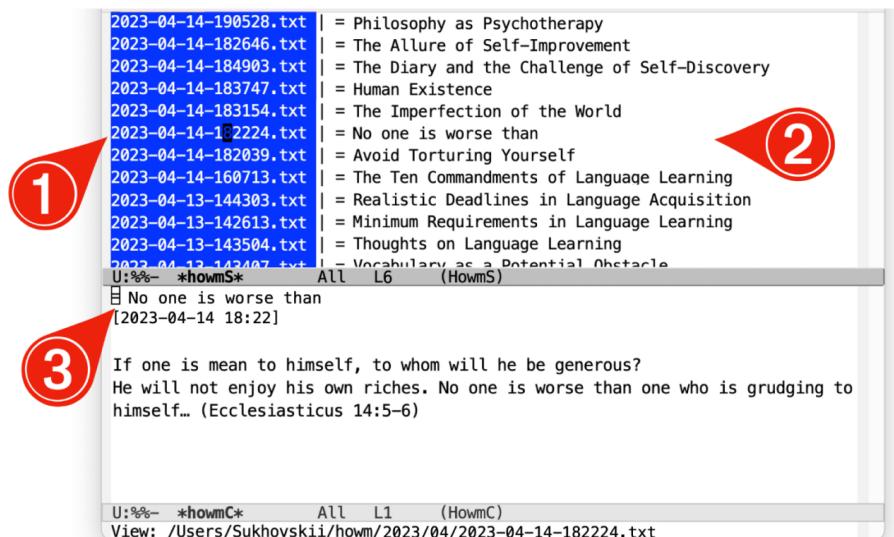
However, the folder structure isn't visible in the Howm interface. Notes are grouped together without any division.

At some point, you will have a lot of notes. How do you find the right one? How do you make an overview of your notes? Howm manages these tasks with the help of lists.

Chapter 7

Working with Lists

Commands like viewing recent notes, listing all notes, performing searches, or navigating through links will switch Howm into a dual-window layout. This interface is central to most interactions with your notes.



- 1 On the left side of the top window, a column displays the numeric names of the notes, highlighted in blue. Orange highlights will mark any notes created today.
- 2 The right side of the list displays note titles; if a title isn't set, the first line of the note is shown instead.
- 3 The bottom window contains a preview of the note the cursor is currently positioned on in the list. As you move the cursor, the preview updates dynamically.

Since lists are a key element of working with Howm, it's worth taking a closer look at the features and functions they offer. No matter which list you're viewing, the basic principles for interacting with it remain the same.

You can call up the list of all notes from the Howm menu by pressing the `[a]` key, or from anywhere else using the shortcut `[C-c , a]`. Notes are arranged by the time of their last edit, with the most recently modified ones at the top.

The idea is that the files you've worked on most recently are the ones you're most likely to need again—the core principle behind Yukio Noguchi's system.

The command opens a screen like the one in the illustration above. Now, let's talk about how to navigate through the list.

You can move the cursor using the `[n]` and `[p]` keys, down and up, respectively.¹ The bottom window flips through the notes as you move the cursor around the list.

Sometimes, a note's full content doesn't fit in the preview window. While in the list, you can scroll through the note preview. Jump to the next chunk of text using the Spacebar, or move slowly, line by line, using the `[j]` key. To scroll backward, use Backspace (Delete key on Macbooks) or the `[k]` key, respectively.

If you want to jump to the bottom window, press `[RET]`. The cursor will then move to the note, allowing you to make any necessary additions or edits.

Notice that the top window with the list will remain open. This is useful for all sorts of quick manipulations with text. You can return to the upper list using the `[C-x o]` command, which is the conventional Emacs way of switching between windows.

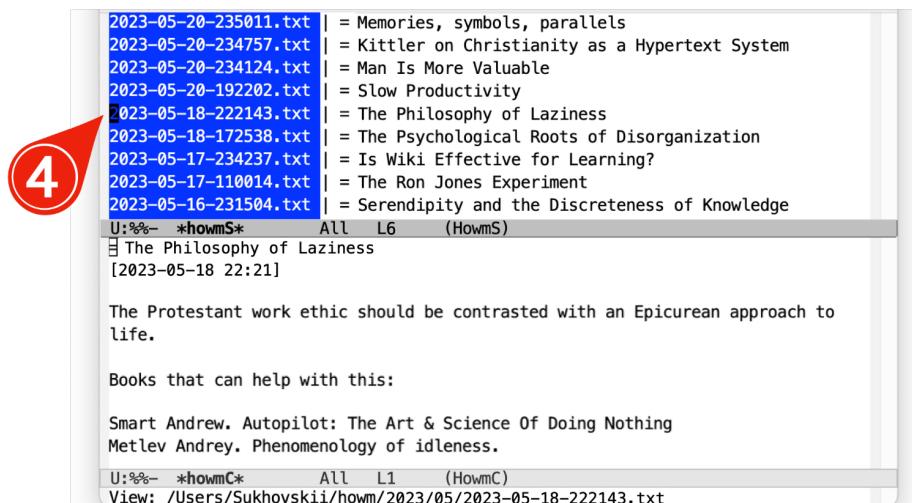
Sometimes you may want to remove the upper list and focus solely on the note content. Place the cursor on the line with the target note and press `[C-u RET]`. Or, if you're already in the note window, press `[C-x 1]`. This is a standard Emacs command that closes all windows except the current one.

If you want the exact opposite—no preview window and just an overview of the list of notes—you can close the bottom window by simply pressing `[1]`. To restore the preview, press `[2]` on your keyboard.

Alternatively, you can switch back and forth by pressing `[v]`. However, if you move the cursor, Howm will reopen the preview window. To make better use of this feature, you can disable the note preview while scrolling through the list-only screen (see the *Advanced Settings*, Tip #7, p. 63).

There's another command that also displays all notes, but in the order they were created. You can access this list from the Howm menu by pressing the `[A]` key, or from within an open note using the `[C-c , A]` shortcut. In the latter case, the list will be contextual—centered on the current note.

¹The standard Emacs commands `[C-n]` and `[C-p]` work too.



- ④ A list with the cursor on the current note, positioned in the middle. It is surrounded by notes created before and after. This is sometimes useful to reconstruct your train of thought.

Having a list of all your notes is important for the various types of sorting we'll cover in the next chapter. But it's also useful on its own.

For example, you can quickly search by title. Just use the standard Emacs text search **C-s** and start typing the name of the note. You don't even need to begin with the first word. Press **C-s** again to jump between search results.

You can also check how many notes are in your wiki. To do this, use the line count command: **C-x 1**. The total will appear in the minibuffer.²

You probably don't need the full list of notes every time. For daily use, the recent notes list tends to be more useful. You can access it from the Howm menu by pressing the **1** key, or from anywhere else using the **C-c , 1** shortcut.

Howm will then display notes that were created or edited within the last seven days. This gives you a quick overview and makes it easy to switch between notes you're currently working on. If there are no recent notes in your wiki, Howm will prompt you to create one.

Do you need a longer list? Here is an example of how you can request a 20-day summary: **C-u 20 C-c , 1**.

By combining variations of this command, you can view the files you worked with this month or quarter. This is useful for summarizing results and self-analysis.

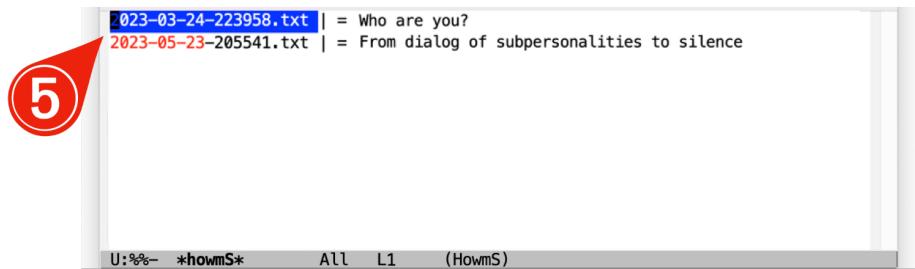
While the cursor is positioned in any list, you can use single-key commands—just like from Howm's menu. Press **a** to view all notes, or **1** for recent ones. You can create a new note with **c**, or return to the main menu with **,**. We'll cover other single-key commands in the corresponding chapters.

²Since there are two lines from the system file in the entire list, subtract them from the resulting number.

What if you want to see all the notes entered into the database today? They are already highlighted in orange. However, if you want a separate list, use the `C-c , .` command.

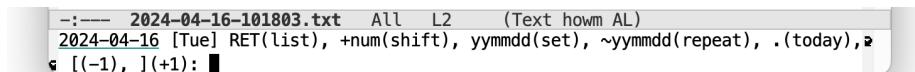
Keep in mind, that Howm will display only the list of created notes; edited notes are not counted in this case.

If you use links to numeric filenames (more on that later), the list will also include older notes that are linked to today's notes. That's because Howm searches for timestamps in both the filenames and the note content.



- 5 Older notes will be highlighted in blue, while today's notes will have a white background.³

In fact, if you place the cursor on the date in square brackets in the note heading and press `RET`, the minibuffer dialog will appear.



Press `RET` again to evoke the list of notes associated with that date. We'll cover the rest options in this minibuffer in the chapter *Making and Listing Plans*.

To see a list of notes created yesterday, type the command `C-c , :`.

You may be interested in notes that you created some number of days ago (let's take 50 days as an example). Then the command `C-u 50 C-c , .` comes to the rescue.

Once you invoke the list of notes for a particular date, you can navigate back and forth using special keys.

Use the brackets `(`, `{`, `[` to move backward by day, month, and year, respectively. This allows you to effortlessly revisit writing from a year or two ago. Use the corresponding closing brackets to move forward in time.

Speaking of wandering around—in Howm, you can also trigger a “random walk” through links by pressing the `w` key from the main menu. We'll come back to this in the chapter *Come-from Links (keywords)*.

³The same applies to the tasks in Howm because they use time-stamps. If you retrieve today's notes, you will notice that the list also includes old records with plans for this day.

Chapter 8

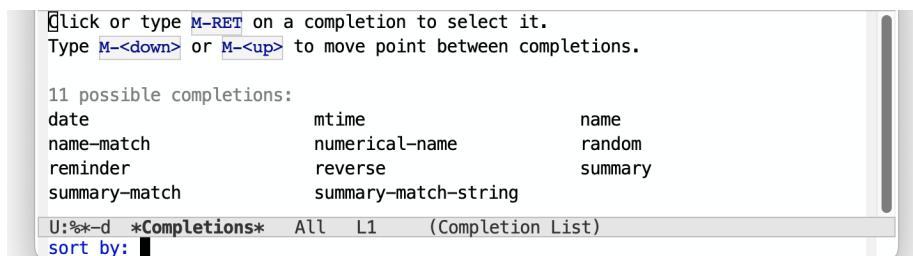
Sorting Notes in Lists

Thus far, we've discussed lists of notes sorted by either last modified time or creation date.

If you want to rearrange the items, press the `S` while in the list. Howm will prompt you to choose the criteria for sorting your notes.



Since you don't know the options yet, just press `Tab` to see them all.



Type the first letter of a sorting criterion and press `Tab`. Emacs will try to autocomplete it. If needed, type a few more letters and press `Tab` again. Once the option you want is displayed, just press `RET`.

