Facet - User Guide

Capture first, organise later

> *A complete guide to Facet - the modern thinking workspace for Windows. From your first captured thought to multi-view databases, rules, and automations.*

Facet · getfacet.co.uk · 2026

Facet is a personal information manager that is designed around a single, guiding principle: **capture first, organise later.** You type whatever's on your mind into a single quick-entry box, and Facet helps you arrange it afterwards through categories, rules, and multiple views of the same data.

This guide shows how to use Facet, starting with a quick hands-on tour, then a section-by-section reference, and finally a set of worked recipes and answers to common questions. It assumes no prior knowledge - if you can type a sentence and click a menu, you can follow along.

The Facet website contains other reading matter; a whitepaper explaining the Facet philosophy, a detailed 'Cookbook' and some technical notes for developers.

---

§ Installing and Licensing Facet

The Facet software can be downloaded from here https://getfacet.co.uk/downloads and you can get a fully functioning 14 day Trial Licence or a permanent licence from here https://getfacet.co.uk/license.

When downloaded, some Anti-Virus packages will scan the installer. This is perfectly normal behaviour and shows that your AV software is doing its job!

When you install the software you may get a warning from Windows SmartScreen. To install the program simply select "More info → Run anyway" to install the software. Once installed, Facet will be available on the Windows Start menu, from where it can be added to your Taskbar or Desktop like any other Windows application.

Like any Windows application, you can add a Shortcut to Facet to your Desktop, Pin it to the Start Menu, or pin it to the Task Bar. You might also find it useful to allow Windows to start Facet whenever the PC is restarted, so it's always available in the background. This is done as follows:

1. Press Win + R on your keyboard to open the Run dialog.Type shell:startup and press Enter to open your personal Startup folder.
2. Locate Facet (you can search for it in the Start menu). Right-click the Facet icon, select More, and click Open file location. (If you don't see this option, right-click the icon and choose "Create shortcut," then copy the shortcut that appears.)
3. Copy and paste the Facet shortcut into the Startup folder you opened in Step 2. Facet will now automatically launch every time you start or restart your PC.

Updating Facet

When a new version of Facet is released, you update simply by downloading the latest installer and running it - there is no need to uninstall the current version first.

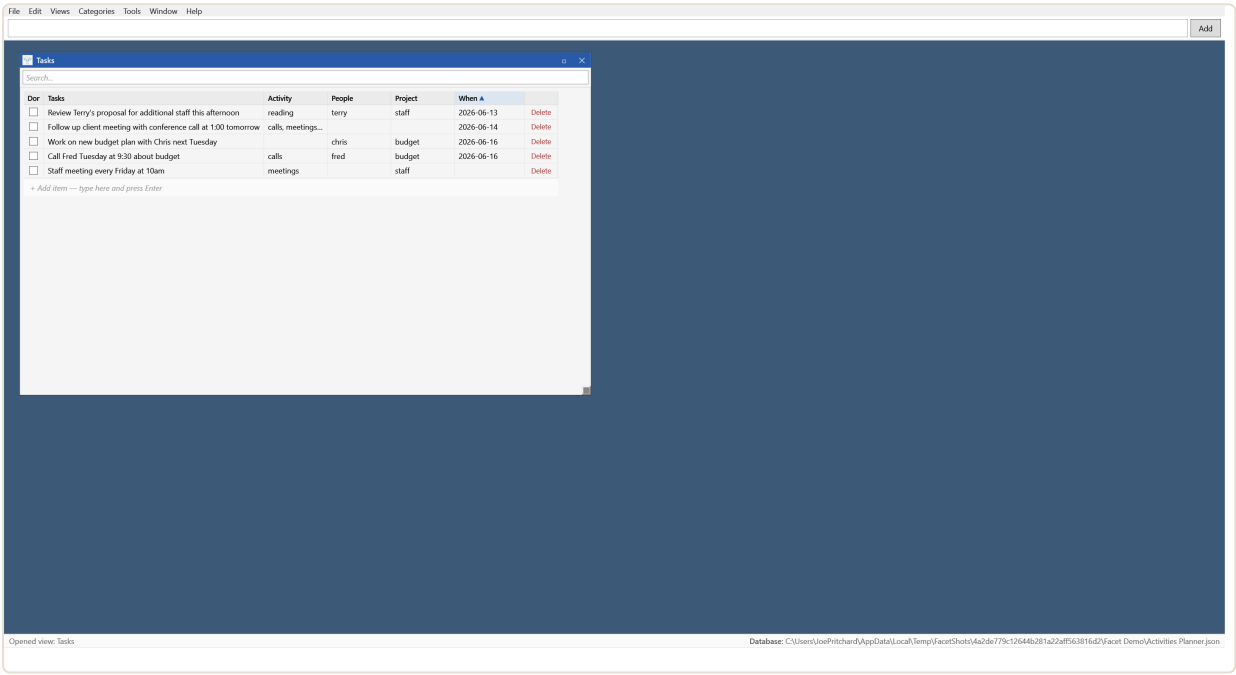
- **Your application is replaced.** The installer writes the new version over the top of the existing one, so you end up with a single, up-to-date copy of Facet in the same place. Your Start menu entry, Taskbar pin, and any Desktop shortcut keep working.
- **Your data is left untouched.** Your databases, settings, and backups live separately from the program files, so updating Facet never touches them. Everything is exactly where you left it when you next open the app.
- **Your licence is preserved.** The new version recognises your existing licence - Trial or permanent - automatically. You don't need to re-enter your licence key or re-activate after an update.

§ Getting started - a quick tour

The fastest way to understand Facet is to try it out watch one thought travel from your head onto the screen and into a view. This tour takes about ten minutes.

1. Open Facet for the first time

When Facet launches it opens your database and shows the **main window**: a menu bar, a **quick-entry box** across the top, the blue **MDI workspace** where view windows float, and a **status bar** along the bottom.



If this is a brand-new install, create something to play with: **File → Create New Database…**, give it a name, and pick a starter template such as **Activities Planner** (the one shown above) or **Radio Log**. Select to include the sample data. Each template arrives with categories, rules, and a handful of views already set up, so you can see how the pieces fit before building your own.

2. Capture a thought

Click the quick-entry box at the top of the page, type a whole sentence, and press **Enter**:

> *Call Fred next Tuesday about the budget*

That is the only decision Facet asks you to make up front - *what* the thought is. You do **not** choose a folder, a project, or a priority first. The item lands in the database immediately, Facet reads "next Tuesday" and sets the due date for you, and the rules engine attaches any categories whose patterns match ("call" → an *actions/calls* tag, "Fred" → a *people/fred* tag, and so on).

The sample data for the Activities planner has been pre-configured with the following data:

People

1. fred
2. chris
3. terry

Actions

1. calls
2. email
3. follow-up
4. meetings
5. readings

Try entering a few more items and watch them appear. Nothing needs filing; everything that you enter is captured.

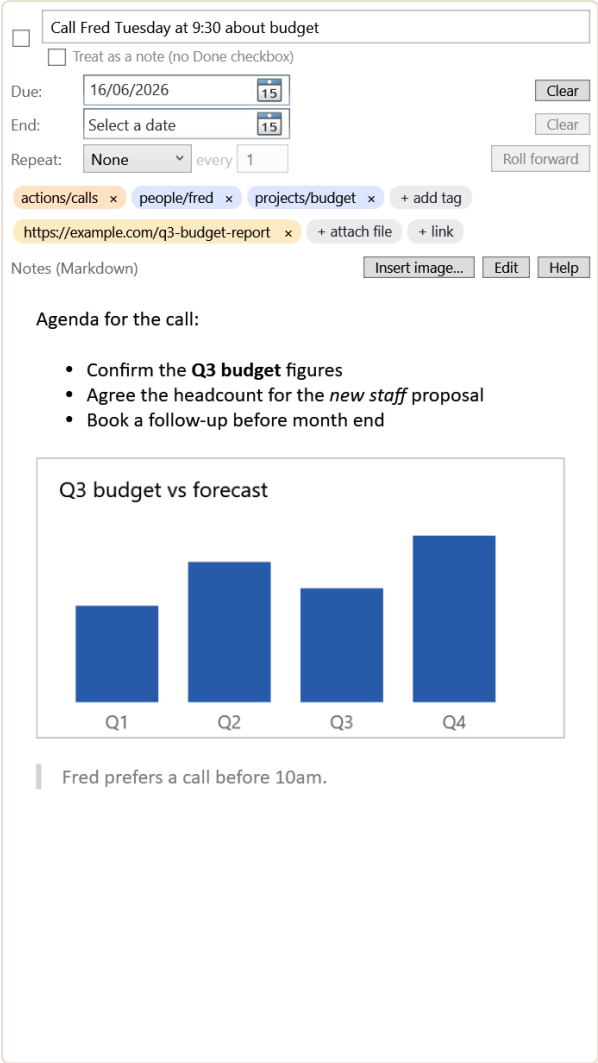
3. Look at the same items from different angles

A **view** is a lens through which you see your items, not a file folder or container for them. The same item shows up in every view whose filter it satisfies. Open the **Views** menu and try opening two or three at once. Simply click on the View that you want to see, and it will be opened on the desktop. You can drag the windows around the workspace, resize them, or close the ones you don't want. Facet remembers the changes you make to the layout of Views and the desktop layout for next time.

Because views all use the same underlying item, editing an item in one view changes it everywhere. If you mark *Call Fred* done in the calendar, it is marked as done in every other view too.

4. Open the detail of an item

Click an item's heading (the **Task** cell in a columnar view) to open the **Detail** panel. Here you can adjust the due date, set a repeat pattern, add or remove category tags, attach files or links, and write long-form **Markdown** notes.



5. Make it yours

Everything in Facet is configurable. **Categories** organise your tags and labels; **Rules** auto-apply them; **Views** slice the data; **Automations** let you script repetitive jobs.

You can change any of these, save the database, and Facet will remember what you have done. Your requirements change; your Facet database changes with them.

The rest of this guide walks through each of those in turn.

Facet is designed to make organising your data a seamless activity:

- **Capture without filing.** Type first, organise later - that is the whole idea. You don't have stop to decide where something "goes", and lose the moment.
- **Undo fearlessly.** Every change is a single **Ctrl+Z** away, so experiment; you can always undo your changes. The engine is built to make trying things out easy.

§ 1. Concepts in one minute

Facet has a vocabulary of it's own, and it's worth listing the terms here:

- **Item** - one piece of information: a task, a note, a reminder, an idea - anything you can put into words or represent with an image or link. Has a short heading, an optional due date (and optional end date), an optional repeat pattern, and an optional Markdown notes body.
- **Category** - a tag you can attach to items. Categories live in a tree (e.g. `People > Fred Smith`, `Projects > Pricing policy`), and one item can carry as many categories as you like.
- **View** - a way of looking at items. A single item is seen from different angles: by due date, grouped under a category, as a matrix, or on a calendar.
- **Rule** - a stored instruction - "when X, do Y" that fires automatically when items are added or changed. Used for auto-tagging data and other functions within Facet.
- **Automation** - a named sequence of actions you trigger manually, by keyboard shortcut, on a schedule, or when something happens. Automation goes beyond what rules are capable of. A Facet Automation can create items, open views, branch, loop, prompt you for input, and act on items in bulk.
- **Database** - A Facet database is where the data is stored. You can have as many databases as you please, and switch between them.

The point: you enter free-form text, and you choose *afterwards* how to view it. Items are not filed into folders - they appear wherever a view's filter brings them up.

---

§ 2. The main window

When Facet opens you see:

- **Menu bar** - File, Edit, Views, Categories, Tools, Window, Help.
- **Quick-entry box** (top) - type an item and press Enter.
- **MDI workspace** (centre, default blue background) - floating windows for each open view, plus a Detail window when an item has been selected. If you drag or resize a window past the edge of the workspace, scroll-bars appear so you have a workspace as large as you need!
- **Status bar** (bottom) - The last action performed is on the left, and the current database path is shown on the right. Warnings and errors - for example, a condition that stopped an automation from running - appear in **red**. Ordinary information messages are grey.

The title bar shows `Facet – <database name>`, where is the name you gave to the database when you created it.

Standard Window operations live under the **Window** menu. The usual Cascade, Tile Horizontal, Tile Vertical, Close All operations all behave as you might expect for a Windows application. Individual view windows can be moved, resized, maximised inside the workspace, and closed with the ✕ in their title bar.

---

§ 3. Adding items

There are several ways to enter an item into your Facet database.

Quick entry

Type into the box at the top of the main window and press Enter (or click **Add**). Example:

> *Call Fred next Tuesday about pricing*

The item lands in the database immediately, picks up any due date it can find in the text, and is run through the rules engine so matching categories get attached automatically.

If the text contains a web address - for example *"Interesting page about tea at https://tea.co.uk"* - Facet recognises it and attaches it to the item as a clickable **link** (shown on the Detail panel), so you can open the page later without copying the address out by hand. The heading text itself is left exactly as you typed it.

