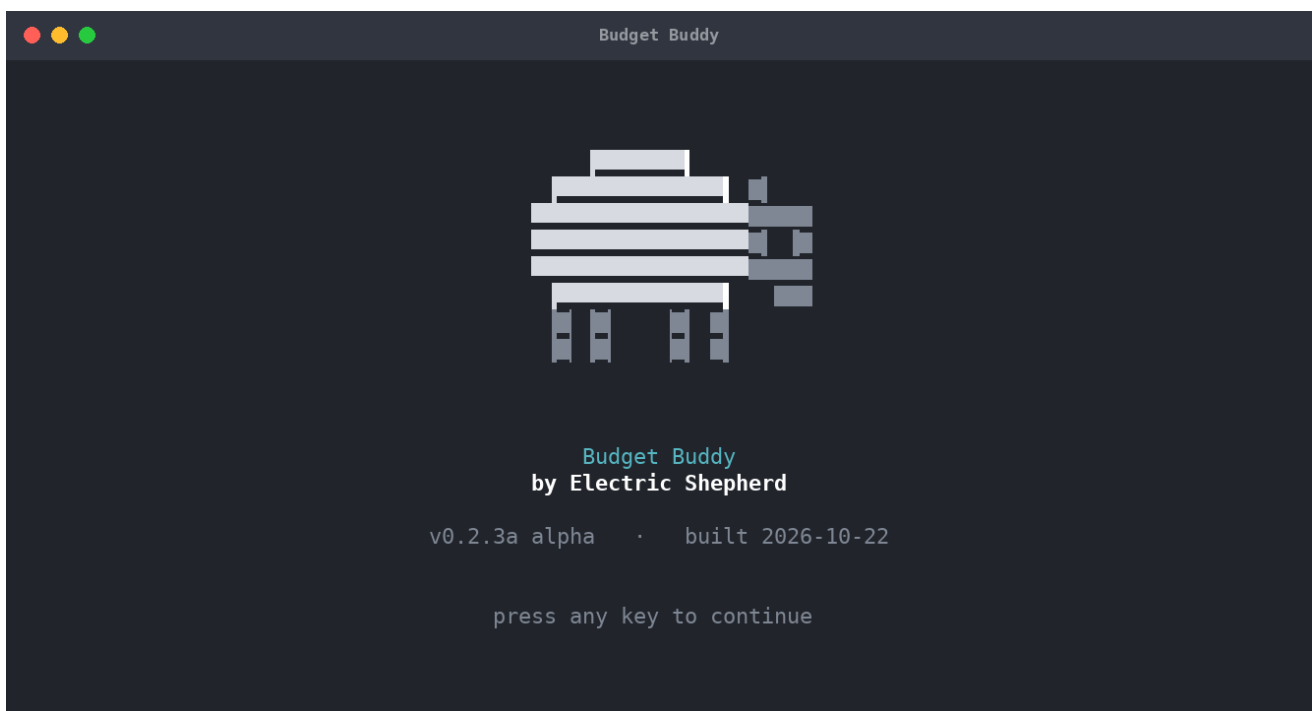


Budget Buddy

User Manual

Version 0.2.3a (alpha) · Built 2026-10-22



Electric Shepherd

A budget planner for your terminal

Contents

1. Welcome to Budget Buddy	6
1.1 What is in version 0.2.3a	6
1.2 What is new in version 0.2.3a	7
1.3 What you need	7
1.4 Installing and starting	8
1.5 The splash screen	8
1.6 About the version number	9
1.7 Where your data lives	9
1.8 Conventions in this manual	10
2. A Tour of the Screen	11
2.1 The sidebar layout	11
2.2 The page: views, sorting and the footer	13
2.3 The top-bar layout	14
2.4 Choosing a size	17
2.5 The pages	18
2.6 Views inside a page	19
2.7 Colours and symbols	19
2.8 The clock	21
2.9 Resizing	22
3. Keys, Forms and Menus	23
3.1 Moving around	23
3.2 The same keys on every list	24
3.3 Forms	25
3.4 Dialogs	28
3.5 The mouse	28
3.6 Messages	29
3.7 Quitting	29
4. Quick Start: Your First Budget	30
4.1 Start with an empty program	30
4.2 Add your accounts (page 7)	30
4.3 Add your income (page 2)	32
4.4 Add your bills (page 3)	33
4.5 Add your debts (page 4)	35
4.6 Add a savings goal (page 6)	36
4.7 Read the Plan (page 1)	37
5. The Plan	39
5.1 Needs attention	40
5.2 Where every dollar goes	40
5.3 Spending by category	43
5.4 Getting out of debt	43
5.5 Savings goals	43
5.6 Notes	43
5.7 Other months	44
6. Income	45
6.1 Adding and editing income	46
6.2 How income is counted	46
6.3 Deleting income	47
7. Expenses: Bills, Daily Spending and Allotments	48
7.1 Bills	48
7.2 Marking a bill paid	49
7.3 Deleting and fixing mistakes	50
7.4 Setting money aside for a bill	50

7.5 Daily spending	50
7.6 Allotments	53
8. Debts	56
8.1 Adding and editing a debt	57
8.2 Recording a payment	57
8.3 Setting money aside	58
8.4 Strategy	58
9. Payoff	59
9.1 Scenarios	59
9.2 Each debt	59
9.3 The schedule	60
9.4 What if I paid more?	61
9.5 Choosing a strategy	62
10. Savings Goals	63
10.1 Adding and editing a goal	63
10.2 Adding money to a goal	64
10.3 The emergency fund	65
11. Accounts	66
11.1 Why balances change through entries	67
11.2 Adding an account	67
11.3 Setting a balance to match your bank	68
11.4 Deposits and withdrawals	68
11.5 Transfers	69
11.6 Undoing and deleting	69
12. Account History	70
12.1 Full screen and all accounts	71
12.2 What the totals line adds up	72
12.3 Dates, balances and the order of entries	72
12.4 Sorting	72
12.5 Deleting entries	73
12.6 Filtering the history	74
12.7 Showing the year	76
13. Importing a Bank CSV	77
13.1 Getting the file	77
13.2 Starting an import	77
13.3 The preview	77
13.4 How the balance is handled	78
13.5 After the import	80
13.6 Importing the same file twice	80
13.7 Backing out an import	80
13.8 What files can be read	80
14. Exporting History to CSV	82
14.1 The file	82
15. Cash Flow	84
15.1 When you will run short	85
15.2 What it does and does not include	85
16. Projections	86
16.1 By paycheck	86
16.2 By bill	87
17. Settings	89
17.1 The program's own settings	89
17.2 The settings file	93
17.3 The rules of the Plan	93

18. Sorting and Searching	95
18.1 The current column	95
18.2 Sorting	95
18.3 Searching	96
18.4 Good to know	97
19. How the Numbers Work	98
19.1 Monthly equivalents	98
19.2 The allocation order	98
19.3 Debt interest and payoff	98
19.4 Dates and schedules	98
19.5 Set-asides and the forecast	99
19.6 Allotments	99
20. Command Line and Environment	100
20.1 Options	100
20.2 Writing the options	101
20.3 Exit status	101
20.4 Environment variables	102
20.5 Printing a screen with --dump	103
21. Your Data File and Backups	104
21.1 Where it is and how it is saved	104
21.2 What is in it	104
21.3 The settings file	105
21.4 Compatibility	105
21.5 Backing up and moving	105
21.6 Starting over	106
22. Troubleshooting	107
22.1 The crash log	109
23. Limits and Known Issues	110
23.1 By design	110
23.2 Known rough edges	110
23.3 Reporting a problem	111
Appendix A. Key Reference	112
A.1 Everywhere	112
A.2 Income, Expenses, Debts, Savings, Accounts	112
A.3 Plan	113
A.4 Expenses	113
A.5 Debts, Payoff	113
A.6 Savings and Accounts	114
A.7 Account history	114
A.8 Import preview	115
A.9 Cash Flow, Projections, Settings	115
A.10 Forms, menus and confirmations	115
A.11 Typing dates and money	116
Appendix B. Categories and Types	117
B.1 Expense categories	117
B.2 Income types	117
B.3 Debt types	118
B.4 Savings goal types	118
B.5 Account types	118
B.6 Repeat frequencies	118
Appendix C. Input Syntax	119
C.1 Dates (in forms)	119
C.2 Money and percentages	119
C.3 History filter	119

Appendix D. Glossary	120
Appendix E. Release Notes	121
E.1 Version 0.2.3a (alpha)	121
E.2 Version 0.2.2a (alpha)	123
E.3 Version 0.2.1a (alpha)	124
E.4 Version 0.2a (alpha)	124
E.5 Version 0.1.1a (alpha)	125
E.6 Version 0.1a (alpha)	125
E.7 Known issues	126
Appendix F. Keys That Moved in 0.2.3a	127
F.1 Keys that changed	127
F.2 What did not change	128
F.3 New in this release	128

1. Welcome to Budget Buddy

Budget Buddy is a budget planner that lives in your terminal. You tell it about your bank accounts, your income, your bills, your debts and your savings goals. It then does the arithmetic most people dread: it splits every dollar of income between those things in a sensible order (essentials first, then debt, then savings), tells you **when** each bill needs money set aside, warns you before cash runs short, and shows how quickly your debts can disappear.

It is one small program. It has no dependencies, needs no network connection, and keeps everything in a single plain-text file that you can read, back up or edit by hand.

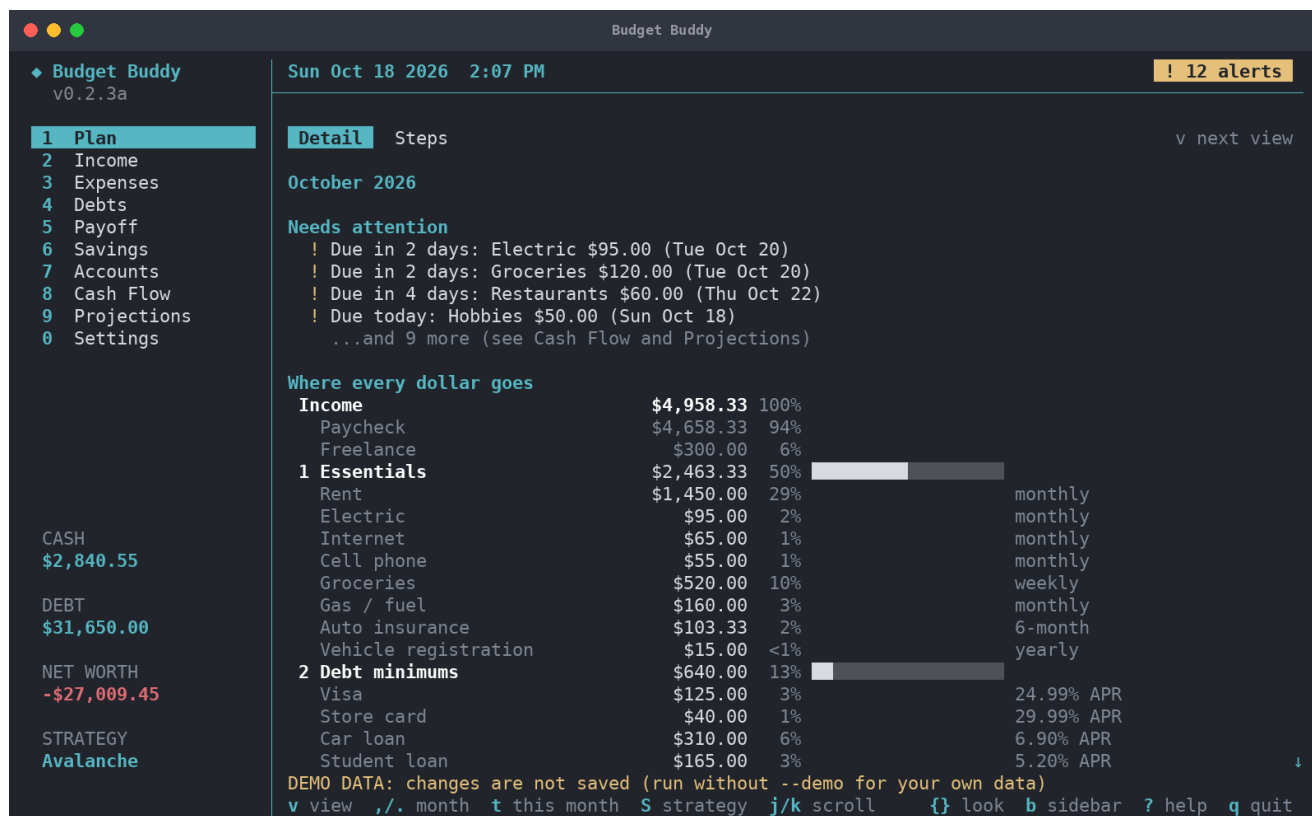


Figure 1. The Plan page, showing the sample budget that budget --demo loads.

1.1 What is in version 0.2.3a

Version 0.2.3a belongs to the second alpha series of Budget Buddy. Everything described in this manual is in the build you have. The program has ten pages and a handful of helper screens:

- **Plan:** the month at a glance. Alerts, where every dollar goes, spending by category, when you will be debt-free, and savings progress. It has two views, Detail and Steps.
- **Income, Expenses, Debts, Savings, Accounts:** the lists you maintain. They all share the same keys for adding, editing, cloning, deleting, sorting and searching.
- **Expenses** has five views: **Bills**, **Daily spending** (a ledger for coffee, groceries and gas), **By category**, **Allotments** (a monthly spending limit per category) and **Suggested** allotments.

- **Payoff:** compares paying only the minimums, following the plan, and a what-if with extra money each month, for all debts together or for each debt.
- **Cash Flow:** a day-by-day forecast of your spendable cash that calls out any shortfall.
- **Projections:** looks ahead. Today it tells you which bills to fund from which paycheck (“from Friday's paycheck, move \$725 for rent”), and the same by bill. It is where future views, such as a calendar, will live.
- **Account history:** every change to a balance, with sorting, powerful filtering and a live totals line.
- **Bank CSV import** with a preview before anything changes, and **CSV export** of your history.
- **Settings:** how the program looks and starts (colour look, clock, splash, start page, sidebar and mouse), and the rules the plan follows, changed through the same menus and dialogs as everything else.

1.2 What is new in version 0.2.3a

Budget Buddy now works like Inventory Central, the other Electric Shepherd program: if you use one, you already know the other.

- **The same keys.** One set of keys moves between pages and views, and one set sorts, searches and edits every list (Chapter 3). A few keys changed to make this possible, and Appendix F lists them.
- **A new screen.** In a window of 100 columns by 24 rows or more, a **sidebar** lists every page with its number key, keeps your cash, debt, net worth and strategy in view, and puts the date, the time and an alert badge above the page. Smaller windows keep a **top bar**, so an ordinary 80 by 24 terminal still gets its full-width page (Chapter 2).
- **Views.** A page with more than one way of being seen names them on a bar at the top, and  steps through them.
- **The mouse.** Click a page, a view, a row, a column heading, a key hint or a button (section 3.5).
- **Thirteen colour looks**, from the default Full color to the painted Winter, Harbor, Dusk, Shepherd, Spring, Autumn, Ember and Samhain, and the three monochrome looks (section 17.1).
- **A settings file** for the program's own settings, kept apart from your data (section 17.2), and a splash screen with the Electric Shepherd sheep (section 1.5).

The release notes in Appendix E have the full list, and the notes for earlier versions.

Note. Budget Buddy is a planning aid, not financial advice. Debt interest is modelled monthly from the APR, so real statements will differ by a few dollars. This is an alpha release: keep your own backup of the data file (section 1.7) until you are comfortable with it.

1.3 What you need

- A **Linux** computer (64-bit Intel or AMD, “x86_64”) and any terminal that understands ANSI escape codes (practically every modern terminal does). Other Unix-like systems such as macOS are expected to work but have not been tested; native Windows is not supported (use WSL). The program supplied with this release needs glibc 2.39 or newer, as found in Ubuntu 24.04, Debian 13 and Fedora 40 and later. A version for ARM (arm64) computers is not supplied with this release.

- A terminal at least **50 columns by 12 rows**. An ordinary **80 by 24** window is fully supported. At **100 columns by 24 rows** or more you get the sidebar layout, and about **160 columns by 34 rows** shows every column of every table.
- The standard Unix `stty` command (Budget Buddy uses it to control the terminal), `sh` and `date` (for your local date; it falls back to UTC without it). Every Linux and macOS system has all three.

1.4 Installing and starting

Budget Buddy is a single program, `budget`, supplied in a zip file named for the kind of computer it is for, such as `budget-0.2.3a-linux-x86_64.zip`. The zip also holds this manual as a PDF. Unpack it and run the program:

```
unzip budget-0.2.3a-linux-x86_64.zip
cd budget-0.2.3a-linux-x86_64
./budget --demo      # look around with sample data (nothing is saved)
./budget             # your own data
```

If your unzip tool leaves the program without its executable permission, run `chmod +x budget` once. To run Budget Buddy from anywhere, copy `budget` to a folder on your PATH (for example `~/local/bin`). The first time you run it without `--demo` it starts empty, ready for your own information (Chapter 4 walks through that).

Tip. Run `budget --demo` first. It loads a believable sample budget (two accounts, two incomes, twelve bills, four debts, two savings goals, fifteen recent purchases and a monthly allotment for each everyday category) so you can look through every page. Changes made in demo mode are never saved, and a reminder is shown on the message line.

1.5 The splash screen

Every time Budget Buddy starts it shows a splash screen: the Electric Shepherd sheep above the product name, the company, the version and the build date. It **waits for a key**: press any key to continue (the key is used up, so it does not also act on the first page). If the terminal is resized while it is showing, it redraws.

A window too small for the sheep gets a framed title instead, which you can also choose in Settings (the **Classic** splash), or you can turn the splash off there. To start without it for one run, use the `--no-splash` (or `-n`) option, or set `BUDGET_NO_SPLASH=1` in your environment (add it to your shell profile to turn it off for good). The splash only appears when the interactive program starts; `--dump`, `--help` and `--version` never show it (section 17.1).

The same product name and version are shown on **every** screen: at the top of the sidebar, or at the bottom right, on the line just above the key hints, in the top-bar layout (see Chapter 2). `budget --version` prints them as a single line:

```
$ budget --version
Budget Buddy v0.2.3a (alpha, built 2026-10-22)
```

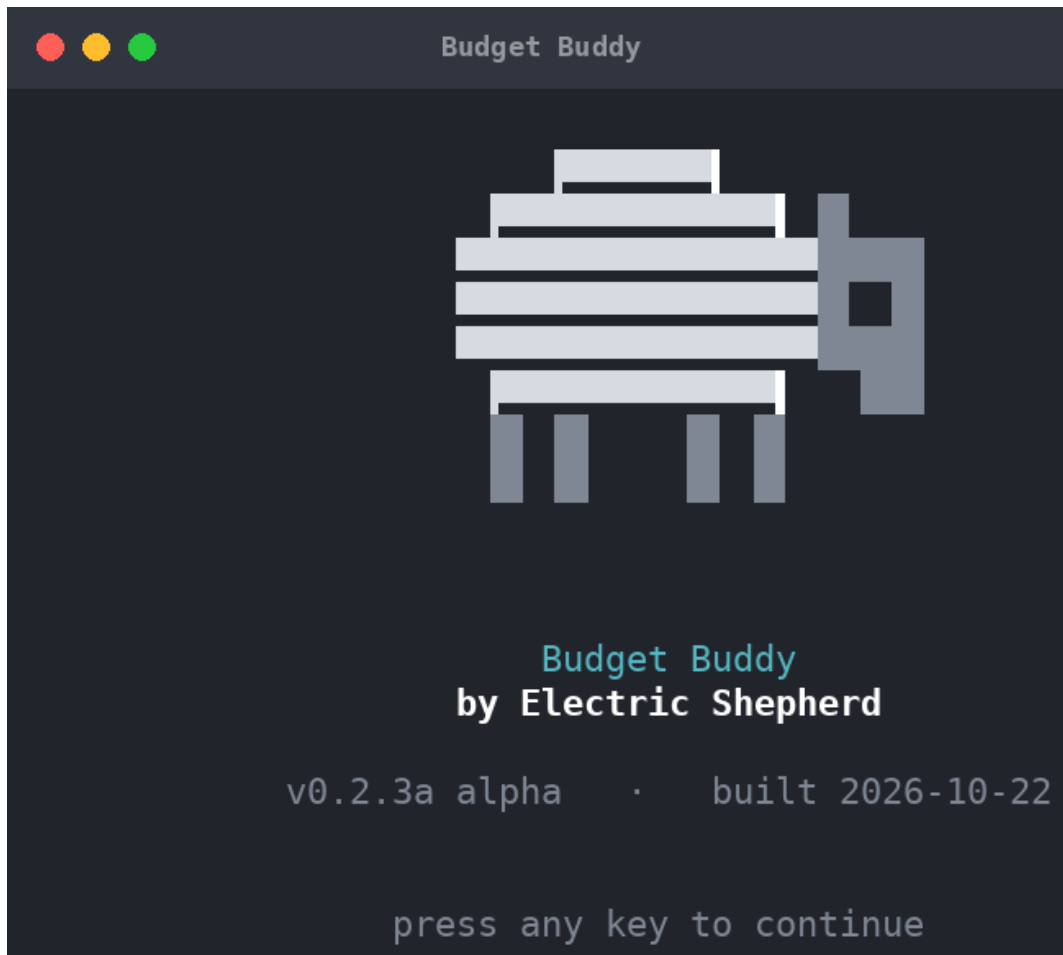


Figure 2. The start-up splash. The version (v0.2.3a, an alpha) and the date the program was built are shown below the name.

1.6 About the version number

Budget Buddy's short version reads v0.2.3a: major version 0, minor version 2, patch release 3, and the letter **a** for alpha (**b** will mean beta). A release with no patch number is written without it (v0.2a), and a final release has no letter (v1.0). The same version appears on the splash screen, on every screen and in budget --version, so you can always tell which build you are running. The build date is stamped in when the program is made, in UTC.

1.7 Where your data lives

Everything you enter is saved in one plain-text file after **every** change; there is no save command to remember. By default it is:

```
~/.local/share/budget-tui/budget.dat  
$XDG_DATA_HOME/budget-tui/budget.dat    (if XDG_DATA_HOME is set)
```

The folder is still called budget-tui, as in earlier builds, so data you already have keeps loading. You can use a different file with --file PATH or the BUDGET_FILE environment variable.

- **Crash-safe saves.** The file is written to a temporary file, flushed to disk, and then renamed over the old one, so a crash or power cut can never leave half a file.
- **An automatic backup.** The first save of each session keeps the previous file as `budget.dat.bak` beside it.
- **Private to you.** The data file and its backup can be read only by the account that owns them (permissions 600), and a folder the program has to create is private too (700).
- **One copy at a time.** While Budget Buddy is open, a second copy cannot open the same file.
- **Settings are kept apart.** How the program looks and starts (colour look, clock, splash, start page, sidebar, mouse) is in a small settings file of its own, `~/.config/budget-tui/settings` (section 17.2), so your money data is never touched by choosing a look.
- **A bad file is never overwritten.** If the file is damaged or is not a Budget Buddy file, the program refuses to start with a message and leaves it untouched.

Tip. To back up your budget, copy the `.dat` file. To move to another computer, copy it there and point `--file` at it (or put it in the default folder).

1.8 Conventions in this manual

- Keys to press are shown like `a`, `Enter`, `Esc` or `Ctrl-S`. Capital letters mean Shift, so `O` is different from `o`.
- Text in this typeface is something you type, a file name, or a command.
- Screenshots are real screens of the program, drawn at the window size the chapter describes. Most show the sidebar layout, in a window wide enough that every column of the table shows (often 120 columns); your terminal may show a slightly narrower or wider layout (Chapter 2). The sample budget in the screenshots is the one that `--demo` loads, and its date is Sunday, October 18, 2026, at 2:07 PM.
- Tip, Note and Careful boxes flag shortcuts, background and things that can lose or change data.

2. A Tour of the Screen

Every screen has the same frame, so once you have learned one you know them all. Budget Buddy chooses between two layouts to suit the size of your terminal, and re-chooses whenever you resize it. Both show the same information and take the same keys; only the arrangement differs. The frame, the keys and the way tables are laid out follow the other Electric Shepherd programs, so if you have used Inventory Central you will find yourself at home.