Let's see how these options can help you navigate through your collection of notes.

date

The function lets you sort notes by their creation date, with the most recent at the top. To see the list in reverse order (from oldest to newest), press `R`. To switch back, press `R` again. The `R` command inverts any list, putting the last element first. If you want an immediate reverse order of edits, creation dates, or alphabet, type `C-u S` and then select the desired option.

reverse

This function is identical to the previous one—it flips the list upside down. But it's quicker and easier to just press the `R` key.

summary

To the right of the numeric file name in the list is the heading—if you’ve added one. The “summary” command lets you sort notes alphabetically by these headings. Titles, as you might guess, can carry extra layers of metadata. By adding certain prefixes to a note’s name, you can mark its status or build sequences of ideas.

mtime

If, after all these rotations, you want to return to the list based on the edit date, select the `mtime` (modification time) command. Alternatively, you can simply restart the list.

random

Tired of order? Shuffle your notes using the `random` command. You might come across a forgotten but interesting note.

name-match

This command lets you arrange your notes in any order you like. Howm will prompt you to enter the numeric name of the file you want to move to the top of the list. There’s no need to type the full name—usually, the last four digits will suffice.

summary-match

This option lets you manually reorder notes using their titles. In the prompt, enter the title or part of it, and Howm will move that note to the top. You can also group together all notes that share a specific keyword in their titles.

summary-match-string

With this option, you can use regular expressions to create more advanced filters.¹

name

This option places all files with non-numeric names at the end of the list, in alphabetical order. We’ll explore this type of note further in the *Descriptive Names* chapter.

reminder

This command lets you work with task lists, if you use them in Howm. We’ll revisit this option later in the *Making and Listing Plans* chapter.

numerical-name

A command that lets you sort your correspondence if you decide to integrate a knowledge base with an email client (see the *Advanced Settings*, Tip #12, p. 64).

¹The usage of Regex is a special topic that is beyond the scope of this brief tutorial.

Chapter 9

Linking Notes in Howm

A knowledge system is more than just a collection of notes; it is a network of references between them.

As mentioned earlier, links in Howm behave in an unusual way. Pressing **RET** on a link doesn't take you directly to another note—instead, it initiates a search. The screen then splits into two parts: the upper window displays the search results, while the lower window previews the note currently under the cursor. This intermediate step allows you to examine other links to the target note.

To create a link in Howm, you need to use one of two special symbols:

Three greater-than signs: **>>>**

Or three less-than signs: **<<<**

Anything on the line to the right of these spruce-like symbols will be interpreted as a link/search trigger. Therefore, you need to start a new line to continue regular text after the link (It's also possible to make links appear inline with text. Refer to the chapter on *Advanced Settings*, Tip #23, p. 66).

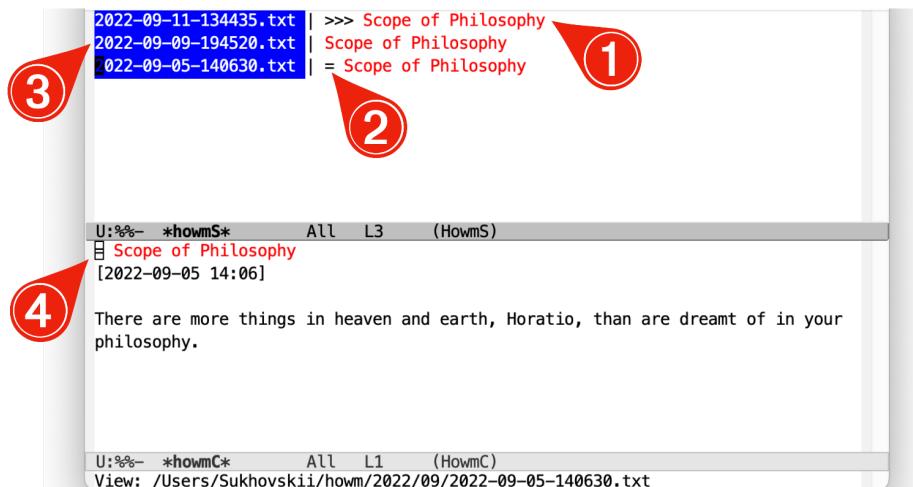
Because these two types of links function differently, it's better to divide the material into two chapters. This chapter will cover the first type.

One way to interlink notes is by using titles—just place the title of the target note after the **>>>** sign in the referencing text.

Here's how to do it quickly:

1. While in the target note, use **C-c , K** to copy the title.
2. Visit the referencing note, type **>>>** sign, followed by a space (for better readability).
3. Then use **C-y** to paste the title.

Assume you've created a link: `>>> Scope of Philosophy` [RET] on the link, and you'll see something like this:



- 1** The link you just created.
- 2** The target note (recognizable by an equals sign) might appear in the middle or at the end of the list, since the notes are ordered by last edit.
- 3** Any note that contains the phrase “Scope of Philosophy” will be included as well.
- 4** A preview of the note.

This list works just like any other in Howm, so all the operations we discussed in the last two chapters are available here as well.

The advantage of linking to the title of a note or any other word combination in the text is that you see unrecorded but potentially important relations. The downside is semantic noise if there are too many items in the search result.

Another way to create a link is to use a numeric file name.

The procedure is very similar. Use the command `C-u C-c , K` to copy the absolute path to the file. Then paste it into a favorite note with `C-y`, prefacing with a `>>>` sign.

The link will appear like this:

```
>>> ~/howm/2022/09/2022-09-09-194520.txt
```

Following the link will display a slightly different list of notes.



5 A note referenced by other entries will always be at the top. This is handy because you can follow the link by simply pressing **[RET]** twice.

6 Notes that contain the link.

Such a link is more reliable. For example, in the previous case, if you change the title, you will encounter broken links. A numeric file name serves as its unique and unchangeable ID, independent of the content.

Another argument in favour of such links is that they are the ones used in Howm when you create a chain of notes.

Of course, just the dates in the list are not very informative. To display the titles, press **[T]** and to switch back press it again.

The absolute path to the file allows you to use Org mode's hyperlink system. You can refer to your notes from .org documents outside the Howm directory by using the **file://** prefix.

```
file://~/howm/2023/09/2022-09-05-140630.txt
```

Keep in mind that using a numeric filename will show a list of all notes containing that exact link. However, Howm will not include links that refer to the note's title—and vice versa. So, it's better to stay consistent with your linking approach.

By using a numeric file name, you can organize backlinks where they are needed. To do this, simply place a link to the note itself.

1. Copy the file name with the command **[C-u C-c , K]**.
2. In the same note, type **>>>** and press **[C-y]**.
3. Clicking on the link will open a list of notes that contain the numeric name of the current file (refer to it).

You can also quickly check for backlinks on the current note, making no changes to it. It's even simpler to do. You just need sequential commands:

```
C-u C-c , K C-c , s C-y RET
```

If you want to link a file outside the Howm directory, you can use the previously mentioned prefix: **file://**

Simply specify the full path to the file:

```
file://~/Folder/Somethig.org
```

However, it is more convenient to link files using Howm's syntax:

```
>>> ~/Folder/Something.org
```

In the first case, clicking the link takes you straight to the target document. In the second, Howm opens a two-window frame with a preview of the file and a list of notes linked to it. This allows you to work on any document Emacs can open—without leaving Howm.

Howm also recognizes URLs copied from your browser's address bar. Emacs will open them in your default browser.

What if you want to reference a Howm note from another Emacs document? For example, I don't keep book summaries inside my Howm knowledge base, but I often need to mention some ideas from my notes in the summary. Ideally, everything should form a unified hypertext system. There is a way to create external links to Howm notes using Org mode—arguably one of the best tools for working with long texts.

Here's how you can proceed:

1. Open the note you want to reference.
2. Open another Emacs frame `C-x 5 2`.
3. Open the desired Org file in a new frame.
4. Go back to the note `C-x 5 o`.
5. Place the cursor on an empty line and click `C-c 1`.
6. Go back to the Org document.
7. Insert a link `C-c C-l`.

I wouldn't say that the linking system in Howm is the most ergonomic, but it is quite functional. You can get used to it and adapt it to your tasks.

One limitation of Howm when working with links is immediately apparent: there is no autofill for this task. You have to find both notes you want to connect and copy the address of one to the other. For people spoiled by automation, this may seem like a terribly slow process.

In fact, once you understand the algorithm, the process will not pose any difficulties. It is most convenient to leverage the potential of lists already available in Howm.

Any note that has been modified is moved to the top of the list. Knowing this, you can easily place the required notes next to each other.

The following steps are recommended:

1. Open Note *A*, press Space and then Backspace, and save the "changes" with `C-x C-s`.
2. Locate Note *B* using the command `C-c , s`.
3. Copy either the numeric name of the file or the title.
4. Bring up a list of recent edits with `C-c , l`.
5. Note *A* will be at the top. Navigate to it and paste the link.

This life hack applies to old notes that you've decided to link. If you are already working with Note *A*, chances are high that the first point with fake changes is unnecessary.

As entries grow, there may be several links in one note, and in different paragraphs. You can move between them very quickly.

C-c , n → Next link.

C-c , p → Previous link.

In the next chapter, we'll talk about the function of <<< sign.

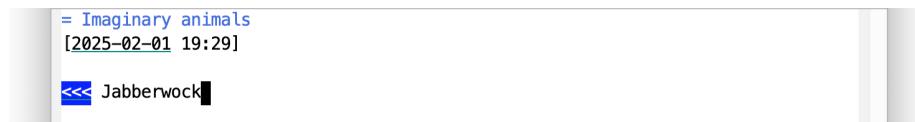
Chapter 10

Come-from Links (keywords)

Three greater-than signs >>> symbolizes a transition from the current note to another—this is a “goto” link.

Three less-than signs <<< have the opposite meaning. They mark a “come-from” link—you’re not trying to go anywhere; instead, you want all other notes containing a certain keyword to lead here.

To create this link, place the desired word or phrase after the <<< sign (anything to the right on the line will be treated as part of the link). The appearance of the sign will change.



In order to activate a link you need to save the file **C-x C-s**.

Suppose there’s another note containing the word *Jabberwock*. When you open it, you’ll see the word is underlined and has become a link.



Place the cursor on the link and press **RET**. You will be forwarded to the standard two-window screen with the list on top.



Since the note with the “come-from” link is always listed first, you just need to press **RET** again to jump to it.

Keep in mind that links are case-sensitive—the word *jabberwock* written in lowercase in other notes won’t be recognized as a link or included in the list. So either stay consistent or configure Howm to ignore case differences (see the *Advanced Settings*, Tip #24, p. 66).

Typing keywords in a note doesn’t dynamically create links—you need to save the changes first. Alternatively, you can activate the link in an unsaved file using the **C-c , r** command or by pressing **r** in the Howm menu. However, when you follow such a link, the note won’t appear in the list—it isn’t in the system yet.