Quick-entry text may include `{name}` placeholders - `{today}`, `{tomorrow}`, `{now}` and a few more - which are filled in with the current date or time as the item is created. See *Automations* (section 7) for the full list.

Date phrases the parser understands include:

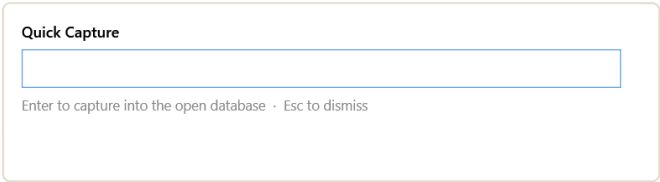
- Standard: `today`, `tomorrow`, `yesterday`, `next Tuesday`, `a week on Friday`, `the first Monday in June`, `23 December`.
- Time-of-day shorthand: `this morning / afternoon / evening`, `tonight`, `tomorrow morning / afternoon / evening / night`, `yesterday evening`, etc. - these resolve to the appropriate **day** (`this afternoon` is today, `tomorrow morning` is tomorrow); the hour is ignored. A time-of-day word may also trail a specific date or weekday (`26th May in evening`, `Wednesday afternoon`) - the day is kept and the hour ignored, so the item still gets a date.
- Bare weekdays resolve **forwards**: `on Wednesday` means the coming Wednesday, never the one just gone.
- Numeric: `2026-06-01`, `1/6/2026`, etc. (locale-aware fallback).

Try entering a few items with different date phrases to see what happens. It can be very informative if you open a Calendar view, and then enter some items - you can watch them pop up on the Calendar view.

Quick capture (global hotkey)

Provided that Facet is running, press **Ctrl+Alt+N** from within *any* application - Facet doesn't need to be focused

- and a small **Quick Capture** box appears. Type a line and press **Enter**; the item is added to the open database (placeholders and date phrases work just as they do in the main quick-entry box). **Esc** dismisses the box without capturing.

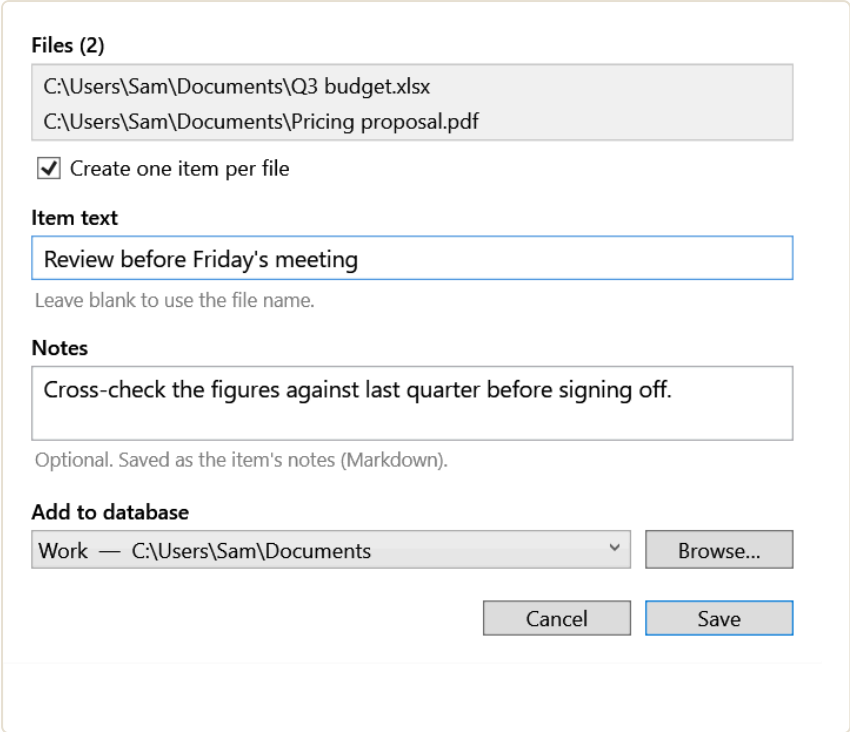


The hotkey is on by default. Turn it off - or check whether another app has already claimed the combination - under **File → Facet Settings → General**.

Add to Facet from File Explorer

Facet can add a right-click **Add to Facet** command to Windows File Explorer, so you can file a document into a database without switching to the app.

- **Turn it on** from **File → Add to Facet in Explorer menu** - a checkable menu item whose tick shows whether the command is currently installed. It is a per-user setting and needs no administrator rights; untick it to remove the command again.
- **Use it:** right-click a file - or a selection of several files - in Explorer and choose **Add to Facet**. On Windows 11 the command sits under **Show more options** (the classic context menu).
- A small dialog appears. Pick the **target database** (it defaults to the one you are using, with your recent databases listed), and optionally type a line of text for the new item plus any **Notes** (Markdown) to store on it. For a multi-file selection you can choose **one item per file** or **one item with a link to each file**.
- When you type item text, Facet appends the captured file's name in brackets - so *"Read this"* on `report.pdf` becomes **Read this (report.pdf)**. Leave the text blank and the file name alone becomes the heading. (Web links, which have no meaningful file name, are not given a bracketed suffix.)
- Each captured item gets the file attached as a **link** - click it later to open the document - and is run through the rules engine for auto-tagging, just as if you had typed it in. Note that the contents of the attached file are **not** run through Facet's rules engine, so if you want any searchable context for the files, add a meaningful item name.
- **Facet doesn't need to be open.** If it is running, the item appears within a second or two; if it is closed, the capture is queued and picked up the next time you open that database. Nothing is lost either way.



Per-group "+ Add item"

Each group inside a view has a small **+ Add item** link. Clicking it lets you add an item *into that group*. The item inherits the group's context:

- Adding under a **category-grouped** view's heading attaches that category to the new item.
- Adding into a **date bucket** sets the due date (you'll get a small date prompt for the ambiguous "This week" and "Later" buckets).

Add from Clipboard

If you copy some text on to the Windows clipboard from any application, you can automatically turn it into an Item within Facet. Simply **Right-click** anywhere on the Desktop within Facet. A dialogue will appear. Enter the item text, press OK, and an Item will be created with the text on the clipboard added to the Notes field. Facet will open the newly added item for you to view.

Detail panel

Selecting any item opens the **Detail** window. Note that this is **one shared window - it re-points to whatever you select**. If you close it, it stays closed: Facet remembers whether it was open and only re-opens it on the next launch when it was open as you left. **File → Open Detail** brings it back any time. Here you can edit:

- The heading.

- **Treat as a note** For items that are reference notes or messages rather than tasks, tick this to hide the **Done** checkbox for this item everywhere (views, the calendar, and this panel). A whole category can do the same for every item filed under it - see *Categories*. Once an item is marked as a note, the **Due**, **End**, and **Repeat** rows below disappear too: a note has no schedule and no recurrence.
  (Reference databases - those with completion turned off in **Properties** - keep the date fields visible, since dates are still meaningful even when "done" doesn't apply.)
- **Due** date and optional **End** date.
- **Repeat** - none / daily / weekly / monthly / yearly, with an interval (e.g. every 2 weeks). Three refinements appear as you choose a pattern:
  - **Weekly** shows a row of weekday toggles (**M T W T F S S**) - pick the specific days it falls on (e.g. Monday, Wednesday, Friday). Leave them all off to repeat on the due date's own weekday.
  - **Monthly** lets you repeat **on the day of the month** (the 15th, say) or **on the Nth weekday** - "the 2nd Tuesday", "the last Friday".
  - **Ends** - every repeat can run **Never**, **On** a fixed date, or **After** a number of occurrences. The calendar's ghost chips and **Roll forward** both respect the end you set.

  The **Roll forward** button advances Due (and End) to the next occurrence - it does *not* create separate copies; the same item moves forward each time you click it (and a count-limited series clears itself once the last occurrence is reached). On a **Calendar** view, future occurrences appear as faded "↻" chips on every matching date in the visible month, so you can see at a glance when the next occurrences fall - click a ghost chip to jump to (and edit) the source item.
- **Tag chips** - These show the categories attached to this item. Click a chip's **×** to remove it; click **+** to add one. A tag you add by hand stays put even if you later edit the heading - only rule-assigned tags are recomputed when the text changes.
- **Links** - attach local files, folders, or web addresses to the item. Click **+ attach file** to browse for a file, **+ link** to type a URL, or simply **drag a file from Explorer** onto the Detail panel. Each link shows as an amber chip: click it to open the target with its default application, or click the **×** to detach it (the file itself is never deleted). Links are first-class - separate from any Markdown links you put in the Notes body - so an item can point straight at the document it's about.

- **Notes (Markdown format)** - long-form notes. Toggle between **Edit** and **Preview** with the button at the top of the notes pane. **Help** opens the Markdown syntax reference. As well as the basics (headings, **bold**, *italic*, lists, links, images, quotes, code and horizontal rules), the notes support a good slice of *extended* Markdown: tables, fenced code blocks, `~~strikethrough~~`, `==highlight==`, `++underline++`, subscript/superscript (`H~2~O`, `X^2^`), tick-box task lists (`- [ ]` / `- [x]`), and automatic links from bare web addresses. Footnotes, definition lists and emoji shortcodes are not rendered. **Tab** indents inside the notes - two spaces, which is what nests a list item under the one above it - and **Shift+Tab** takes an indent back. Select several lines first to shift them together. **Ctrl+Tab** moves on to the next field, if you are working without the mouse.

Edits commit when the field loses focus (so tabbing through is safe).

To delete an item, use the **Delete** button at the right-hand end of its row in any view (after every column), the **Delete item…** entry on its right-click menu, or press **Delete** with the item selected and no text field focused. Deletion always asks for confirmation; tag-to-category links are left intact.

Copying an item

Edit → Copy Item duplicates the selected item - its tags, dates, repeat pattern, links, and notes - appending `- (Copy)` to the heading. It's a quick way to use an existing item as a template for a similar one, and it's a single undo step.

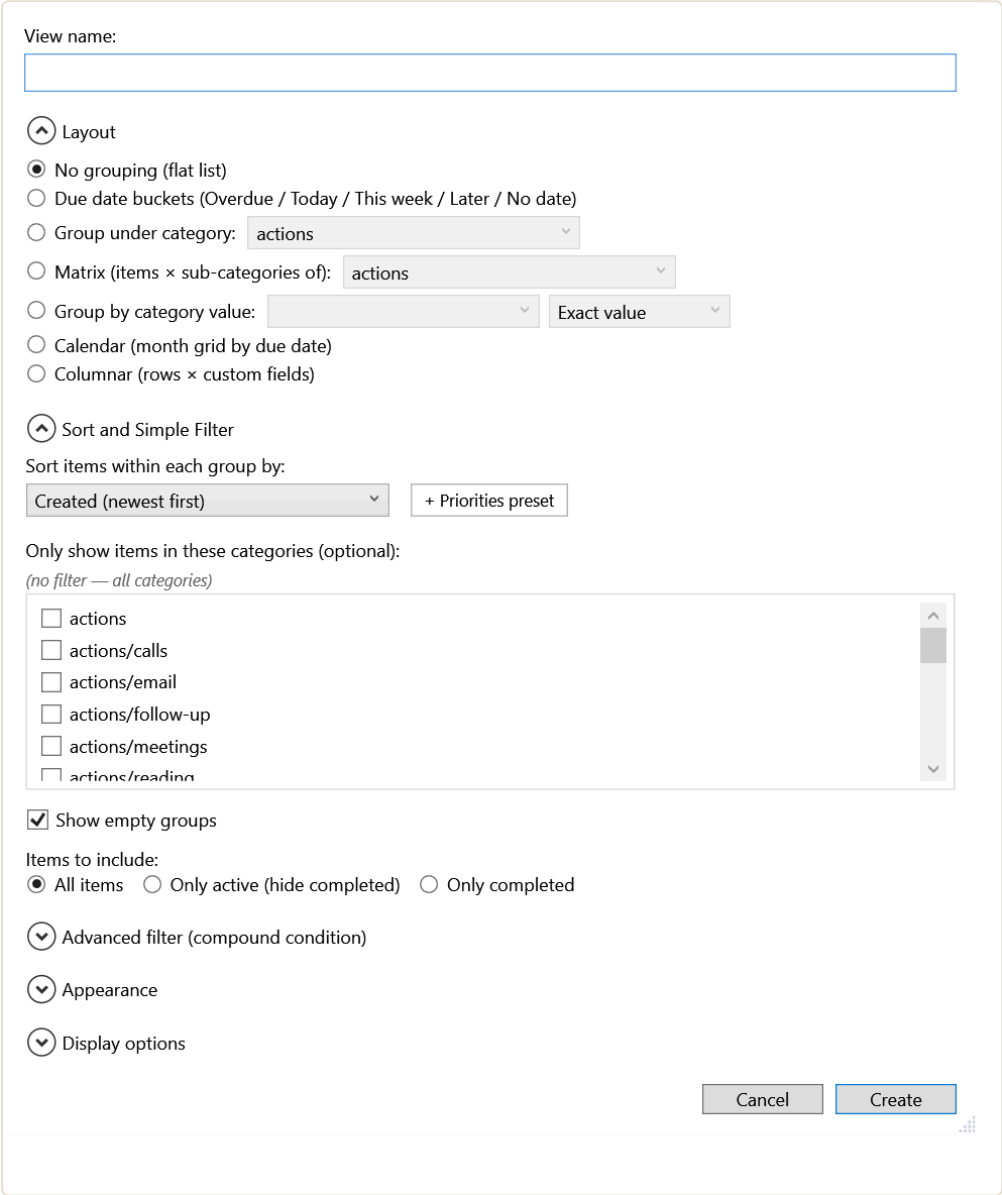
---

§ 4. Views

Creating a view

Views are the surprisingly powerful tools at the heart of Facet. As with all aspects of Facet, experiment to see what you can do with them!