Layout	When you get it	What it looks like
Sidebar	The terminal is at least 100 columns by 24 rows and the sidebar is on (it is, unless you turn it off with b).	All ten pages are listed down the left, with four figures that stay in view. The date, time and alerts sit above the page.
Top bar	Anything narrower or shorter, down to 50 columns by 12 rows, or whenever the sidebar is off. A plain 80 by 24 terminal uses this layout and keeps its full-width page.	The page names run along the top, with one line of figures under them.

2.1 The sidebar layout

Figure 3 shows the Plan with the parts of the frame numbered.

#	Part	What it shows
1	Pages	The ten pages, each with its number key and its full name. The current page is a highlighted bar. Press the number, or Tab / Right and Shift-Tab / Left , to move between pages, or click one with the mouse (section 3.5).
2	Product and version	“Budget Buddy” and the version, at the top of the sidebar, on every screen.
3	Date and time	Today's date and, unless you hide it, the time of day (section 17.1).
4	Alert badge	“! 12 alerts” in yellow, or in red when something is overdue. It counts what the Plan lists under <i>Needs attention</i> (Chapter 5). When there is nothing to report there is no badge.
5	Views bar	The views of the current page, with the one you are on highlighted. Press v for the next view and v for the previous one. Pages that have only one view have no bar (section 2.2).
6	Page	The page itself: a document (Plan, Payoff, Cash Flow, Projections) or a table you can move around in. It takes all the room that is left, 74 columns at the smallest sidebar window. Dialogs open in the middle of it.
7	Four figures	Cash (your spendable accounts), Debt , Net worth and the debt Strategy (Avalanche or Snowball, Chapter 9), at the bottom of the sidebar on every page. A figure that does not exist yet is shown as a dash: Cash needs an account, Debt a debt. Cash is green (red if negative); Net worth is red when it is negative.
8	Message line	Confirmations (green) and problems (red) appear here after you do something, and clear on your next key. In demo mode a reminder that nothing is saved is shown here.
9	Key hints	The keys that work on this screen right now. They change with the page and with the dialog that is open.

Everything that colour says in this layout is also said another way: the current page is a reversed bar, the alert badge is a reversed pill, and the highlighted row of a table is reversed. That is why the program stays readable in every colour look (section 2.7).

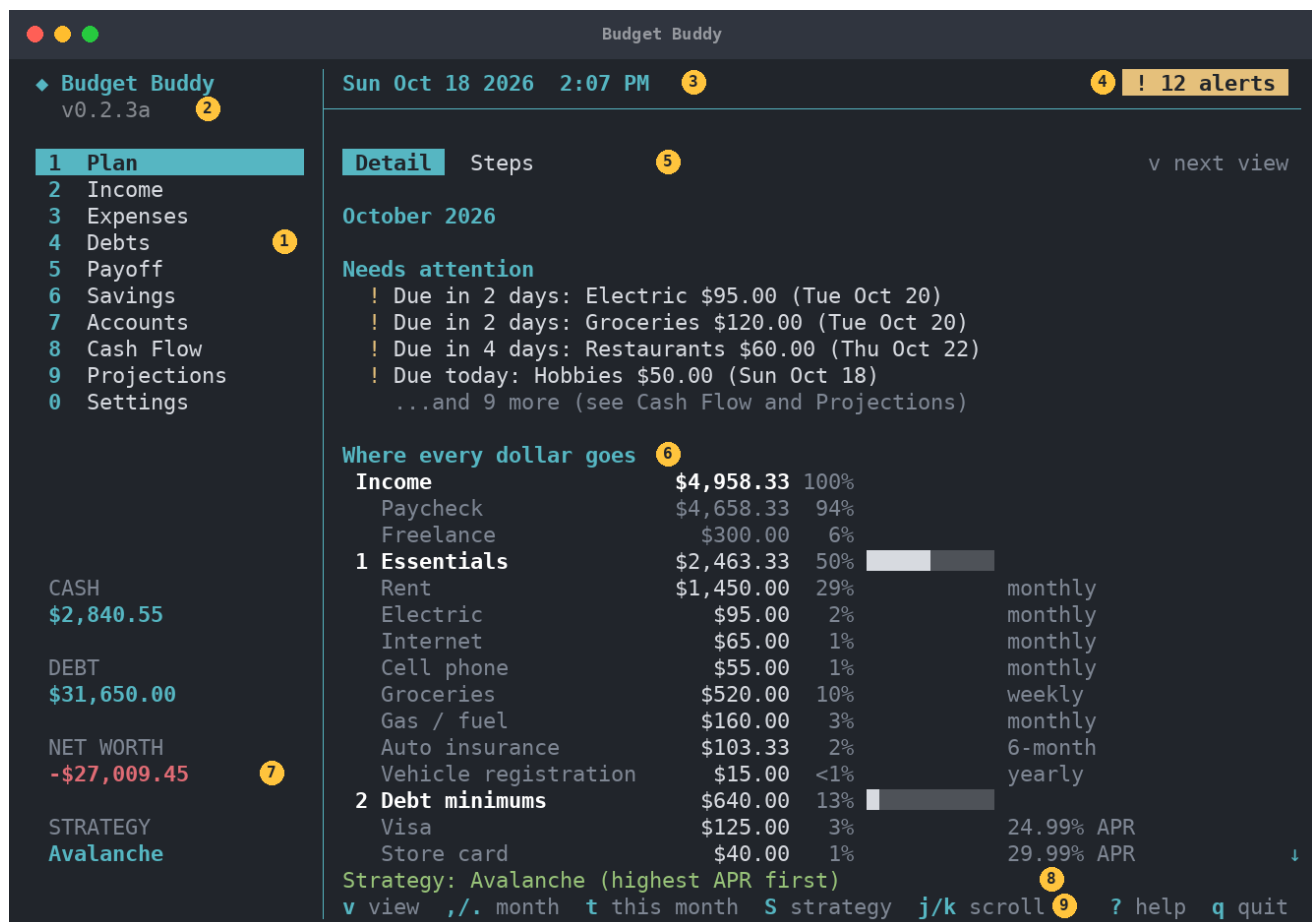


Figure 3. The parts of the sidebar layout, on the Plan page. A confirmation ("Strategy: Avalanche") is showing on the message line.

Turning the sidebar off

b switches the sidebar on and off, on any page. Off, you get the top-bar layout even in a big window (Figure 4), which is what you want if you would rather have the room for wide tables. The choice is remembered: it is the **Sidebar** setting (section 17.1). In a window too small for the sidebar, **b** still changes the setting, and the sidebar appears when the window grows.

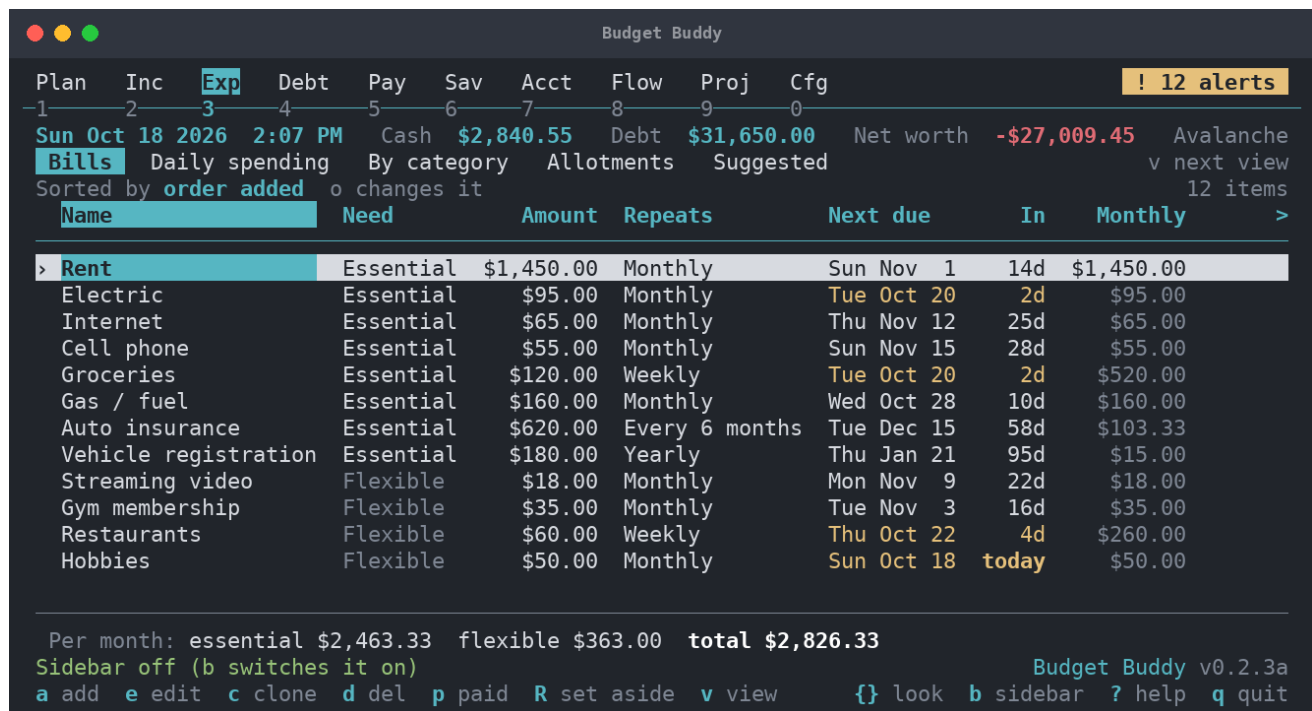


Figure 4. A 100 by 24 window with the sidebar turned off. The page names run along the top and the page has the full width; the key hints show **b** to bring the sidebar back.

2.2 The page: views, sorting and the footer

The pages that hold a table of rows (Income, Expenses, Debts, Savings and Accounts) share one layout, shown in Figure 5. Every one of them takes the same keys for moving, sorting and searching (Chapters 3 and 18).

#	Part	What it shows
1	Views bar	One name for each view of the page. Expenses has five: <i>Bills</i> , <i>Daily spending</i> , <i>By category</i> , <i>Allotments</i> and <i>Suggested</i> . v and V step through them, and a click on a name goes straight to it. When the window is too narrow to name them all the bar shows the current one followed by ►, and the hint “v next view” stays at the right if there is room. A page with a single view has no bar.
2	Sorted-by line	“Sorted by order added” until you sort; afterwards the column and the direction, as in “Sorted by Need ▲”. It also says that o changes it. A search that is active is named here too (section 18.3).
3	Item count	“12 items”, or “3 of 12 items” when a search hides some.
4	Current column	The column the sort, search-in-column and column keys act on. It is shown as a highlighted heading, with its cell highlighted in the row you are on. h and l move it left and right. The arrow ▲ or ▼ marks the column the page is sorted by.
5	Row marker	A > marks the highlighted row, which is also shown in reverse video.
6	Hidden-column markers	A > at the right edge of the headings means columns are hidden because the window is too narrow; a < at the left means columns have scrolled off that side. The current column always stays in view, so moving it with l brings hidden columns in. A ↓ or ↑ at the edge of the table means there are more rows that way.

#	Part	What it shows
7	Footer	A rule, then a line of totals for the page, for instance the monthly total of your bills.

Budget Buddy

Sun Oct 18 2026 2:07 PM ! 12 alerts

1 Plan
2 Income
3 Expenses
4 Debts
5 Payoff
6 Savings
7 Accounts
8 Cash Flow
9 Projections
0 Settings

CASH
\$2,840.55

DEBT
\$31,650.00

NET WORTH
-\$27,009.45

STRATEGY
Avalanche

Bills Daily spending By category Allotments Suggested
Sorted by Need ▲ o changes it
Name Need Amount Next due In
> Rent Essential \$1,450.00 Sun Nov 1 14d
Electric Essential \$95.00 Tue Oct 20 2d
Internet Essential \$65.00 Thu Nov 12 25d
Cell phone Essential \$55.00 Sun Nov 15 28d
Groceries Essential \$120.00 Tue Oct 20 2d
Gas / fuel Essential \$160.00 Wed Oct 28 10d
Auto insurance Essential \$620.00 Tue Dec 15 58d
Vehicle registration Essential \$180.00 Thu Jan 21 95d
Streaming video Flexible \$18.00 Mon Nov 9 22d
Gym membership Flexible \$35.00 Tue Nov 3 16d
Restaurants Flexible \$60.00 Thu Oct 22 4d
Hobbies Flexible \$50.00 Sun Oct 18 today

Per month: essential \$2,463.33 flexible \$363.00 total \$2,826.33
DEMO DATA: changes are not saved (run without --demo for your own data)
a add e edit c clone d del p paid R set aside ? help q quit

Figure 5. The Bills page, with the current column moved to **Need** (1 twice) and the page sorted by it (o).

2.3 The top-bar layout

Narrower or shorter terminals, and any window with the sidebar off, put the pages along the top instead, as in Figure 6. This is the layout of a standard 80 by 24 window, and it gives the page the whole width.

- **Page names** run along the top. When the full names do not fit (that happens below 91 columns) the short names are used: Inc, Exp, Debt, Pay, Sav, Acct, Flow, Proj and Cfg. The alert badge sits at the right of the same line when there is room for it.
- **The rail** under the names is a thin rule that carries each page's number key. The current page is a highlighted bar in the names above it.
- **The figures line** shows the date and time, then Cash, Debt and Net worth, and the debt strategy at the right edge. If the window is too narrow for all of it, the spacing shrinks first, then the strategy goes, then the year, then the figures themselves (Debt, then Cash, then Net worth).
- **The page** starts right below. On a 28-row window or taller a blank row is left under the figures line; at 80 by 24 the page gets every row it can.

- **Message line, product and version, key hints.** The last two rows work as before. “Budget Buddy v0.2.3a” is shown at the right end of the message line. A message is never cut short for it: if a message needs the room, the label steps aside until the message clears.

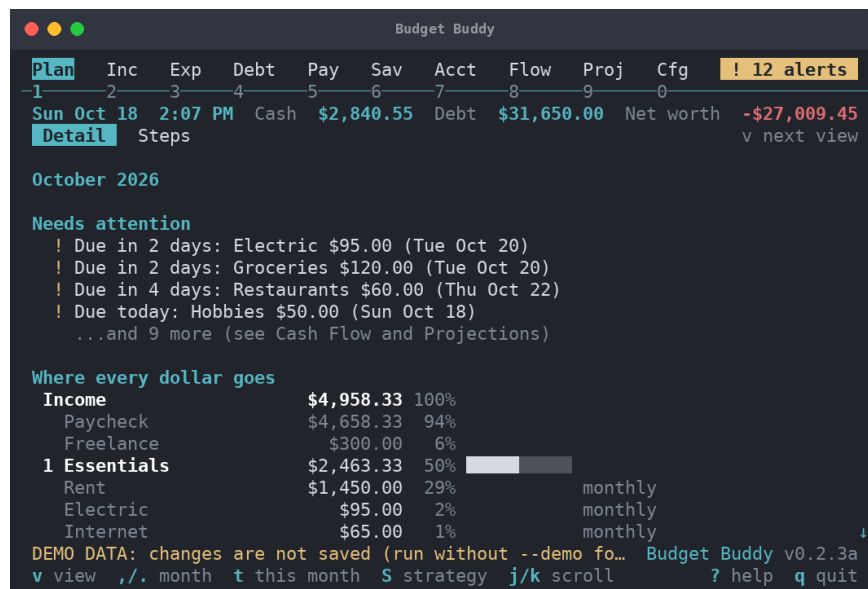


Figure 6. The top-bar layout on a standard 80 by 24 terminal, showing the Plan.

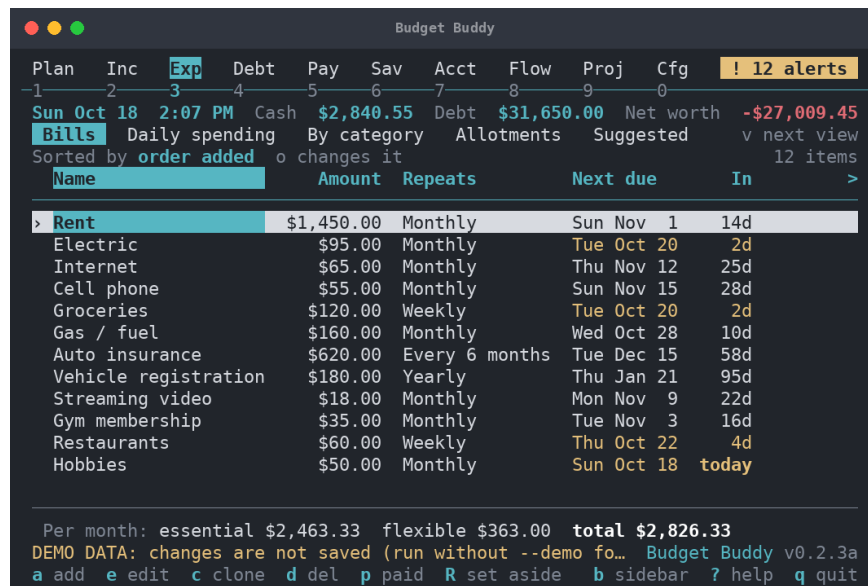


Figure 7. Bills at 80 columns. The Need column is dropped, the > at the right edge of the headings says so, and the key hints that do not fit are omitted.

Narrow windows drop the least important table columns first, and key hints that do not fit are left out (press ? for the full list). Figure 7 shows the Bills page at 80 columns; the same page in a 120-column window shows more of its columns (Figure 12).

On a very short window, under 16 rows, the rail is dropped to save a row and the page names are shown as a single line with the current one highlighted (Figure 8).

```

Budget Buddy
Plan Inc Exp Debt Pay Sav Acct Flow Proj Cfg
Sun Oct 18 2:07 PM Cash $2,840.55 Debt $31,650.00 Net worth -$27,009.45
Bills Daily spending By category Allotments Suggested v next view
Sorted by order added o changes it 12 items
Name Amount Repeats Next due In >
> Rent $1,450.00 Monthly Sun Nov 1 14d
Electric $95.00 Monthly Tue Oct 20 2d
Internet $65.00 Monthly Thu Nov 12 25d
Cell phone $55.00 Monthly Sun Nov 15 28d
Per month: essential $2,463.33 flexible $363.00 total $2,826.33
DEMO DATA: changes are not saved (run without --demo fo... Budget Buddy v0.2.3a
a add e edit c clone d del p paid R set aside b sidebar ? help q quit

```

Figure 8. An 80 by 14 window. The rail is dropped, and the current page is highlighted in the page names.

At the smallest size, 50 by 12 (Figure 9), the page names shrink to their short forms without numbers, the figures line keeps what fits, and the table shows as many columns as there is room for. Everything still works.

```

Budget Buddy
Plan Inc Exp Debt Pay Sav Acct Flow Proj Cfg
Sun Oct 18 2026 2:07 PM Net worth -$27,009.45
Bills Daily spending v next view
Sorted by order added o changes it 12 items
Name Amount Next due >
> Rent $1,450.00 Sun Nov 1
Electric $95.00 Tue Oct 20
Internet $65.00 Thu Nov 12
Per month: essential $2,463.33 flexible $363.00 total $2,826.33
DEMO DATA: changes are not saved (run without --demo fo... Budget Buddy v0.2.3a
a add e edit c clone d del ? help q quit

```

Figure 9. The smallest window Budget Buddy draws, 50 by 12.

If the terminal is smaller than 50 columns by 12 rows, Budget Buddy asks you to make it bigger instead of drawing a broken screen:

```

Budget Buddy

Window too small
Budget Buddy needs at least 50 x 12
(this window is 49 x 12)

```

Figure 10. A terminal that is too small.

2.4 Choosing a size

Terminal size	What you get
Under 50 × 12	The “Window too small” message, nothing else.
50 × 12 up to 80 × 24	The top bar with short page names, and as many columns and hints as fit.
80 × 24	The top bar with full-width pages. Every screen in this manual works here.
100 × 24 and up	The sidebar. The page is 74 columns wide at 100 columns of terminal (Figure 11), and gets wider as the window does.
About 160 × 34	Every column of the widest table (Bills needs about 160 columns), and most of the Plan without scrolling.

Budget Buddy
v0.2.3a

Sun Oct 18 2026 2:07 PM ! 12 alerts

1 Plan
2 Income
3 **Expenses**
4 Debts
5 Payoff
6 Savings
7 Accounts
8 Cash Flow
9 Projections
0 Settings

CASH
\$2,840.55
DEBT
\$31,650.00
NET WORTH
-\$27,009.45
STRATEGY
Avalanche

Bills		Daily spending	By category	Allotments	Suggested	12 items
Sorted by order added		o changes it				
Name	Amount	Repeats	Next due		In	>
> Rent	\$1,450.00	Monthly	Sun Nov 1	14d		
Electric	\$95.00	Monthly	Tue Oct 20	2d		
Internet	\$65.00	Monthly	Thu Nov 12	25d		
Cell phone	\$55.00	Monthly	Sun Nov 15	28d		
Groceries	\$120.00	Weekly	Tue Oct 20	2d		
Gas / fuel	\$160.00	Monthly	Wed Oct 28	10d		
Auto insurance	\$620.00	Every 6 months	Tue Dec 15	58d		
Vehicle registration	\$180.00	Yearly	Thu Jan 21	95d		
Streaming video	\$18.00	Monthly	Mon Nov 9	22d		
Gym membership	\$35.00	Monthly	Tue Nov 3	16d		
Restaurants	\$60.00	Weekly	Thu Oct 22	4d		
Hobbies	\$50.00	Monthly	Sun Oct 18	today		

Per month: essential \$2,463.33 flexible \$363.00 total \$2,826.33
 DEMO DATA: changes are not saved (run without --demo for your own data)
 a add e edit c clone d del p paid R set aside ? help q quit

Figure 11. The smallest sidebar window, 100 by 24, showing Bills.

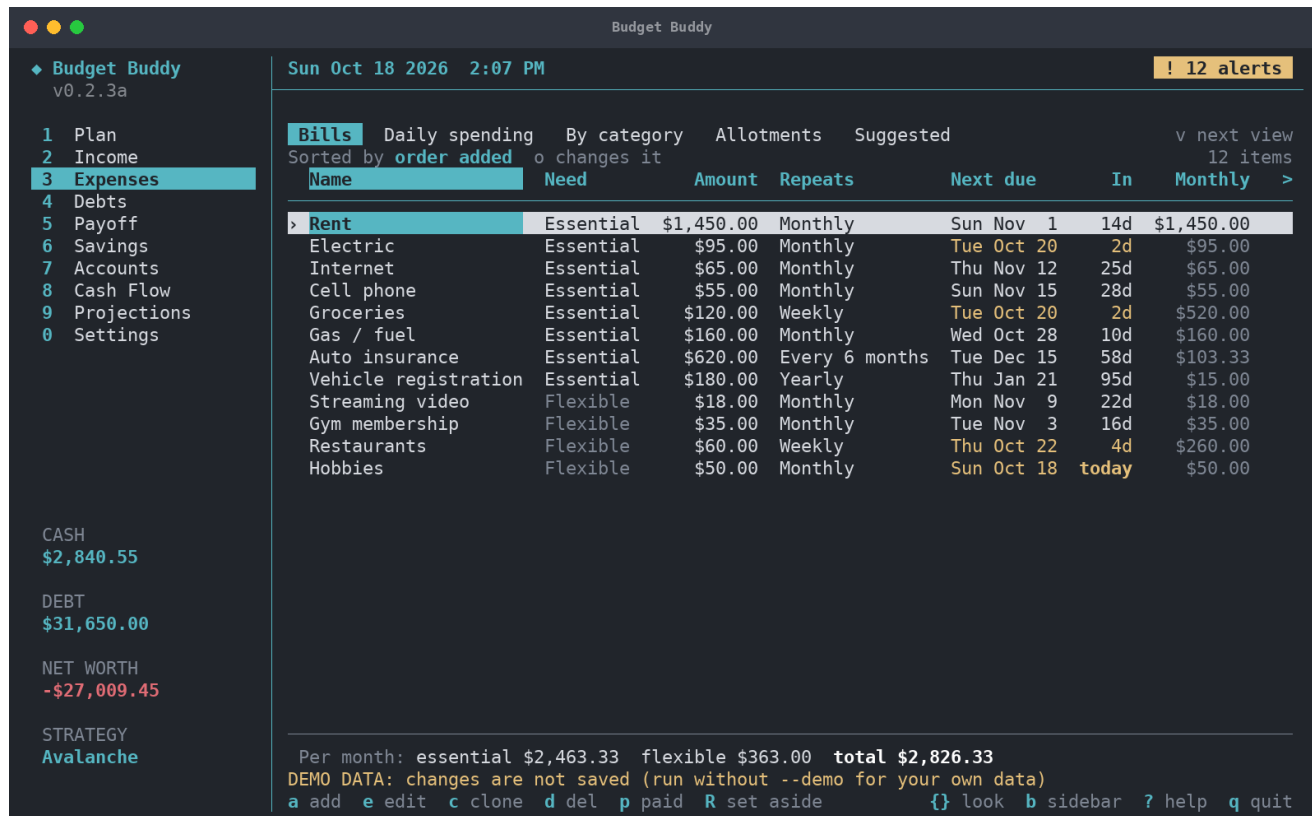


Figure 12. A 120 by 34 window. More columns of the Bills table are shown; the > says that some are still hidden.