“Come-from” links are useful for creating various types of indexes—by subject, topic, person, or source. A particularly helpful feature is the ability to assign an alias to a keyword—a string containing synonyms or related concepts:

```
<<< heuristics <<< serendipity <<< intuition
```

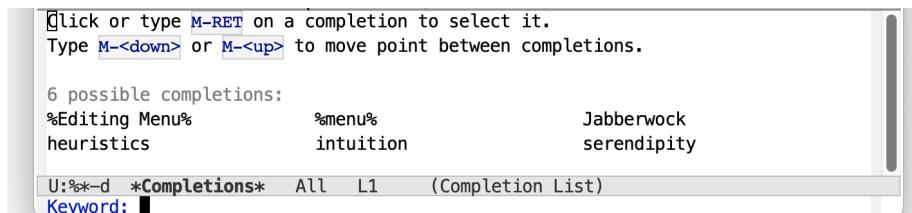
Once you have created a chain like this, all the specified terms in other notes will lead to this particular text. The search list will combine the results for all three words.

“Come-from” links can also serve a similar purpose to tags.

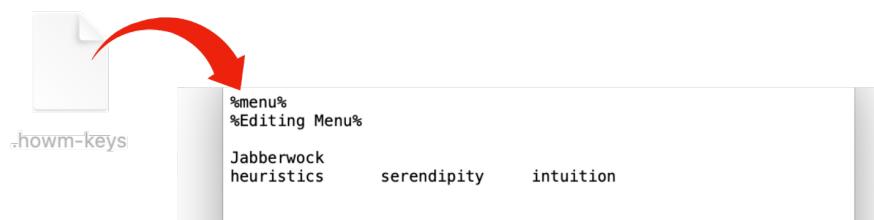
You may want to distinguish tags from ordinary words to retrieve only specific notes. One way to do this is by appending a special symbol to the word. For example, the link **<<< :intuition:** will ignore casual mentions of the word “intuition” and list only the notes explicitly marked with the tag.

Of course, remembering all tags can be cumbersome, especially as their number grows. Howm lets you query available options by pressing **i** in the main menu, or **C-c , i** from anywhere else.

The command opens a minibuffer prompt. Start typing the tag name, then press **Tab** to autocomplete. Hit **RET** to insert it into the current note. To view the full list of keywords, simply press **Tab** without typing anything.



Whenever you add a new “come-from” link to your notes, Howm stores it in a special file called `.howm-keys`



The file is located in your home directory. It is hidden, and without special command you won't see it in a typical file manager.

You can open this file in Emacs using the command sequence: **C-x C-f .howm-k Tab RET**. However, a visit is necessary only if you've changed some links.

Deleting a “come-from” link in the target note won't disable associated links elsewhere, since Howm relies on the data in the `.howm-keys` file. To deactivate the link, you'll need to remove or edit the corresponding keyword in that file as well.

Although you can do it manually, Howm has a special command for this. Use **M-x howm-keyword-remove RET** and type the word to delete (say, *Jabberwock*). Howm will display a prompt in the minibuffer:

```
Safe to back up "/Users/Your-name/howm/.howm-keys" yourself before
truncating. Truncate now? (yes or no)
```

Type **yes** and press **RET** if you're confident in your decision. Howm will then show another prompt:

```
Removed "Jabberwock"; but you may need to edit existing notes. Search
for it? (y or n)
```

Press **y**, and Howm will search for the corresponding “come-from” link in your wiki. If there is one, delete it, and you're all set.

There is also a subtype of <<< links that uses double square brackets—similar to wiki markup. Here, they behave in a specific way.

Clicking **RET** on a word or phrase in double square brackets creates a new note; the bracketed text becomes the title, preceded by <<< sign.



You might find this function useful in your workflow. However, there are a few caveats.

1. You can't link to a note's regular title using square brackets.
2. Since titles here combined with “come-from” links, Howm appends them to `.howm-keys` file. Over time, this can clog the keyword system with unnecessary information.
3. This function is disabled in Howm's Markdown and Org-mode versions because it conflicts with their syntax.

Finally, you can also use “come-from” links for a bit of fun, or even a moment of serendipity. There's a special Howm function—“Random Walk”—which can be launched from the menu by pressing the **w** key, or from any note using the command **C-c , w**. This starts the process of following “come-from” links randomly. You'll be jumping from note to note. The cycle is endless, but you can stop it at any time by pressing **C-g**. Naturally, this feature is only useful if you have enough notes with “come-from” links. Otherwise, it won't function as intended.

Chapter 11

Search Functions

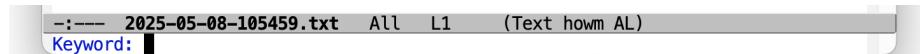
One day, your list of notes will get too long to find anything just by scrolling. That's when you'll need the search function.

There are two ways to search through notes in Howm.

Keyword search—plus usual search

To initiate this type of search from Howm's menu or from any list, just press **s** or use the **C-c , s** command from elsewhere.

A special minibuffer dialog will open then.



You can search through your notes using keywords saved in Howm's `.howm-keys` file. Type the first few letters of the desired term, press **Tab** to autocomplete, and **RET** to confirm. Alternatively, press **Tab** with an empty prompt to display the full list of available options.

The search result will be the same as if you had used a “come-from” link in a note.

If there are no keywords in your query, Howm will perform a regular search instead.

Usual search—plus regex search

A second approach also allows you to search for any text fragments. You can invoke it from the main menu or a list by pressing **g**, or from elsewhere using the **C-c , g** command.

This will open the following minibuffer dialog:



Instead of keyword functionality, this search supports regular expressions.

For example, by writing: `Woozle \| Wizzle` you will get all notes that contain either of these words. This is known as a logical OR.

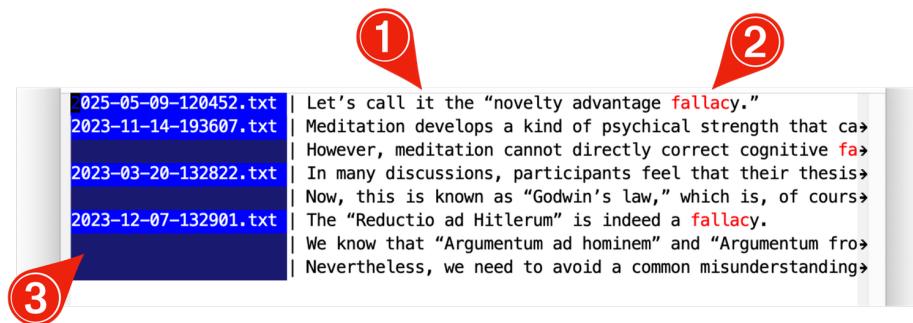
There may be more intricate commands. If you'd like to explore them, consider reviewing the regular expression (regexp) syntax. These features are optional for working in Howm.

For simple search queries, you can use the commands `C-c , s` and `C-c , g` interchangeably.

Let's now examine how Howm behaves during the search process.

While you're typing in the prompt, Howm will narrow down the list of options. Keep in mind that the search is case-sensitive.¹ Any word containing the typed letter sequence—regardless of its position—will be added to the list. The minibuffer displays the number of matches in your notes collection next to your search query, shown as `[n hits]`. When you're done, press `RET` to confirm.

The result will be the typical dual-window interface we've already encountered in Howm, although the list will differ.



- ① A note snippet with the word you're searching for.
- ② The target word is marked in red.
- ③ If a word appears on multiple lines within a note, each occurrence will be listed separately. Repeated mentions are shown as empty dark blue cells in the left column. You can fold or unfold these lines by pressing the `u` key.

Sometimes it's easier to browse the search results by note titles rather than text snippets. You can toggle between these two views by pressing the `T` key.

A single-word search—depending on the size of your note collection—can produce a very long list. It's often better to narrow it down. For example, you might be interested only in notes that mention both philosophy and Aristotle. This is known as a logical AND.

In Howm, such a query must be performed sequentially. After the initial search, press `G` and enter another word to narrow down the list of results.

There is an upside to this. You can conduct one round of searching and then decide how to refine your results. This process can be repeated several times.

The commands covered in the *Sorting Notes in Lists* chapter are applicable here as well.

When working on an extensive project, you're likely to repeat the same queries frequently. Howm keeps a search history to make this process easier. You can access it by pressing `h` in the main menu or by using the `C-c , h` command from anywhere else.

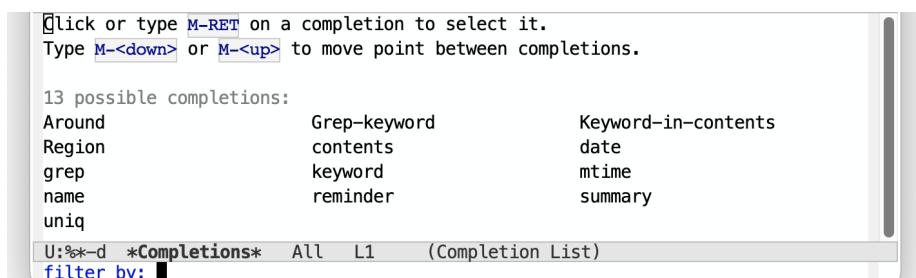
¹More specifically, lowercase searches are case-insensitive, while the use of uppercase letters makes the search case-sensitive.

Chapter 12

Filter Functions

An abundance of information can be overwhelming—unless you know how to sift out the irrelevant. Howm lets you specify search criteria or request particular collections of notes.

To start the filter function, press the `f` key in any list. This opens a prompt in the minibuffer. To view all available options, press `Tab`.



Alternatively, if you know the filter name, type its first few letters, press `Tab` to autocomplete, and `RET` to confirm.

Let's now consider the available options in order of importance.

name

This filter applies to the numeric file names shown in the left blue column. For example, specifying `2023-03` will list all notes created in March 2023. To find notes from a specific day, enter the full date in the format `YYYY-MM-DD`.

date

This filter is similar to the previous one, but it allows you to specify a range—for example, to search for notes from December 2022 to February 2023.

mtime

Here you can also specify a time range, but the output will show a list of notes you've worked on—whether created or modified—during that period.

summary

This filter applies to the right-hand column. It lets you narrow the list of note titles by specifying a particular word. You can also refine search results by adding terms that must appear within the same line of the note snippets (see p. 31).

keyword

This filter also applies to titles and snippets in the right-hand column, using keywords to narrow the list.

grep

This command lets you search within a given collection of notes and returns a shorter list based on your query. The result will be the same as if you had used the `G` key.

Grep-keyword

This option narrows down the list of notes by searching for keywords within your entries.

contents

This command filters the list by content search without altering the display of the remaining notes. To invoke it, simply press `F` from any list.

Keyword-in-contents

This option is similar to the previous one, but filters based on keyword search. Press the `K` key in any list to call it.

reminder

This option lets you filter tasks in a list based on specific criteria (see the *Making and Listing Plans* chapter or details).

Region

In Emacs, you can select text or list items by pressing `C-SPC` and then moving the cursor. Highlight the notes you want, then press `f` and select the Region option. Howm will remove everything else from the list.

Around

A filter that allows you to take any note (by moving the cursor over it) and query a list of other notes you have worked with before and after the specified one. This is useful when you need to recall context.

You can request a specific number of notes to be shown in context. For example, to display 15 notes before and after the active one, press `C-u 15`, then the `f` key, and choose *Around*.

uniq

A command that duplicates a function bound to the `u` key.

**

You can narrow down the list of notes not only with a logical AND, but also with a logical NOT. For example, you might want to exclude notes about Hegel from a search on philosophy. Use the command `C-u f`. Howm will prompt you to specify the criteria for elimination.