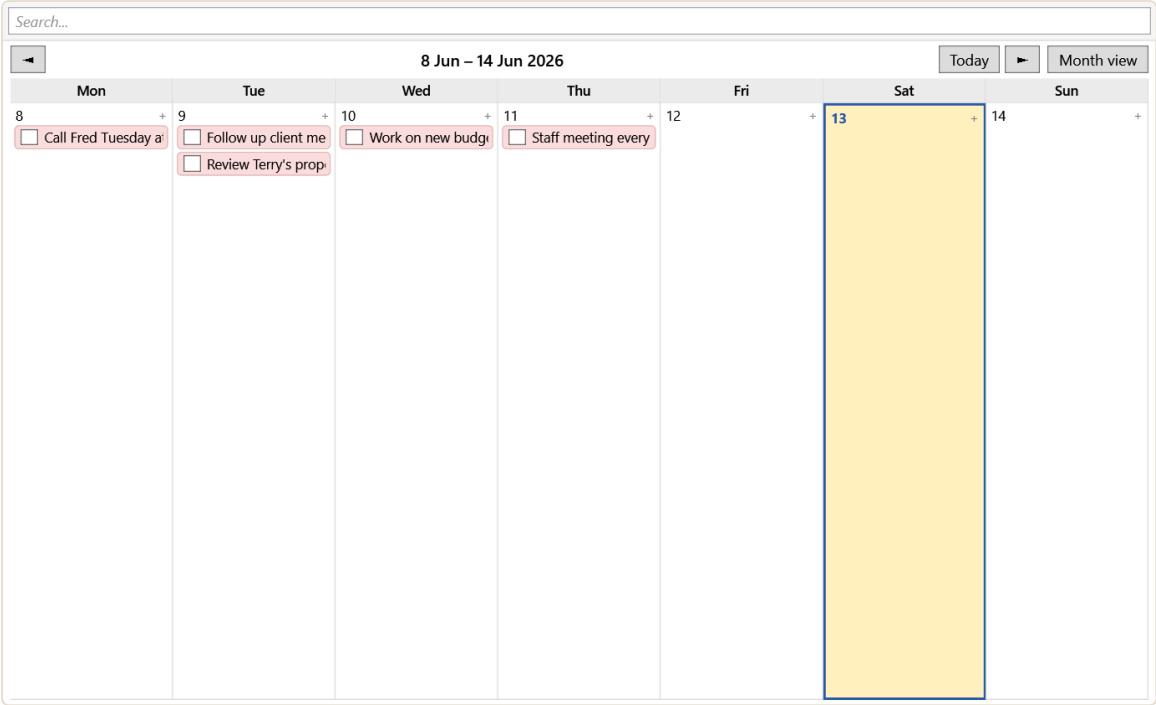
Views → New View… opens a dialog where you choose:



- **Name** - anything you like.

- **Layout** - one of:
  - **Flat list** - no grouping.
  - **Due date buckets** - A 'Date Bucket' is a date or range of dates that is used to gather together items. Values are : Overdue / Today / This week / Later / No date. *This week* means the rest of the current calendar week (today through the last day of the week); items due next week - even a day or two away - sit under *Later*. The week's first day is set per database on the **Properties** page (Monday by default).
  - **Group under category** - pick a parent category; the view shows one group per direct child, and rolls items up from all descendants.
  - **Matrix** - items as rows, the direct children of a chosen category as columns. Click a cell to toggle that tag on that item.
  - **Calendar** - a month grid. Items appear on their due date; drag an item onto a day to reschedule it. Use ◄ / ► to move months and **Today** to jump back - the current day is shaded yellow with a blue border so it stands out at a glance. The **Week view** button (top right) switches to a single week of tall, full-height cells - handy when a day has more items than a month cell can show; ◄ / ► then step a week at a time and the button flips to **Month view** to switch back. The week begins on the day set by **Week starts on** in the database's Properties. **Overdue items** (active tasks whose due date is before today) are tinted pale red so they catch the eye, and hover an item over a day cell during a drag and the cell glows pale blue to show where the drop will land. Item names that are too long to fit are trimmed with an ellipsis - **hover an item to see its full name** in a tooltip. When a day holds more items than its cell can show, the extra ones collapse into a "**+N more**" line (hover it to read the hidden headings, or switch to Week view to see them all). Items with no date appear in a **No date** strip beneath the grid - drag any of them onto a day cell to schedule it; only genuinely undated items appear there (an item dated in another month simply waits on its own month rather than crowding this one, and **Note** items - those with their tick box switched off - are left out of the strip, since they are reference material rather than things to schedule; a Note that *does* carry a date still shows on its day). Items with a **Repeat** pattern (e.g. weekly) also appear as faded "↻" chips on every future occurrence inside the visible period - click one of these 'ghost chips' to jump to the source item. Ghost chips are read-only; reschedule by editing the source item.
  - **Group by category value** - pick a typed category (Number / Date / Text - see *Typed categories* under section 5) and a **grain**. Text groups by exact value. Date offers Exact / by day / by week / by month / by year. Number offers Exact / by 1 / by 10 / by 100 / by 1000 - the bucket label reads e.g. `14200–14299`. Items lacking a value fall into a `(no value)` section, always last.

- **Sort and Simple Filter** (expander) - sort and basic filtering options:
  - **Sort items within each group by** - text, due date, etc.
  - **Only show items in these categories (optional)** - multi-select list of categories. Tick any number; items must carry a tag under **any** ticked category to appear. Leaving every box clear shows every item (no filter).
  - **Show empty groups** - keep groups visible even if they have no items.
  - **Items to include** - All / Only active / Only completed.
- **Advanced filter (compound condition)** - more complex conditions built with the same editor used for rules. The view shows items that satisfy *both* this and the simple category filter.
  To list only items due on a particular day - for example **today** - add a single **Due date** condition here: set the operator to **on** and the date to **today**. The view then shows just the items whose due date is today, and stays correct tomorrow because *today* is evaluated live each time the view refreshes. The same condition with **on or before / today** gives you everything due by the end of today (including anything overdue), and **within / 7 / days of / today** gives a rolling week. *(If you would rather see your whole list broken into Today / This week / Later instead of filtered down to one day, use the **Due date buckets** layout above.)*
- **Appearance** - here you can set the title-bar background, title font colour and body background.
- **Display options** (expander) - five switches that tailor the view, all of them off by default.
  - **Show statistics strip** - adds a "N items · M done · K overdue" line at the top of the view. Notes (items with the **Treat as a note** flag, or those in a category that suppresses the Done checkbox) are excluded from all three counts, so the line describes just the task-shaped items.
  - **Number items within each section** - `1.`, `2.`, `3.` prefix.
  - **Draw a thin rule between sections** - visual divider in dense grouped views.
  - **Hide column headers** - columnar layouts only; suppresses the header row in narrow windows.
  - **Section order** - alphabetical (default), reverse alphabetical, most-items first, fewest-items first, or **Manual order…** which reveals a textbox where you list category paths one per line (top becomes first section). Useful for priority-style views where High / Medium / Low isn't alphabetical.



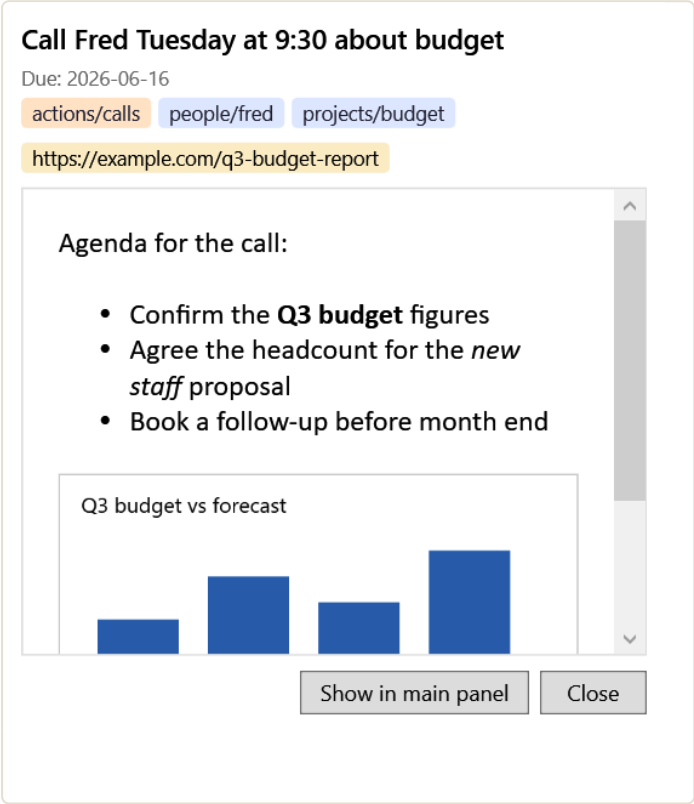
Using a view

- The **Search box** at the top of every view allows you to filter items. The text entered is used to filter items by the Item Name or contents of the Notes field.It's 'real time' (there is a short delay after a key is pressed). It also matches against typed-category text values. If you take a look at the 'Radio Log' example, you'll find it searches the QTH, Operator, Callsign, SIO fields and numeric values, so a Radio Log search for "ZS6" or "14250" finds the right contact. Press **Esc** or the ✕ to clear the search.
- **Drag** items between groups to reassign categories or change due date. Dragging from an ancestor group correctly strips the descendant tag that was causing the rollup.
- **Delete an item** with the **Delete** button at the right-hand end of its row - after every column in a columnar view - or from its right-click menu. **Right-click a view's title bar** (or its entry in the Views menu) to allow you to **Edit view…** and **Delete view…**.
- **Select several items at once** in any list or columnar view: **Ctrl+click** to add or remove a row, **Shift+click** to select a run, and a plain click to go back to one. Selected rows are highlighted. **Right-click** the selection for bulk actions - **Mark done / not done**, **Set** or **Clear due date**, **Add tag…**, and **Delete** - each applied to every selected item as a single undo step. (Multi-select isn't available in the Calendar and Matrix layouts, which have their own gestures.)

The item right-click menu

Right-click a single item in a list or grouped view for its own menu:

- **Open in new window** - opens the item in its own standalone window: a read-only card showing its heading, dates, repeat pattern, category and link chips, and rendered notes. You can open as many as you like and leave them side by side to compare or refer to items while you work on others - they update live as the items change. Editing still happens in the main Detail panel; **Show in main panel** (in the card) selects the item there.
- **Copy whole item** - the same duplicate-as-template action as **Edit → Copy Item**.
- **Move to database… / Copy to database…** - send the item to another of your databases (see *Move or copy an item to another database* under section 9).
- **Delete item…** - with a confirmation prompt.



Sorting a columnar view

In a columnar (table) view, **click any column header to sort by that column**; click the same header again to reverse the order. The active sort column tints pale blue and shows a ▲ or ▼ glyph next to its label. The choice is saved with the view (so it survives close/reopen and is covered by **Ctrl+Z**), and works for Text, Due, Completion, Created, Modified, Tags, Notes, Repeats, Completed, Number, and Category columns (typed Number / Date / Text categories sort by value; plain category columns sort by the leaf segment of the matching tag). A Number column with no Header set isn't sortable - it has no key to sort by.

Numeric columns and TOTAL

A columnar view can include **Number** columns (decimal values per item). Set the column **Kind** to *Number* in the view editor and give it a **Header** (which doubles as the data key - *don't* rename it later, or the existing values will go missing). Optionally set a **Currency** prefix like `$`, `£`, or `kg`. Decimals default to 2 and thousands separators are on by default (edit the JSON to change).

- **Edit** a Number cell the same way as other cells (double-click or F2). Type a raw number - currency prefix and thousands separators get stripped on commit, so `$1,234.56` is accepted as-is.
- An empty value clears the entry (renders blank, not "0").
- When the view has at least one Number column and at least one item, a **TOTAL** footer row appears: the leftmost text column gets the label, each Number column shows the sum, other columns stay blank.
- For a **typed Number category** (see *Typed categories* in section 5), the same TOTAL footer covers any Category column targeting it - *unless* you untick **Show TOTAL row in columnar views** on that category in the Categories dialog. Switch it off for values that don't sum naturally (a Radio Log frequency, a sequence number, a percentage): the column's footer cell goes blank, and when no other column qualifies, the whole TOTAL row is suppressed.
- One item can carry many named numbers - a `Pledges` column in one view and a `Hours` column in another are independent fields on the same item.

