

# Facet — Cookbook

*Recipes for the things just outside the manual*

*A growing collection of worked recipes for Facet — the modern thinking workspace for Windows. Where the User Guide explains what each feature does, the Cookbook shows how to combine them to solve a real problem.*

The [User Guide](#) is the reference: it walks through every menu, dialog, and setting in turn. This Cookbook is the companion volume. It assumes you already know your way around the basics — capturing items, views, categories, rules, and automations — and instead shows how to wire those pieces together into something more than the sum of its parts.

Each recipe is self-contained. It states a **goal**, lists the **ingredients** (the Facet features it leans on), and gives **steps** you can follow with a real database. Most build on a starter template you can create from **File** → **Create New Database...**, so you can try them without inventing data first.

Nothing here is a hidden or unsupported feature — every recipe uses the app exactly as the User Guide describes. The value is in the *combination*: the patterns that aren't obvious from reading any single section on its own.

***This document grows.** New recipes are added as the app is tested and as common questions come up. If a recipe you want isn't here yet, the building blocks almost certainly are — check the User Guide section named in each recipe's Ingredients line.*

## § 1. A live "Today" dashboard that never goes stale

*Goal: one view that always shows exactly what's due today and what's overdue — without you ever re-filtering it.*

**Ingredients:** Views → Advanced filter (Due date condition); the *today* keyword evaluated live (User Guide §4).

The **Due date buckets** layout is the quick way to split work into Today / This week / Later. But sometimes you want a single, tight list — just today and anything late — that you can leave open all day and trust.

1. **Views** → **New View...**, name it *Today*, and pick the **Flat list** layout.
2. Open the **Advanced filter** section and add one **Due date** condition: set the operator to **on or before** and the date to **today**.
3. Add **Items to include** → **Only active** in the Sort and Simple Filter section so completed work drops away.
4. Set the sort to **Due date** so the most overdue items rise to the top.

Because *today* is re-evaluated every time the view refreshes, this list is correct tomorrow, next week, and next month — you never touch it again. Leave it open in a corner of the workspace as your standing to-do list.

*Variation — a rolling week. Use the operator **within / 7 / days of / today** instead. Now the view shows everything due in the next seven days, and the window slides forward on its own each day.*

## § 2. A Kanban board from a pick-one group

*Goal: drag items between status columns — To Do, Doing, Done — the way you would on a sticky-note board.*

**Ingredients:** a pick-one category group (User Guide §5); the Matrix layout (§4); drag-to-retag (§4).

Facet has no dedicated "board" view, but a **Matrix** over a **pick-one group** behaves like one, and it takes about two minutes to build.

1. **Categories** → **Manage...** Add a parent category `status`, then three children: `status/todo`, `status/doing`, `status/done`.
2. Select each child and give them all the **same Pick-one group** name — say `workflow`. The shared name means an item can only ever carry one of the three at a time, so moving it to *Doing* automatically clears *To Do*.
3. Give each a distinct **chip colour** (grey, amber, green works well).
4. **Views** → **New View...**, choose the **Matrix** layout, and point it at the `status` parent. You get one column per status and one row per item.
5. **Click a cell** to set that item's status. Because the categories are mutually exclusive, ticking *Doing* unticks whatever it had before — the item "moves" across the board.

To capture straight onto the board, add a rule (**Categories** → **Rules...**) that assigns `status/todo` to any new item, so everything lands in the first column ready to be moved along.

---

### § 3. Tickle every overdue task into a "Waiting" state

*Goal: each morning, sweep anything that has slipped past its due date into a visible "needs attention" category — in one click, or automatically.*

**Ingredients:** an automation with an *items matching...* target (User Guide §7); a scheduled trigger; Test run.

1. **Tools** → **Automations...**, **New**, name it *Flag overdue*.
2. Add an **Assign category** action. Change its **target** from *the item in context* to **items matching...**, and in the condition editor below it add a **Due date** condition: operator **before**, date **today**. Set the category to assign as `status/waiting` (create it if you need to).
3. Before trusting it, click **Test run**. Facet reports every item it *would* tag, in a dialog, changing nothing — confirm the list looks right.
4. Set the **Trigger** to **On a schedule** → **daily** at, say, 08:00. A scheduled automation catches up if Facet was closed at that time, so opening the app at 09:00 still runs the 08:00 sweep.

Now build a **Today** dashboard (Recipe 1) or a category-grouped view that surfaces `status/waiting`, and every late item announces itself the moment you sit down.

*The whole sweep is a **single undo step** — one **Ctrl+Z** puts every tag back if you don't like the result.*

## § 4. A weekly-review launcher

*Goal: press one shortcut and have Facet lay out the exact set of views you review every Friday — front and centre.*

**Ingredients:** an automation with several **Open view** actions (User Guide §7); a keyboard-shortcut trigger; the *centre on screen* option.

1. Build the views you want for a review — perhaps *Today, This Week, By Project, and Waiting*.
2. **Tools** → **Automations...**, **New**, name it *Weekly review*.
3. Add one **Open view** action per view, in the order you want them stacked. Tick **centre on screen** on the last one so Facet comes to the foreground with the review layout ready, even if you were in another application.
4. Set the **Trigger** to **Keyboard shortcut** and press your combination (e.g. **Ctrl+Shift+R**).

One keystroke now opens your whole review board. Add a **Show message** action at the end — *“Review: clear Waiting, reschedule overdue, pick three for Monday”* — to give yourself the checklist alongside the views.

---

## § 5. One item, many independent numbers

*Goal: track different numeric fields on the same items in different views — say Pledges in a fundraising view and Hours in a time view — without them colliding.*

**Ingredients:** Number columns keyed by header (User Guide §4); the TOTAL footer.

A Number column's **Header** doubles as the data key, so two columns with different headers are two genuinely separate fields on the same item.

1. In a columnar view, add a **Number** column with Header `Pledges` and a currency prefix `£`. The TOTAL footer sums it.
2. In a *different* columnar view over the same items, add a **Number** column with Header `Hours`. It is a wholly separate value — editing one never touches the other.
3. Because each item carries a small dictionary of named numbers, you can keep adding fields ( `Miles` , `Score` , `Weight` ) view by view, each with its own total.

***Watch the header.** The header is the key. Rename a Number column after you have typed values into it and the old values go missing — they're still keyed under the old name. Decide the header first, then enter data.*

## § 6. Capture templates with placeholders

*Goal: stamp the date (or other run-time values) into items as you type them, without reaching for the Detail panel.*

**Ingredients:** `{name}` placeholders in quick entry (User Guide §3 and §7).

The quick-entry box, the Quick Capture hotkey, and every per-group + **Add item** box all expand `{name}` placeholders as the item is created.

- Type `Standup notes {today}` and the item is born with today's date baked into its heading.
- `{now}`, `{tomorrow}`, `{yesterday}`, and `{datetime}` work the same way.
- An unrecognised `{name}` is left exactly as written — so `{budget}` stays literal if you ever want it to.

For something you log every day, pair this with a manual or scheduled **automation** that runs a **Create item** action containing the template text — `Daily log {today}` — so the skeleton item appears on its own.

---

## § How to read a recipe

Every recipe in this Cookbook follows the same shape:

- **Goal** — the real-world outcome, in one sentence.
- **Ingredients** — the Facet features involved, with the User Guide section to consult if a step is unfamiliar.
- **Steps** — numbered, in order, written so you can follow along in a live database.
- **Variations and notes** — quoted asides that extend or warn.

When a recipe and the User Guide seem to disagree, the User Guide wins — it tracks the current build. Tell us at [getfacet.co.uk](http://getfacet.co.uk) if you spot a drift.