You have a choice of filters:

name

Exclude notes by numeric file name. You can remove a single note, or filter out an entire month or year.

date

Similar to the previous one, but it lets you specify a date range to exclude.

mtime

Filtering out notes based on their last edited date. For example, you want to remove from the list notes on philosophy that you worked with a year ago.

summary and keyword

Exclude notes using words and keywords found in text snippets or titles.

grep and contents

Cross out notes based on the occurrence of specific words in their content.

Grep-keyword and Keyword-in-contents

These work like the previous two filters but exclude notes based on keywords.

The **Around**, **Region**, **reminder**, and **uniq** functions are useless in this case, so don't pay attention to them.

All these clever tricks for sorting and filtering help you quickly navigate your collection of notes and create a workspace with only the texts you need.

Chapter 13

Deleting Notes

Rarely, you might need to delete a note; for example, if entry *A* exactly duplicates entry *B*.

Howm doesn't offer a single-click delete option. Perhaps that's for the best. Amputating texts should be a deliberate act.

First, check whether the note you want to delete is linked to other entries. Careless deletion may result in broken links in the system.

The easiest way to check is as follows:

Call these sequential commands: `C-u C-c , K C-c , s C-y RET`. This will search your collection for any mentions of the note. If none are found, only the note itself should appear in the list.

Using hyperlinks on headings instead of numeric names makes the procedure almost identical. Just start with the `C-c , K` command.

Once you're confident that removing the note won't harm your knowledge base, you can proceed with the surgical operation.

To do this, switch to Dired mode. If the cursor is positioned on a targeted note, press the `X` key. The screen will change, but the cursor will still point to the same note.



In the new window, you have access to all features of Dired mode.

Press `v` to look in the file if reassurance is needed. Press `q` to return to the Dired mode.

To delete a note, press the `D` key. Emacs will prompt you to confirm the irreversible action. Deleted notes are gone permanently—they aren't moved to the trash—so be careful!

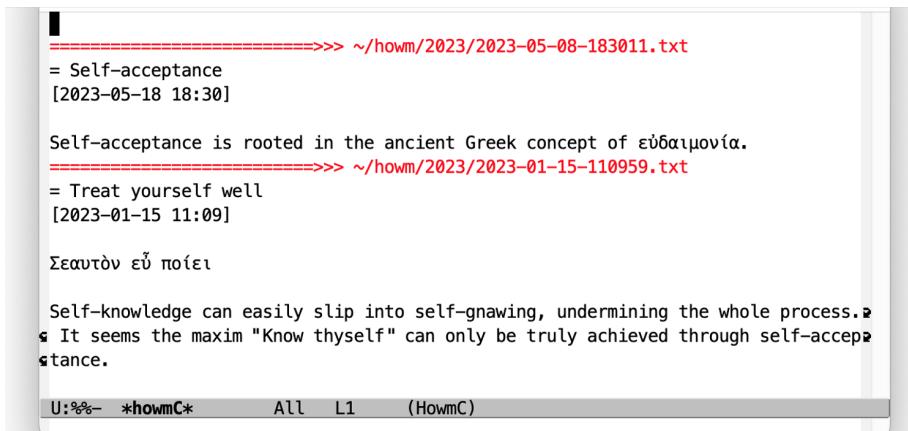
Pressing `X` in any list activates Dired mode, allowing you to perform various file operations. However, these features aren't necessary for using Howm.

Chapter 14

Stitching Notes

Sometimes it's helpful to overcome the discreteness of your texts and treat them as a continuous document. This aligns with Howm's motto: "Write fragmentarily and read collectively."

To view notes as a single, concatenated scroll, press `0` or `@` in any list. Entries will be temporarily stitched together, with markers indicating the seams between them.



You can jump from note to note using the `Tab` and `M-Tab` keys. To exit the scroll-view press `0` or `@` keys again. This feature is useful for quickly searching and reviewing notes.

Kazuyuki Hiraoka suggests the following technique:

1. Run an initial search using the `C-c , s` command.
2. Combine the results into a single scroll view.
3. Use the `C-s` command to search within the merged text.

To arrange notes in a specific order—say, for drafting an article—use this trick:

1. Add a unique label to the entries you want to include in the article—for example `art23`.
2. Use the search function to compile a list of the selected notes.
3. Arrange the list in the desired order using the sorting function:
 - (a) Press the `S` to open the sorting dialog.
 - (b) Use the `name-match` option to move the first note to the top, then repeat for each one.
 - (c) Press `R` to reverse the list.
4. Create a single scroll by pressing `0`.
5. Select the text with `C-x h`, then copy it with `M-w`.

Now you can switch to your editor of choice, paste the text, and polish it there.

You can keep labels in your notes as markers of used entries—this can help you avoid redundancy in future articles.

A concatenated scroll lets you check the total word and character count in your notes collection.

1. Call up the list of all notes with command `C-c , a`.
2. Press the `0` key.
3. Call `M-x count-words` and press `RET`.

The minibuffer will display a report. The figure isn't entirely accurate because special symbols are included in the count. However, the discrepancy is typically minor.

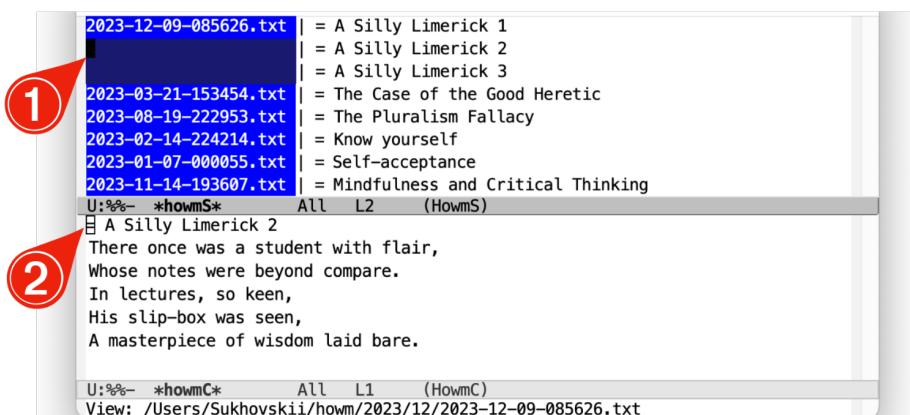
Chapter 15

Handling Long Notes

Sometimes you may need to write a long text rather than a short entry. Even then, it's a good idea to divide the material into sections. You can use additional titles to organize extensive notes.

```
= Title 2  
Some text  
= Title 3  
Some text
```

Howm's lists display multi-titled notes differently from simple ones.



- ① An empty dark blue cell in the left column indicates that the titles belong to the same note whose numeric filename is listed above.
- ② When the cursor is on a list, the preview window shows each segment as a separate note—content under other titles remains hidden until you navigate to that entry.

The *Recent* section in Howm's menu displays each titled fragment as a separate note, all sharing the same numeric filename.¹

¹Keep in mind that modifying this kind of note causes all its titles to appear in the Recent list, which can lead to clutter.

To quickly navigate between sections of a long note, use the following commands:

| | |
|----------------|--------------------------|
| C-c , N | → Next title. |
| C-c , P | → Previous title. |
| C-c , H | → Go to the first title. |
| C-c , L | → Go to the last title. |

To create a new section in a note, start a new line with an equals sign and add a title if needed.

You can also use the **C-c , C** command—this appends a title template to the end of the note, regardless of the cursor's current position.

Sometimes it's useful to focus on a single section of a multi-titled note while hiding the others. To do this, place the cursor on the desired title and run **M-x howm-narrow-to-memo RET**.

Since this feature is disabled by default, Howm will open a window with explanation and prompt you to choose how to proceed.

Once you press **y**, Howm will add the following line to your `.emacs` file:

```
(put 'howm-narrow-to-memo 'disabled nil)
```

Your note will be narrowed to the selected section.

To switch back to the full text, press **C-x n w**.

You can also toggle between hiding and showing other sections by running **M-x howm-toggle-narrow RET**.

Although there's no dedicated keybinding for the narrowing command, it's worth creating one if you plan to use this feature regularly (see the *Advanced Settings*, Tip #14, p. 64).

Sometimes you may need to search for a specific phrase within a long note. The simplest way is to use the standard **C-s** command.

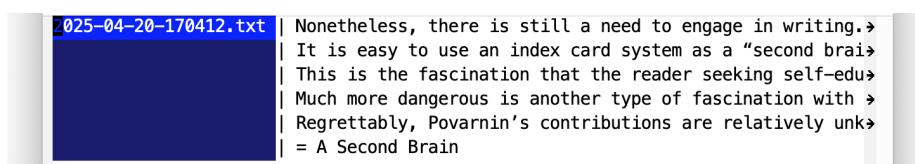
What if you want to see a list of all matches? Howm includes a special feature for that.

While in a note, use the **C-c , o** command and enter your query. You'll see the standard Howm interface—with a list in the top window and a preview below. This time, however, you'll be navigating within a single note.

Alternatively, you can press **o** in Howm's menu to search in the last opened file.

Another helpful technique for working with long notes is the use of marks. To create one, press **C-SPC C-SPC** with the cursor positioned at the desired location.

Afterward, you can use the **C-c , x** command to view the list of marks in the current note, or press **x** in the main menu to see the marks from the last opened document.



Howm remembers all marks made during a session, but you need to open a specific note to see its associated list.

Finally, you can use unique letter combinations as a kind of bookmark. For instance, place a label like qq in one note, and create a link <<< qq in another. Clicking the link will open the document at the corresponding location.²

²The Q-trick was proposed by Merlin Mann in 2010. He used it when working with notes in Simplenote and Notational Velocity.

Chapter 16

Working with Buffers

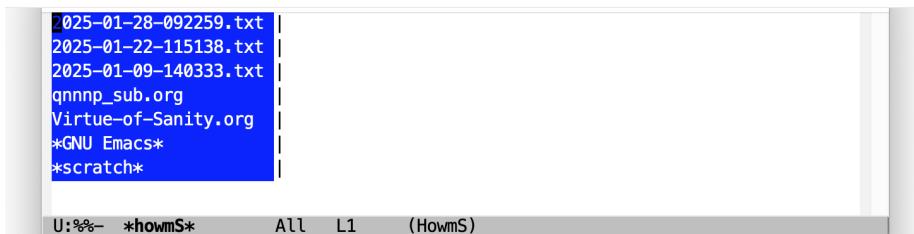
During your work session, Emacs keeps every opened document in a special list of buffers. You can access it using the standard command `C-x C-b`.

This feature is useful for many tasks, so it's worth exploring the Emacs documentation to see how it can enhance your workflow.

Howm also benefits from Emacs's list of buffers. For example, you can use the `C-c , SPC` command to switch from a Howm note to the last non-Howm document.

To switch back, use the `C-x ←` command.

You can work with buffers directly from Howm. Press `b` in the main menu or in any Howm list, or use the `C-c , b` command elsewhere. This opens the familiar dual-window layout, with a preview at the bottom—except the top window now displays a list of buffers.



Since notes are listed in the order they were visited, this feature can serve as “breadcrumbs”—a way to track your path through the knowledge base.

Numeric filenames in the list aren't very informative, but you can adjust Howm to display titles instead (see *Advanced Settings*, Tip #8, p. 63).