The top bar is not a fallback: it is the right choice for a split terminal or a small window, and nothing is lost. Its figures line shows cash, debt, net worth and the strategy on every page.

2.5 The pages

Key	Page	What it is for
1	Plan	The month's plan: alerts, where every dollar goes, spending by category, debt-free date, savings progress. Views: Detail and Steps.
2	Income	Paychecks and other income, with the next pay date and the average monthly amount.
3	Expenses	Bills with due dates, the Daily spending ledger, spending By category, and the Allotments pages.
4	Debts	Balance, APR, minimum and due date for each debt, and what the plan pays this month.
5	Payoff	Debt-free dates and interest for minimums only, the current plan, and a what-if. One view for all debts and one for each debt.
6	Savings	Goals with progress bars, deadlines and an estimated finish month.
7	Accounts	Your bank accounts, balances and net worth; the entry point for history, import and export.
8	Cash Flow	A day-by-day forecast of your spendable cash, with any shortfalls called out.
9	Projections	Looking ahead: when to move money aside, by paycheck or by bill.
0	Settings	The rules the plan follows, and how the program looks and starts.

B works from **any** page: a quick “set this account's balance to match my bank” (section 11.3).

2.6 Views inside a page

A page can show its content in more than one **view**. The views are named on the bar at the top of the page, and **v** and **V** step through them, wrapping around at the ends. (Earlier versions used a different key on each page; Appendix F lists what changed.)

- **Plan:** *Detail* (every line item) and *Steps* (the steps only, so the whole plan fits on one screen).
- **Expenses:** *Bills*, *Daily spending*, *By category*, *Allotments* and *Suggested* (Chapter 7).
- **Payoff:** *All debts*, then one view for each debt, showing its schedule (Chapter 9). **m** switches between monthly and yearly rows.
- **Projections:** *By paycheck* and *By bill* (Chapter 16).

2.7 Colours and symbols

- **Green** is money in or good news; **red** is money out, a negative number or a problem; **yellow** is a warning or money that has not been assigned; the heading colour (cyan in the default look) marks headings and keys; **grey** is supporting detail.
- The highlighted row of a table is shown in reverse video, with a › marker at its left edge.
- ▲ and ▼ in a column heading mean the table is sorted by that column, ascending or descending (Chapter 18).
- ↓ and ↑ at the edge of a page tell you there is more to scroll to. On documents (Plan, Payoff, Cash Flow) scroll with **j** / **k** or the arrow keys.
- **Colour looks.** Settings has a **Colors** setting with thirteen looks (Figures 13 and 14): the terminal's own full colour, which is the default, a blue and teal look, eight painted looks with their own backgrounds, and three single-colour looks for terminals where colour is distracting or hard to see. **{** and **}** step through them from any page. Section 17.1 has the details.
- **Plain terminals.** Set NO_COLOR=1 to start in Mono white (the -c option wins over it, and neither is saved), or BUDGET_ASCII=1 to draw boxes and bars with plain ASCII characters. The program draws in ASCII on its own if your locale is not UTF-8.

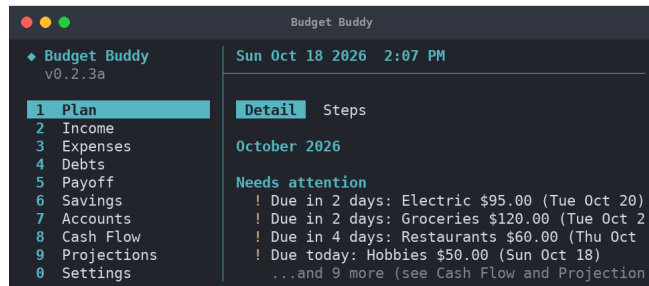
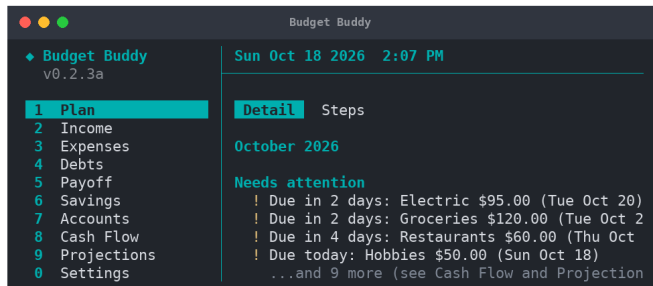
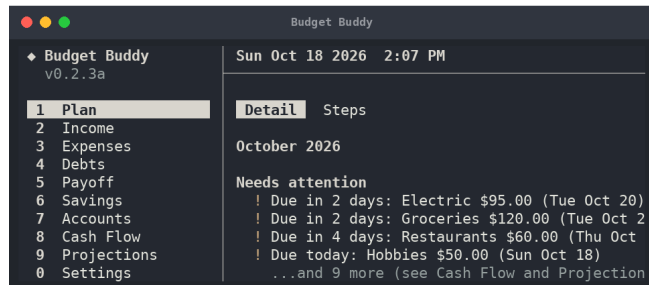
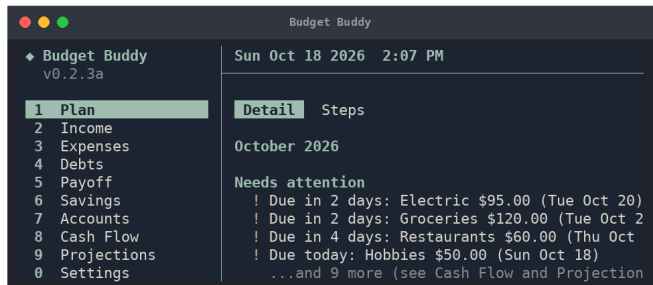
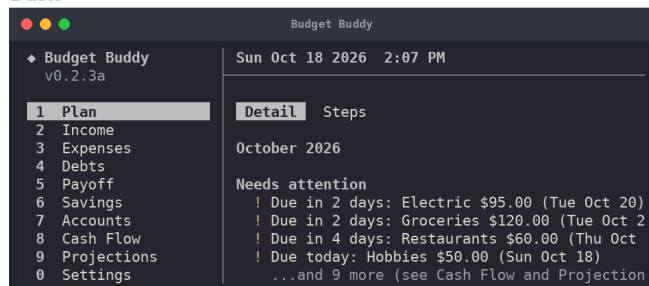
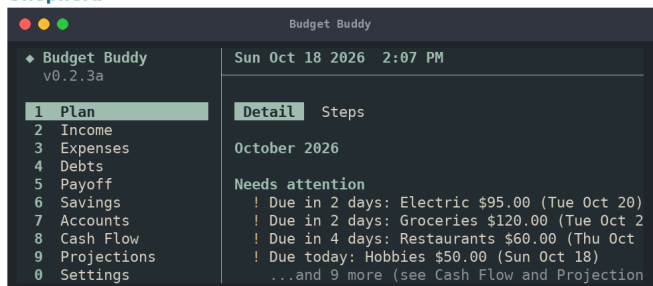
Full color**Blue/teal****Winter****Harbor****Dusk****Shepherd**

Figure 13. The first six looks, in the order `} steps through them: Full color, Blue/teal, Winter, Harbor, Dusk and Shepherd. Each shows the top of the Plan.`

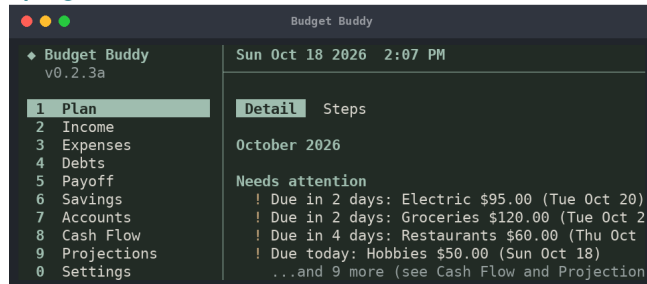
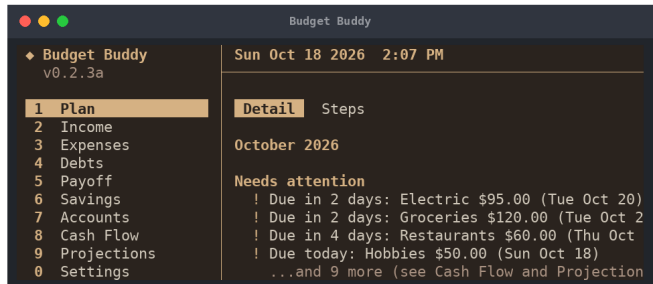
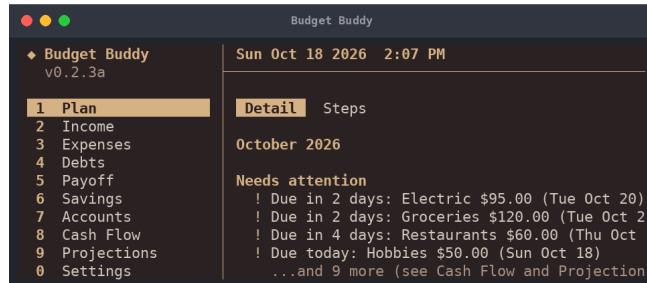
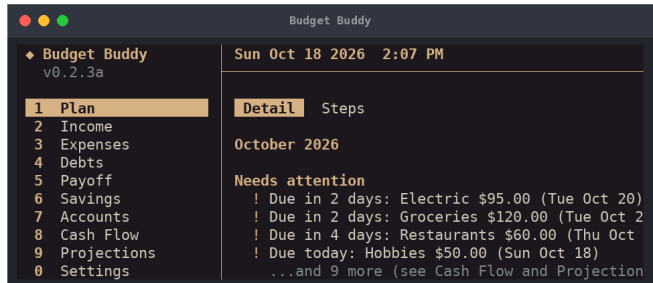
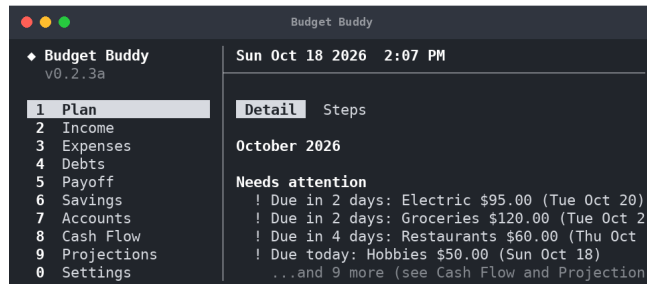
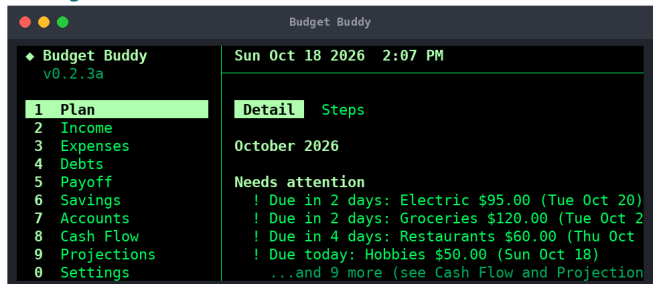
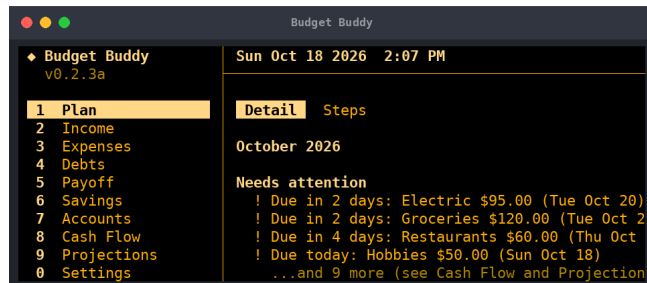
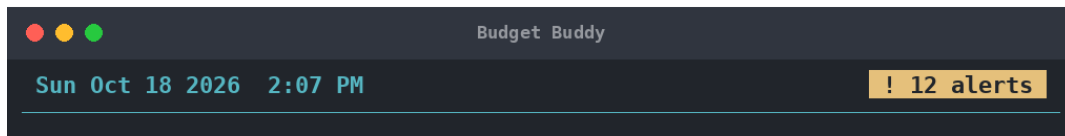
Spring**Autumn****Ember****Samhain****Mono white****Mono green****Mono amber**

Figure 14. The other seven: Spring, Autumn, Ember, Samhain, Mono white, Mono green and Mono amber. Alerts, the current page and highlighted rows are told apart by brightness and reverse video, so every look stays readable.

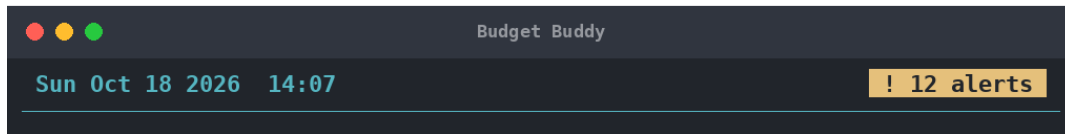
2.8 The clock

The time of day sits beside the date, in the sidebar layout above the page and in the top-bar layout on the figures line (Figure 15). Choose 12-hour or 24-hour form, or hide it, in Settings (section 17.1). Budget Buddy reads the system clock about once a minute, just after the minute changes, so a session left open overnight stays right.

12-hour



24-hour



Hidden

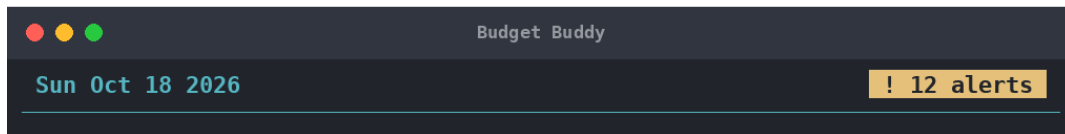


Figure 15. The date line with the clock set to 12-hour, 24-hour and hidden.

2.9 Resizing

Budget Buddy notices when you resize the terminal and redraws at once, choosing the layout again. A dialog that is open stays open and moves to the middle of the page; beside the sidebar it never covers the list of pages.

3. Keys, Forms and Menus

Budget Buddy is driven from the keyboard, and the same handful of habits work everywhere. Learn them once here and the rest of the manual is only about *what* each screen does. Press **?** or **F1** at any time to see the key reference in the program itself (Figure 16; the full list is also in Appendix A). These keys are the ones the other Electric Shepherd programs use, so they will be familiar if you use those too. If you used an earlier version of Budget Buddy, Appendix F lists the keys that changed.

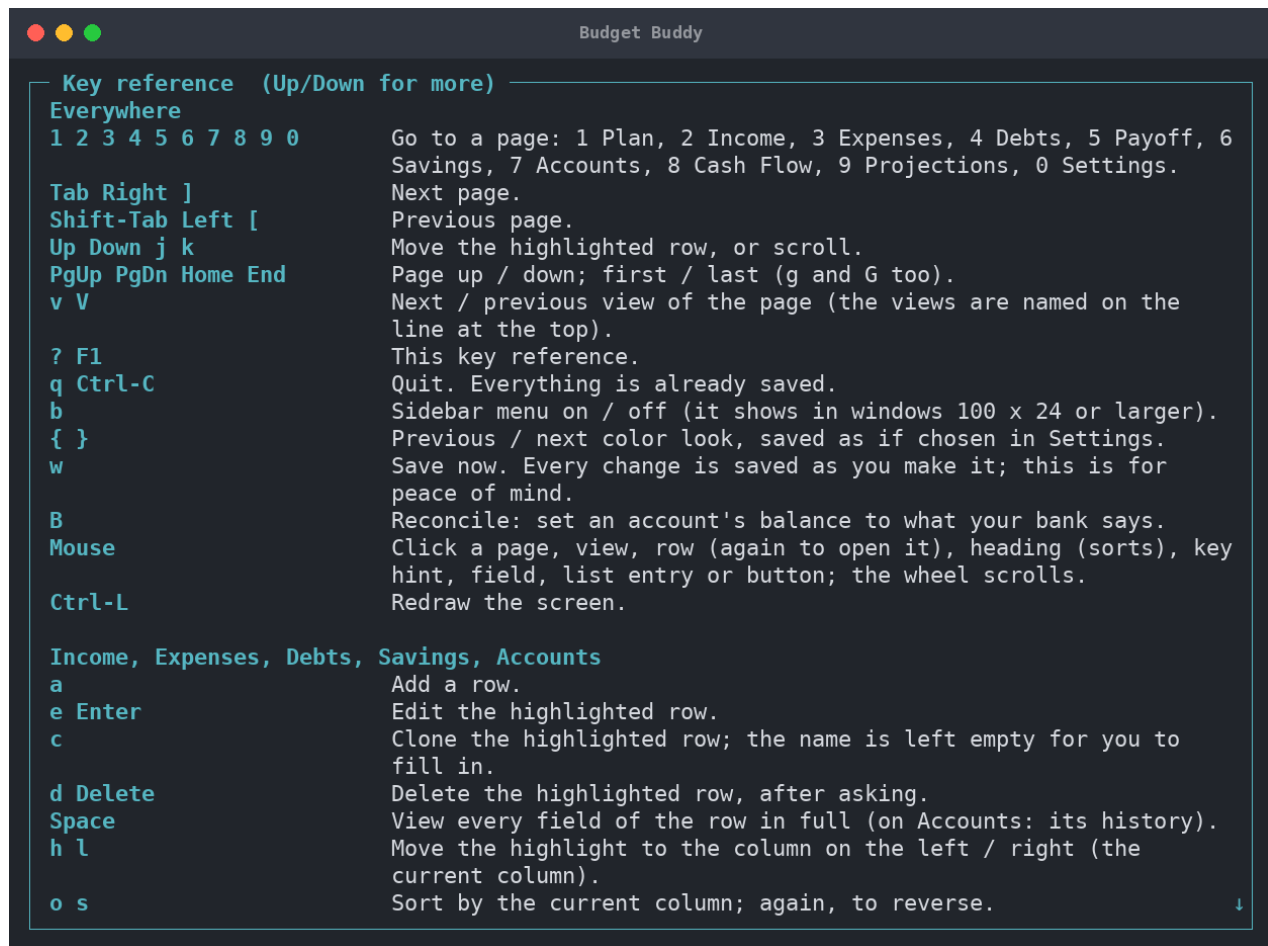


Figure 16. The built-in key reference, opened with **?**. Scroll it with **Up** and **Down**; **Esc** closes it.

3.1 Moving around

Keys	What they do
1 to 9 , 0	Jump straight to a page. 0 is Settings.
Tab / Shift-Tab , Right / Left ,] / [	Next / previous page (when no dialog is open).
Up / Down or k / j	Move the highlighted row in a table, or scroll a document.
PgUp / PgDn	Move or scroll a page at a time.

Keys	What they do
Home / End or g / G	Jump to the top or bottom.
h / l	On a table: move the current column left or right (see below).
v / V	The next / previous view of the page. The views are named on the bar at the top of the page (section 2.2).
b	Turn the sidebar on or off (section 2.1).
{ / }	The previous / next colour look. The choice is saved, as if you had made it in Settings.
B	Reconcile : set an account's balance to match your bank (works on every page).
w	Save now. Every change is saved as you make it, so this is only for peace of mind.
Ctrl-L	Redraw the screen, if something else has scribbled on it.
? or F1	Open the key reference.
q or Ctrl-C	Quit. Everything has already been saved.

3.2 The same keys on every list

Income, Expenses (bills and the daily ledger), Debts, Savings and Accounts are all tables, and they all answer to the same keys:

Key	Action
a	Add a new row. A form opens.
e or Enter	Edit the highlighted row.
c	Clone the highlighted row: a form for a new row that starts as a copy, with the name left empty for you to fill in.
d or Delete	Delete the highlighted row. You are always asked first.
Space	View every field of the row in full (Figure 18). On Accounts, Space opens the account's history instead.
h / l	Move the current column left or right.
o or s	Sort by the current column; press again to reverse (Chapter 18).
O	Choose a sort column and direction from a dialog.
Alt-1 to Alt-9	Sort by the 1st to 9th column from the left.
r	Flip the sort direction.
/	Search every column for some words (Chapter 18).
f	Filter on the current column only.
Esc	Clear the search or filter.

The **current column** is the one the keys **o**, **s** and **f** act on. It is drawn with a highlighted heading, and the cell in it is highlighted on the row you are on (Figure 5). The current column always stays on screen, so on a narrow window **l** brings hidden columns into view.

Actions always apply to the row you can see highlighted, however the table is sorted or searched. When you add a row it becomes the highlighted one. On the **Accounts** page **Enter** edits the account, like everywhere else; its history is on **Space** or **H** (Chapter 12).

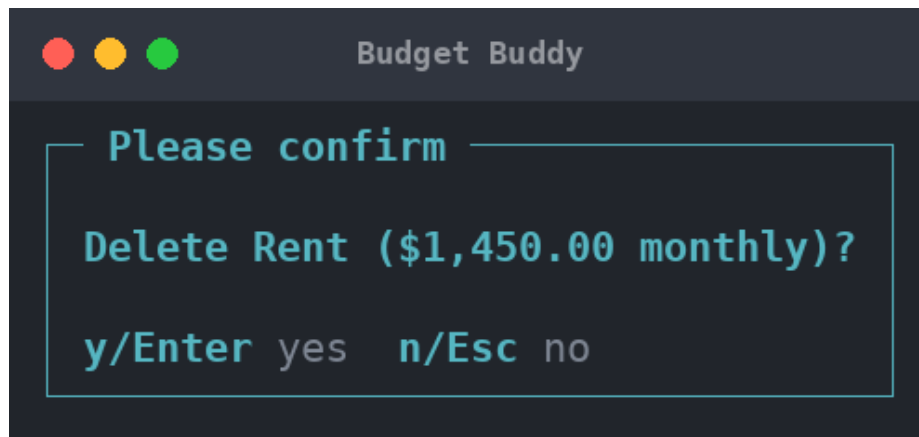


Figure 17. Deleting asks first. **y** or **Enter** confirms; **n**, **Esc** or **q** cancels.

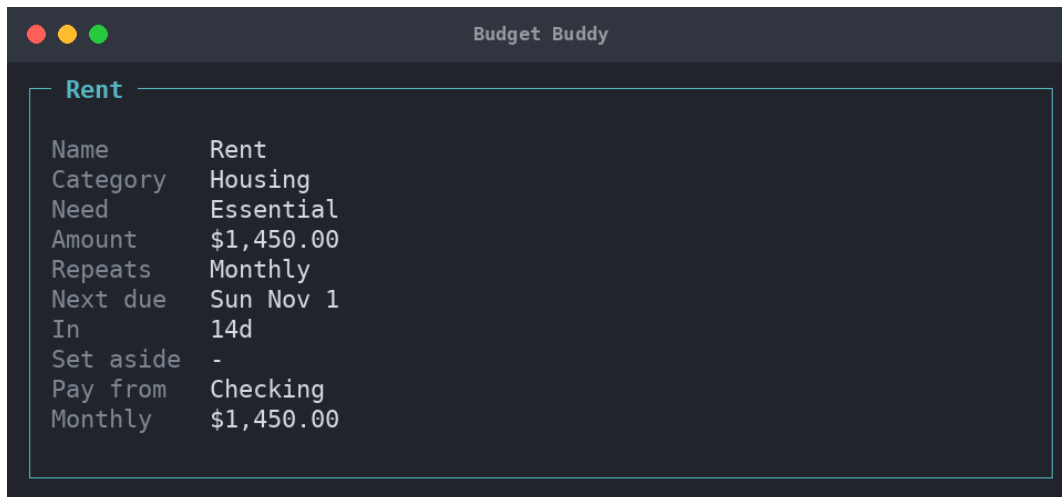


Figure 18. **Space** on a bill shows every field of it, including the ones the table has no room for. **e** edits the row; **Esc**, **q**, **Space** or **Enter** closes the view.