Wrapping long column text

By default a columnar cell shows a single line and trims the rest with an ellipsis. Tick **Wrap** for a column in the view editor's column list to let its cells wrap onto several lines instead - most useful for **Task** and **Notes** columns. Give the column a **Width** as well, so there is a fixed edge to wrap against; a wrapped column left without a width falls back to a sensible default.

Inline cell editing (columnar views)

In a columnar view, the Task, When, and category cells are editable in-place - you don't have to open the Detail panel.

- **Click the Task cell** to open the Detail panel for that row (matches the gesture in every other view). Clicking any other cell just selects the row without showing Detail, so in those cases your focus stays where you're working.
- **Double-click** an editable cell, or click to select then press **F2**, to start editing. **Enter** or **Tab** commits, **Esc** cancels.

- **Delete** on a focused cell:
  - On a **Task** cell - deletes the whole item (same as the row-level DEL in other layouts).
  - On a **Category** cell whose target is a **plain** category - strips that tag and any descendants from the item.
  - On a **Category** cell whose target is a **typed** category (Number / Date / Text - see *Typed categories* in section 5) - clears the per-item value only; the category membership stays so the item still appears in category-grouped views and rule conditions. Re-typing a value brings the cell back.
  - On a **Tags**, **When**, or **Number** cell - clears just that cell (breaks the assignment / clears the value) and leaves the item alone.
  - Use **Ctrl+Z** in every case to put it back.
- **Resize a column** by dragging the thin handle on the right edge of any header cell. The cursor switches to a horizontal-resize arrow when you hover near the boundary. Widths persist with the view, and Ctrl+Z undoes a resize like any other change. (Hide column headers is on? Re-enable headers to expose the handles, or edit the view to set the Width numerically.)
- **Reorder columns** by dragging a header onto another header. A blue bar shows up on the left or right edge of the target column to tell you where the dragged column will land - drop on the left half to insert *before* the target, the right half to insert *after*. To move a column to the end, drop on the right half of the last column. Ctrl+Z undoes the move.
- **Categories type-ahead** - as you type, Facet fills in the rest of the longest matching category name as a selected suffix. Press **Enter** to accept, or keep typing to override.
- **Multiple categories per cell** - a Category cell shows every matching tag, comma-separated. Press **Ins** while a Category or Tags cell is focused to *add* a new tag without disturbing the existing ones (typing in normal edit mode replaces the set).
- **Dates** - a When cell accepts natural language ("Tuesday", "next Friday") as well as numeric formats; the cell parses on commit.
- Editing a tag that doesn't yet exist as a category creates it (with ancestors) automatically - no need to dip into the category manager first.

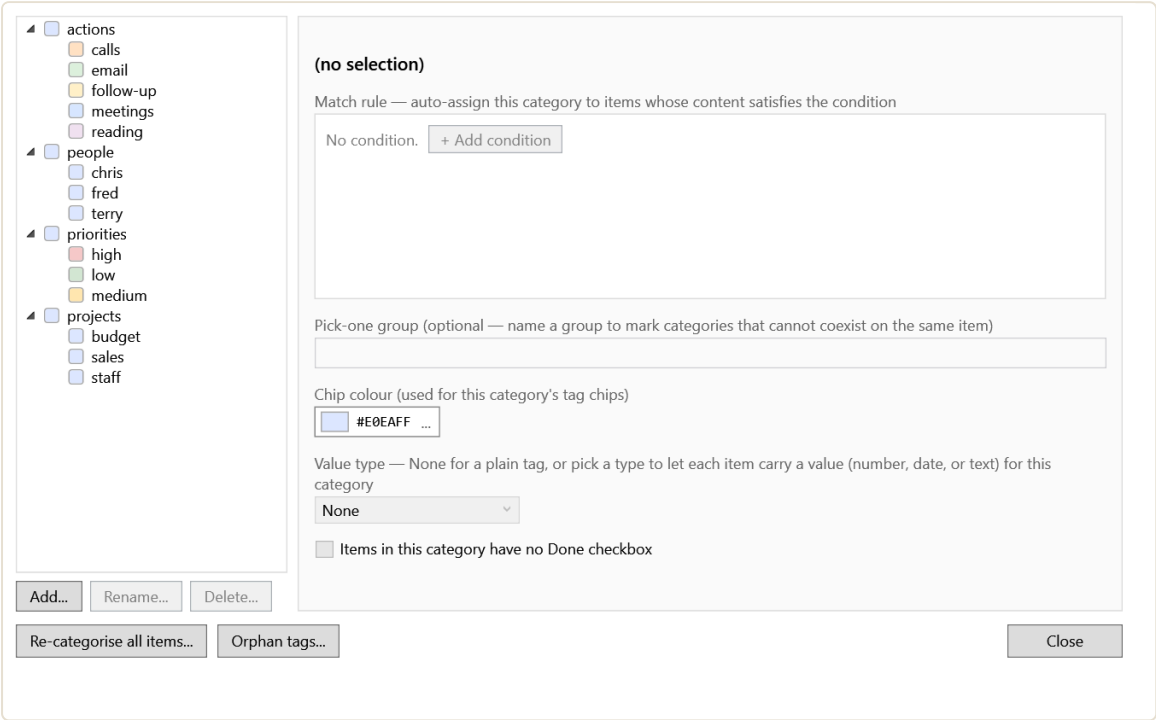
Saved view positions, sizes, z-order, and open/closed state - together with the Detail window's open/closed state - persist between sessions, so the workspace comes back the way you left it.

If you use the same Facet database on different computers with different screen sizes, then you may find that some elements are 'off the screen' on the smaller screen size. They've not been lost; they're temporarily out of sight.

---

§ 5. Categories

Categories are at the heart of Facet, and an understanding of how they work is essential - fortunately, they're not that complicated to use! **Categories → Manage…** opens the category manager:



- Left pane: the category tree. Use **Add…**, **Rename…**, **Delete…** for basic operations. Rename and delete cascade - every item tag, view filter, view grouping, and rule reference is updated when you change something here.

- Right pane (per selected category):
  - **Match rule** - a condition that, when true for an item, auto-assigns this category. (This is the per-category form of the rule engine.)
  - **Pick-one group** - give a name; any categories sharing that name cannot coexist on the same item (useful for status-like values).
  - **Chip colour** - colour for the category's tag chips and the matrix column header.
  - **Items in this category have no Done checkbox** - tick this to treat every item filed under the category (or a category nested beneath it) as a note rather than a task: its completion checkbox is hidden everywhere. Combined with the category's match rule, items lose the checkbox automatically as they flow in - the category-wide form of the Detail panel's *Treat as a note*.
  - **Value type** - turns the category into a *typed slot* that holds a per-item value as well as the bare membership. Choices: **None** (the default - plain tag), **Number**, **Date**, or **Text**. See *Typed categories - per-item values* below for the full story. Changing the type on a category that already has values warns first before discarding them.
  - **Show TOTAL row in columnar views** - appears only for **Number**-typed categories. On by default (the column auto-totals in a columnar view); untick for values that don't sum naturally - a Radio Log frequency, a sequence number, a percentage. The Radio Log starter template ships with this flag off for `frequency` to demo it.
- **Re-categorise all items…** - re-runs every rule across every item in the database. Use this after editing rules. Tags you assigned by hand are kept; only rule-assigned tags are recomputed. To have this happen for you, switch on **Auto-refresh categories on category edit** on the database's Properties page (see *Databases* below) - the refresh then runs automatically each time you close the category manager after a change.
- **Orphan tags…** - shows item tags that don't correspond to any category. You can either promote them (turn the tag into a real category) or strip them.

Categories from a view

You can drag-rearrange items between category groups in a view, and that edits their tags directly. The view is also an editing surface - you don't have to open the manager just to move things around.

Typed categories - per-item values

A category can hold a **per-item value** as well as the bare membership. Set **Value type** on the category to one of:

- **Number** - a decimal (price, frequency, hours, weight, …). The Detail panel surfaces a numeric editor; a columnar Category column targeting this category renders the value (formatted by the column's Currency / Decimals / Thousands) instead of the tag leaf, and the TOTAL footer row sums it alongside any classic Number columns.
- **Date** - a date (DateTime under the hood; the time part is ignored at render time). The Detail panel shows a date picker; columnar cells accept natural-language input ("next Friday") the same way the *When* column does.
- **Text** - free text. Useful for short codes (radio operators' "559" SIO reports, room numbers, model numbers) where parsing as a number would lose context.

Assignment and value are independent. Removing the tag chip removes the category from the item; clearing the value (DEL in a cell, blank in the editor) leaves the tag intact. A columnar **Category** column whose **Prefix** targets a typed category automatically switches to the typed editor - no separate column kind needed.

Where typed values show up:

- **Detail panel** - a *Category values* expander beneath the tag chips lists one editor row per typed category currently assigned to the selected item. Hidden entirely when no typed category applies.
- **Columnar views** - set a Category column's Prefix to a typed category's path; the cell renders (and inline-edits) the per-item value.
- **Sort** - the View dialog's *Sort items within each group by* dropdown gains an entry per typed category, ordering items ascending by value (items without a value sort last).
- **Group by** - the *Group by category value* layout option (see *Views*) buckets items by typed-category value with type-appropriate grain options.
- **CSV round-trip** - typed-category columns survive *File → Export to CSV…* and *File → New Database from CSV…* via the `[number] [date] [text]` header convention (see those sections).

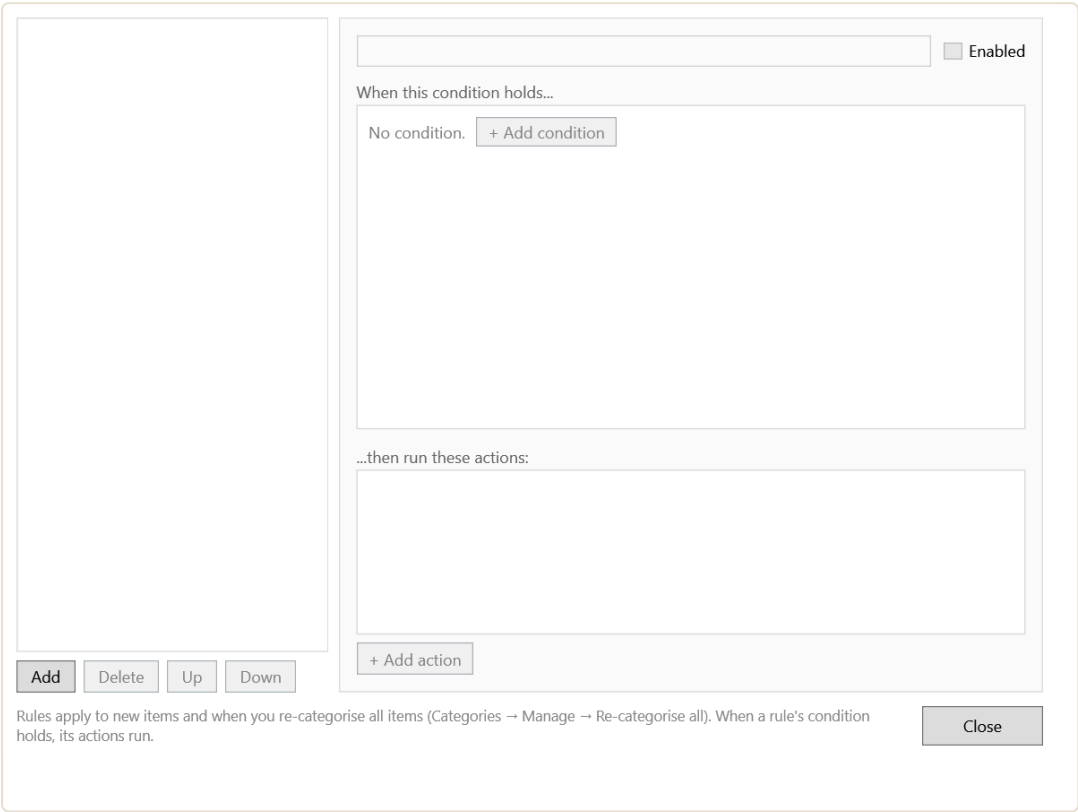
The **Radio Log** starter template (under **File → Create New Database…**) shows all these category types in use. A typed Date for when each contact was logged, typed Number for frequency in kHz, typed Text for the SIO report, operator name, and grid square, plus mode and band as plain pick-one categories. Open it and play - you will no doubt think of other applications for this sort of layout.

---

§ 6. Rules

Rules are at the heart of how Facet automatically processes the data you enter. By setting appropriate rules you can assign an item to a category, remove it from a category - even delete the item. Once set up, rules are run automatically by Facet.

Categories → Rules… opens the rules dialog. A rule has a **name**, an **Enabled** checkbox, a **condition tree**, and a list of **actions**.