As you work on your notes, you'll accumulate many open buffers. Some are still in use, while others just get in the way. So how do you clear the clutter?

You can press `Q` while in any list or in the Howm menu, or use the `C-c , Q` command elsewhere. Once confirmed, this will close all Howm buffers except unsaved ones. You'll then be taken to the last unsaved Howm note—or, if none remain, to the most recently opened non-Howm buffer.

If you want to close only a specific buffer, use the standard Emacs command `C-x k` instead.

Chapter 17

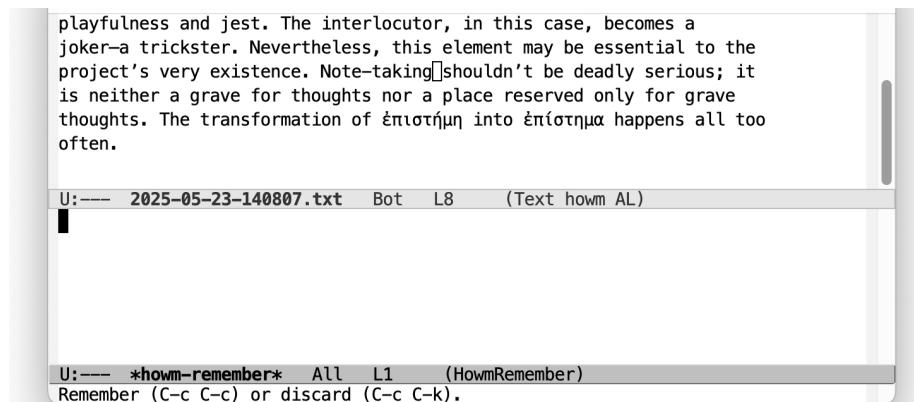
Quick Notes

Sometimes, while working on one note, an idea for another pops up. You don't want to break your flow—but you also don't want to lose the insight.

Howm has a quick note feature designed for exactly this situation. It lets you jot down a prompt without leaving your current text. You'll get a new, empty note with a title and timestamp—just enough to capture something you want to remember.¹

To call up the quick note, press `e` in the main menu or in any list. Otherwise, use the `C-c , e` command.

Howm will open a completely blank window below the current one, with command hints in the minibuffer.



Quick notes are meant for very short entries—what you type becomes the title. If you want to write a full paragraph instead, it's better to start a regular note.

The content in the window you opened isn't associated with a file yet. To create an actual note, press `C-c C-c`. Howm will save the text and kill (close) the buffer, returning you to the note you were working on before.

Or, if you change your mind, press `C-c C-k`. Howm will close the buffer without saving anything.

¹Calling this feature from another note also adds a backlink next to the date.

To review these memos, open the list of recent notes or use the main Howm menu.

One specific detail about quick notes: by default, a blank line is inserted between the title and the timestamp. This behavior exists for backward compatibility, as it was originally required in some situations. You can customize the quick notes template to remove the blank line (see *Advanced Settings*, Tip #15, p. 64).

Chapter 18

Duplication

Some writing tasks are routine—we often type the same text repeatedly, such as the sections needed for bio cards (e.g., years of life, achievements, etc.). So why not create a template?

In Howm, you can use the *duplicate function* for this purpose. The `C-c , D` command creates a clone of the current note. The copy is identical to the original, except for a new numeric filename.

You can perform this action from the main menu by pressing `D`. In this case, the last open buffer will be duplicated.

In notes with multiple titles, only the section of text where the cursor is located will be duplicated. This behavior actually opens up some interesting possibilities.

First, you can create a document containing a collection of templates, each separated by a title.

Second, you can easily duplicate a part of any note by placing an equal sign above and below the selected fragment.

A document containing templates can be given a special code name (such as `T_01`) to make it easily accessible via search.

If you change your mind after invoking the duplicate command, press `C-/` to undo the action and return to the original note with `C-x ←`. No traces will remain in the system.

Chapter 19

Descriptive Names

You can add any plain text file—regardless of its name—to the Howm folder. The program will handle it just fine. However, for proper list functionality, the first line of the note should follow Howm’s syntax: start with an equals sign, and optionally include a title.

If you call up a list, you’ll see the descriptive name among the standard numeric filenames.

| | |
|-----------------------|--|
| 2023-05-18-222143.txt | = The Philosophy of Laziness |
| 2023-05-20-192202.txt | = Slow Productivity |
| Writing=Thinking.txt | = Luhmann on the Necessity of Writing |
| 2023-05-20-172538.txt | = The Psychological Roots of Disorganization |
| 2023-05-20-190658.txt | = The Error of Instrumental Attribution |

You can also create a file with a descriptive name in Howm. Use the `C-c , I` command; the minibuffer will prompt you to enter the filename and extension.

```
U:--- 2025-05-27-140932.txt  All  L1  (Text howm AL)
Memo file: ~/howm/
```

Howm then creates a template for the new note.

Of course, consistency is key in knowledge management. You should have a very good reason to deviate from Howm’s standard naming system.

It’s best to store files with descriptive names in a designated folder.¹

This folder must be created in advance. You can use a standard Emacs command to do this:

1. `M-x make-directory RET`
2. Type `~/home/ingenious-thoughts RET`

After creating the directory, you can specify its path when prompted:

```
~/home/ingenious-thoughts/
```

¹Howm treats all files placed in its directory as part of the wiki, regardless of which subdirectory they’re in. They will appear together in all listings.

Actually, you don't even need to place these files in the Howm folder—you can simply specify a path to them (see *Advanced settings*, Tip #11, p. 64).

You can also navigate to any text file outside the Howm directory. To do this, press **M** while in a list, or use the **C-c , M** command elsewhere.

To activate Howm syntax in that file, type: **M-x howm-mode**

Chapter 20

Task Types in Howm

If a personal wiki is your everyday tool, it makes sense to manage your intellectual work—or even all your tasks—right there. Howm offers a flexible planning system designed for the naturally disorganized. The goal is to strike a balance between the yin of chaos and the yang of control.

Lists are the core tool for working with information in Howm, and they're naturally well-suited for planning. But any to-do list tends to grow to a size incompatible with life expectancy. The key is to ensure that only the relevant plans are visible at any given time.

The principle here is the same as with organizing notes: spending too much time on categorization tends to be counterproductive. Hiraoka writes that he took the drastic step of abolishing “importance” altogether, relying instead on the concept of “seasonality.” The metaphor underscores that a sense of timeliness is a key factor in decision-making.

In Howm, you have several tools at your disposal: lists with schedules, deadlines, to-dos, reminders, and deferrals.

While all tasks and events have assigned dates, the meaning of those dates varies.

The *to-do* date marks the beginning of a timeframe for a task.

The *deadline* date marks the end of the task completion period.

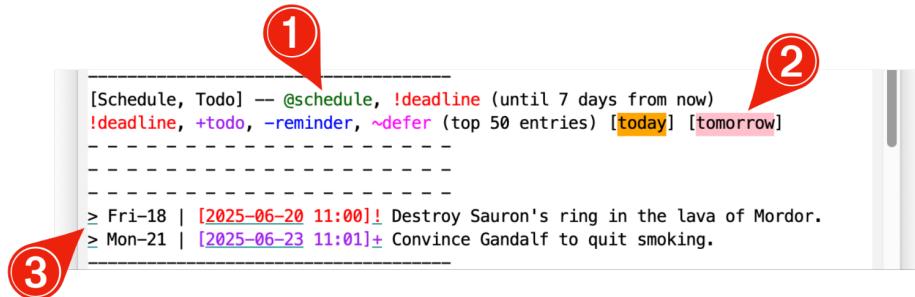
The *schedule* specifies the exact day for an event or task.

There are two additional, clever ways to manage your plans.

Reminders allow you to keep some ideas in view temporarily. Then—based on their date—they gradually disappear from the visible list.

If you want specific items to periodically catch your attention, use the *defer* function—the date determines when the cycle begins.

Howm has a special syntax that cleverly reflects the behavior of each function. Additionally, color cues help you distinguish between types at a glance. All this information is available in the main menu for quick reference.



- ➊ Appending @ after a bracketed date automatically renders its text green. A plus sign changes the text to purple, and so on.
- ➋ Tomorrow's date is always highlighted in pink, and today's in orange.
- ➌ You can get a quick overview of your plans from the main menu. Howm generates a list of to-dos, reminders, and similar items. This data is extracted from all your notes, no matter where you wrote it down—nothing gets lost. Naturally, the task section will be empty when you first start using Howm.

If you use the `M->` command to scroll to the bottom of the menu, you'll find mnemonic cues for Howm's syntax.

How to remember:

- Schedule at(@) the date
- Reminder sinks down(-).
- Todo floats up(+).
- Deadline needs attention!
- Defer waves(~) up and down.
- Done is the end(..).

All tasks in Howm follow this format:

`[YYYY-MM-DD HH:mm] + task symbol + space + task description`

The time is optional.

But what exactly do these “sinks” and “floats” in the syntax mean?

Hiraoka's imagery portrays the list as water, with tasks sinking to the bottom or surfacing to the top.

Therefore, task lists are not static but fluid. This makes it easy to identify what's currently relevant: unnecessary items remain deep, while essential ones float near the surface.

The “buoyancy” of each item is dynamically calculated based on its properties and the current date—a higher value indicates higher priority.

There are three distinct priority groups. Items with a priority close to 77,777 appear at the top of the list, while those near -77,777 remain at the very bottom. Items with a priority around 0 are situated in the middle.

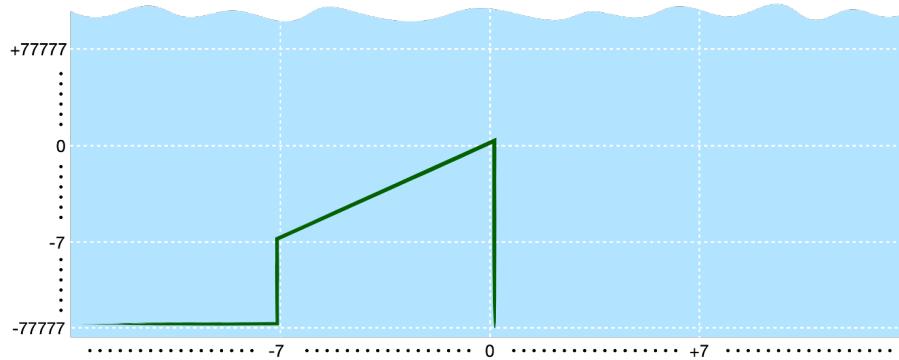
All upcoming tasks receive a priority around -77,777. The order among these sunken items is meaningless—it simply keeps them out of sight, unless your list is very short.

As the designated day approaches, items show different behavior.

Schedule

The scheduled event is visible on the list seven days ahead of the set time. Overdue events are submerged the following day—which makes sense, since you don't need a reminder about a party that happened yesterday.