3.3 Forms

Adding or editing anything opens a form: a small dialog with one line per field, and the buttons **[Save]** and **[Cancel]** under them. A field marked with the > marker at its left edge is the one you are editing, and a short explanation of it appears near the bottom of the dialog. The last line of every form lists its keys.

Key	What it does in a form
Tab / Down	Next field; after the last one, the Save and Cancel buttons.
Shift-Tab / Up	Previous field.
Enter	On a text field: the next field, and save from the last one. On a menu field: open the menu. On a button: press it.
Ctrl-S or F2	Save from anywhere in the form. If something is wrong, the form stays open and tells you what.

Key	What it does in a form
Esc	Close an open menu or list first; press again to cancel the form without saving.
Left / Right	On a menu field: the previous / next choice, without opening the menu. On a text field: move the cursor.
Home / Ctrl-A , End / Ctrl-E	Start / end of the text.
Ctrl-U / Ctrl-K	Delete everything before / after the cursor.
Delete / Backspace	Edit text in place.
Ctrl-N / Ctrl-P	On a field that has suggestions: show the list, then step through it.

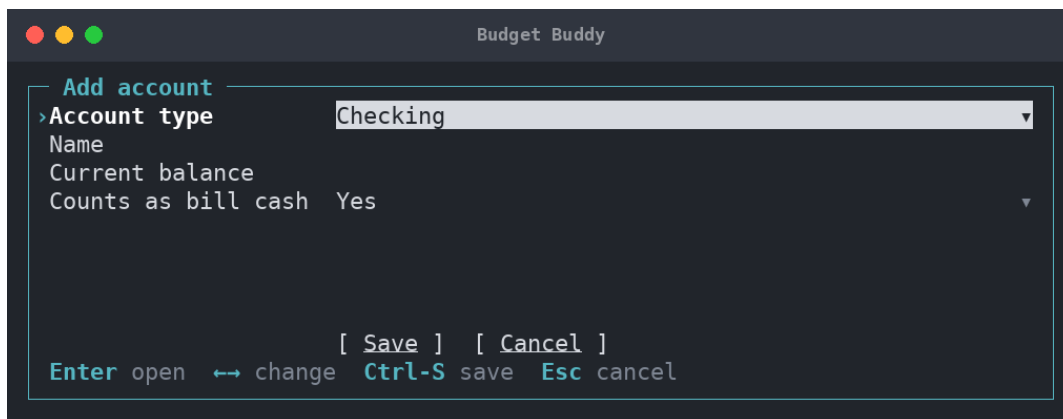


Figure 19. A form (adding an account). The > marks the current field; a field with a ▼ is a menu.

A field that already has a value is shown **selected**: just start typing to replace it, or use the arrow keys, **Home**, **End** and **Delete** to change part of it.

Menus

Fields with a ▼ at the right are menus. Open one with **Enter** or **Space**, or simply start typing. The list narrows to the entries that contain what you type, with the matching letters underlined; **Up** / **Down** move through it and **Enter** chooses.

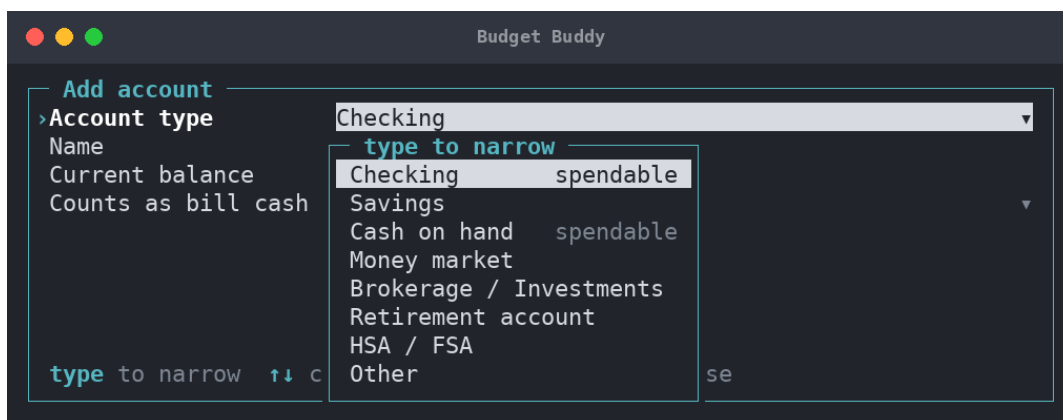


Figure 20. A menu open on the Account type field. Some entries carry a note, such as spendable.

Many **Name** fields are menus too, with suggestions that depend on the category you picked (“Rent”, “Mortgage” and “HOA dues” for Housing). They are only suggestions: type any name you like, and press **Enter** to keep what you typed.

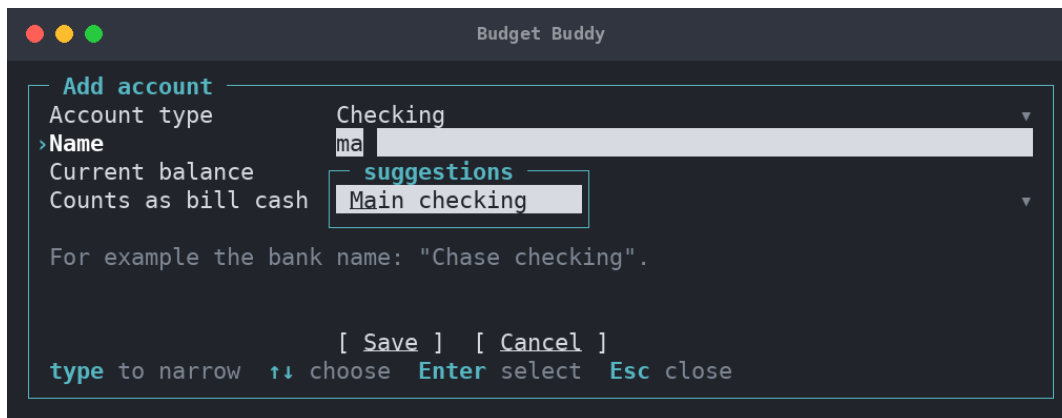


Figure 21. Name suggestions for a Checking account, filtered as you type.

Dates

Date fields understand many shorthands. While you type, the form shows the weekday and how far away the date is, so you can confirm you meant what you typed (see Figure 27 in Chapter 4).

You type	Meaning
2026-11-01	A full date, year first.
11/1 or 11/01	Month and day; the year is filled in (next occurrence, for upcoming dates).
Nov 1	A month name and day.
15	The next 15th.
+7	Seven days from today.
+2w	Two weeks from today.
today, tomorrow	What they say.

Dates must fall between the years 1900 and 2200. Anything outside that range, however it was typed (+99999999 for instance), is refused with a message.

Money and percentages

Money accepts 1500, 1,500.50 and \$42; leave off the dollar sign and commas if you like. A comma is only ever a thousands separator, in groups of three (1,500). If you write the cents after a comma, as in 12,50, the form stops and tells you to type 12.50, so an amount can never change by a factor of a hundred without your noticing. Percentages accept 24.99 or 24.99%. Amounts are stored to the cent.

When something is wrong

If a field cannot be understood, the form stays open and a red message explains what to try. Fix the field and press **ctrl-s** again; nothing is saved until the form is correct.

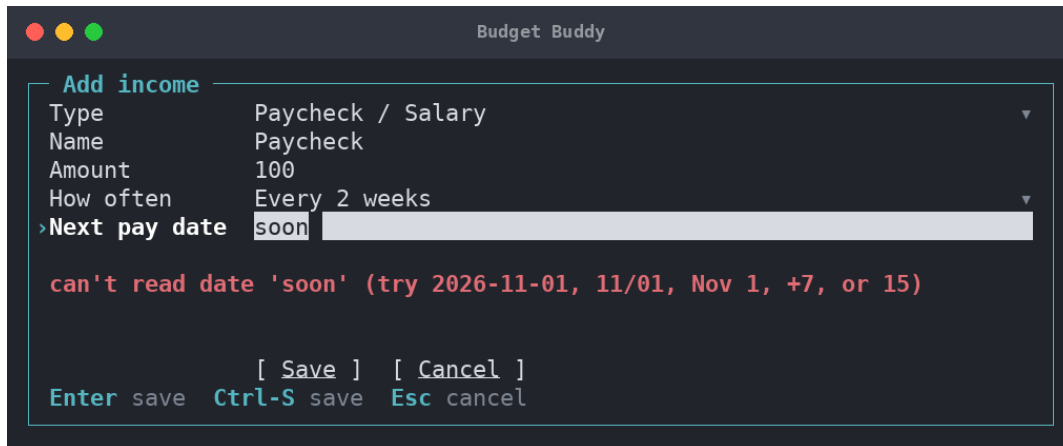


Figure 22. A date that could not be read. The message suggests formats that work.

3.4 Dialogs

Besides forms, a few dialogs take their own keys. Each lists them on the line at the bottom of the screen.

Dialog	Keys
Confirmation	<code>y</code> , <code>Y</code> or <code>Enter</code> says yes. <code>n</code> , <code>N</code> , <code>Esc</code> , <code>q</code> or <code>Ctrl-C</code> says no. Nothing happens until you answer.
Key reference	<code>Up</code> , <code>Down</code> , <code>j</code> , <code>k</code> , <code>PgUp</code> , <code>PgDn</code> , <code>Space</code> , <code>Home</code> and <code>End</code> scroll. <code>Esc</code> , <code>q</code> , <code>?</code> , <code>F1</code> and <code>Enter</code> close it.
View of a row	<code>Up</code> and <code>Down</code> scroll. <code>e</code> edits the row. <code>Esc</code> , <code>q</code> , <code>Space</code> and <code>Enter</code> close it.
Menu (Settings)	<code>Up</code> and <code>Down</code> move, <code>Enter</code> applies, <code>Esc</code> puts the old choice back (section 17.1).

3.5 The mouse

Everything can be done from the keyboard, but the mouse works too, in a terminal that reports it (nearly all current ones do). A click does what the matching key would do:

Click on	What happens
A page name (sidebar or top bar)	Go to that page.
A view name on the views bar	Show that view.
A column heading	Sort by that column; click again to reverse.
A row of a table	Highlight the row, and make the column you clicked the current column. Click the same row again to open it, as <code>Enter</code> does.
A key hint at the bottom	Press that key.
A field , a list entry or a button in a form	Go to the field, choose the entry, or press the button. In a Settings menu the first click highlights an entry (and shows a colour look on the page behind it) and the second chooses it.

The **wheel** moves three rows at a time, on a page, in a list, in the key reference or in a history. The mouse only acts on what is on screen: while a dialog is open nothing behind it takes a click, and the key reference, a confirmation and the import preview take no clicks except on their key hints.

While the mouse is on, dragging to select text usually needs **Shift** held in the terminal; if you would rather have the mouse back, turn off the **Mouse** setting (section 17.1).

3.6 Messages

After most actions a one-line message appears on the message line above the key hints (green for success, red for a problem), such as “Added Rent”. It disappears when you press the next key. A message is never shortened to make room for the product label in the top-bar layout: the label steps aside until the message clears.

3.7 Quitting

Press **q** (or **ctrl-c**) from any screen that is not showing a dialog. Budget Buddy saves after every change, so there is normally nothing to confirm. If the terminal window is closed, or the program is interrupted or killed, the screen is restored and your data is as you last left it.

The one exception is a save that **failed** (a full disk, for example). Then a red **NOT SAVED:** line stays at the bottom of the screen until a save works, and quitting tries once more to save before asking “Your changes could not be saved. Quit anyway and lose them?” Answer **y** only if you accept losing the changes made since the failure. If you leave anyway, the program ends with exit status 1 and prints the reason.

4. Quick Start: Your First Budget

This chapter builds a small budget from nothing, one screen at a time. It takes about five minutes. Do the steps in this order; each one feeds the next. The screens here match what you get if you follow the same entries; yours will show your own names and numbers.

Note. Not ready to enter real figures? Run `budget --demo` instead. It starts with a complete sample budget (the one used in the screenshots in the rest of this manual) and never saves your changes.

4.1 Start with an empty program

Run `budget`. The splash screen appears, with the sheep, the name and version, and the line “press any key to continue”. Press a key and, because there is no data yet, the Plan shows a short welcome that tells you where to begin:

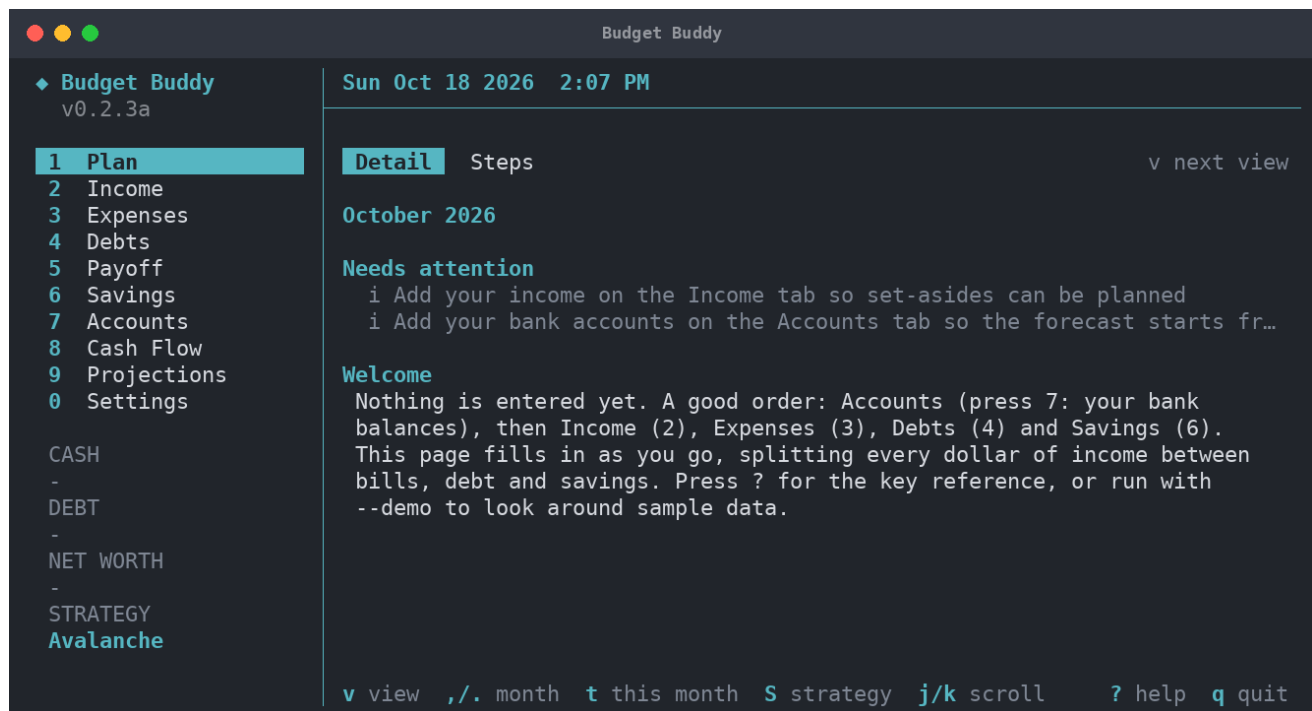


Figure 23. A new, empty budget. The Plan has nothing to plan yet; the alerts marked *i* are only pointers to where to begin, so there is no alert badge. The four figures in the sidebar show a dash until there is something to show.

4.2 Add your accounts (page 7)

Accounts come first because your bills and the cash-flow forecast are checked against the money you actually have. For each bank account:

1. Press **7** to open **Accounts**, then **a**.
2. Choose the **Account type** from the menu: Checking, Savings, Cash on hand, and so on.
3. Pick or type a **Name** (for example “Main checking”).
4. Type the **Current balance**, exactly what the bank shows today.

5. **Counts as bill cash** says whether this account's money is available to pay bills. Checking and Cash on hand default to Yes; savings and investments to No.
6. Press **ctrl-S** to save (or press **Enter** on the last field, or on the **Save** button).

Add account

Account type: Checking

Name: Main checking

> Current balance: 2840.55

Counts as bill cash: Yes

What the account holds today. Later changes go through Reconcile / Deposit / Withdraw so there is a history.

[Save] [Cancel]

Enter next Ctrl-S save Esc cancel

Figure 24. Adding a checking account: type, name and today's balance.

Repeat for each account. After the first one the Accounts page looks like Figure 25, and with a savings account added it looks like Figure 26. The page is sized for a terminal 100 columns wide, the smallest that shows the sidebar; a wider window shows more columns.

Budget Buddy
v0.2.3a

Sun Oct 18 2026 2:07 PM

Sorted by **order** added o changes it

Account	Type	Spendable	Balance	Earmarked	Updated
> Main checking	Checking	yes	\$2,840.55	-	Oct 18

1 item

CASH
\$2,840.55

DEBT
-

NET WORTH
\$2,840.55

STRATEGY
Avalanche

Spendable cash \$2,840.55 what bills are checked again...

Debt owed \$0.00

Net worth \$2,840.55

Added account 'Main checking'

a add e edit d del + in - out t transfer b sidebar ? help q quit

Figure 25. One account entered. The summary under the table now shows Spendable cash and Net worth.

Sorted by	order	added	o	changes it	Balance	Earmarked	Updated
	Account	Type		Spendable			
	Main checking	Checking	yes		\$2,840.55	-	Oct 18
>	Savings	Savings	no		\$1,800.00	-	Oct 18

CASH	\$2,840.55		
DEBT	-		
NET WORTH	\$4,640.55		
STRATEGY	Avalanche		

Spendable cash	\$2,840.55	what bills are checked again...
Other accounts	\$1,800.00	savings, investments, retir...
Debt owed	\$0.00	
Net worth	\$4,640.55	

Added account 'Savings'

a add e edit d del + in - out t transfer b sidebar ? help q quit

Figure 26. Checking (bill cash) and Savings (not bill cash). Savings is earmarked later for savings goals.

4.3 Add your income (page 2)

1. Press **2**, then **a**.
2. Choose a **Type** ("Paycheck / Salary"). The name, and the usual frequency for that kind of income, fill in.
3. Type the **Amount** of one payment, after tax.
4. Choose **How often** (every 2 weeks, twice a month, monthly, and so on).
5. Type the **Next pay date**. +6 means six days from today.
6. **Ctrl-S** saves.

If you choose **Twice a month**, one more field appears for the day of the month of the second payment (31 means the last day of the month).

Add income

Type: Paycheck / Salary

Name: Paycheck

Amount: 2150

How often: Every 2 weeks

> Next pay date: +6

→ Sat Oct 24 2026 (in 6 days)

[Save] [Cancel]

Enter save Ctrl-S save Esc cancel

Figure 27. A paycheck of \$2,150 every two weeks, with the next pay date typed as +6. The line beginning → shows what the date means.

Budget Buddy
v0.2.3a

Sun Oct 18 2026 2:07 PM

Sorted by order added o changes it 1 item >

Name	Amount	Repeats	Next	Monthly
> Paycheck	\$2,150.00	Every 2 weeks	Sat Oct 24	\$4,658.33

Average per month **\$4,658.33** (one-time items not counted)
Added income 'Paycheck'

a add e edit c clone d del Space view o sort ? help q quit

CASH
\$2,840.55
DEBT
-
NET WORTH
\$4,640.55
STRATEGY
Avalanche

Figure 28. The Income page after saving. Budget Buddy works out the monthly average. The > at the right of the headings says there are more columns than fit at this width.

4.4 Add your bills (page 3)

1. Press **3**, then **a**. The first field is a **Category** menu.
2. Choosing a category fills in the **Priority** (essential or flexible) and the usual frequency, and offers typical names.
3. Type the **Amount** and the **Next due date**.
4. **Pay from** is the account this bill normally comes out of; it becomes the default when you mark the bill paid.
5. Save with **ctrl-s**. Repeat for rent, utilities, phone, insurance, subscriptions...

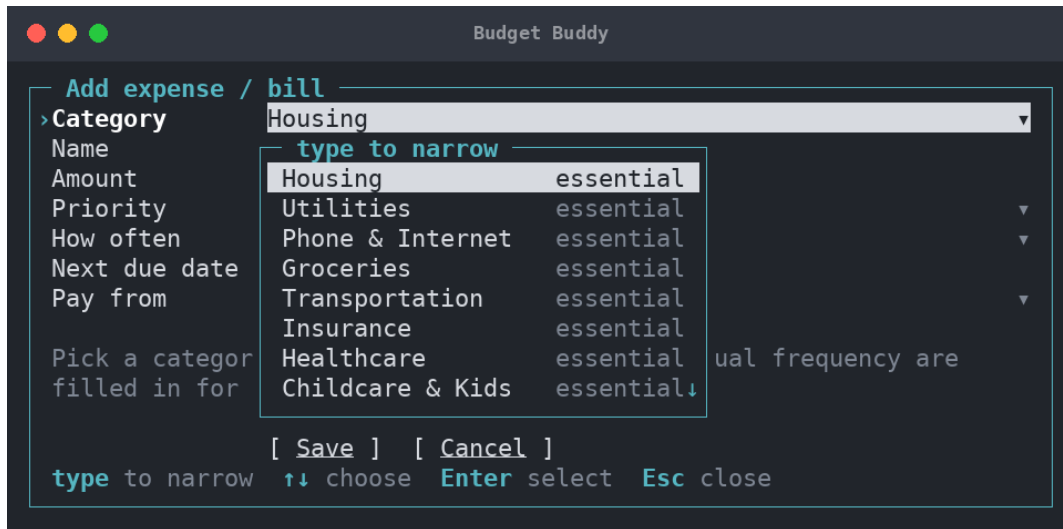


Figure 29. The category menu on a new bill. The note beside each category says whether it is normally essential or flexible.

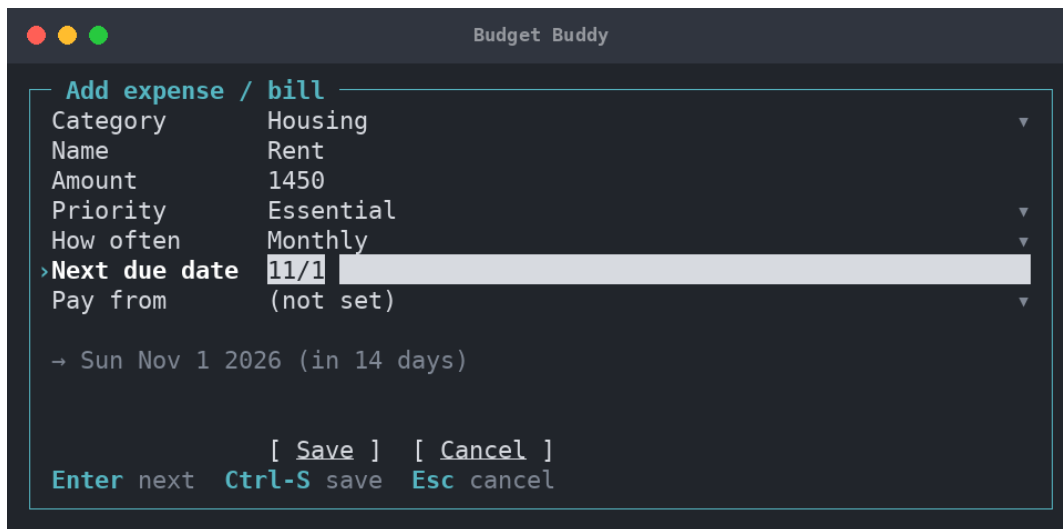


Figure 30. Rent: \$1,450 monthly, next due 11/1.