Conditions

Built up with AND / OR / NOT and these attributes:

- **Text matches** - word-boundary substring match against the item's text.
- **HasCategory** - item carries a given category (with optional descendant rollup, so "HasCategory: Projects" can mean "any descendant of Projects").
- **HasDue** - item has a due date.
- **DueDate compared to value** - equality and range comparisons against a date.

Actions

- **Assign category** - add a tag.

- **Remove category** - remove a tag.
- **Set due date** - with start/end and an "only if unset" option so the rule doesn't fight a date the user typed.
- **Discard item** - delete the item entirely.

When rules run

- On every new item, and whenever an item is edited.
- On **Re-categorise all items…** in the category manager.

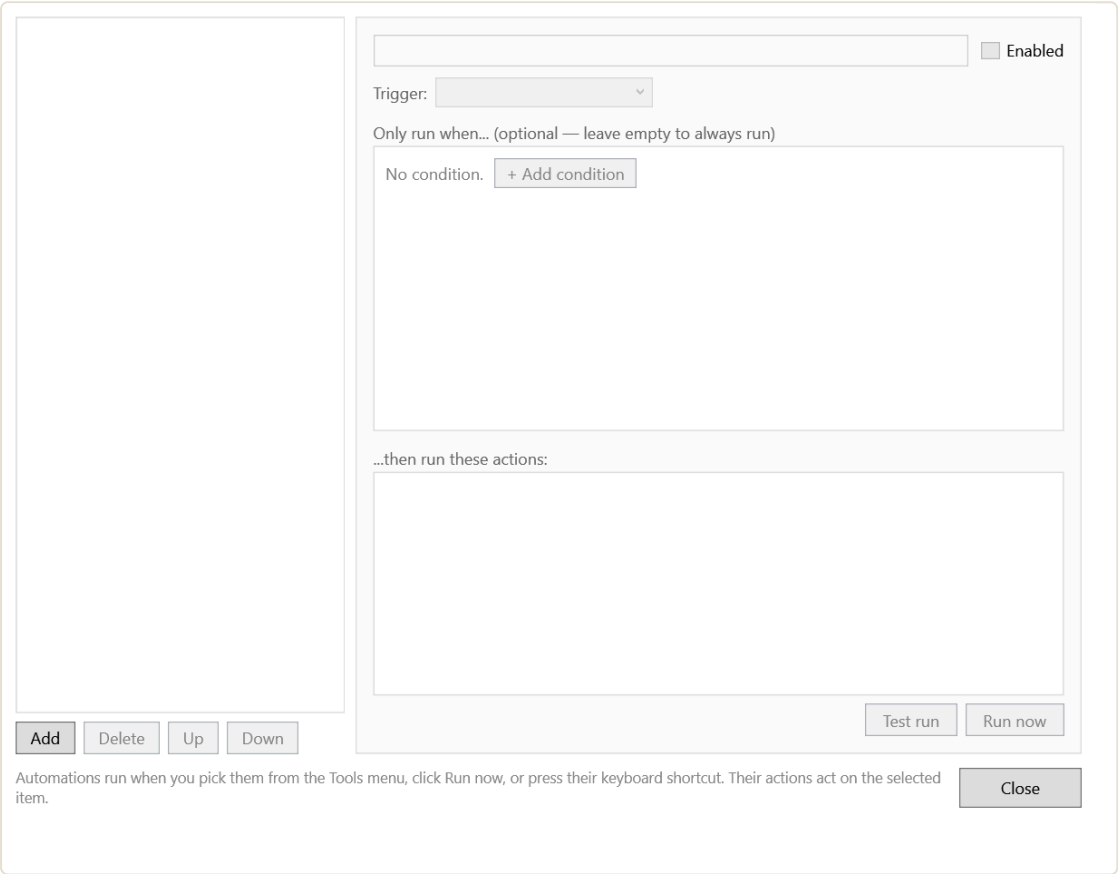
The engine ensures that rules that process category changes all settle in one pass. Side-effect actions (set-date, discard) are then executed.

---

§ 7. Automations

A **rule** (section 6) auto-assigns categories. An **automation** goes further: it is a named sequence of actions you trigger by hand, by keyboard shortcut, on a schedule, or when something happens in the app. Rules answer "what categories does this item have?"; automations answer "when X, do Y" - tagging, setting dates, creating items, opening views, showing messages - and can branch, loop, and ask you questions as they run.

Tools → Automations… opens the manager. Each automation has a **name**, an **Enabled** checkbox, a **trigger**, an optional condition, and a list of **actions**.



Triggers - when an automation runs

Choose one from the **Trigger** dropdown:

- **Run manually** - runs only when you ask. Manual automations are listed in the **Tools** menu (below "Automations…") - click one to run it. The **Run now** button in the manager runs the selected automation immediately, handy while you build it.
- **Keyboard shortcut** - click the box and press a combination (Ctrl with a letter or number, optionally Shift). Pressing it runs the automation - except while you are typing in a text box, and built-in shortcuts (Ctrl+S, Ctrl+Z, …) are refused.

- **When something happens** - runs on an app event: an item is created, completed or deleted; the selection changes; a view is opened or closed; a database is opened; or the app starts.
- **On a schedule** - runs **daily**, or **weekly** on a chosen day, at a time you set. A scheduled automation catches up if the app was closed when its time passed: open Facet at 10:00 and a 09:00 daily automation runs then. It runs at most once a day. *Editing the schedule re-arms it:* if a daily automation already fired today and you then change its time (or day), Facet clears the "already ran today" mark so it can run again at the new time - so adjusting a 16:15 trigger to 16:20 fires again at 16:20.

"Only run when…" - an optional condition

Under the trigger is a condition editor - the same one rules use. Leave it empty and the automation always runs; fill it in and the automation runs only when the condition holds for the item in context.

Actions - what an automation does

+ Add action adds a step; an automation runs its actions top to bottom. The basic kinds:

- **Assign category** / **Remove category** - add or remove a tag.
- **Set due date** - with an optional end date and an "only if unset" guard.
- **Discard item** - delete the item. Any automation that can delete items is flagged ⚠ **deletes items** in the list.
- **Show message** - pop up a dialog with text you choose. When this fires from a scheduled automation while you're working in another application, Facet also **flashes its taskbar button** so you notice the dialog even though it opened behind the app you're using - click the flashing button to bring it forward.
- **Run automation** - call another automation by name.
- **Create item** - add a new item with the text you give.
- **Open view** - open, or focus, a saved view. Tick **centre on screen** to have the view appear centred in the workspace with Facet brought to the foreground - useful for a scheduled automation that should grab your attention even while you're working in another application. Left unticked, the view simply opens (or is focused) without pulling Facet forward.

Procedural actions - branching, loops and prompts

Five further action kinds turn a flat list into a small program:

- **If** - give a condition; the actions under **Then** run when it holds, those under **Else** when it doesn't. Both branches are ordinary action lists, so an If can contain more Ifs.

- **For each item** - give a matching condition; the actions under **Do** run once for every item that matches. Inside the loop, that item is the one the body's actions act on, and its heading is available as a variable whose name you choose.
- **Set variable** - store a value (which may itself contain placeholders - see below).
- **Ask for value** - pause and ask you for some text, storing the answer in a variable. Cancelling the prompt ends the run.
- **Stop** - end the run immediately; no later actions run.

Targets - which items an action touches

The item actions (assign / remove category, set due date, discard) each carry a **target** dropdown:

- **the item in context** - the selected item; for an event automation, the item the event was about; inside a **For each**, the current loop item.
- **items matching…** - choosing this drops a condition editor below the action; the action then runs on *every* item that matches. This is how you work in bulk - "set every overdue item to *waiting*", or "discard everything matching this condition".

Variables and placeholders

Text in a **Show message**, **Create item** or **Set variable** action can contain `{name}` placeholders, replaced when the automation runs:

- **System constants**, always available - `{today}`, `{now}`, `{datetime}`, `{tomorrow}`, `{yesterday}` (`{date}` and `{time}` are aliases for the first two).
- `{item}` - the heading of the item in context.
- **Run variables** - any name created by **Set variable** or **Ask for value**.

An unrecognised `{name}` is left exactly as written.

The same `{name}` placeholders also work in the **quick-entry box** and the per-group **+ Add item** boxes (section 3) - type "Trip notes " and the new item is created with today's date filled in. The value is baked in once, as the item is entered.

Trying an automation safely

While you build an automation, **Test run** (next to **Run now**) does a *dry run*: it reports, in a window, everything the automation would do - every tag, date, new item, deletion, message - **without changing anything**. Use it to vet a bulk or item-deleting automation before you trust it.

Tools → Automation Log… lists the automations that have run this session, newest first, each with the time and how it ended (*Ran*, *Stopped early*, or an error).

Running, undo, and safety

- Every run is a **single undo step**, however many actions or items it touched - one **Ctrl+Z** reverses the whole thing.
- Automations cannot loop: if A runs B and B runs A, Facet stops the chain and notes it in the status bar. While an automation runs, the changes it makes do not trigger other event automations.
- A run that changes nothing adds no undo entry.

---

§ 8. Search

Two distinct search surfaces:

- **Per-view search box** - at the top of every view; filters that view as you type. The query is saved on the view (so it survives close/reopen and is what "Save as view…" captures). As well as the item's heading and notes, the box matches against per-item **typed-category text values** (Operator, Callsign, QTH, SIO, and any other free-form Text-typed category) and **numeric values** (Number columns and Number-typed category values, formatted plainly so a search for `14250` finds a frequency value).
- **Global Find** - **Ctrl+F** opens a Find pane that searches across *all* items in the database, text and notes. Click a result to jump to it. Use **Save as view…** at the bottom of the pane to persist the current query as a flat-list view.
- **Search Other Databases** - **Tools → Search Other Databases…** searches every *other* Facet database in your database folder and recent list at once, **without opening or importing any of them**. Results are grouped by database; select one for a read-only preview, and **Go to in its database** opens that database and jumps straight to the item. Use it when you can't remember which database something is filed in.

---

§ 9. Databases

You can have multiple Facet databases. Under **File**:

- **Open Database…** - point Facet at an existing `db.json`.
- **Create New Database…** - make a new one, optionally pre-seeded with sample data. A fresh database comes with a starter set of views, including a **Today** list - active items due today or overdue, soonest first - for a quick daily review.
- **New Database from CSV…** - build a brand-new database from a spreadsheet file. See *Import from CSV* below.
- **Import…** - merge items and categories from another Facet database into the active one. Items match by Id (duplicates skipped, never overwritten); missing categories and their ancestor chain are created. Rules are *not* imported (so a foreign rule can't silently reshape this database's tagging). The whole merge is one undo step.
- **Export…** - write the current database to a standalone JSON file that can be opened or imported elsewhere.
- **Export to CSV…** - write the current database as a CSV file: one row per item, columns for category roots and number fields. The shape is the same one **New Database from CSV…** expects, so the export round-trips cleanly back through import. See *Export to CSV* below.
- **Publish as Facet package…** - bundle the database together with everything that travels with it - its Explorer capture inbox and its bundled-images folder (see **Bundle images**, below) - into a single **Facet package** (a `.facpub` file). Handy for moving a database to another machine, emailing it, or taking a snapshot - and anyone can open it read-only with the free **Facet Reader** (see *Publish and share* below).
- **Create from package…** - the reverse: pick a `.facpub` made by **Publish as Facet package**, choose where to save it and what to call it, and Facet rebuilds a complete database - images and inbox included - then opens it.

Saves happen automatically (debounced). **Ctrl+S** or **File → Save Now** forces an immediate save. Every database is backed up once a day - see *Daily backups* below.

Daily backups

Facet keeps its own safety net. Once a day it writes a backup of **every database it knows about** - the ones in your database folder and the ones in your recent list - whether or not you have opened them that day. The pass runs in the background shortly after Facet starts (and again just after midnight if you leave Facet running), so it never gets in your way.