The following diagram can represent this:

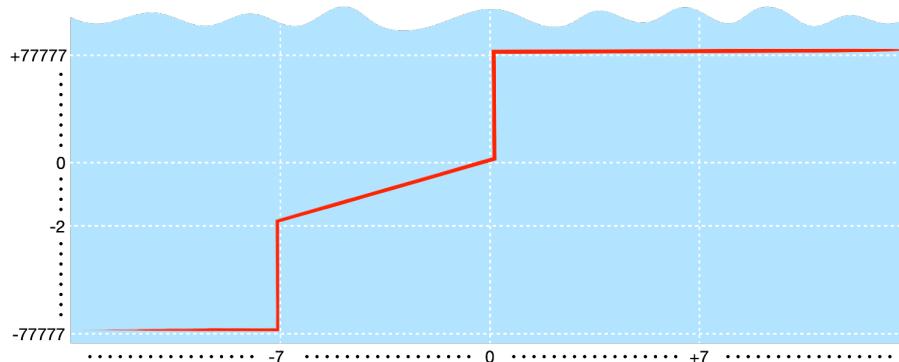


[2024-04-08 19:15]@ Come to Tom Bombadil's party.

Deadline

If there are more than seven days before the target date, the task stays submerged at the bottom of the list. Then, it moves to the middle to ensure visibility. On the due date, it floats to the top—and stays there even after the deadline has passed.

The diagram below can be used to depict this idea:



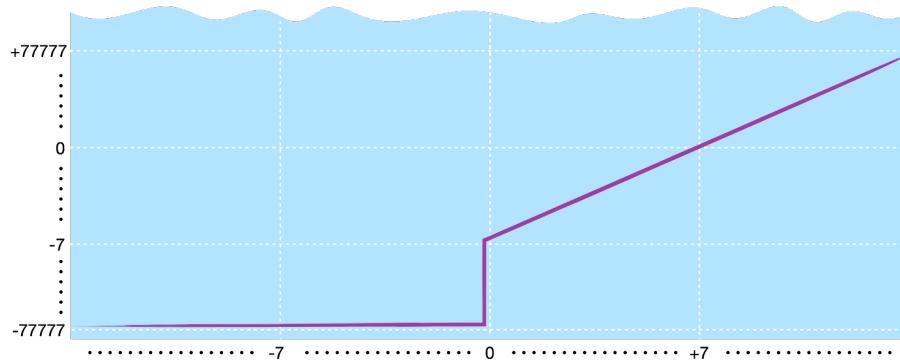
[2024-09-22]! Buy a present for Bilbo's birthday.

When the deadline is reached or surpassed, the exclamation point will receive an additional red highlight: [2024-09-24]!!

To-do

The to-do stays at the bottom of the list until its appointed date. Once that date arrives, it moves closer to the middle for better visibility. Over the next seven days, the task gradually floats upward—first to the middle, then higher and higher. This means that the items at the very top of the list are the oldest to-dos.

This diagram illustrates this behaviour:



[2024-09-18]+ Ask Galadriel for the lembas recipe.

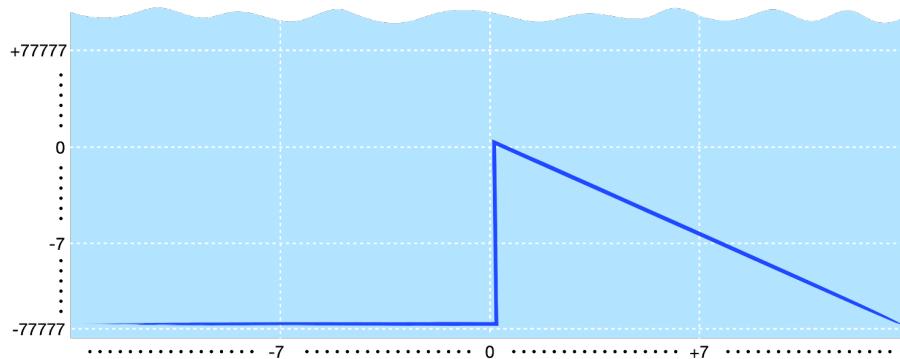
Reminder

Reminders are not intended for planning ahead. As Hiraoka writes:

I never actually set future dates for the “reminder.” I just use it with today’s date for random things I hope to do/study/read... and let most of them sink, since obviously the sum of them exceeds my capacity.

A reminder is meant to sink if it doesn’t win your attention. On its due date, it appears in the middle zone of the list and then sinks a little each day, eventually submerging to the bottom. The descent rate is one unit per day.

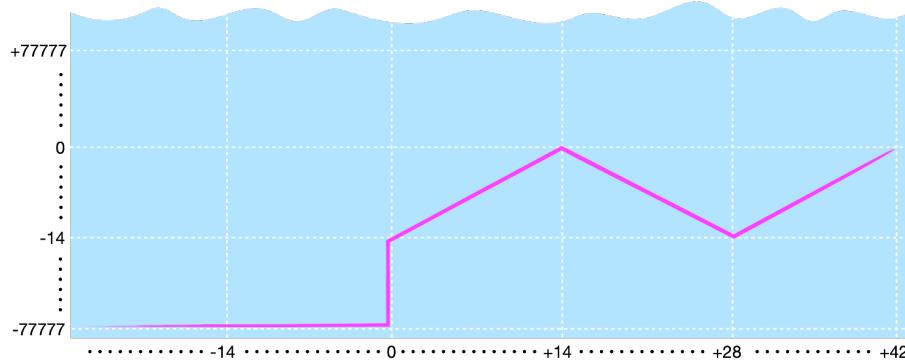
This diagram can be used to illustrate the idea:



[2024-09-13]- Try the new ale at the Prancing Pony.

Defer

The Defer function moves an item to 14 points below 0 on the appointed date. From there, it takes 15 days for the item to float into the middle zone. After reaching it, the item gradually sinks back to -14. The task repeats this floating-and-sinking process over a 30-day cycle, oscillating within the middle zone.



[2024-09-16]~ Stop fantasizing that I'm Frodo.

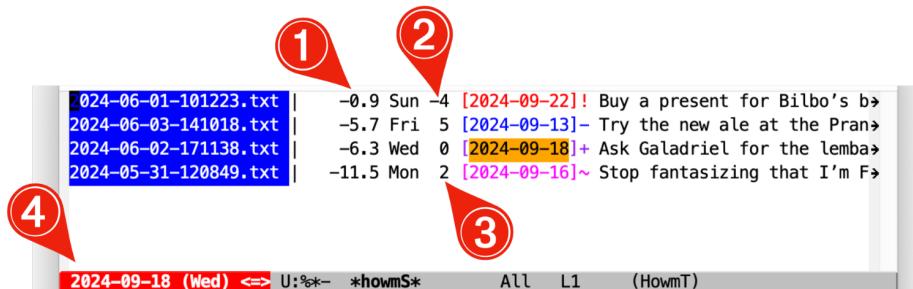
It's best to avoid using the Defer function for repetitive tasks (there's another option we'll discuss later). It acts more like a recurring reminder.

**

Grasping the concept of task buoyancy can be challenging, given its abstract nature. Fortunately, Howm has a built-in simulator that lets you see these principles in action.

Use this command to start the simulation: `M-x howm-simulate-todo`

This will open a special interface in the top window.



- ① Importance ranking for tasks. To display these numbers in task lists by default, modify your `.emacs` file (see Tip #20 on p. 65).
- ② Days until the target date.
- ③ Days past the target date.
- ④ A date-changing simulator that lets you pretend to be in the past or future.

If you want to see how the task list will appear tomorrow or a few days from now, just click the `>` button as many times as needed. To jump ahead 30 days, use the command `C-u 30 >`. To go back in time, press the `<` key. Press `=` key to return to the current date.

Now you can see how shifting the date affects a task's buoyancy and its position in the list.

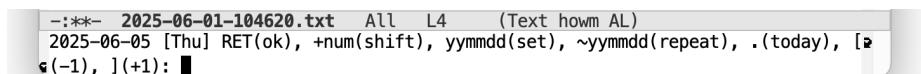
To exit simulation, press `q` key.

Chapter 21

Making and Listing Plans

The previous chapter explored the key principles behind Howm’s dynamic task list. Now, let’s look at the tools that make planning in Howm easy and efficient.

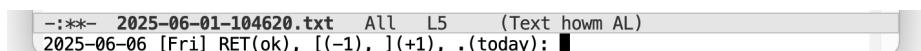
To start, you don’t need to type task dates manually. The command `C-c , d` inserts the desired date into the open note.¹ When you run this command, Howm opens a dialog in the minibuffer.



By default, Howm assumes you want today’s date. If that’s the case, simply press either the `.` key or `RET`.

While in the dialog, you can switch dates back and forth by pressing the `[` and `]` brackets, respectively.

As you do so, the dialog will display:



Set the desired date and press `RET`. You can reset your changes to today’s date by pressing the `.` key.

What if you have plans for January 5, 2028? Just enter 280105 in the first dialog and press `RET`.

In most cases, there’s no need to enter the full date—Howm understands a variety of abbreviations.

Numbers from 1 to 31 are treated as dates in the current month.² The number 803 is interpreted as August 3rd of the current year.

By typing `+14` and then pressing `RET`, you will add fourteen days to today’s date.

If you need both the date and time, use the `C-c , T` command. It inserts today’s date and time without any prompts, in the format: `[YYYY-MM-DD HH:MM]`

¹You can also insert a date into a previously opened note using Howm’s menu—just press `d`. The date will appear at the current cursor position in that note.

²If it’s June and you type 31, Howm will interpret it as 30 + 1 and assume you meant July 1st.

Additionally, you can use a clever function called action-lock. Type `{_}` and press **RET** with the cursor inside—it will insert today’s date and time. This function can be used in various ways (see *Advanced Settings*, Tip #21, p. 65).

Now that you’re a pro at setting dates, it’s time to start jotting down real tasks.

You can create a dedicated note for your plans or scatter them throughout your wiki. In a note, you can list tasks consecutively or mix them with comments—it doesn’t matter. Howm lets you navigate swiftly through any setup.

There are two ways to view plans: through the Howm menu or by calling up special lists. Let’s explore both options.

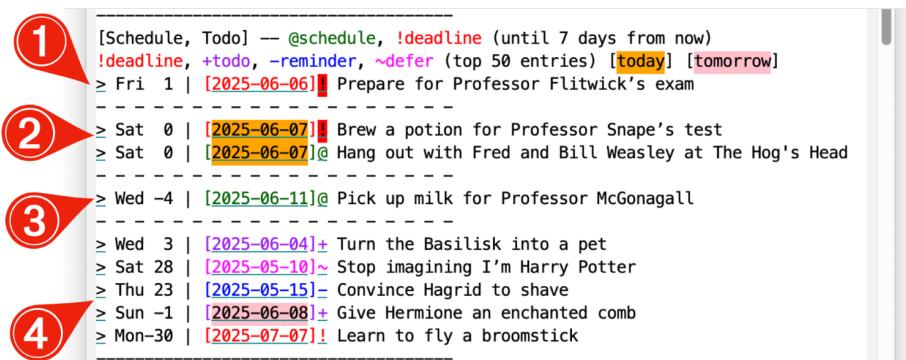
Howm’s menu

Howm’s menu offers a convenient way to stay updated on the tasks and events at hand.

The list in the main menu is not exhaustive—it has a depth of 50 items and covers a seven-day schedule. But that’s the point: to keep tasks that belong to another “season” out of sight.

Naturally, if you have only a few tasks, they’ll all be listed—regardless of their dates.

There are four segments for plans in Howm’s menu.