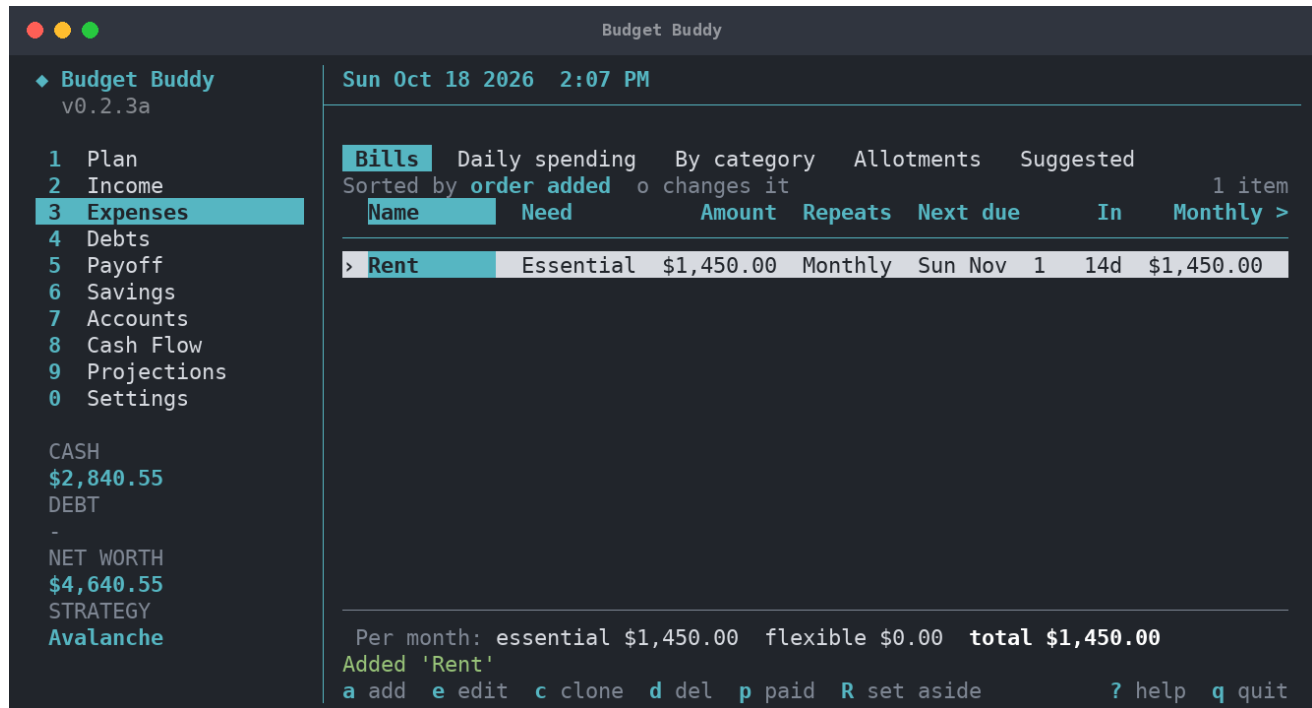


Figure 31. The Bills page after saving the first bill.

Tip. Enter every recurring bill, including yearly and twice-a-year ones such as insurance and registration. Budget Buddy spreads those over the paychecks before they are due so they never ambush you.

4.5 Add your debts (page 4)

For each debt enter its **Current balance**, **APR** (the annual interest rate, such as 24.99), **Minimum payment**, how often it is due and the **Next due date**. Use APR 0 for an interest-free loan.

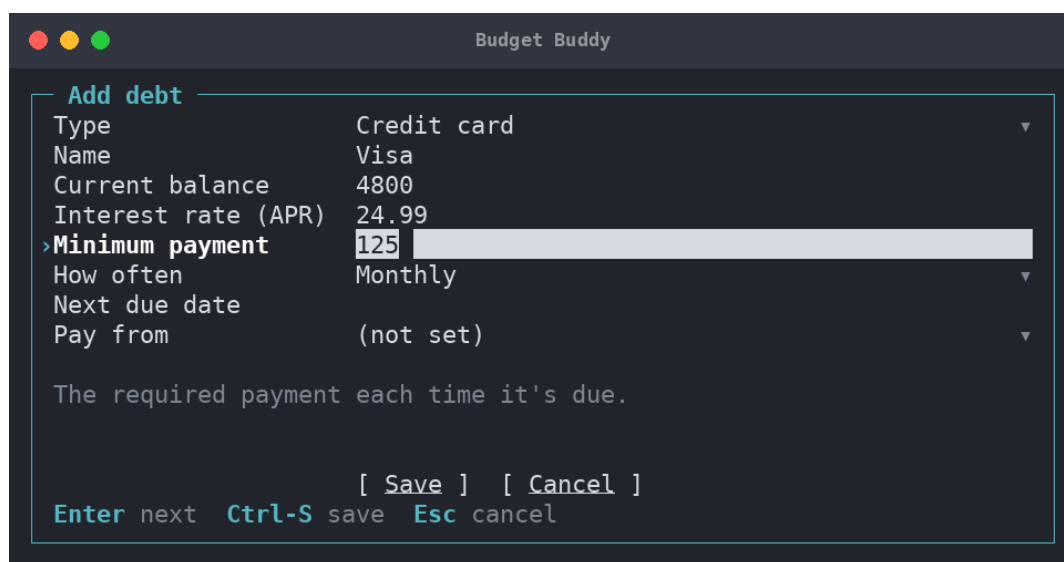


Figure 32. A credit card: \$4,800 at 24.99% APR, \$125 minimum.

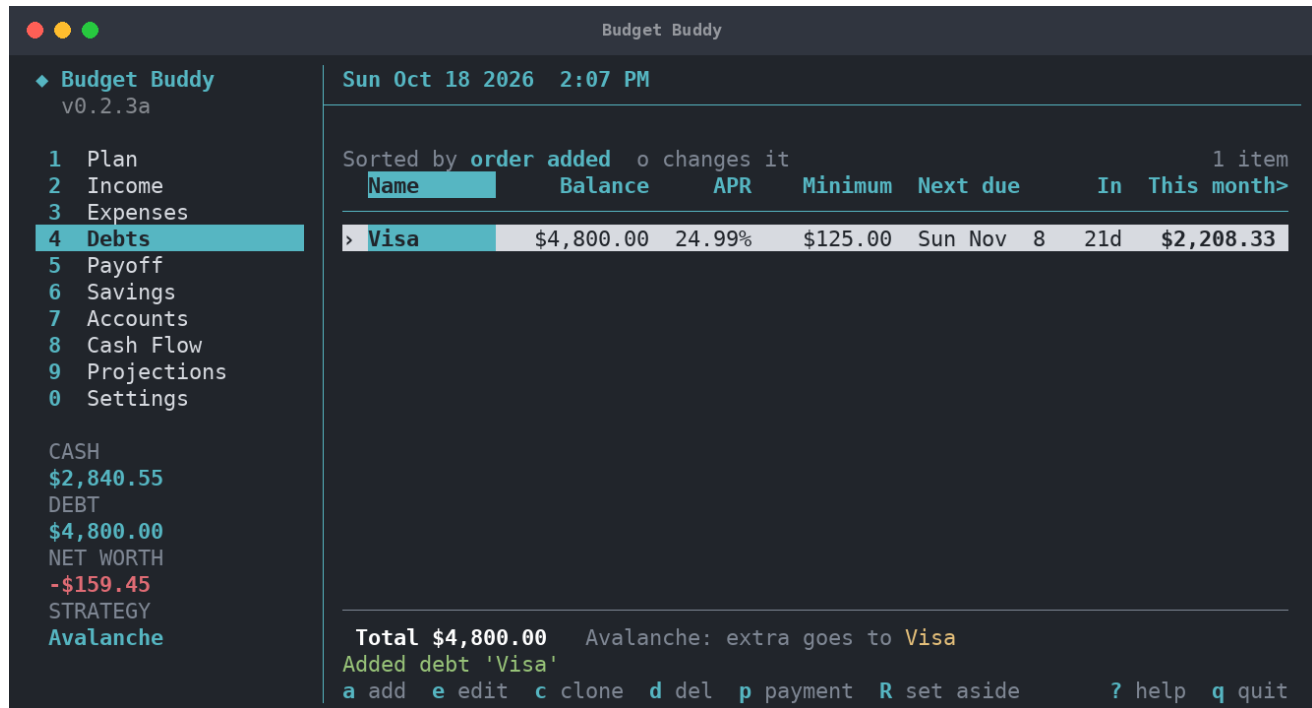


Figure 33. The Debts page. This month's payment is minimum plus any extra the plan assigns.

4.6 Add a savings goal (page 6)

A goal has a **Type**, a **Target amount**, what you have **Saved so far**, and an optional **Deadline**. With a deadline, the plan sets aside the monthly amount needed to reach the target in time. **Priority** (1 to 9, 1 first) decides which goal is funded first when money is short, and **Held in** names the account where the money is kept.

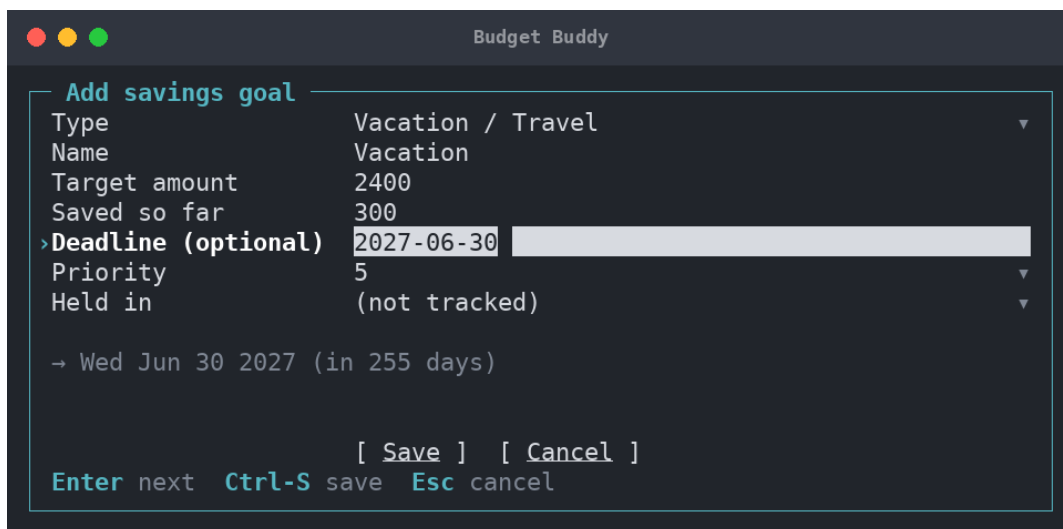


Figure 34. A vacation fund: \$2,400 by June 30, 2027, with \$300 saved so far.

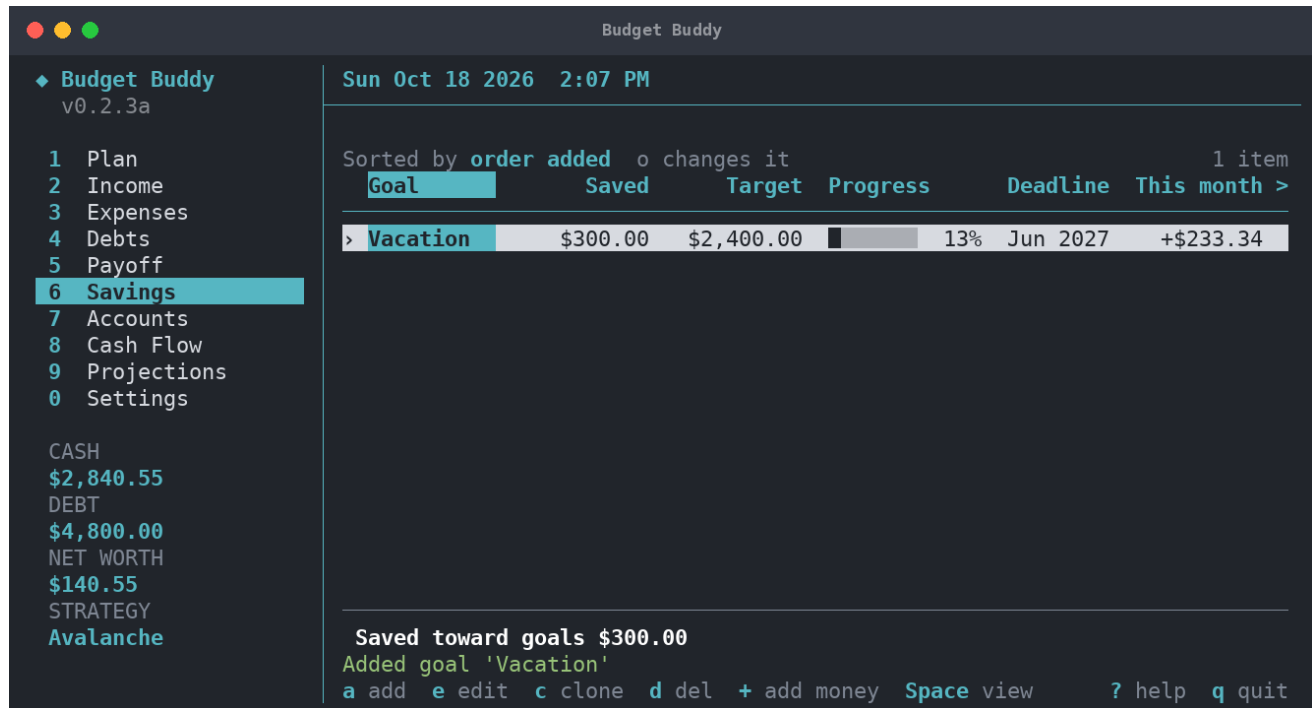


Figure 35. The Savings page. The progress bar and the estimated finish month appear once the plan has run.

4.7 Read the Plan (page 1)

Press **1**. The sidebar now shows your cash, debt and net worth. Budget Buddy has split your income, set the bills against their due dates and projected your debt payoff:

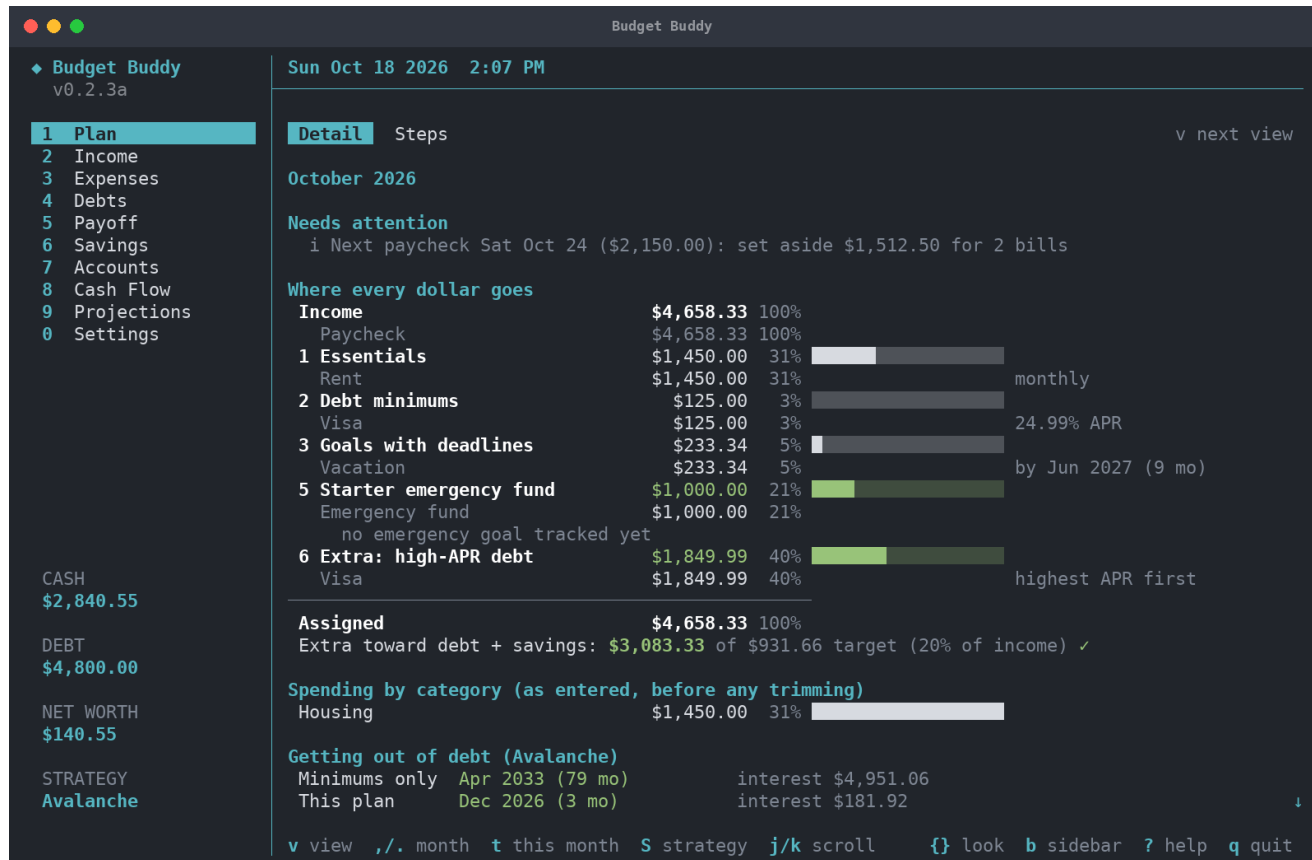


Figure 36. The Plan for the budget built in this chapter.

Next, look at **Projections** (9) to see which paycheck should fund which bill, and **Cash Flow** (8) to see whether your cash ever runs short. The following chapters explain every screen in detail.

5. The Plan

The Plan (**1**) is the heart of Budget Buddy and the screen to read first each time you open it. It is a scrolling document that answers: *what needs my attention, where does every dollar of this month's income go, how does my spending break down, when will I be out of debt, and how are my savings goals doing?* It has no table to select from; scroll with **j** / **k**, the arrow keys, **PgUp** / **PgDn**, **Home** and **End**. It has two **views**, named on the bar at the top of the page: *Detail*, which shows every line item, and *Steps*, which shows only the steps so that the whole plan fits on one screen.

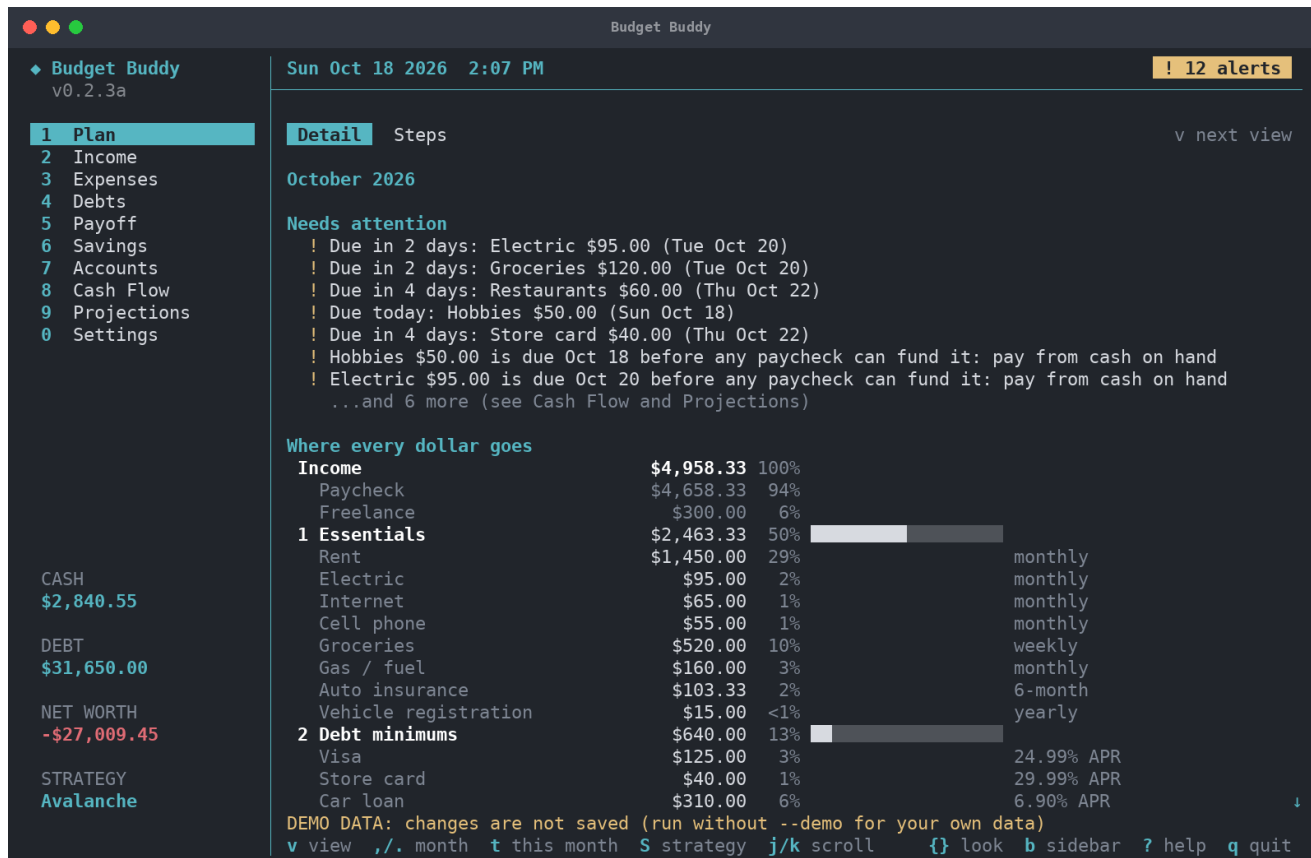


Figure 37. The top of the Plan, with the sample budget.

Key	What it does
, / .	Show the previous / next month. (< and > do the same.)
t	Return to the current month.
v / V	Switch between the Detail and Steps views (Figure 39).
S	Switch the debt strategy between Avalanche and Snowball.
j k PgUp PgDn Home End	Scroll.

5.1 Needs attention

The first block lists things you should do or know about, most urgent first. Each line starts with a marker: **!** in red or yellow for something to act on, and **i** in grey for information. It only appears for the current month. The checks are:

Message	When it appears
OVERDUE: <i>name amount</i> was due <i>date</i>	A bill or debt payment whose due date has passed and has not been marked paid. Red.
Due today / tomorrow / in N days	A bill or debt due within the next 7 days. Yellow.
Cash gap: balance falls below zero on <i>date</i> ...	The cash-flow forecast (Chapter 15) finds a day when your spendable cash drops below zero, or below your cushion if you set one in Settings. It says how low it goes, when it recovers, and how much more you would need on hand today. Red.
<i>Bill</i> is due <i>date</i> before any paycheck can fund it: pay from cash on hand	A bill due within 21 days that falls before the next paycheck arrives, so the money has to come from what you have now. Yellow.
Next paycheck <i>date</i> : set aside <i>amount</i> for N bills	Information: how much of the next paycheck to move aside (see Chapter 16).
<i>Account</i> was last updated N days ago	A spendable account whose balance has not been reconciled for more than 14 days. Press B to fix it (section 11.3).
Add your income / bank accounts...	Reminders while you are still setting up.

When there are more alerts than fit, the list ends with “...and N more”; the Cash Flow and Projections pages show the rest.

5.2 Where every dollar goes

This is the plan itself. Your average monthly income is listed first, one line per source, then the order in which it is spent. Each numbered step gets what is left after the steps above it, and the bar and percentage show its share of income.

Step	What it is
1 Essentials	Every expense marked Essential: rent, utilities, groceries, insurance and so on. Yearly and six-monthly bills count at their monthly share.
2 Debt minimums	The minimum payment on every debt, each at its monthly equivalent.
3 Goals with deadlines	The monthly amount needed to reach each deadline goal on time.
4 Flexible spending	Dining out, subscriptions, hobbies and other Flexible bills. If the plan needs money for debt and savings it can trim these, but never by more than the <i>Max trim</i> setting.
5 Starter emergency fund	A first cushion (default \$1,000), until it is built. It is not shown once you have it.
6 Extra: high-APR debt	Extra payments on debts at or above the high-interest threshold (default 8% APR), in avalanche or snowball order.
7 Emergency fund	Builds the emergency fund to the target number of months of essentials and minimums (default 3).

Step	What it is
8 The rest	Split between remaining debt and savings goals (default half and half). If one side cannot use its half, the other gets it.
Unassigned	Anything left over is shown in yellow, so it never just disappears.

A numbered step is skipped when it does not apply, which is why Figure 37 jumps from step 4 to step 6: the starter fund is already complete. Under the line in Figure 38, **Assigned** shows the total that has been allocated, and the line below it says whether the extra money going to debt and savings meets your target (a ✓ when it does).