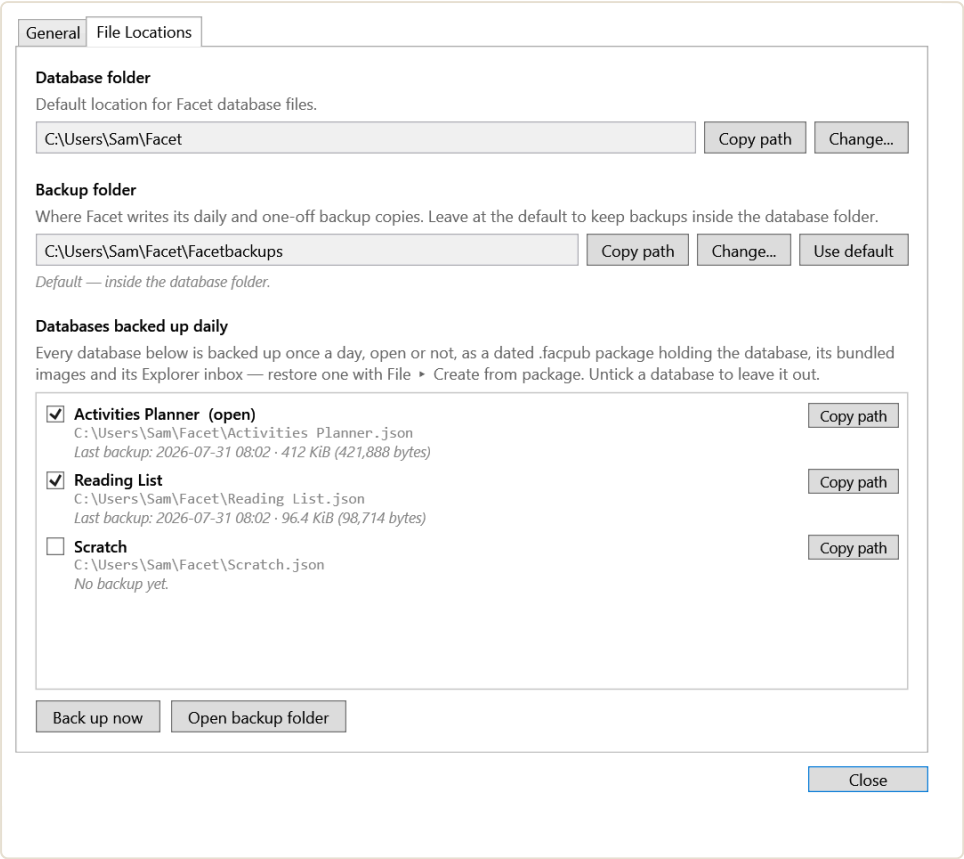
A backup is a **Facet package**: a dated `.facpub` file holding the database *and everything that travels with it* - its bundled-images folder and its Explorer capture inbox - so nothing is missing when you come to put it back.

Backups are grouped by database, one folder each:

```
Facetbackups\
    Activities Planner\
        source.txt                                  which database this folder b
        Activities Planner-2026-07-30.facpub
        Activities Planner-2026-07-31.facpub
        Activities Planner-2026-07-31-143205.facpub  a "Back up now" taken at 14
    Reading List\
        Reading List-2026-07-31.facpub
```

- **Where they go.** By default, a `Facetbackups` folder beside the database itself. **File → Facet Settings… → File Locations** can point every database's backups somewhere else instead - a second drive, a synced folder, a NAS share. Two databases that happen to share a file name keep separate folders there, so they can never overwrite each other's backups.
- **Choosing what gets backed up.** The same **File Locations** tab lists every database in the daily pass with a tick-box each; untick one to leave it out. New databases are backed up by default. For the database you have open, the same switch is on its **Properties** page (**Back this database up every day**).
- **Backing up on demand. Back up now** on that tab saves any pending edits and then backs up every ticked database whose latest backup does not already match it, and tells you what it did. If a database has changed since today's backup was written — you have just entered a few items — you get a second package stamped with the time as well as the date (`Activities Planner-2026-07-31-143205.facpub`) rather than "already up to date": the morning's package stays as it is, and the new one reflects the database as it stands. Press it twice with no editing in between and the second press correctly does nothing. **Open backup folder** opens the location in Explorer.
- **Restoring one. File → Create from package…**, pick the dated `.facpub`, and choose where to save it and what to call it. Facet rebuilds the whole database - images and inbox included - and opens it. (Restoring alongside your live database rather than over the top of it is deliberate: you can compare the two before deciding.) A `.facpub` is also an ordinary zip, so you can peer inside one with Explorer if you prefer, and anyone can open one read-only in the free **Facet Reader**.
- **Nothing is ever deleted.** Facet does not prune old backups; if the folder grows larger than you want over the months, delete the older files yourself.

Backups are a safety net, never a gate: if the backup location is unreachable - a disconnected drive, a share that has gone away - Facet says so on the status bar and carries on saving your work as normal.



If the database's drive goes missing. Keeping a database on a USB stick or a network share is fine, and Facet copes when the drive isn't there:

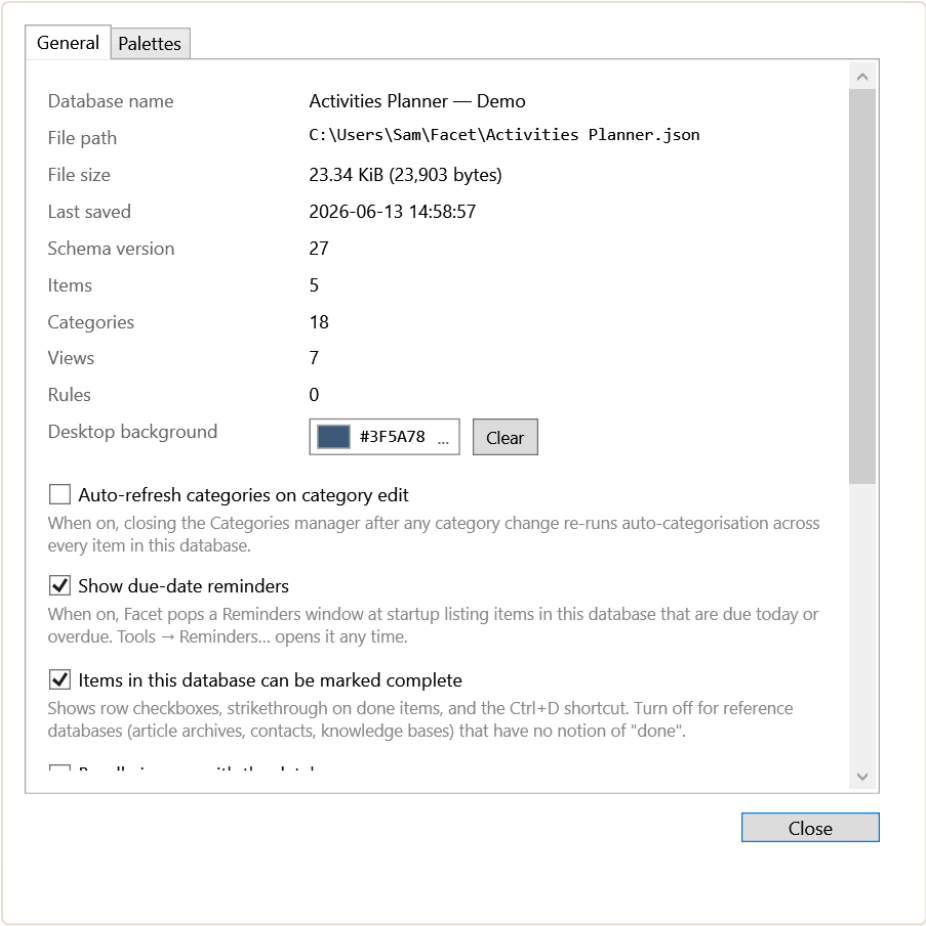
- **At startup**, if the location can't be reached, Facet opens your default database instead and tells you which location failed. Your chosen location is remembered, not overwritten - plug the drive back in, restart Facet, and your real database is back with nothing to fix. (Use **File** → **Open Recent** to hop back to the stand-in database if you want it again.)
- **While Facet is open**, if the drive is pulled out mid-session your work stays on screen rather than vanishing with an error. Facet says it couldn't save, writes a dated recovery copy in a `rescued` folder next to its settings, and names that file again if you close Facet while the drive is still missing. Reconnect the drive and **File** → **Save Now** to carry on as normal.

Two more **File** entries are useful when you're poking around:

- **File → Properties…** - the menu entry is labelled with the open database's name - shows the database name, file path, size on disk, last-saved timestamp, schema version, and counts of items, categories, views, and rules. It also holds every setting that belongs to *this* database, split across two tabs - **General** and **Palettes**.
  On the **General** tab:
  - The **Desktop background** picker recolours the MDI canvas behind your view windows - pick any swatch from the active palette, or click **Clear** to revert to the built-in default. The choice is per-database, so opening "Work" and "Home" databases can each have a distinctive backdrop.
  - Tick **Auto-refresh categories on category edit** to have Facet re-run categorisation across every item automatically whenever you close the category manager after a change, so you needn't remember **Re-categorise all items…** yourself.
  - **Show due-date reminders** controls whether the Reminders window pops at startup for this database.
  - **Items in this database can be marked complete** turns the completion checkboxes, strikethrough, and **Ctrl+D** on or off - switch it off for reference databases that have no notion of "done".
  - **Week starts on** sets the first day of the week for this database (Monday by default). It controls where the Calendar grid begins and the boundary of the **This week** due-date bucket.
  - **Bundle images with the database** - when on, an image you insert into an item's notes is copied into a `<database>-images` folder beside the database file and referenced relatively, instead of pointing at wherever the original happened to live. That keeps the database and its pictures together: they survive being moved as a pair, and **File → Publish as Facet package** can pack them up. Off by default, so existing notes keep referencing images where they already sit; turning it on affects images added from then on.
  - **Back this database up every day** - on by default. Leave it ticked and this database joins the daily backup pass described under *Daily backups*; untick it for a scratch or throwaway database you don't want copies of. The line beneath the tick-box shows exactly which folder its backups go to.
  - **Undo history depth** caps how many undo steps this database keeps (1–500); lowering it trims the oldest steps straight away.

  The **Palettes** tab manages the colour palettes for this database. A palette is a named set of colours that appear as quick-pick swatches in every colour picker - category chips, the desktop background, and so on. Pick a palette from the list to see its swatches; **New…**, **Rename…**, and **Delete…** manage the set, and **Use this palette in the database** makes the

selected one active. Click a swatch to edit its colour or × to remove it, and **+ Add** appends a new one. Because palettes live on the database, each database carries its own set - they travel with the file when you move or publish it.



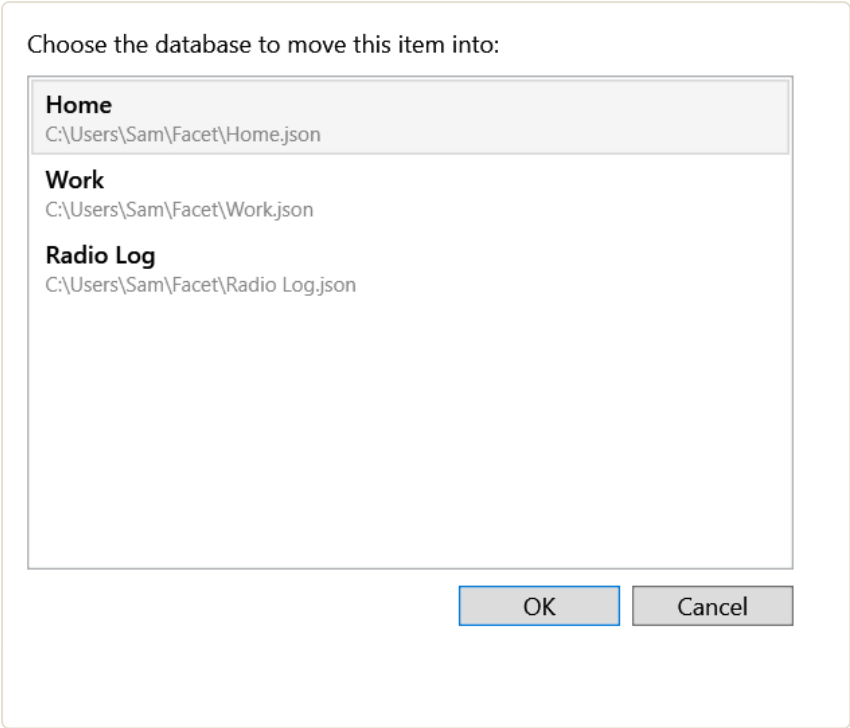
- **File → Abandon Changes…** discards everything you've changed since the last save and reloads the database from disk. The undo history is wiped too. Useful if you've made an experimental mess and the debounced autosave hasn't fired yet. There's a confirmation prompt; the default button is **No** so a stray Enter won't trash work.

Move or copy an item to another database

To send a single item to one of your other databases, **right-click it in a view** and choose **Move to database…** or **Copy to database…**. Facet lists the other databases it can find (those in your database folder and your recent list); pick one and the item is written into it:

- A fresh copy is made, with its own identity, so it reads as a new entry there.
- Any **categories** the item uses that the target lacks are created in it, parent categories included, keeping their colour and any typed-value kind.
- **Bundled images** in the item's notes (see *Bundle images*) are copied into the target's images folder and their links rewritten so they still show.

The target database is written straight to disk - it doesn't have to be open. **Copy** leaves the original in place; **Move** removes it from the current database once the target has been written successfully (so a problem writing the target never loses the item). The removal on a Move is a single **Ctrl+Z** step in the current database; the change to the *other* database isn't on this database's undo stack, since it's a separate file. A line in the status bar confirms what happened.



Import from CSV

File → New Database from CSV… turns a spreadsheet into a fresh Facet database - categories, views, and one item per row, inferred from the columns. The spirit of the original Lotus Agenda Definition File: hand Facet a flat table and get a structured database back.