The screenshot shows a terminal window displaying the output of the Howm command. The output is divided into four segments, each preceded by a red circle with a number:

- 1** [Schedule, Todo] -- `@schedule, !deadline` (until 7 days from now)
`!deadline, +todo, -reminder, ~defer` (top 50 entries) `[today] [tomorrow]`
- 2** `> Fri 1 | [2025-06-06]!` Prepare for Professor Flitwick's exam
- 3** `> Sat 0 | [2025-06-07]!` Brew a potion for Professor Snape's test
`> Sat 0 | [2025-06-07]@` Hang out with Fred and Bill Weasley at The Hog's Head
- 4** `> Wed -4 | [2025-06-11]@` Pick up milk for Professor McGonagall
`> Wed 3 | [2025-06-04]+` Turn the Basilisk into a pet
`> Sat 28 | [2025-05-10]~` Stop imagining I'm Harry Potter
`> Thu 23 | [2025-05-15]-` Convince Hagrid to shave
`> Sun -1 | [2025-06-08]+` Give Hermione an enchanted comb
`> Mon-30 | [2025-07-07]!` Learn to fly a broomstick

- 1** The top segment warns you about missed deadlines.
- 2** The second segment shows today’s scheduled events and deadlines.
- 3** The third segment displays events and deadlines for the current week. Items are arranged in chronological order.
- 4** The last segment includes to-do, defer, and reminder-type tasks, along with upcoming deadlines. Items here are sorted by their buoyancy.

Dates are preceded by the weekday and a day offset—a negative number for days before the assigned task, and a positive number for days after it.

To open the corresponding note, place the cursor on the `>` sign and press **RET**.

List for tasks

You can call up a dedicated list of your scheduled events and deadlines.

To do this, simply press `y` in the main menu or in any other list, or use the `C-c , y` command from anywhere else.

You'll get Howm's standard two-window layout, with a list of notes containing your plans on top.

| | |
|------------------------------------|--|
| <code>2025-05-03-174756.txt</code> | Fri 1 <code>[2025-06-06]!</code> Prepare for Professor Flitwick's ex> |
| <code>2025-05-02-164734.txt</code> | Sat 0 <code>[2025-06-07]!</code> Brew a potion for Professor Snape's> |
| <code>2025-05-07-214758.txt</code> | Sat 0 <code>[2025-06-07]@</code> Hang out with Fred and Bill Weasley> |
| <code>2025-05-20-112714.txt</code> | Wed -4 <code>[2025-06-11]@</code> Pick up milk for Professor McGonaga> |
| <code>2025-05-07-214758.txt</code> | Mon-30 <code>[2025-07-07]!</code> Learn to fly a broomstick |

The advantage—especially when tasks are scattered across different files—is that you can quickly check the context of each task in the preview window.

Tasks are listed in calendar order, oldest first, but the cursor is positioned on today's deadlines or events.

Unlike the main menu, these lists are not limited—you have access to all your plans. For example, you can check events scheduled for the next quarter or review those from the past.

To see the list of to-do, defer, and reminder-type tasks, along with deadlines, use the `C-c , t` command or simply press `t` from the main menu or any other list.

| | |
|------------------------------------|---|
| <code>2025-05-03-174756.txt</code> | Fri 1 <code>[2025-06-06]!</code> Prepare for Professor Flitwick's ex> |
| <code>2025-05-02-164734.txt</code> | Sat 0 <code>[2025-06-07]!</code> Brew a potion for Professor Snape's> |
| <code>2025-05-20-112714.txt</code> | Wed 3 <code>[2025-06-04]+</code> Turn the Basilisk into a pet |
| <code>2025-05-03-174756.txt</code> | Sat 28 <code>[2025-05-10]~</code> Stop imagining I'm Harry Potter |
| <code>2025-05-15-082714.txt</code> | Thu 23 <code>[2025-05-15]-</code> Convince Hagrid to shave |
| <code>2025-05-07-214758.txt</code> | Sun -1 <code>[2025-06-08]+</code> Give Hermione an enchanted comb |
| | Mon-30 <code>[2025-07-07]!</code> Learn to fly a broomstick |

Here, tasks are arranged based on their buoyancy, with the cursor positioned at the top of the list. To view them in calendar order, press `S` and select the *reminder* option.

When scrolling through a long list, you can quickly return to today's agenda by pressing the `.` key.

You can integrate plans hosted in Howm with Emacs calendar packages. One package worth exploring is `emacs-calfw`.

There are a few techniques for working with long task lists. You can:

1. Move all tasks containing a certain word to the top of the list. Press `S` key, select the *summary-match* option, type the word, and press `RET`.
2. Narrow the list to tasks matching a specific word. Press `f` key, select the *summary* option, type the word, and press `RET`.
3. Filter tasks by a certain period. Press `f` and select the *reminder* option. Howm will then prompt you for a time range.

4. Search for a specific word across your wiki, then press **[S]** key and choose the *reminder* option. Press **[RET]** to bring all tasks with that word to the top of the list.

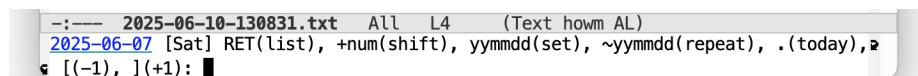
Chapter 22

Fine-Tuning Plans

A good planning system should be flexible. Everything flows, as Heraclitus said. Circumstances change, and so do we. A tool for swiftly adjusting goals is a necessity.

Of course, you can retype your tasks manually, but Howm has some neat features for editing plans.

To change the date, place the cursor inside the square brackets and press **RET**. Howm will open a dialog in the minibuffer.



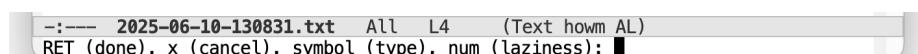
The options available here were discussed in the previous chapter.

You can reassigned the date directly from the main menu—just place the cursor on the bracketed date and press **RET**. Howm will temporarily take you to the corresponding file and open a dialog in the minibuffer. Once the change is complete, you'll automatically return to the menu.

Keep in mind that changes haven't been saved yet. To save them without leaving the menu, use the **C-x s** command.

You can also adjust a task's properties and its behavior in the list.

In the note or the main menu, move the cursor to the symbol after the square brackets and press **RET**. A prompt will appear in the minibuffer.



Let's go through the choices from right to left.

The **num (laziness)** option controls how quickly tasks surface or sink. Enter the desired number and press **RET** to adjust it.

You may want to check the reference in the main menu first. To see it, just scroll down a bit using the **C-v** command.

```
Format of schedule and todo (Please replace {} with []):
{2002-10-21}@1 schedule -- (shown in schedule part. @3 = "3 days schedule to 10@2
@-23")
{2002-10-21}+7 todo -- float up slowly from the date in 7 days
{2002-10-21}!7 deadline -- float up fast from 7 days before the date
{2002-10-21}-1 reminder -- float at the date and sink slowly one unit per 1 day
{2002-10-21}~30 defer -- float at the date and repeat sinking and floating with a
@30 days period
{2002-10-21}. done -- sink forever
(Numbers after marks are default values.)
```

Knowing the default values, you can tell that:

1. A scheduled event [2023-06-25]@4 won't sink for three extra days after its due date.
2. A to-do [2023-06-25]+14 will float up twice as slowly.
3. A deadline [2023-01-20]!14 will float up seven days earlier than usual.
4. A reminder [2023-08-15]-2 will sink twice as slowly.
5. A defer [2023-03-09]~15 will oscillate twice as often.

Next, the `symbol` (type) option lets you redefine an action—a reminder can become a deadline, a scheduled event can be changed into a reminder, and so on. Just enter a new character and press `RET`.

It's time to talk about changing the status of plans. Sooner or later, you'll have to either fulfill or abandon them.

The `x` (cancel) option allows you to calibrate your ambitions. Press the `x` key, and the task's appearance will change.

```
[1990-12-31]+ Read Moby-Dick
↓↓↓
[2023-06-28]. cancel [1900-12-31]:+ Read Moby-Dick
```

The cancel mark shows that you ended the attempt—not the task.

The final and most satisfying option in the minibuffer is `RET` (done). Press `RET`, and the task will be marked as completed.

```
[2023-06-04]+ Attain enlightenment
↓↓↓
[2023-06-28]. [2023-06-04]:+ Attain enlightenment
```

The dot shows that the case is at its end.

Each time you finish a task, Howm congratulates you in the minibuffer with: "Finished 1 task!", "Finished 2 tasks!", and so on.

Completed and canceled tasks are submerged to the bottom of the task list. The main menu doesn't show them at all.

To review them, use the commands `C-c , t` or `C-c , y`, and then press `R` to flip the list.

**

Some tasks can't be completed once and for all—life puts them back on your list. There are many ways to deal with these recurrences.

In Howm, you need to specify the number of times a task should repeat.

To create a recurring task or event, move the cursor to the bracketed date and press **RET**. The familiar minibuffer dialog will appear.

Type the date when the recurrence should stop. For example: ~250601

Then, press **RET**, and Howm will prompt you to set the repetition rate, starting from the initial due date.



RET(all) – repeat daily

num(days) – repeat every n days

w(week) – repeat every 7 days

m(month) – repeat monthly on the same date

y(year) – repeat yearly on the same date

After you enter the parameters and press **RET**, Howm will generate a column of task clones with different dates.

That way, you can reassign the date of an individual item without breaking the rest of the chain.

Chapter 23

Advanced Settings

Welcome to the land of tinkerers, where every Howm feature can be adjusted to your liking.

Personally, I think Howm works well as it is. The *Advanced Settings* chapter doesn't imply that the default setup is somehow inferior; it simply means that advanced knowledge is required.

The easiest way to access the customization section is from the main menu.



- ① Move the cursor over this button and press **[RET]**. This will open Howm's GUI settings panel.

Here, you have control over every aspect of Howm's behavior. However, it's best not to change individual parameters until you have a general understanding of how the system works.

We won't cover every possibility—just the most sensible adjustments.

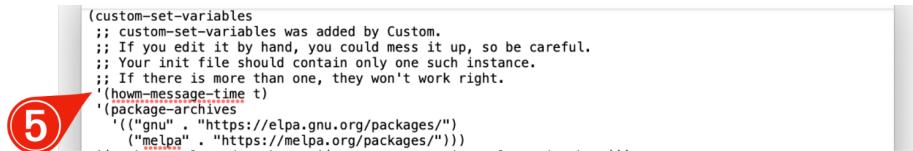
To change something in the settings, you need to follow a certain procedure. For example, suppose you want to add a load time report for searches and the main menu.

Press **[RET]** on the **Devel group**. In the new buffer, click the ► next to **Home Message Time**.



- ② You can change settings by clicking the `Toggle` button.
- ③ State will change from STANDARD to EDITED.
- ④ Click `Apply and Save` for the changes to take effect.

Your preferences will be written to the `.emacs` file, within the `custom-set-variables` section at the beginning of the file.



- ⑤ A line of Lisp code that tells Howm how to operate.

If you change your mind, go back to the settings section and click the `State` button. In the dialog, select *Erase Customization*, then click `Apply and Save`.

Alternatively, you can simply remove the corresponding lines from the `.emacs` file.

Frankly, in many cases it's much easier to edit the `.emacs` file directly. There's nothing particularly complicated about it. The file is just a collection of commands that you can usually copy and paste. (Of course, learning some Lisp is always beneficial.)