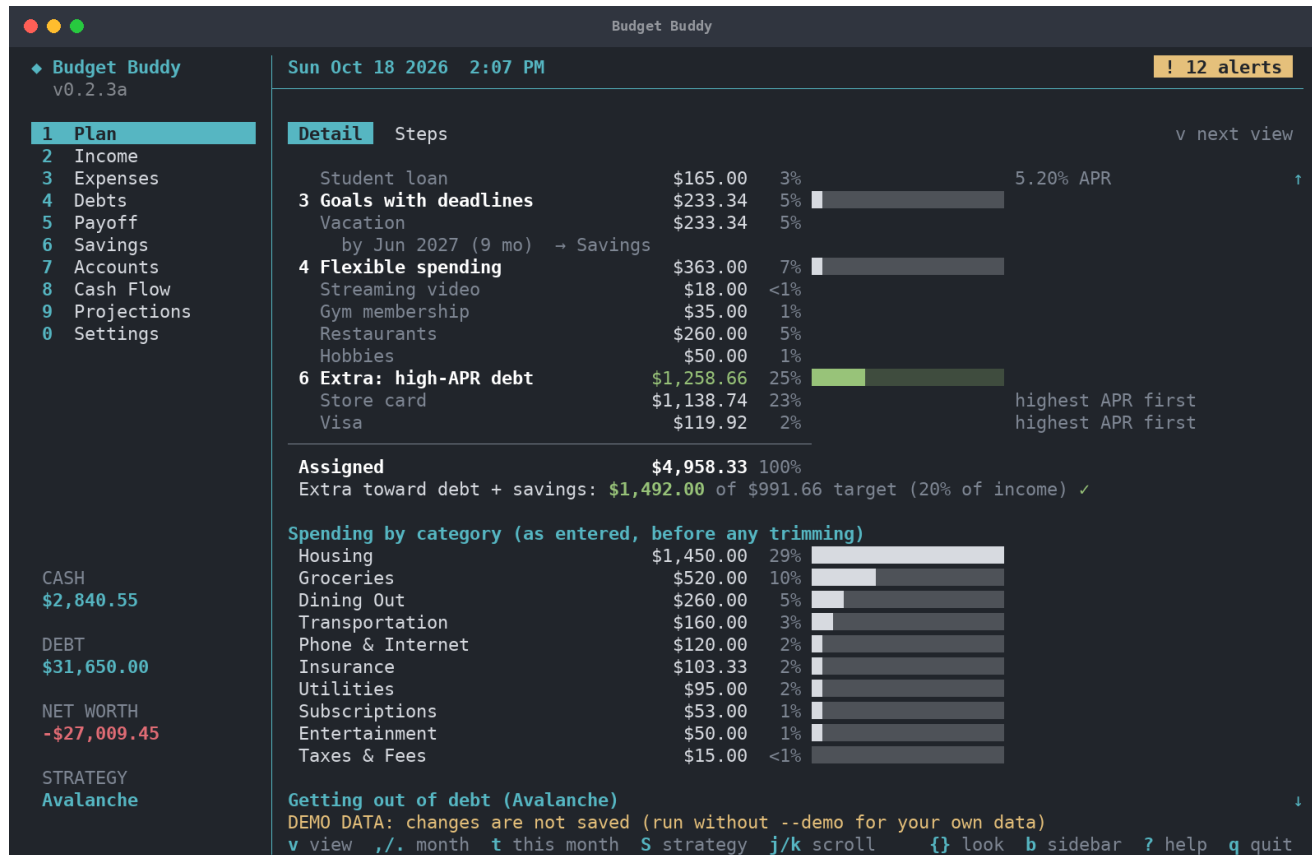


Figure 38. The lower half of the Plan: the end of the spending steps, the total, spending by category, the debt-free dates and savings goals.

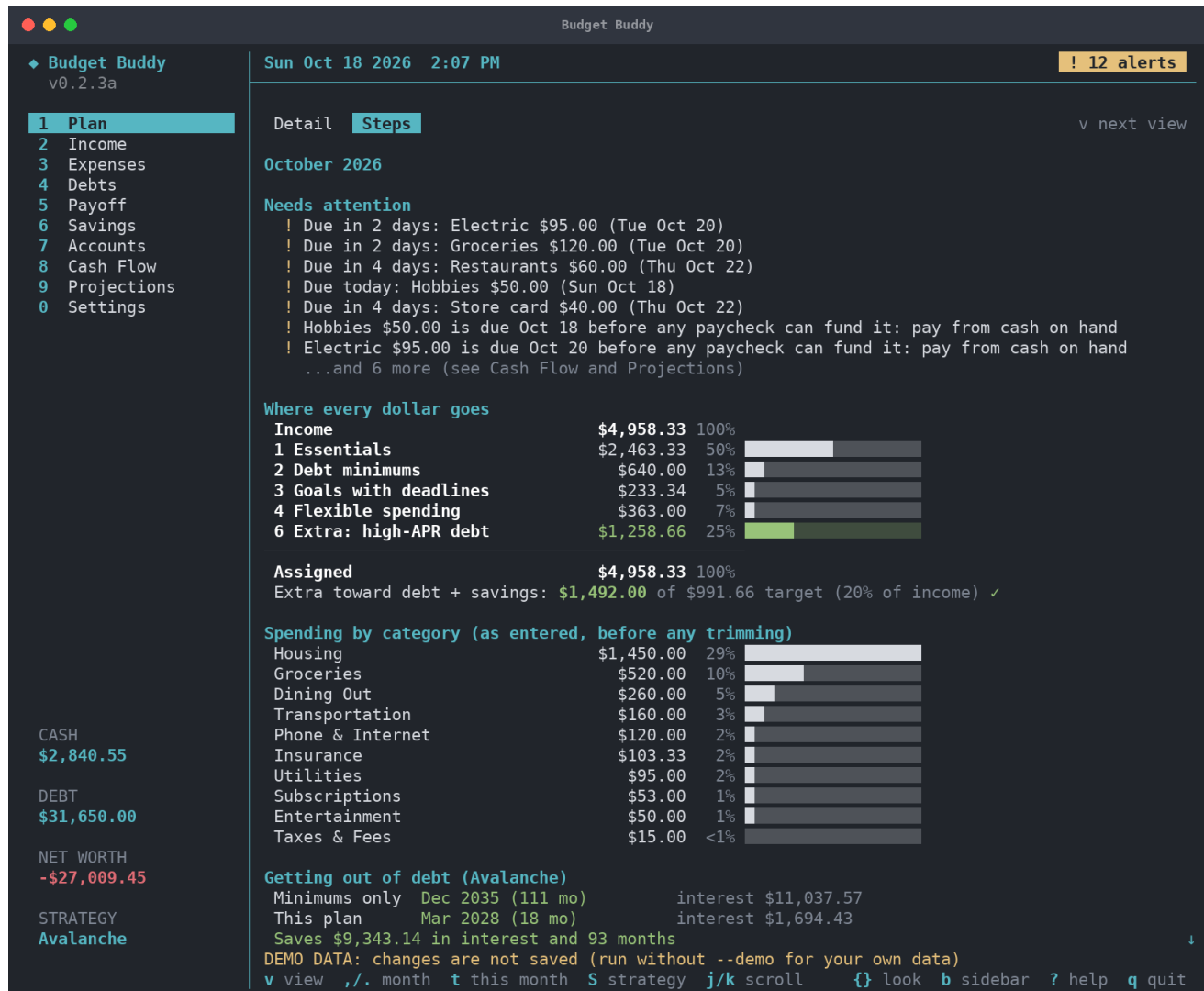


Figure 39. The Steps view (**v**) leaves out the line items so that the whole plan fits on one screen. The alerts and the sidebar figures are the same in both views.

Tip. Press **s** on this screen to try Snowball: the extra money goes to the smallest balance first instead of the highest rate (within the high-interest group; see section 8.4). The Payoff page (Chapter 9) shows what the choice costs or saves.

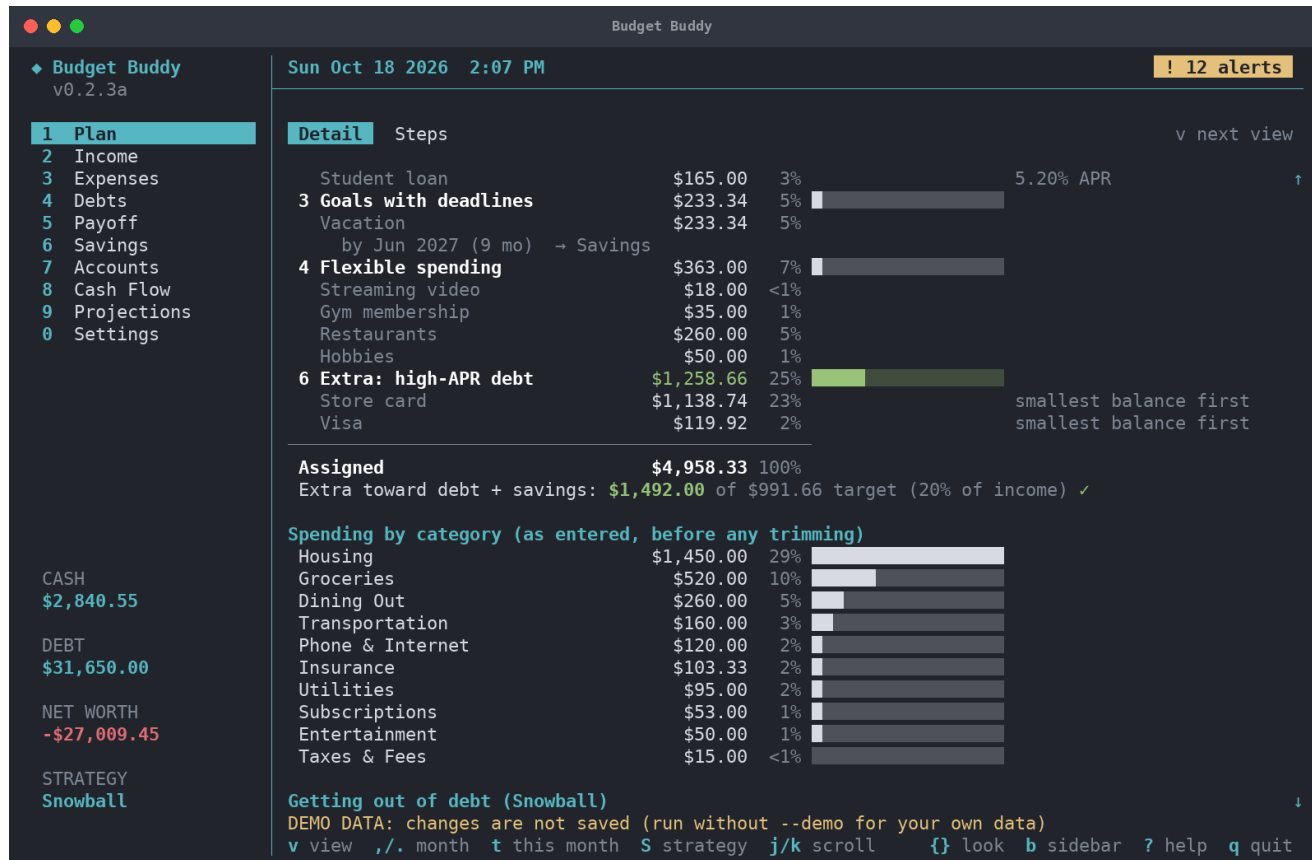


Figure 40. The same budget under Snowball. With these particular debts the amounts come out the same, because the two highest-rate debts are also the two smallest; only the labels (“smallest balance first”) and the strategy name change.

5.3 Spending by category

“Spending by category (as entered, before any trimming)” groups your bills by category with a bar for each, largest first. It is what your bills add up to, so it does not include trimming or what you actually bought; for that, use the Daily spending and Allotments pages (Chapter 7).

5.4 Getting out of debt

Two lines compare the future: **Minimums only** (what happens if you never pay more than the minimum) and **This plan** (minimums plus the extra from step 6), each with a debt-free month and the total interest, followed by what the plan saves you. The Payoff page shows the details.

5.5 Savings goals

One line per goal: what is saved, the percentage of the target, a progress bar, this month's planned contribution and the deadline or estimated finish. A goal with no target (an emergency fund set to “automatic”) shows the amount the plan is aiming for. If you have no emergency-fund goal yet, the plan uses a virtual one and says so (“not tracked yet”); add a real one on the Savings page to track the real balance.

5.6 Notes

When the plan has to make a trade-off it says so at the bottom under **Notes**. The messages you may see are:

- A debt's minimum **does not cover its monthly interest**, so the balance grows unless you pay extra.
- Essentials, minimums and deadline goals **exceed income** by some amount: flexible spending is cut to zero and nothing is left over.
- **Flexible spending was trimmed** by some amount (and a percentage) to free money for debt and savings.
- Only a small amount is left for extra debt and savings, below your **target**, because flexible spending is already at the trim limit.
- **No emergency-fund goal yet**: add one under Savings.

5.7 Other months

`.` moves to the next month and `,` to the previous one. The plan is recalculated for that month: one-time income dated inside it counts, recurring items are shown at their monthly equivalent, and the monthly amount a deadline goal needs is worked out with the months that remain. The heading says (*not the current month; press t to return*), and the alerts block is hidden because it only applies to today.

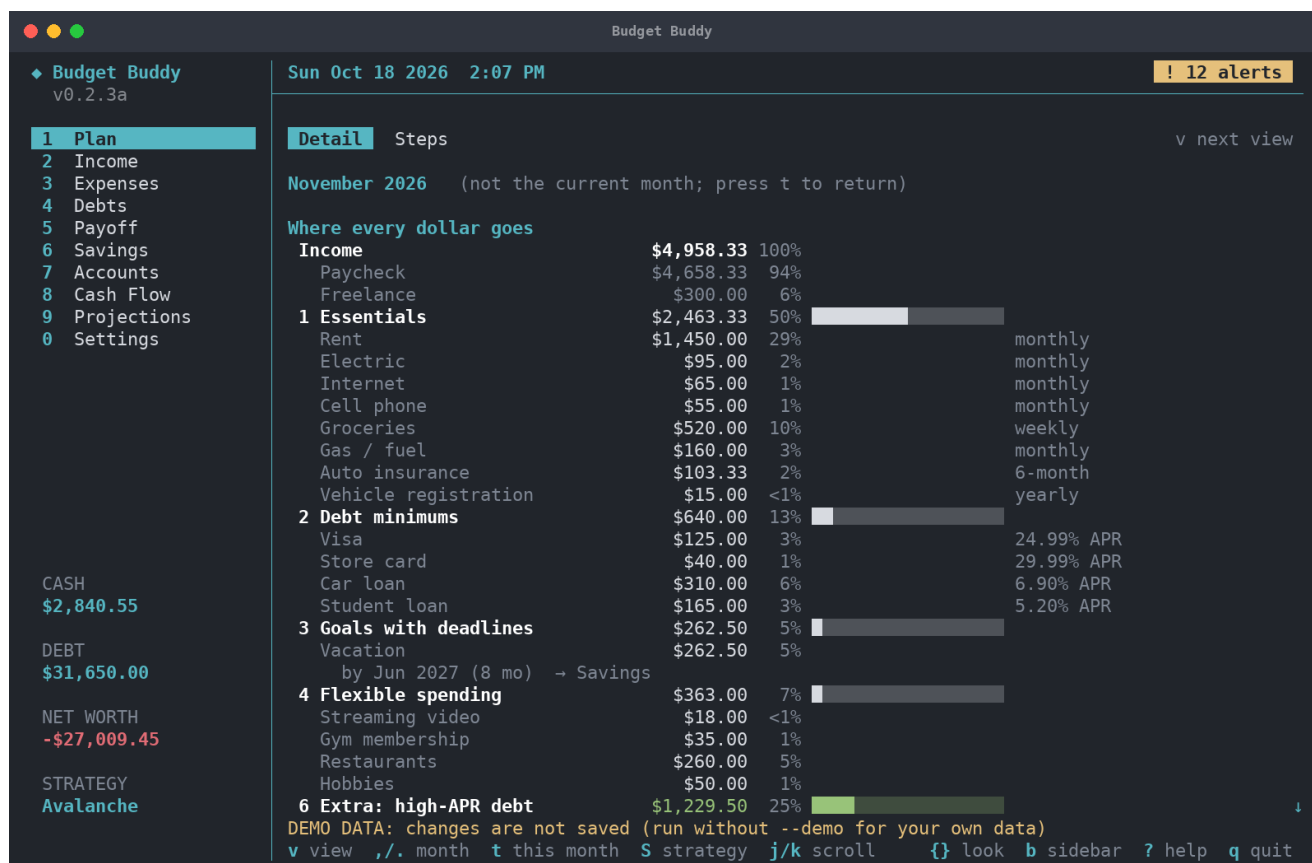


Figure 41. Next month's plan. Note the heading, and that the Vacation goal now has one fewer month to run.

6. Income

The Income page (**2**) lists every source of money: paychecks, freelance work, benefits, a one-time bonus. Budget Buddy needs two things from each: how much, and when. From those it forecasts your cash, plans which paycheck funds which bill, and works out your average monthly income for the Plan.

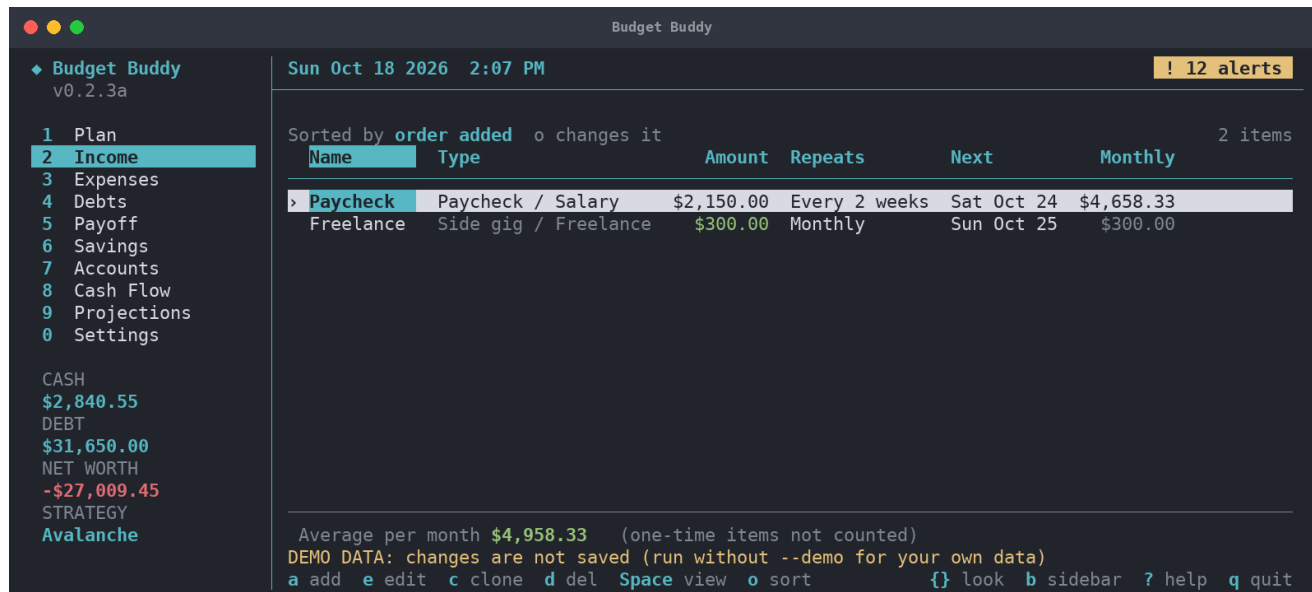


Figure 42. The Income page.

Column	Meaning
Name	What you call it.
Type	The kind of income (Appendix B lists them).
Amount	One payment, after tax.
Repeats	How often it arrives.
Next	The next date it arrives. Recurring income keeps moving forward on its own; you do not have to update it.
Monthly	The average monthly equivalent (see below).

Below the table, **Average per month** adds up the Monthly column. One-time items (a bonus, a tax refund) are not counted in the average: they count only in the month they arrive.

6.1 Adding and editing income

Press **a** to add, **e** / **Enter** to change the highlighted row, or **c** to start a new row as a copy of it (the name is left empty). **Space** shows every field of a row. The form has these fields:

Field	What to enter
Type	Pick from the menu: Paycheck / Salary, Hourly / Weekly pay, Side gig / Freelance, Bonus, Tax refund, Rental income, Benefits, Pension, Alimony / Child support, Interest & Dividends, Gift, Other. The type fills in a sensible name and how often it normally arrives.
Name	Any name. The menu offers suggestions for the type.
Amount	The amount of <i>each</i> payment, after taxes.
How often	One time, Weekly, Every 2 weeks, Twice a month, Monthly, Quarterly, Every 6 months or Yearly.
Next pay date	When the next payment arrives (any of the date forms in Chapter 3).
Second pay day	Only for <i>Twice a month</i> : the day of the month of the second payment. 31 means the last day of the month.

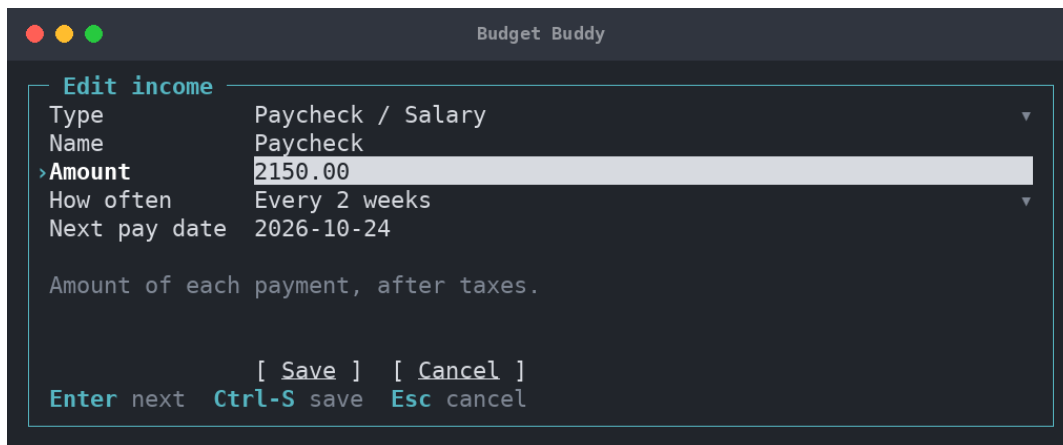


Figure 43. Editing an income. The form is the same as adding one, with the current values filled in.

6.2 How income is counted

Recurring income is converted to a monthly equivalent so that every month of the plan looks the same:

Repeats	Monthly equivalent
Weekly	$\text{amount} \times 52 \div 12$
Every 2 weeks	$\text{amount} \times 26 \div 12$ (so two months a year hold a third paycheck, and the average already includes them)
Twice a month	$\text{amount} \times 2$
Monthly	amount
Quarterly, Every 6 months, Yearly	$\text{amount} \div 3, 6 \text{ or } 12$
One time	not averaged; counted only in the month it arrives

In the sample budget, a \$2,150 paycheck every two weeks is \$4,658.33 a month, and \$300 of monthly freelance work brings the average to \$4,958.33.

Note. A paycheck you get twice in one month does not change the plan, because the plan uses the average. The Cash Flow and Projections pages, which work day by day, do use the real dates.

6.3 Deleting income

Press **d** or **Delete**; Budget Buddy asks you to confirm. Deleting an income changes the Plan, Cash Flow and Projections immediately.

The page is a table like the other lists, so **o**, **/**, **h** / **l** and the rest of the keys in section 3.2 work here too. It has a single view, so there is no views bar.

7. Expenses: Bills, Daily Spending and Allotments

The Expenses page (**3**) has five **views**, named on the bar at the top of the page: **Bills** (things that come due on a date), **Daily spending** (the small purchases you make every day), **By category** (what you have spent in each category this month), **Allotments** (how much you plan to spend in each category per month, and how you are doing) and **Suggested** (what Budget Buddy would set for each). **v** steps to the next view and **v** to the previous one, wrapping around; you can also click a view's name. Every view takes **,** and **.** for the previous and next month where a month matters, and **t** for this month.