The flow

1. Pick the CSV file. Facet auto-detects the delimiter (comma, tab, or semicolon), reads the header row, and profiles each column.
2. The **column-roles preview** appears. Each row in the grid is one CSV column, with the role Facet inferred plus a sample of the values. The summary line at the bottom tells you how many items, categories, and views the import will create.
3. Override any role from the dropdown. The summary updates live.
4. Click **Create**. Facet creates the new database, runs the import, and switches you into it.

How columns are mapped

Inferred role	When it fires	What it produces
Text (item heading)	The longest-on-average text column	Becomes `Item.Text` - exactly one column per import
Due date	At least 80% of values parse as a date	Becomes `Item.Due`; only one Due column is used (first wins)
Number	Every non-empty value parses as a decimal	Becomes a `ColumnDef` in the Table view; values land in `Item.Numbers`. Currency (`$ £ € ¥`) and decimal places are inferred from the data, so `$1,234.56` formats correctly in the TOTAL footer
Category (single)	Low-cardinality text column (≤ 15 distinct values, or up to ¼ of the rows for larger files, hard-capped at 50)	One Category per distinct value under `column-name/value`, set as a pick-one group so an item can only carry one value at a time
Category (multi)	Any cell contains a semicolon	Cell is split on `;` / `,`; each token becomes a Category under the column prefix, not a pick-one group (items can carry several)
Completion	Header is `done`, `complete`, `closed`, `finished`… AND values are boolean-shaped (`yes/no`, `1/0`, `x/blank`, `true/false`)	Drives `Item.IsComplete`; no category is created
Typed category	Header ends `[number]`, `[date]`, or `[text]` - the round-trip convention written by **Export to CSV…**	The category exists at the bare path (suffix stripped) with the matching **Value type**; values land in `Item.CategoryNumbers / CategoryDates / CategoryTexts` and the tag is added to the item. Also available manually in the role dropdown as **Typed category (number/date/text)**
Ignore	Anything else (high-cardinality strings the heuristic didn't promote to Text)	Dropped

Headers are kebab-cased into category paths: `"Pricing Policy"` becomes `pricing-policy`. Chip colours cycle through the database's active palette.

What you get out

For every import, Facet generates four kinds of view:

- **Inbox** - flat list of every item.
- **Table** - a columnar view that mirrors the CSV's layout: Text, Due, every Number column (with the inferred currency and decimals), then a Category column per category column in the source. Has the statistics strip switched on.
- **By** - one grouped view per Category column. Sections use the category chip colour as a background.
- **By Date** - only generated when a Due column was chosen.

After the import you have a regular Facet database. Add views, rename categories, rewrite the Table view's columns - nothing is locked.

Tips and limits

- One Text column only. If your CSV has several long-prose columns, the longest-on-average wins Text and the others fall to Ignore - promote the one you actually want from the preview dialog.
- One Due column only. If two columns parse as dates, only the first is used
  - the rest flip to Ignore.
- The category-column cap is `min(50, max(15, rows / 4))` distinct values. A 100-row CSV with a 25-value "Region" column will categorise; a 100-row CSV with a 60-value "Notes" column will not.
- Multi-tag splits on `;`. Comma-separated tags work too but only if the column has at least one semicolon-bearing cell - otherwise commas are treated as prose.
- The import is "create-new" only. To merge a CSV into an existing database, import it into a new one first, then use **File → Import…** to merge.
- A typed-category header (e.g. `frequency [number]`) is detected before any value-shape heuristic, so a sparse Date column still imports as the typed category rather than collapsing to plain Date or being ignored. If the target category already exists in the destination with a different value type, the existing type wins - the CSV doesn't reshape it.

Example files

The repo ships several CSVs you can try under `docs/csv-examples/`:

- `tasks.csv` - Title, Due, Status, Priority, Assignee. The everyday case.
- `expenses.csv` - currency Amount column drives the TOTAL footer in the Table view.

- `reading-list.csv` - multi-tag `Topics` column and a `Done` Completion column.
- `tricky-quoting.csv`, `edge-cases.csv`, `semicolon-european.csv` - edge-case fixtures exercising the RFC-4180 parser, the full inference table, and semicolon-delimited European files.
- `reading-list-500.csv` - a 500-row volume test for scrolling and filtering performance.

A matching manual test plan (`docs/csv-examples/TESTING.md`) walks through ten numbered smoke tests with expected outcomes.

Export to CSV

File → Export to CSV… does the opposite of the importer: it writes the current database to a CSV file shaped so the importer can read it back.

What gets exported

- **One row per item.** Items with no Text are skipped (they can't be re-imported anyway).
- **Standard columns first:** `Text`, `Due` (only when at least one item has a Due date), `Done` (only when the database has completion semantics - Information-Sifter-style databases skip it).
- **One column per category root.** If your database has categories like `status/open`, `status/closed`, and `topics/architecture`, the export emits a `Status` column and a `Topics` column. Cells contain the leaf path under that root; multiple categories under the same root join with `;` (matching the multi-tag import format).
- **One column per number key.** Every distinct key seen in any item's `Numbers` dict becomes a column. Missing values render as empty cells.
- **One column per typed category with values.** Header shape is `<path> [number|date|text]` - e.g. `frequency [number]`, `heard [date]`, `sio [text]`. Number cells use invariant culture (`14250.5`); Date cells use ISO `yyyy-MM-dd`; Text cells pass through as-is. Typed categories with no values on any item are skipped (no empty column).

The output is UTF-8 CSV with `,` delimiter and RFC-4180 quoting (cells containing commas, quotes, or newlines are double-quoted; embedded quotes are doubled).

Round-trip behaviour

Export → re-import produces an *equivalent* database, not a byte-identical one:

- **Preserved:** item Text, Due, IsComplete, Categories (kebab-cased on the way back in), Numbers values, typed-category Value type (Number / Date / Text - reconstructed from the `[type]` header suffix), per-item typed values.

- **Not preserved:** item Ids, Created/Modified timestamps, item ordering inside categories, rules, views, palettes, window placements, schema version stamping.

If you want a lossless backup of a Facet database, use **File → Export…** (JSON), not the CSV path.

Deep hierarchies

Categories deeper than two segments (e.g. `projects/launch/q3`) export the inner slash as part of the cell value (`launch/q3` under the `projects` column). On re-import each slash-separated piece is kebab-cased on its own, so `projects/launch/q3` comes back with its full parent/child structure intact - the intermediate `projects/launch` category is recreated too. Hierarchies of any depth now round-trip cleanly.

Publish and share

A Facet database is more than its `.json` file: it may have an **Explorer capture inbox** sitting beside it (queued "Add to Facet" items waiting to be ingested) and, if **Bundle images** is on, a `<database>-images` folder of pictures its notes point at. A **Facet package** - a single `.facpub` file - carries all of that as one unit.

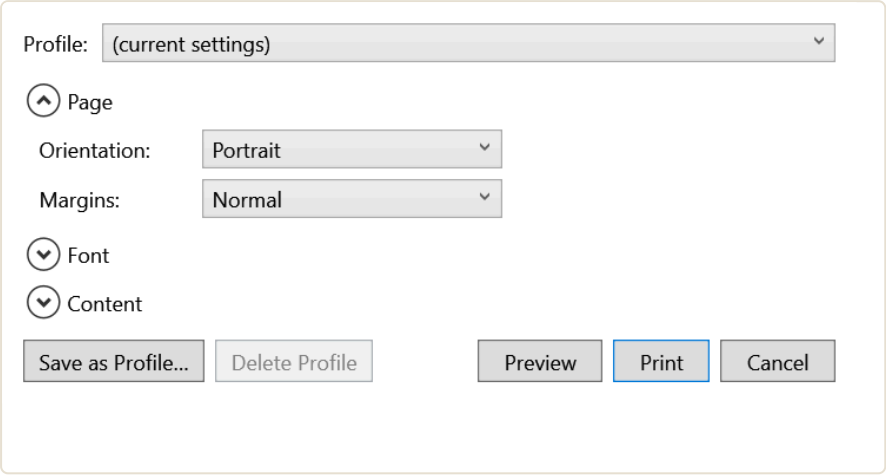
- **File → Publish as Facet package…** saves the open database first, then packs the database file, its inbox folder, and its images folder into a single `.facpub`. Pick the file name and location; that's it.
- **File → Create from package…** does the reverse. Choose a `.facpub` (made this way, or by an older build as a `.zip`), then choose where to save the new database and what to **name** it. Facet unpacks the database, its `<name>-images` folder and its inbox, **renames** everything to the name you gave, and fixes up the relative image references in your notes so the bundled pictures still show. The new database opens automatically. Facet won't overwrite an existing database of the same name - pick a different name or folder if it warns you.

This pairs naturally with **Bundle images**: bundle first so your pictures live beside the database, and a published package is then fully self-contained - it will open with every image intact on another machine.

Sharing with Facet Reader. A `.facpub` is the file you hand to someone who doesn't have Facet. **Facet Reader** is a free, separate download that opens a package **read-only**: the recipient can browse it, search it, sort and filter, use your views, and read every item - but cannot change the content, views, categories, rules or automations, and never alters the file you sent. They just double-click the `.facpub` (or use **File → Open Package** in Reader). The full Facet editor opens the same packages too, so on your own machine a double-click opens the package in the editor.

§ 10. Printing

Every print command opens the **Print Options** dialog first, where you choose how the output looks, then click **Print** or **Preview**.



Under **File → Print**:

- **Print Current View…** (**Ctrl+P**) - print the active view's groups and items. All layouts are supported (date / category / matrix / calendar / flat); the output honours the view's filter, sort, and search. A **columnar (Table)** view prints as a real table - every column reaches the page, with a TOTAL row beneath any Number columns - rather than a flat list of headings.
- **Print Selected Item…** - print the selected item's heading, dates, tags, links, and rendered Markdown notes.
- **Print Templates…** - build and print a saved document that combines several views with your own Markdown text (see *Print templates* below).

Choosing **Preview** in the dialog opens a paginated viewer (page navigation, zoom, search, Print toolbar) rendered through XPS, so it matches the printer output exactly.

Print Options

The dialog groups its settings in collapsible sections:

- **Page** - orientation (Portrait / Landscape) and margins (Normal / Narrow / Wide).
- **Font** - the body font family and size.
- **Content** - whether to include a running header (the view or item name on every page), page numbers in the footer, the metadata block (a view's group/sort line, or an item's dates and tags), and - for item prints - the notes body.

Your last-used settings are remembered for the next print.

Printing profiles

Set the options the way you like them and click **Save as Profile…** to store them under a name. The **Profile** dropdown at the top of the dialog recalls a saved profile in one click; **Delete Profile** removes the selected one. Profiles are stored with the database, so each database can carry its own set.

Print templates

A **print template** combines several views - and your own Markdown text - into a single saved document. Open **File** → **Print** → **Print Templates…** to build one.

- The left pane lists your saved templates; **New** and **Delete** manage the list.
- A template is an ordered list of **sections**. Use **+ View** to add a section that prints a view's items, or **+ Text block** to add a Markdown heading or paragraph. Reorder with the ↑ / ↓ buttons; drop a section with ✕.
- Select a section to edit it: a View section offers a dropdown of your views; a Text block opens a Markdown editor (`# Heading`, paragraphs, lists - the same syntax as item notes).
- **Format with** picks the Printing Profile that supplies the fonts, margins and page options for the whole document.
- **Preview** and **Print** render every section in order. Each view starts on a fresh page unless it directly follows a text block - so a heading and the view it introduces stay together.

Print templates are stored with the database.

---

§ 11. Reminders

Facet can remind you about items that need attention. When the database has **reminders** switched on, a **Reminders** window appears at startup - and on the first check after midnight if you leave Facet running - listing every active item that is **due today** or **overdue**, soonest first. Overdue items are shown in red. **Double-click** a row to jump to that item.

- **Tools → Reminders…** opens the window any time, whether or not anything is due.
- Reminders are a per-database setting: toggle **Show due-date reminders** on the database's **Properties** page (see *Databases*). It's on by default.
- Completing an item, or clearing its due date, drops it from the list at the next check.

---

§ 12. Undo / Redo

Every mutation in Facet is undoable.

- **Ctrl+Z** - undo
- **Ctrl+Y** or **Ctrl+Shift+Z** - redo
- Or **Edit → Undo / Redo**

The Edit menu shows live labels (e.g. *Undo - Rename category*). Multi-step operations (an import, a re-categorise-all, a drag that triggers rule side-effects) are wrapped as a single step.

Text edits use focus-based capture, so tabbing through a field without changing anything won't push a step onto the stack.

The depth cap is per-database (default 50) and configurable in Settings.

---

§ 13. Help and the View Manager

Several windows are reachable from the **Help** and **Views** menus.

- **Help → User Guide…** (**F1**) opens this guide in a window with a click-to-scroll table of contents on the left. Use **Ctrl+F** inside the content pane to search the text.
- **Help → Keyboard Shortcuts…** (**Ctrl+/**) opens a static cheat-sheet of every shortcut, grouped by area.
- **Help → Markup (Markdown) Syntax…** opens the reference for the Markdown you can use in item notes and print-template text blocks - the same cheat-sheet as the **Help** button in the notes editor.
- **Help → About Facet…** shows the version and licence information.
- **Views → Manage Views…** (**F8**) opens the **View Manager** - a list of every view, with **Open / Close / Edit / Rename / Delete / New** on one screen. Double-click a row to open or bring the view to the front, and **drag a row up or down to reorder the list** - the order you set is saved and is the same order the **Views** menu uses. Handy once you've accumulated more than a handful of views and the Views menu starts to feel crowded.