There is one caveat, though—you won't be able to undo such changes through the settings. You'll have to go back into the `.emacs` file and make the corrections manually.¹

The rest of this chapter will cover code examples that modify Howm's behavior.

¹If you make some changes through the settings panel and others directly, conflicts may arise. It's a good idea to leave some reminders for yourself to avoid confusion. You can add comments inside the `.emacs` file by starting the line with a semicolon.

Be careful when copying and pasting code snippets. L^AT_EX formatting can sneak into your `.emacs`, so always double-check the results.

Adjusting the Menu

Tip #1

When the number of notes exceeds a thousand, Howm becomes a little pensive (though not unbearably so) when loading the main menu. You can optimize the process by adding the following lines to your `.emacs` file:

```
(setq howm-menu-refresh-after-save nil)
(setq howm-menu-expiry-hours 1)
```

There are side effects: the menu will stop updating automatically when you save notes. But it's not a big deal. You can always refresh it by pressing the `R` key.

Tip #2

You can set how many recent notes appear in the main menu (20 by default) by adding this code:

```
(setq howm-menu-recent-num 10)
```

Tip #3

You can change Howm's menu layout through direct "surgery." After all, it's just a `.txt` file.

Press the `m` key while in the Howm menu to open the file for editing. You can rearrange any elements in this file—for example, swap the positions of recent and random notes.

Tip #4

You're also free to add items to the menu. It could be a link to a special note, a logon point. Just write a regular Howm link somewhere in the file:

```
>>> Profound Thoughts
```

Tip #5

Another option is to add a search filter for a specific word. To do this, place the following entry somewhere in the file:

```
%here%(howm-menu-search "your word")
```

The menu will display a list of matching notes.

Adjusting Lists

Tip #6

The menu file is located in the Howm folder, so it appears in lists alongside other entries. If you find this distracting, you can move it to another directory. Just don't forget to specify the new location in your `.emacs` file.

```
(setq howm-menu-file "~/path-to-the-file
/0000-00-00-000000.txt")
```

Tip #7

To prevent the preview window from opening automatically while scrolling through the only-list screen, add the following code:

```
(setq howm-view-keep-one-window t)
```

Tip #8

When viewing a list of open buffers, numeric names aren't very informative. Howm can display note titles instead. Add the following lines to your `.emacs` file to make it happen:

```
(add-hook 'howm-mode-hook 'howm-mode-set-buffer-name)
(add-hook 'after-save-hook 'howm-mode-set-buffer-name)
```

Heads up! In long notes with many titles, this setting may behave strangely.

Tip #9

To make Howm open the list of notes in a second frame, add the following lines to your `.emacs` file:

```
(setq howm-view-close-frame/tab-on-exit t)
(setq howm-view-window-location 'frame)
```

This is useful when you want to check something in your wiki without closing the current window. Once you're done with the second frame, just press `q` to close it.

Adjusting File System

Tip #10

To keep `.howm-keys` and `.howm-history` files neatly packed in the Howm folder, add the following to your `.emacs` file:

```
(setq howm-home-directory "~/howm/")
(setq howm-directory "~/howm/")
(setq howm-keyword-file (expand-file-name ".howm-keys"
  howm-home-directory))
(setq howm-history-file (expand-file-name ".howm-history"
  howm-home-directory))
```

Tip #11

You can specify additional directories for Howm to search for notes:

```
(setq howm-search-path '("/path/to/notes/directory" "/path/
  to/other/notes/directory"))
(setq howm-search-other-dir t)
```

Tip #12

If you're using an email client like Wanderlust under Emacs, you can configure Howm to search for files in the corresponding directory. Just make sure to specify the correct file path:

```
(setq howm-search-path '("~/Mail" "~/News"))
(setq howm-search-other-dir t)
```

Adjusting Notes

Tip #13

If you want Howm to create one file for each day, add this line to your .emacs configuration:

```
(setq howm-file-name-format "%Y/%m/%Y_%m_%d.txt")
```

In this case, Howm will append dated titles to an existing note instead of creating a new file. The result will be a long note with sections.

Tip #14

To quickly narrow down the content of a long note, assign a keybinding.

```
(define-key howm-mode-map (kbd "C-c ,")
  'howm-narrow-to-memo)
```

Now, after using the `C-c ,` command, you'll see only the relevant section.

Tip #15

To remove the blank line after the title in a quick note template, use this code:

```
(setq howm-remember-insertion-format "%s")
```

Tip #16

If you want Howm to save files in .org format, add the following to your .emacs file:

```
(setq howm-file-name-format "%Y/%m/%Y-%m-%d-%H%M%S.org")
```

Tip #17

For even deeper integration with Org mode, do this:

Remove the line from the previous tip, and place these lines above (`(require 'howm)`):

```
(require 'howm-org)
(setq howm-file-name-format "%Y-%m-%d.org")
```

This will change the note template and main menu layout to Org mode syntax. Notes will be saved in `.org` format, with one file for all daily entries.

Keybindings will also be rewritten: wherever you used a comma in Howm commands, use a semicolon instead.

Tip #18

Org mode files outside the Howm directory don't recognize Howm's commands. To enable the keybindings, add the following line to your `.emacs` file:

```
(add-hook 'org-mode-hook 'howm-mode)
```

Even Howm's links will work in any Org document—though you can only link to files inside the Howm directory, since that's where the search is performed.

There is one downside, however: switching between Howm and non-Howm buffers will no longer work for Org mode files.

Adjusting Tasks

Tip #19

You can include a tally of completed plans in the main menu. Just place the following line wherever you'd like to see it:

```
%here%howm-congrats-count
```

Howm keeps a tally of tasks completed during the session and resets the count to zero when Emacs is restarted.

Tip #20

To indicate the buoyancy of each task in lists and Howm's menu, add the following line to your `.emacs` file:

```
(setq howm-menu-todo-priority-format "(%8.1f)")
```

Tip #21

Howm uses commands called action-locks. You're already familiar with the `{ }` sign, but it's also possible to set up other “switch units” to perform various actions.

One such unit is already available as an example. Just type `{ }` in any note, place the cursor inside the brackets, and press `[RET]` a few times. The symbol will cycle through an asterisk, then a dash, and back to a space.

You can assign any characters you like to this function. For example, you can use specific symbols to indicate the completeness of notes or your progress on tasks.

```
(setq action-lock-switch-default '("{" }" "{○}" "{○}" "{●}" "{●}" "{●}"))
```

To make it work correctly, place this line in the `.emacs` file before the `(require 'howm)` line.

Tip #22

When you press `[RET]` on a special task symbol, Howm prompts you for the next action. You can assign the `[RET]` key to mark the task as finished instead.

Just add the following line to your `.emacs` file:

```
(setq howm-action-lock-reminder-done-default "")
```

To access the prompt, press `[C-u RET]`.

Adjusting Links

Tip #23

It's possible to make links appear inline. You just need a symbol to mark where the link ends—for example, a semicolon. Add the following lines to your `.emacs` configuration file to enable this behavior:

```
(setq howm-keyword-body-regexp "[^;\\r\\n]+")
(setq howm-ref-body-regexp "[^;\\r\\n]+")
```

Make sure to place it before: `(require 'howm)` line.

Tip #24

The “come-from” links feature is case-sensitive, meaning it only turns words into links if their case matches the keyword. To make it work regardless of case, add the following line to your `.emacs` file:

```
(setq howm-keyword-case-fold-search t)
```

Chapter 24

Questions and Concerns

Q: Will Howm slow down when I have numerous notes?

A: It depends on your operating system, Emacs version, and computer performance. Usually, Howm handles thousands of notes without any noticeable difficulties.

Q: Wouldn't important notes get lost without a detailed structure?

A: The system allows important notes to stay prominent, while less relevant ones naturally drift to the periphery. A certain degree of chaos in the knowledge base proves to be an advantage. Conversely, overly rigid classification and structure burden the user with the constant need to tidy everything up.

Q: Is Howm more efficient than other knowledge management systems?

A: It depends on your needs and preferences. There are no features in Howm that a user couldn't replicate in other applications if desired. The advantage of Howm lies in a reasonable balance between simplicity and features. Another advantage is its integration into the Emacs ecosystem. If you prefer working with Plain Text files, Howm is a sensible choice.

Chapter 25

Howm Command List

The meaning of commands becomes apparent within the context of working with notes, so consider this list a crib sheet. This table contains only shortcuts specific to the Howm package; I skipped other useful Emacs shortcuts.

| Shortcut | Command | Action |
|-------------|------------------------------------|---|
| C-c , , | M-x howm-menu | Call the main menu |
| C-c , c | M-x howm-create | Create a new note |
| C-c , e | M-x howm-remember | Create a quick note |
| C-c , I | M-x howm-create-interactivly | Select a note name and localization |
| C-c , D | M-x howm-dup | Duplicate note |
| C-c , a | M-x howm-list-all | List all notes |
| C-c , l | M-x howm-list-recent | List recent notes |
| C-c , . | M-x howm-find-today | List today's notes |
| C-c , : | M-x howm-find-yesterday | List of yesterday's notes |
| C-c , A | M-x howm-list-around | Show the note in temporal context |
| C-c , K | M-x howm-keyword-to-kill-ring | Copy a note title |
| C-u C-c , K | --- | Copy a numeric filename |
| C-c , s | M-x howm-list-grep-fixed | Search/recall keywords |
| C-c , g | M-x howm-list-grep | Search with regexp |
| C-c , o | M-x howm-occur | Search inside one note |
| C-c , h | M-x howm-history | Show search history |
| C-c , i | M-x howm-insert-keyword | Insert a keyword into a note |
| C-c , p | M-x action-lock-goto-previous-link | Jump to the previous link |
| C-c , n | M-x action-lock-goto-next-link | Jump to the next link |
| C-c , x | M-x howm-list-mark-ring | Show the list of marks |
| C-c , b | M-x howm-list-buffers | List open buffers |
| C-c , Q | M-x howm-kill-all | Close all buffers (except unsaved) |
| C-c , SPC | M-x howm-toggle-buffer | Switch between howm/non-howm files |
| C-c , t | M-x howm-list-todo | List to-dos |
| C-c , T | M-x howm-insert-dtime | Insert the current date and time |
| C-c , y | M-x howm-list-schedule | Call up a schedule |
| C-c , d | M-x howm-insert-date | Insert the current date |
| C-c , C | M-x howm-create-here | Insert a heading with date |
| C-c , N | M-x howm-next-memo | Jump to the next heading in a long note |

| Shortcut | Command | Action |
|-----------------|---------------------------------|---|
| C-c , P | M-x howm-previous-memo | Jump to the previous heading in a long note |
| C-c , L | M-x howm-last-memo | Jump to the last heading in a long note |
| C-c , H | M-x howm-first-memo | Jump to the first heading in a long note |
| C-c , r | M-x howm-refresh | Activate links on keywords in text |
| C-c , R | M-x howm-menu-refresh | Update the main menu |
| C-c , M | M-x howm-open-named-file | Open a file outside Howm directory |
| C-c , w | M-x howm-random-walk | Follow random come-from links |

Naturally, you can quickly check Emacs's key bindings for any mode. While in Howm, type **C-h b** to see a list of available shortcuts.