7.1 Bills

Budget Buddy
v0.2.3a

Sun Oct 18 2026 2:07 PM ! 12 alerts

Bills Daily spending By category Allotments Suggested v next view 12 items

Sorted by **order added** o changes it

Name	Need	Amount	Repeats	Next due	In	Monthly
> Rent	Essential	\$1,450.00	Monthly	Sun Nov 1	14d	\$1,450.00
Electric	Essential	\$95.00	Monthly	Tue Oct 20	2d	\$95.00
Internet	Essential	\$65.00	Monthly	Thu Nov 12	25d	\$65.00
Cell phone	Essential	\$55.00	Monthly	Sun Nov 15	28d	\$55.00
Groceries	Essential	\$120.00	Weekly	Tue Oct 20	2d	\$520.00
Gas / fuel	Essential	\$160.00	Monthly	Wed Oct 28	10d	\$160.00
Auto insurance	Essential	\$620.00	Every 6 months	Tue Dec 15	58d	\$103.33
Vehicle registration	Essential	\$180.00	Yearly	Thu Jan 21	95d	\$15.00
Streaming video	Flexible	\$18.00	Monthly	Mon Nov 9	22d	\$18.00
Gym membership	Flexible	\$35.00	Monthly	Tue Nov 3	16d	\$35.00
Restaurants	Flexible	\$60.00	Weekly	Thu Oct 22	4d	\$260.00
Hobbies	Flexible	\$50.00	Monthly	Sun Oct 18	today	\$50.00

Per month: essential \$2,463.33 flexible \$363.00 total \$2,826.33
 DEMO DATA: changes are not saved (run without --demo for your own data)
 a add e edit c clone d del p paid R set aside {} look b sidebar ? help q quit

Figure 44. The Bills page.

Column	Meaning
Name, Category	What it is and what it is filed under (Appendix B lists categories).
Need	Essential or Flexible. Essential bills are paid first by the plan; Flexible ones can be trimmed.
Amount, Repeats	One payment, and how often.
Next due, In	The next due date and the days until it. A bill due within a week is shown in yellow; one that is late shows in red as “3d late”.
Set aside	Money you have already put aside for the next payment (section 7.4).
Pay from	The account the bill normally comes out of.
Monthly	The average monthly cost. A weekly \$120 grocery bill is \$520 a month; a \$620 six-monthly insurance bill is \$103.33.

Under the table, **Per month** totals the essential bills, the flexible bills and everything together. A narrow window drops the right-hand and least important columns, and a **>** at the edge of the headings says so; **Space** shows every field of the highlighted bill, and **h** / **l** move the current column so that a hidden one scrolls into view (Chapter 2).

Adding and editing a bill

Press **a** to add, **e** / **Enter** to edit, or **c** to start a new bill as a copy of the highlighted one (its name is left empty). The first field is the category, because picking it fills in most of the rest.

Figure 45. A new, empty bill form. The hint explains the category field.

Field	What to enter
Category	Housing, Utilities, Phone & Internet, Groceries, Transportation, Insurance, Healthcare, Childcare & Kids, Education, Taxes & Fees, Pets, Subscriptions, Dining Out, Entertainment, Shopping, Personal Care, Travel, Gifts & Donations or Other. The category sets the Priority and frequency to the usual values, and the Name menu to typical names.
Name	Any name; suggestions are offered.
Amount	Amount of each payment.
Priority	Essential: always paid first. Flexible: can be trimmed to free money for debt and savings.
How often	One time, Weekly, Every 2 weeks, Twice a month, Monthly, Quarterly, Every 6 months or Yearly.
Next due date	The next date it is due.
Second due day	Only for <i>Twice a month</i> : the day of the month of the second payment.
Pay from	The account it normally comes out of; used as the default when you mark it paid.

A bill due on the 31st really falls on the last day of shorter months and goes back to the 31st afterwards. Press **d** or **Delete** to delete a bill (you are asked first).

7.2 Marking a bill paid

When you pay a bill, record it with **p**. The dialog shows the amount (the bill's amount, which you can change if the actual payment differed) and the account it came from.

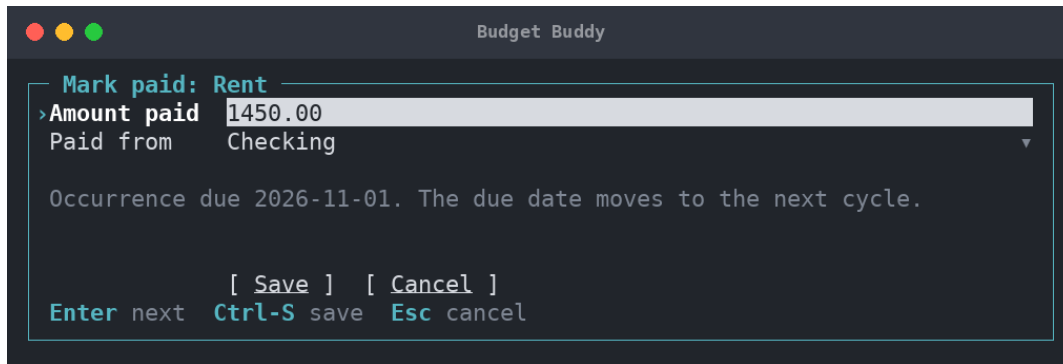


Figure 46. Marking Rent paid. The due date moves to the next cycle.

- The bill's **due date moves to the next occurrence**. A one-time bill has nothing further to pay.
- If you choose an account in **Paid from**, the amount is withdrawn from it and appears in that account's history as "Paid *name*", so the balance stays right. Choose (*don't adjust accounts*) if you paid from somewhere Budget Buddy does not track.
- Money you had set aside for the bill is reduced by what you paid.
- The bill counts as **spent** in the allotment for its category, in the month it was due (section 7.6).

7.3 Deleting and fixing mistakes

Deleting a bill removes it from the plan immediately. If you mark a bill paid by mistake and then delete the matching account-history entry (section 12.5), only the account is corrected; the bill's due date does not move back, so set it with **e** if that matters.

7.4 Setting money aside for a bill

If you have already put money aside for the next payment of a bill, press **R** and enter the amount. That money reduces what the Projections page asks you to move from your paychecks. It is shown in the **Set aside** column, and it is used up (reduced by the amount you pay) when you mark the bill paid.

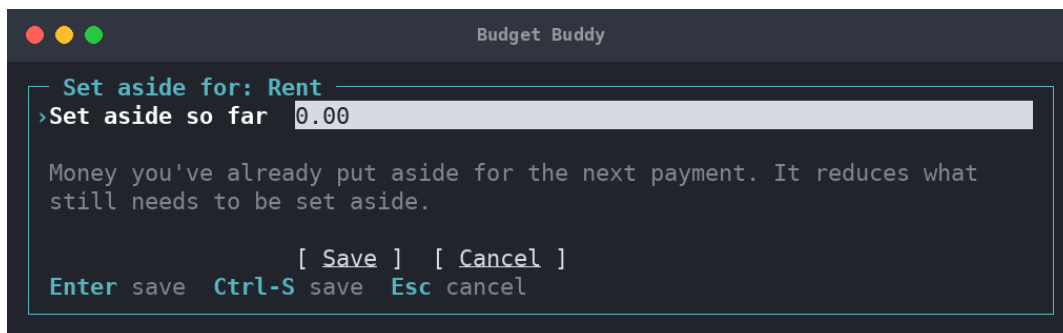


Figure 47. Recording money already set aside for Rent.

7.5 Daily spending

Bills are things with a due date. For everything else (coffee, groceries, gas) use the **Daily spending** ledger. Switch to the **Daily spending** view with **v**.

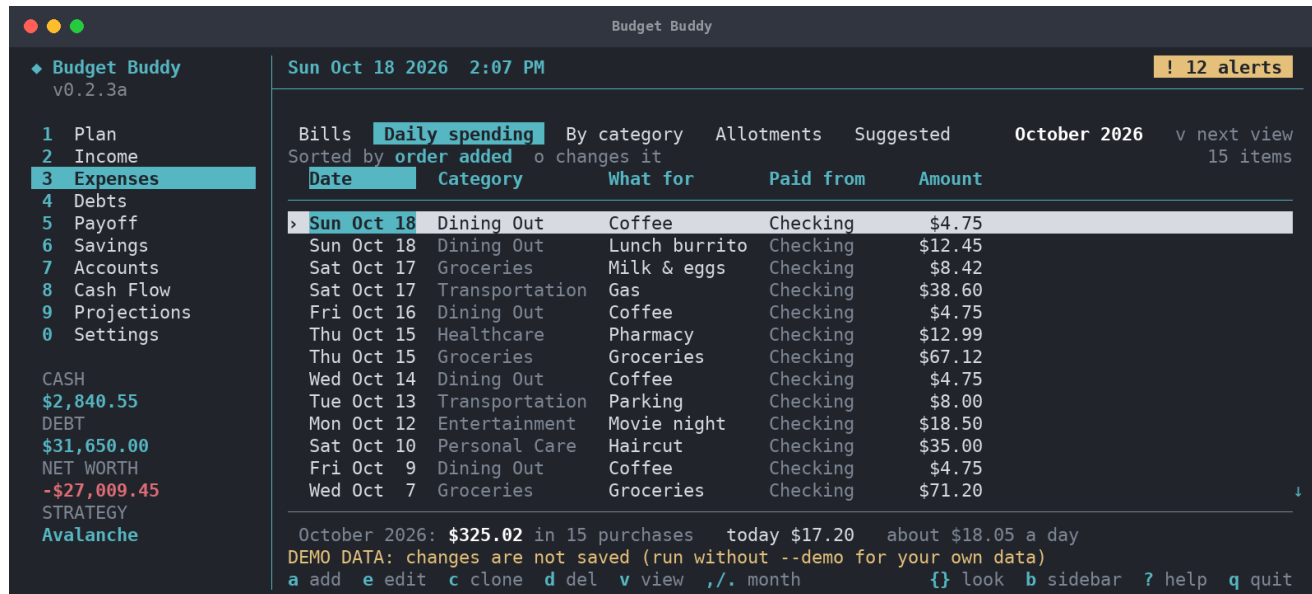


Figure 48. The Daily spending ledger, newest first.

The line at the bottom totals the month: how much you have spent, how many purchases, what you spent today and the average per day so far.

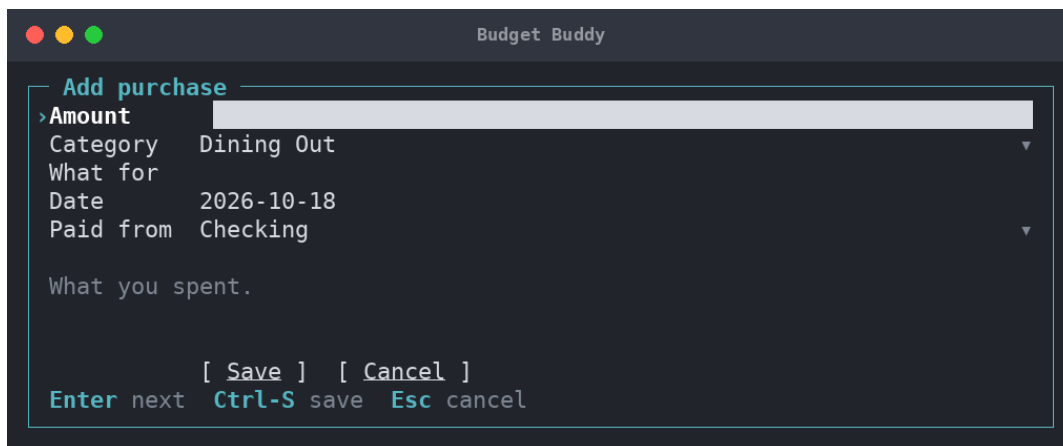


Figure 49. Adding a purchase. The date starts at today.

Field	What to enter
Amount	What you spent. It must be more than \$0; record a refund as a deposit on the account instead.
Category	The same categories as bills.
What for	Optional. The menu suggests things you bought before in that category, with the last amount for reference, and accepts anything you type.
Date	Defaults to today. Back-dated purchases land in the account history on their own date.
Paid from	The account it came out of. Choose (cash / not tracked) for money that never touched a bank account.

A new purchase starts in the category and account of the previous one, so entering a run of purchases takes little more than typing the amount and pressing **ctrl-s**.

Key	Action
a	Add a purchase.
e / Enter	Edit the highlighted purchase.
c	Clone it: a new purchase that starts as a copy, dated today.
d	Delete it (asks first).
, / .	Previous / next month. The month is shown at the right of the views bar.
t	Back to this month.
v / V	The next / previous view (the next one is the totals by category, below).
o o / f	Sort and search (Chapter 18).

Totals by category

The **By category** view, the next one along (**v**), shows what you have spent this month in each category next to what your bills plan for it:

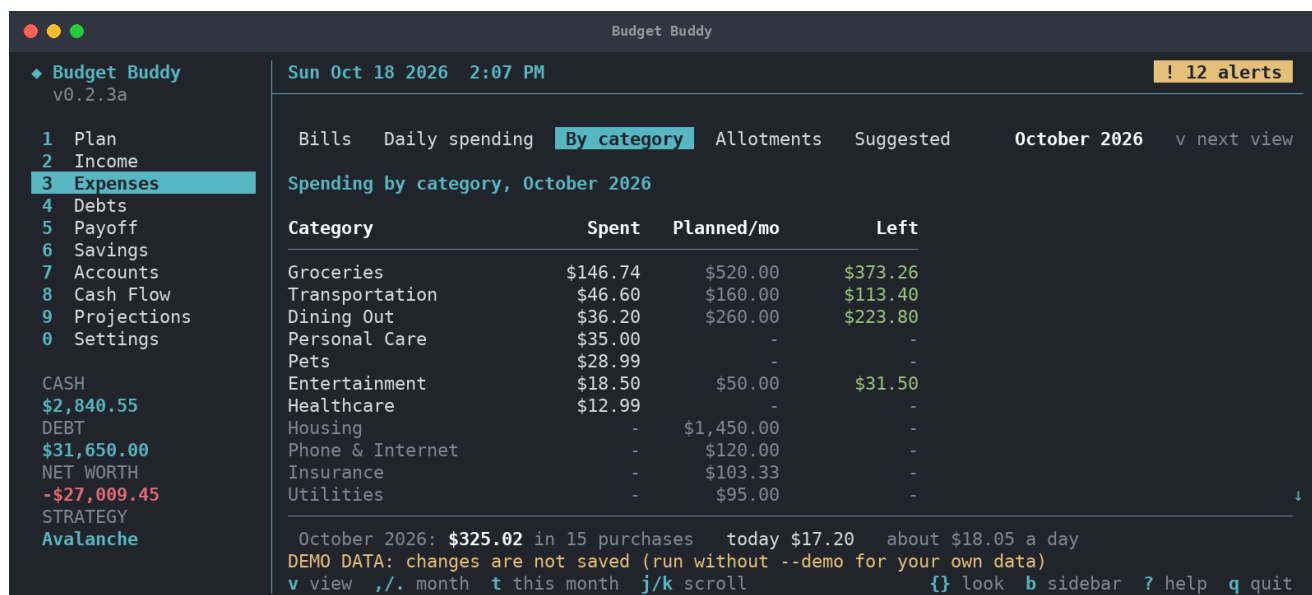


Figure 50. Spending by category for October.

This view is a summary and not a list of rows you can select, so **a**, **e**, sorting and searching do not apply; **v** takes you back to the daily purchases, and **,** / **.** change the month. **Spent** is this month's purchases. **Planned/mo** is what the bills in that category add up to (weekly items count 52/12 weeks a month). **Left** is the difference, or over \$X in red when you have gone past it. Rent and utilities are paid as bills, so they usually show no purchases here.

How purchases affect your accounts

A purchase taken from an account is posted to that account's **history** straight away (it shows as *spend* in the From column, Chapter 12), so the balance drops by the amount. Editing or deleting the purchase corrects the account too. Deleting the entry from the account history removes the purchase from the ledger, so the two can never disagree.

7.6 Allotments

An **allotment** is the most you plan to spend in one category in a month: say \$500 on groceries, \$200 on dining out. Two of the views belong to it: **Allotments** (what you have set, and how you are doing) and **Suggested** (what Budget Buddy would set); **v** and **v** move between them and the others.

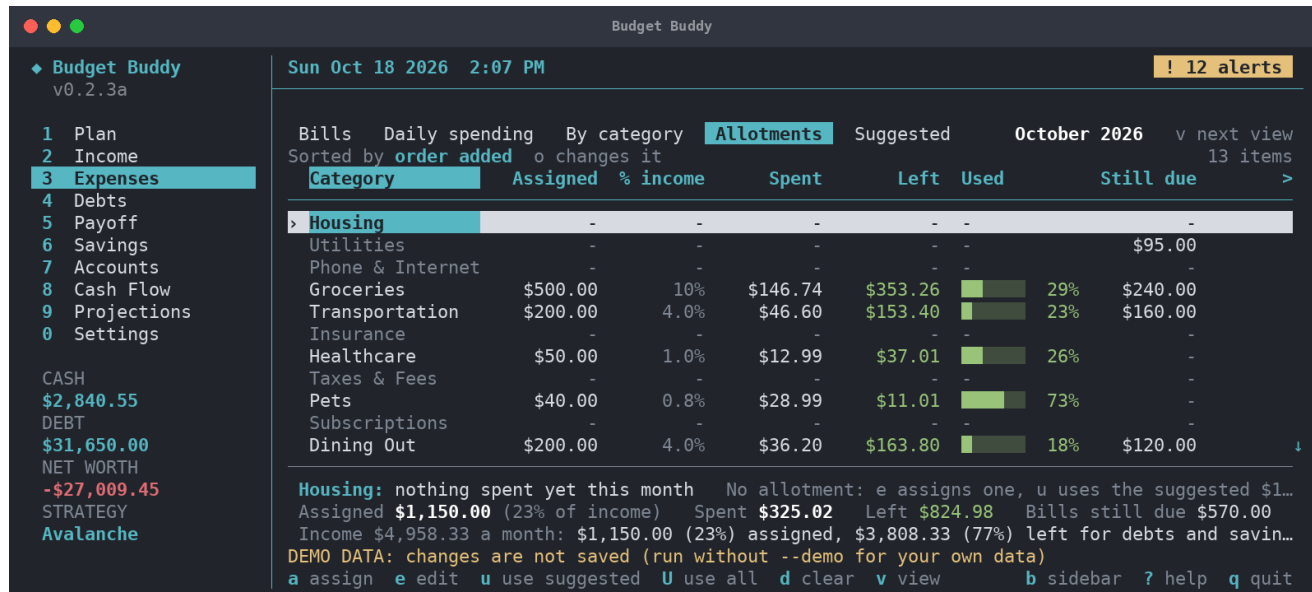


Figure 51. The Allotments view: one row for each category that has an allotment, spending, bills or a suggestion.

Column	Meaning
Assigned	The monthly amount you set. - means you have not set one; \$0.00 is a real allotment that means "budget nothing here".
% income	That amount as a share of your average monthly income (recurring income only).
Spent	What has been spent in the category this month: purchases plus bills marked paid.
Left	Assigned minus spent: green while there is room, yellow once 85% is used, and over \$X in red.
Used	A bar and the percent of the allotment spent.
Still due	Bills in the category that fall due this month and are not yet marked paid.
Purchases, Bills paid	The two parts of <i>Spent</i> .

Narrow terminals drop the right-hand columns first. The lines under the table always describe the highlighted category ("-\$240.00 more due in 2 bills"), the totals, and how the allotments compare with your income.

What counts as spent

- Every purchase in the **Daily spending** ledger dated in the month.
- Every **bill** due in the month that you have marked paid, counted at the bill's amount **in the month it is due**, not the day you paid it. A weekly bill counts once per due date, so a five-Thursday month counts five.

Careful. If you record the same money twice, such as a weekly Groceries *bill* marked paid and the same shopping trips in *Daily spending*, it counts twice. The Purchases and Bills paid columns show where each dollar came from, which makes that easy to spot.

Setting and changing allotments

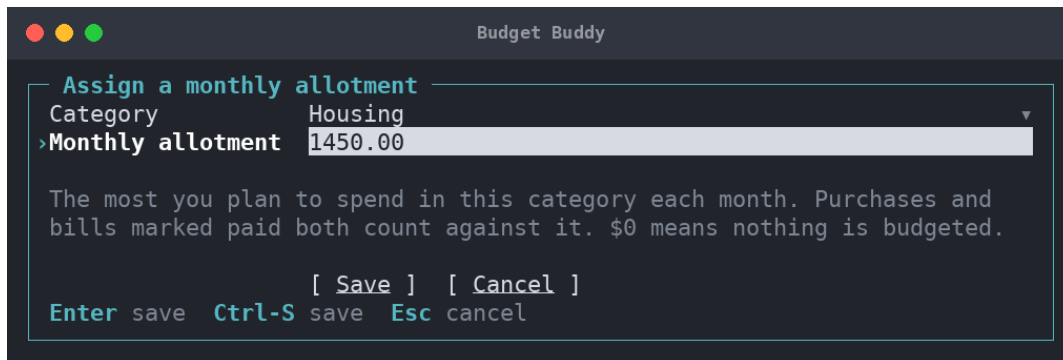


Figure 52. Assigning a monthly amount to a category.

Key	What it does
a	Assign: pick a category (the list shows what is assigned or suggested) and type the amount. It starts on the highlighted category.
e / Enter	Change the highlighted category's allotment. On the Suggested view the amount starts at the suggestion, so you can tune it first.
u	Use the suggestion for the highlighted category right away.
U	Use the suggestion for every category that has none yet . Allotments you set yourself are never replaced.
d	Clear the highlighted category's allotment (asks first).
v / V	Switch between the Allotments and Suggested views; the same category stays highlighted.
, / .	Previous / next month, t for this month.
o O / f	Sort and search; each view keeps its own.

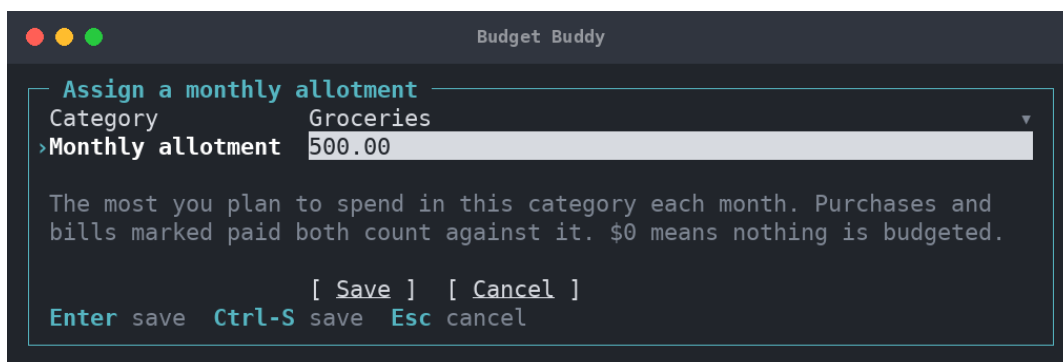


Figure 53. Editing an existing allotment.

Allotments are standing monthly amounts: the same amount applies to whichever month you look at, so an earlier month shows that month's spending against today's allotment. Nothing carries over from month to month.

Suggested amounts

Budget Buddy							
Sun Oct 18 2026 2:07 PM ! 12 alerts							
Bills	Daily spending	By category	Allotments	Suggested	October 2026	v next view	
Sorted by order added o changes it							
Category	Suggested	% income	Assigned	Why	vs assigned	Bills/mo	>
Housing	\$1,450.00	29%	-	your bills	-	\$1,450.00	
Utilities	\$95.00	1.9%	-	your bills	-	\$95.00	
Phone & Internet	\$120.00	2.4%	-	your bills	-	\$120.00	
> Groceries	\$520.00	11%	\$500.00	your bills	+\$20.00	\$520.00	
Transportation	\$160.00	3.2%	\$200.00	your bills	-\$40.00	\$160.00	
Insurance	\$104.00	2.1%	-	your bills	-	\$103.33	
Healthcare	-	-	\$50.00	no history yet	-	-	
Taxes & Fees	\$15.00	0.3%	-	your bills	-	\$15.00	
Pets	-	-	\$40.00	no history yet	-	-	
Subscriptions	\$53.00	1.1%	-	your bills	-	\$53.00	
Dining Out	\$260.00	5.2%	\$200.00	your bills	+\$60.00	\$260.00	

Groceries: its bills come to \$520.00 a month.
 Suggested total \$2,827.00 (57% of income) Leaves \$2,131.33 (43%) for debts and savings
 The plan funds \$2,826.33 for spending. u use this U use all that have no allotment
 DEMO DATA: changes are not saved (run without --demo for your own data)
 a assign e edit u use suggested U use all v view ,/. month b sidebar ? help q quit

Figure 54. The Suggested view, with the reason for each figure.