---

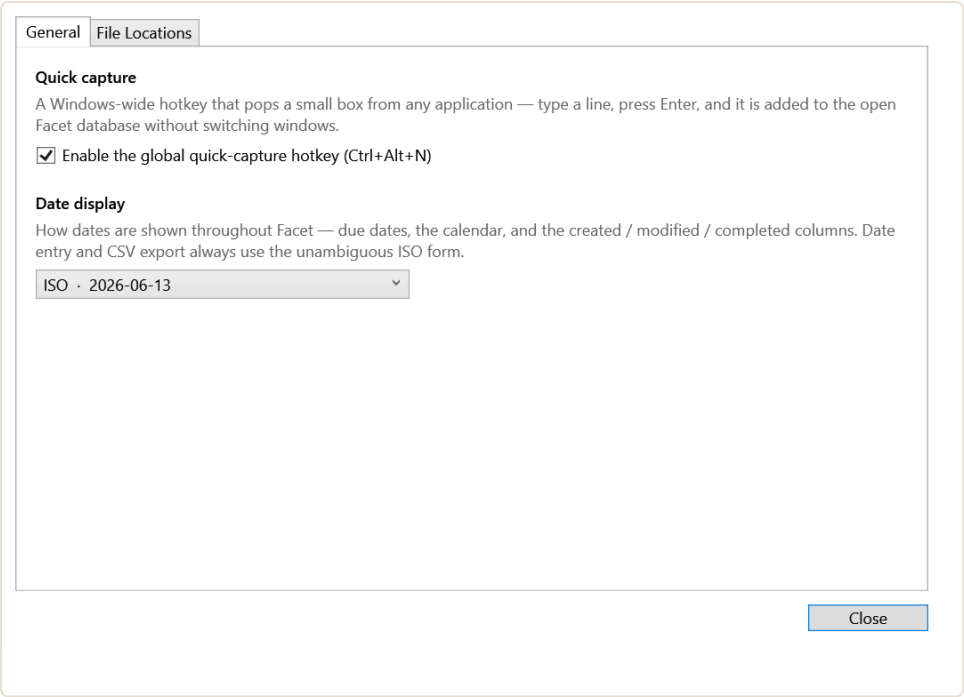
§ 14. Keyboard shortcuts

Shortcut	Action
Enter (in quick-entry)	Add the typed item
Ctrl+Alt+N (global)	Open the Quick Capture box from any application
Ctrl+S	Save now
Ctrl+Z	Undo
Ctrl+Y / Ctrl+Shift+Z	Redo
Ctrl+F	Open / focus the global Find pane
Ctrl+P	Print current view (the dialog also has a Preview button)
Ctrl+D	Toggle completion (done / not done) on the selected item
Delete	Delete the selected item (when no text field is focused)
Esc (in view search)	Clear the per-view search box
F1	Open the user guide
F8	Open the View Manager
Ctrl+/	Show the keyboard shortcuts cheat-sheet
F2 (on a columnar cell)	Edit the cell in place
Ins (on a Category / Tags cell)	Add another tag without replacing existing ones
Delete (on a Category / Tags / When / Number cell)	Clear just that cell
Delete (on a Task cell)	Delete the whole item
Enter / Tab (in a cell editor)	Commit the edit
Esc (in a cell editor)	Cancel the edit
Tab / Shift+Tab (in the Notes box)	Indent / outdent the line or selected lines

Shortcut	Action
Ctrl+Tab (in the Notes box)	Move on to the next field

§ 15. Settings

File → Facet Settings… is a tabbed dialog of **application-wide** options:



- **General** - three application-wide preferences:
  - The **quick-capture hotkey** toggle (see *Adding items*).
  - The **date display** format, which controls how dates are shown across Facet - ISO (`2026-05-22`), day-month-year (`22nd May 2026`), month-day-year (`May 22, 2026`), or your Windows locale's short date. Date entry and CSV export always use the unambiguous ISO form.
  - The **Appearance / theme**: **Light** - the modern Fluent styling shown throughout this guide - or **Classic**, the original, plainer Windows look. Switching *to or from* Classic asks to restart Facet, because it swaps out the whole control style set; the choice is remembered between sessions.
- **File Locations** - the database folder, the backup folder, and the list of databases included in the daily backup (with **Back up now** and **Open backup folder**). See *Daily backups* under *Databases*.

Settings that apply only to the open database - undo depth, completion tracking, reminders, the desktop background, and the colour **palettes** - live on its **Properties** page instead (**File → Properties…**, see *Databases*).

Settings are stored under `%AppData%\Facet`. A one-time migration moves legacy `%AppData%\Agenda` content over on first run.

---

§ 16. Licensing and activation

Facet is licensed software: it needs a valid **licence key** to run. You buy a key once from getfacet.co.uk (checkout is handled by Polar), activate it on each PC you use, and it's yours to keep.

Activating Facet

The first time you open Facet on a machine, the **Activate Facet** dialog appears. Paste the licence key from your purchase email into the box and click **Activate**. Facet registers this PC against your key and remembers it - you won't be asked again on this machine unless the licence changes. Each PC is registered under its **Windows machine name**, so you can recognise it later in your Polar customer portal.

Your key allows a set number of devices; each PC you activate uses one slot.

Working offline

Facet doesn't need a connection every time it starts. After a successful activation it re-checks your licence quietly in the background, but if you're offline it keeps working for up to **21 days** since the last successful check. Connect to the internet within that window and the clock resets automatically.

Moving to another PC

Each activation uses one device slot on your key. To move Facet to a new machine, sign in to your **Polar customer portal**, deactivate the old device to free its slot, then activate Facet on the new PC with the same key. The portal is also where you see which devices a key is on and download invoices.

If activation fails

The dialog tells you why:

Message	What it means
That key isn't recognised	A typo, or the key is for a different product. Copy it straight from your licence email and try again.
This key has been revoked	The licence was cancelled or refunded. Contact support if that's unexpected.
This key has expired	The licence's validity period has ended.
The activation limit on this key has been reached	Every device slot is in use - deactivate one in your Polar customer portal first.
Couldn't reach the licensing server	No connection, or the server is briefly unreachable. Try again when you're online.

Where the licence is stored

Facet keeps your activation in a small `license.json` file. The installed build stores it under `%LocalAppData%\Facet`; the **portable** build keeps it beside the program, so a USB-stick copy stays self-contained.

Help → About Facet… shows your current licence and version information at any time (see *Help and the View Manager*).

---

§ 17. Tips for testing

- Start with **Create New Database…** → **Include sample data** to get a populated workspace to poke at.
- Try entering an item like `Call Fred next Tuesday about pricing` and watch which categories get auto-attached (open **Categories** → **Rules…** to see why).
- Open the **same items** through three different views - a date-bucket view, a category-grouped view, and a calendar view - and edit through each in turn. The whole point of Facet is that they're all the same data.
- After editing rules or categories, hit **Re-categorise all items…** to reapply the new logic across the existing items.
- Undo aggressively. The engine is built to make experimentation cheap.

---

§ Recipes - worked examples

Three short end-to-end walkthroughs that combine the features above. Each builds on a starter template you can create from **File → Create New Database…**, so you can follow along with real data.

Recipe 1 - A personal task tracker

Goal: capture tasks as plain sentences and have Facet sort them by person, project, and what kind of action they are.

1. Create a database from the **Activities Planner** template. It arrives with *actions*, *people*, and *projects* categories, a priority pick-one group, and a columnar **Tasks** view.
2. Capture a few tasks the way you would say them out loud - *"Call Fred Tuesday about the budget"*, *"Email Wendy the launch timeline"*, *"Review Terry's proposal this afternoon"*. Watch the **Activity**, **People**, and **Project** columns fill themselves in: the rules behind the template read the words "call", "email", "review", and the names, and tag each item accordingly.
3. Open the **Calls** view to see only the calls, the **People** view to see work grouped by who it involves, and **This Week** to see what is due soon - all the same items, three lenses.
4. Add a priority by hand: open an item's Detail panel and add the *priorities/high* chip. Because the priorities are a **pick-one group**, adding *high* to an item that was *medium* swaps them - an item can only carry one priority at a time.
5. Want overdue items to stand out automatically? Add a **rule** (section 6): condition *DueDate is before today*, action *Assign category actions/urgent*. Run **Re-categorise all items…** and every late task is flagged.

Recipe 2 - Tracking contacts and accounts

Goal: keep sales or support activity organised by account, contact, product, and issue.

1. Create a database from the **Account Manager** template. Its Tasks view has columns for Contact, Account, Activity, Product, and Issues.
2. Capture activity as it happens - *"Call Tony at Big Co about Spire pricing"*. The template's rules tag the account, contact, product, and activity from the sentence.
3. Open **By Account** to roll every item up under its account, or **Issues** to triage by problem type. The issue categories ship with a red chip colour so they catch the eye.
4. Use **global Find** (**Ctrl+F**) to pull every mention of a product across all accounts, then **Save as view…** to keep that search as a permanent view.

Recipe 3 - A structured log with per-item values (Radio Log)

Goal: record entries that each carry numbers and codes, not just a heading - and total or group by those values.

1. Create a database from the **Radio Log** template. It uses **typed categories** (section 5): a Date for when each contact was logged, a Number for the frequency, and Text for the operator, grid square, and signal report.
2. Add a contact and fill in its values in the Detail panel's **Category values** expander, or inline in the columnar view's cells.
3. Group the log **by frequency band** using the *Group by category value* layout with a Number grain, or sort the table by any typed column by clicking its header.
4. Note that the frequency column does **not** show a TOTAL - the template ships with **Show TOTAL row** switched off for it, because summing frequencies is meaningless. Switch it on for a column that *should* sum (pledges, hours, amounts) and the footer appears.
5. When you're done, **Export to CSV…** - the typed columns round-trip through the `frequency [number]`, `heard [date]`, `sio [text]` header convention, so you can re-import the log later without losing the value types.

---

§ Frequently asked questions

Where does my data live? Each database is a single `.json` file. The default sits under `%AppData%\Facet`; **File → Properties…** shows the exact path for the open database. Daily backups are written to a `Facetbackups` folder alongside it, one sub-folder per database. There is also an `inbox` folder and a further folder for graphical images. Further details of the Facet file structure can be found in the **Facet Developer Notes** on the website.

Are my other databases backed up too, or only the one I'm using? All of them. Once a day Facet backs up every database in your database folder and your recent list, open or not, as a dated `.facpub` package that includes each database's images and inbox. **File → Facet Settings… → File Locations** lists them, shows when each was last backed up, and lets you leave any of them out. See *Daily backups* under *Databases*.

Do I have to save? No. Facet autosaves a moment after every change. **Ctrl+S** forces an immediate save if you want to be sure before closing.

I put an item in the wrong place - how do I move it? Items aren't "in" a place; they appear wherever a view's filter brings them up. To change how an item is classified, change its **tags** - drag it between groups in a view, or edit the chips in its Detail panel. To change *when* it's due, drag it to another date bucket or calendar day, or edit the **Due** field.

Why did a category attach itself to my item? A **rule** or a category **match rule** matched the item's text. Open **Categories → Rules…** or the category's match rule in **Categories → Manage…** to see the pattern. Tags you add by hand are never removed by rules - only rule-assigned tags are recomputed when you edit an item.

Can one item be in several views at once? Yes - that is the whole point. A view is a lens, not a folder. The same item can appear in a date view, a category view, and a calendar simultaneously, and an edit through any one of them updates all of them.

How do I undo something? Ctrl+Z. Even big operations - an import, a re-categorise-all, a bulk action across many selected items - count as a single undo step. **Ctrl+Y** (or **Ctrl+Shift+Z**) redoes. If you **really** want to be sure, save a copy of the database under a new name before you do major changes.

What's the difference between a rule and an automation? A **rule** answers "what categories does this item have?" and runs automatically as items change. An **automation** answers "when X, do Y" - it can create items, open views, set dates, branch, loop, and prompt you, and you trigger it by hand, by shortcut, on a schedule, or on an event. See sections 6 and 7.

I want to try a bulk or deleting automation safely first. Use **Test run** in the Automations manager. It reports everything the automation *would* do - every tag, date, new item, and deletion - without changing anything. It puts the process that will be followed into a dialogue box for you to inspect.

Can I move data in and out of Facet? Yes. **New Database from CSV…** builds a database from a spreadsheet; **Export to CSV…** writes one back out in the same shape. For a lossless backup, use **Export…** (JSON) rather than CSV. **Import…** merges another Facet database into the open one.

How do I get help inside the app? Press **F1** for this guide, **Ctrl+/** for the keyboard cheat-sheet, and look for the **Help** button in the notes editor for the Markdown syntax reference.