Column	Meaning
Suggested, % income	The suggestion, rounded up to a whole dollar, and its share of income.
Assigned, vs assigned	What you have now, and how far the suggestion is from it (+\$20.00 means the suggestion is higher).
Why	<i>your bills</i> : what the bills in the category add up to. <i>bills, trimmed 12%</i> : the same, cut back the way the Plan trims flexible spending. <i>3-month average</i> : what you actually spent per month over up to the last three earlier months, used when that is more than the bills. <i>no history yet</i> : nothing to base it on.
Bills/mo, Avg spent	The two numbers the suggestion comes from.

The average only looks at months since your first recorded purchase, so months before you began tracking do not drag it down, and it never uses the current month, which is not finished. The lines under the table compare the suggested total with your income and show what would be left for debts and savings.

8. Debts

The Debts page (**4**) holds everything you owe: credit cards, loans, medical bills, a loan from a friend. For each one Budget Buddy needs the balance, the interest rate and the required payment. It then adds each minimum to the plan, directs extra money to the debts that cost you the most, and projects how fast they disappear.

Budget Buddy
v0.2.3a

Sun Oct 18 2026 2:07 PM ! 12 alerts

Sorted by **order** added o changes it 4 items

Name	Balance	APR	Minimum	Next due	In	This month	Set aside
> Visa	\$4,800.00	24.99%	\$125.00	Sun Nov 8	21d	\$244.92	-
Store card	\$1,150.00	29.99%	\$40.00	Thu Oct 22	4d	\$1,178.74	-
Car loan	\$11,200.00	6.90%	\$310.00	Sat Nov 14	27d	\$310.00	-
Student loan	\$14,500.00	5.20%	\$165.00	Tue Oct 27	9d	\$165.00	-

CASH
\$2,840.55
DEBT
\$31,650.00
NET WORTH
-\$27,009.45
STRATEGY
Avalanche

Total \$31,650.00 interest about \$255.93/mo Avalanche: extra goes to Store card
DEMO DATA: changes are not saved (run without --demo for your own data)
a add e edit c clone d del p payment R set aside {} look b sidebar ? help q quit

Figure 55. The Debts page.

Column	Meaning
Name, Type	What it is called, and its kind (credit card, auto loan, student loan... Appendix B).
Balance	What you owe now.
APR	The annual percentage rate.
Minimum, Repeats	The required payment each time, and how often.
Next due, In	The next due date and the days until it (yellow within a week; red and “late” when overdue).
This month	What the plan pays this month: the minimum plus any extra money it assigns to this debt. A small card can show far more than its minimum when it is first in line for the extra (here the Store card shows \$1,178.74, enough to clear it).
Set aside	Money you have already put aside for the next payment.

The line below the table gives the total owed, the approximate interest you are being charged each month, and which debt the extra money goes to under the current strategy.

8.1 Adding and editing a debt

Press **a** to add a debt, **e** / **Enter** to edit one, or **c** to start a new debt as a copy of the highlighted one (the name is left empty). **Space** shows every field of a debt, including any column the window has no room for. The fields:

Field	What to enter
Type	Credit card, Store card / Buy-now-pay-later, Auto loan, Student loan, Mortgage, Personal loan, Home equity / HELOC, Medical bill, Tax debt, Family / friend loan, Payday loan or Other.
Name	Any name; suggestions are offered.
Current balance	What you owe right now. Update it from your statement whenever you like.
Interest rate (APR)	For example 24.99 or 24.99%. Use 0 for an interest-free loan.
Minimum payment	The required payment each time it is due.
How often	Usually Monthly; the other schedules are available.
Next due date	When the next payment is due.
Second due day	Only for <i>Twice a month</i> .
Pay from	The account the payment normally comes out of.

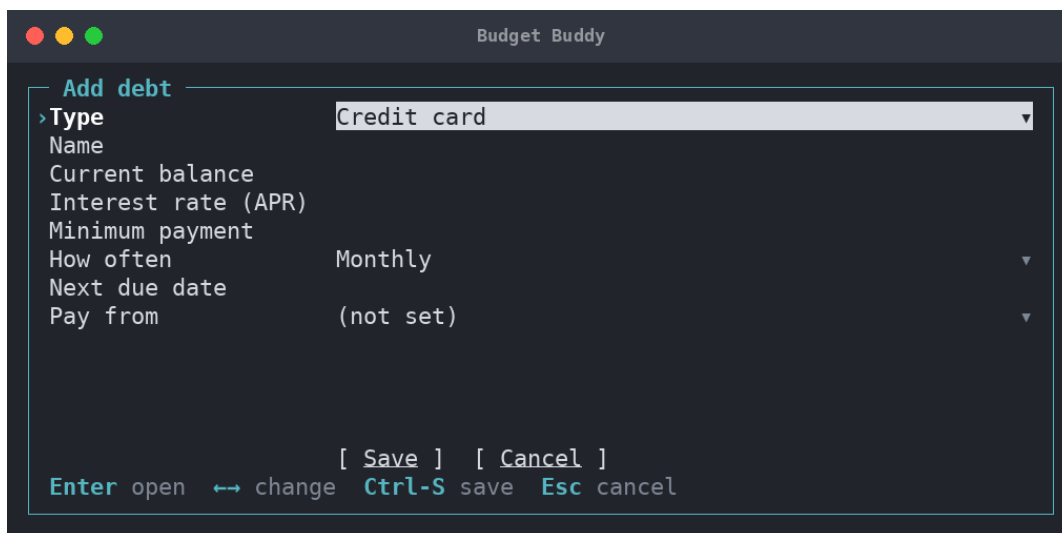


Figure 56. The debt form.

Note. If a debt's minimum does not even cover its monthly interest, the plan says so under Notes (section 5.6): the balance will grow unless you pay more.

8.2 Recording a payment

Press **p** when you make a payment. The dialog suggests the amount the plan pays this month (minimum plus any extra), and you can change it to what you actually paid.

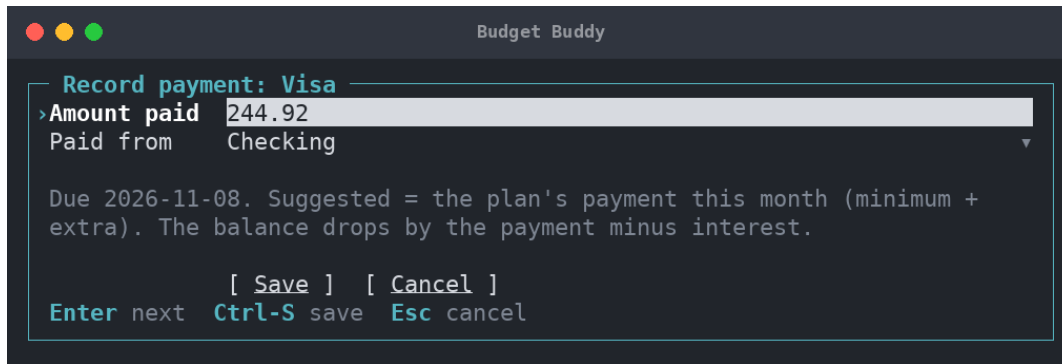


Figure 57. Recording a payment on the Visa card. The suggestion is the plan's payment for this month.

- The payment is split for you: **interest** for one month is taken off first, and the **rest reduces the balance**. So a \$244.92 payment on a card that accrues \$99.96 of interest lowers the balance by \$144.96.
- The due date moves to the next occurrence.
- If you choose an account in **Paid from**, the amount is withdrawn from it and logged in its history as “Paid name”.
- Money you had set aside for the payment is reduced by what you paid.

Careful. Deleting the account-history entry for a payment only corrects the account. It does not restore the debt's balance or move its due date back; edit the debt if you need to.

Tip. Your statements are the final word on what you owe. Budget Buddy models interest monthly from the APR, so every so often press **e** and type the balance from your statement to bring it back in line.

8.3 Setting money aside

As with bills, **R** records money you have already put aside for the next payment, so that the Projections page asks for less.

8.4 Strategy

S switches the debt strategy, and it works on the Debts, Plan and Payoff pages. (In earlier versions it was a lowercase **s**; that key now sorts, like it does on every list.) The choice only changes *where the extra money goes*; minimums are always paid.

Strategy	Extra money goes to	Why choose it
Avalanche	The debt with the highest APR first.	Costs the least interest overall.
Snowball	The debt with the smallest balance first.	Clears whole debts sooner, which many people find motivating.

Debts at or above the **high-interest threshold** (8% APR by default; see Settings) always receive the extra money first, under either strategy. The strategy then decides the order *within* that group, and within the lower-rate group after it. So a 24% card is never left waiting behind a 3% loan just because the loan is smaller.

The setting is also on the Settings page. Chapter 9 shows what each strategy would cost.

9. Payoff

The Payoff page (**5**) answers: *when will I be debt-free, and what does it cost?* It plays the future forward month by month under three scenarios and shows a full schedule you can scroll through. The views bar at the top of the page names its views: **All debts**, then one view for each of your debts.

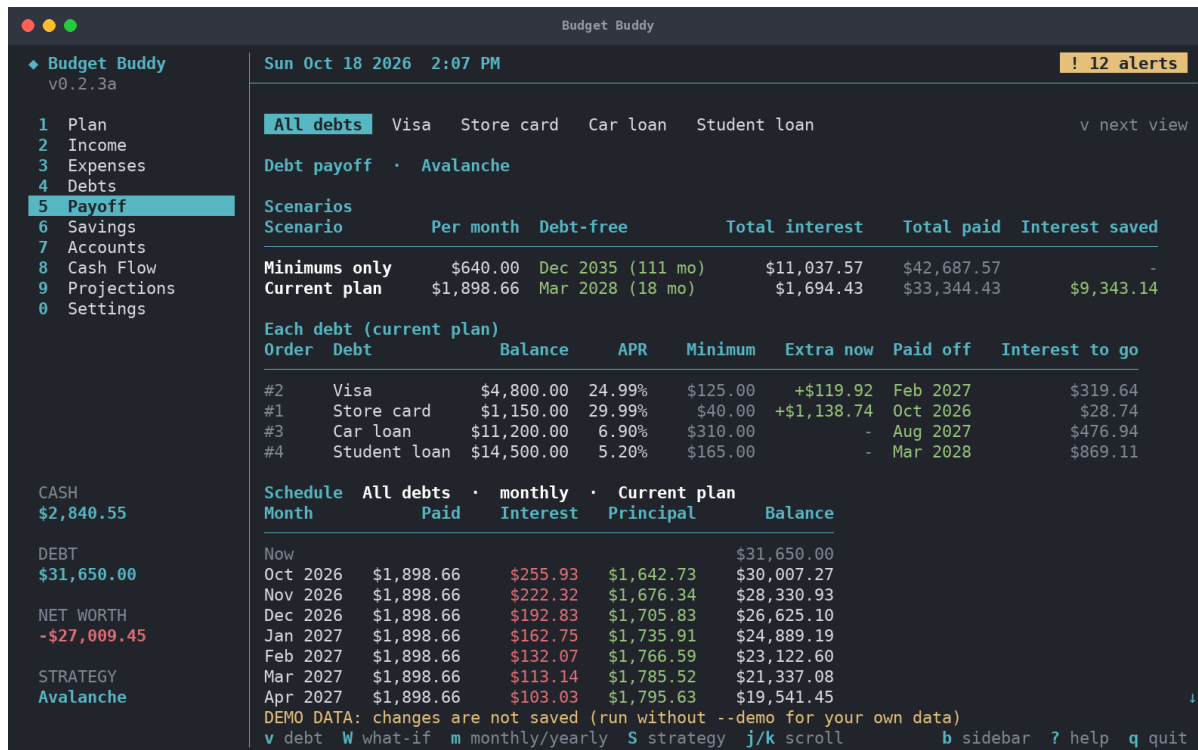


Figure 58. The Payoff page with the sample debts.

9.1 Scenarios

Scenario	What it assumes
Minimums only	You pay only the minimum on every debt and never more.
Current plan	You pay the minimums <i>plus</i> the extra the Plan sets aside for debt each month, following your strategy. As debts are cleared, their payments roll into the next one.
What-if	Appears when you ask for it (section 9.4): the current plan with extra money on top.

For each scenario the table shows the **payment per month**, the **debt-free month** and how many months that is, the **total interest**, the **total paid** and the **interest saved** compared with minimums only. In the sample budget, following the plan saves \$9,343.14 in interest and more than seven years.

9.2 Each debt

The second table lists every debt in the order the extra money reaches it (**Order #1** first), with its balance, APR, minimum, the **extra it receives now**, the month it will be **paid off** and the **interest still to go**. In the screenshot, Store card is #1 (the highest APR), so it gets \$1,138.74 extra and is cleared this month; Visa is #2.

9.3 The schedule

Beneath the tables is a month-by-month schedule: how much is paid, how much of that is interest, how much reduces the balance, and what is left. It starts as **All debts**; press **v** to move to the next debt's schedule and **v** to go back (a click on a debt's name works too), and the tables above keep the current scenario. **m** changes between monthly and yearly rows.

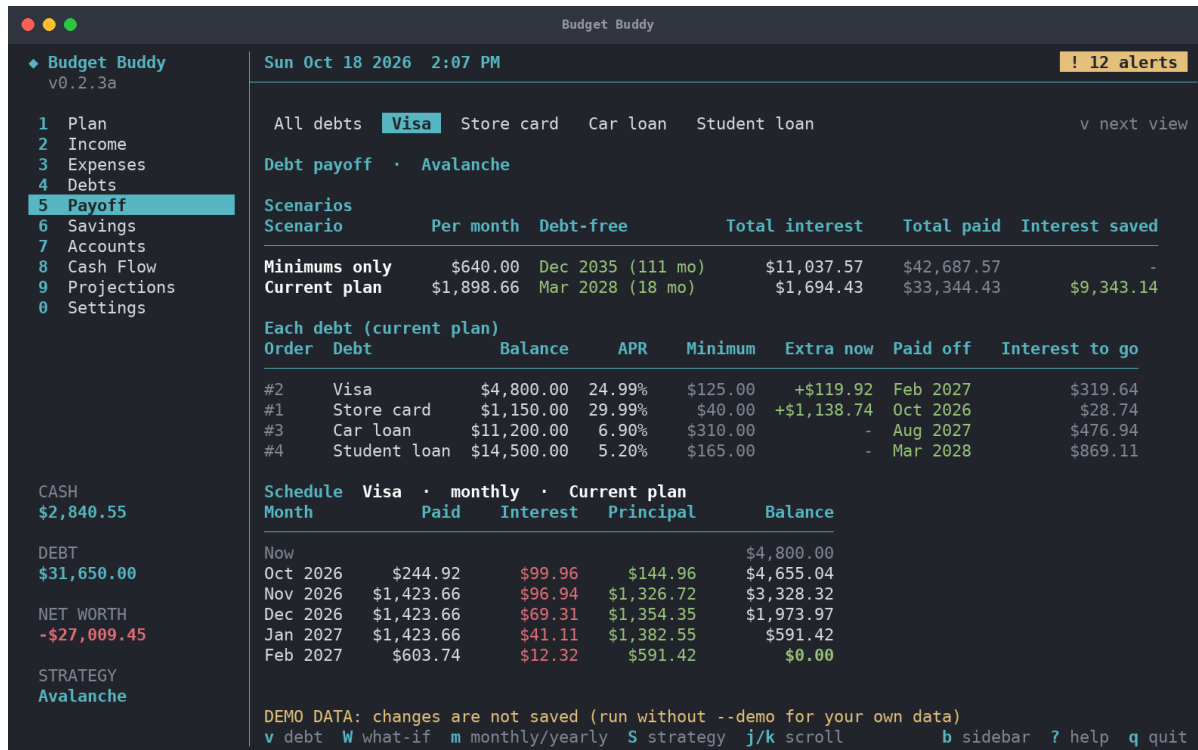


Figure 59. The schedule for one debt (the Visa). It receives the extra money after the Store card is cleared.

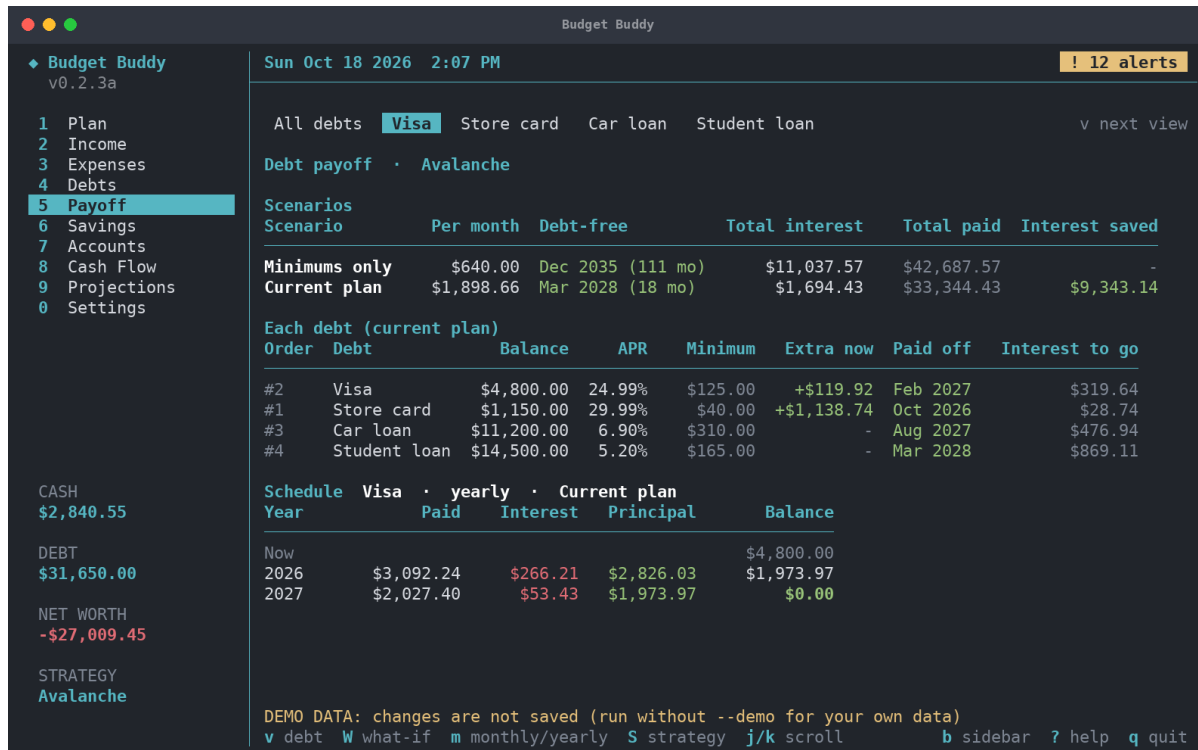


Figure 60. The yearly view: one row for each year.

Key	Action
W	What-if: enter extra money per month.
m	Switch the schedule between monthly and yearly.
v / V	Show one debt's schedule: next / previous (cycling through All debts). . and , do the same, as they did in earlier versions.
S	Switch strategy between Avalanche and Snowball.
j k PgUp PgDn	Scroll.

9.4 What if I paid more?

Press **W**, type an extra amount per month, and press **ctrl-S**. The extra is applied to your debts in strategy order, on top of the plan. Entering 0 clears it.

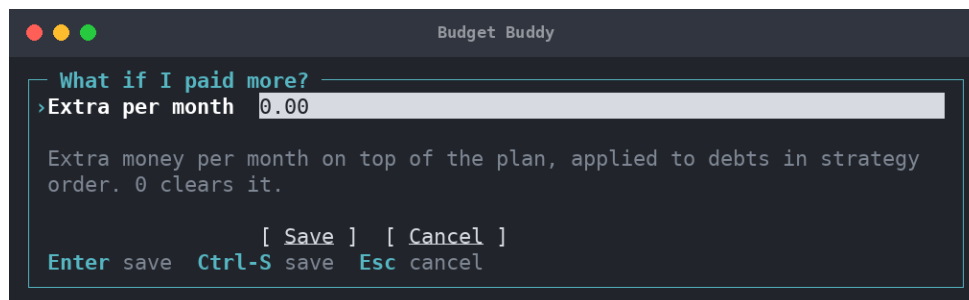


Figure 61. The what-if dialog.

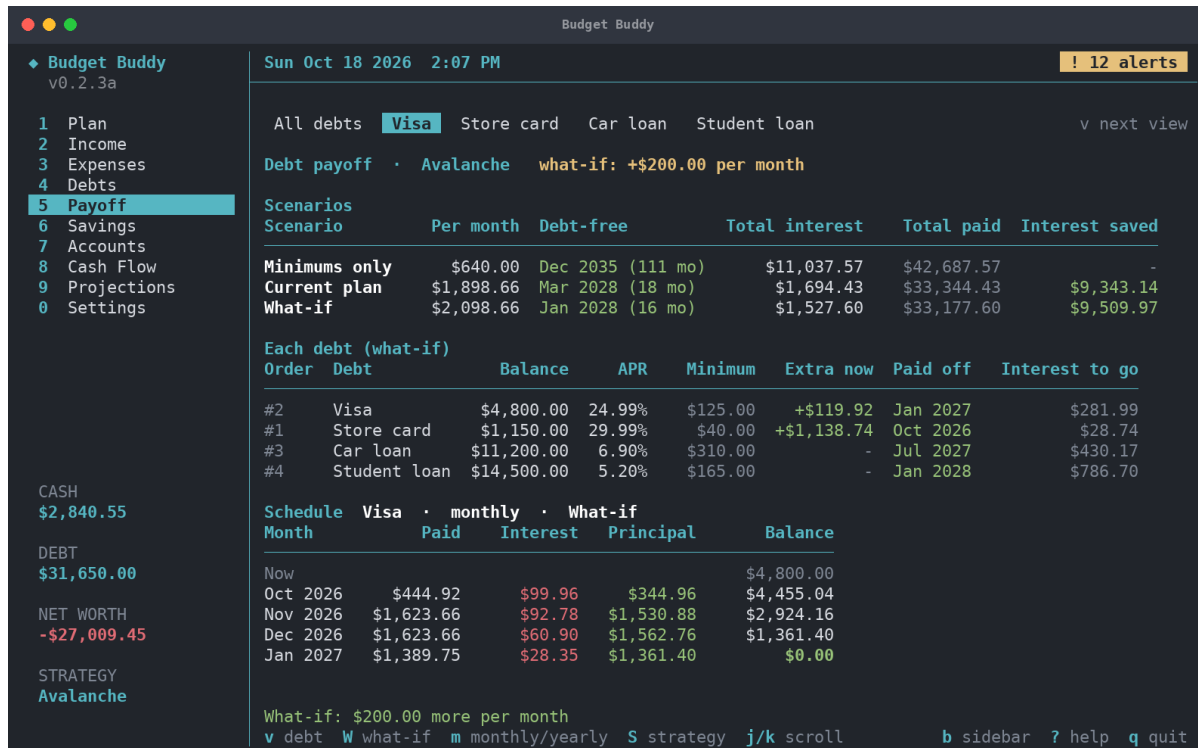


Figure 62. An extra \$200 a month brings the debt-free date forward two months and saves about \$167 more in interest. The extra row and the message line show what is being assumed.

The what-if only changes this screen; it does not change your plan or any of your data, and it is not saved: it lasts until you quit the program.

9.5 Choosing a strategy

Because **s** works here too, you can compare Avalanche and Snowball on the same screen: press **s**, look at the **Current plan** row, press **s** again. For the same payments, Avalanche never costs more interest, and Snowball can clear individual debts sooner. Because debts above the high-interest threshold are always first (section 8.4), the two often give the same or very similar results; they differ most when you have several debts of similar rates but very different sizes. Neither is better in the abstract: use the one you will stick to.

10. Savings Goals

The Savings page (**6**) tracks what you are saving for, how far along you are, and what the plan sets aside for it each month.

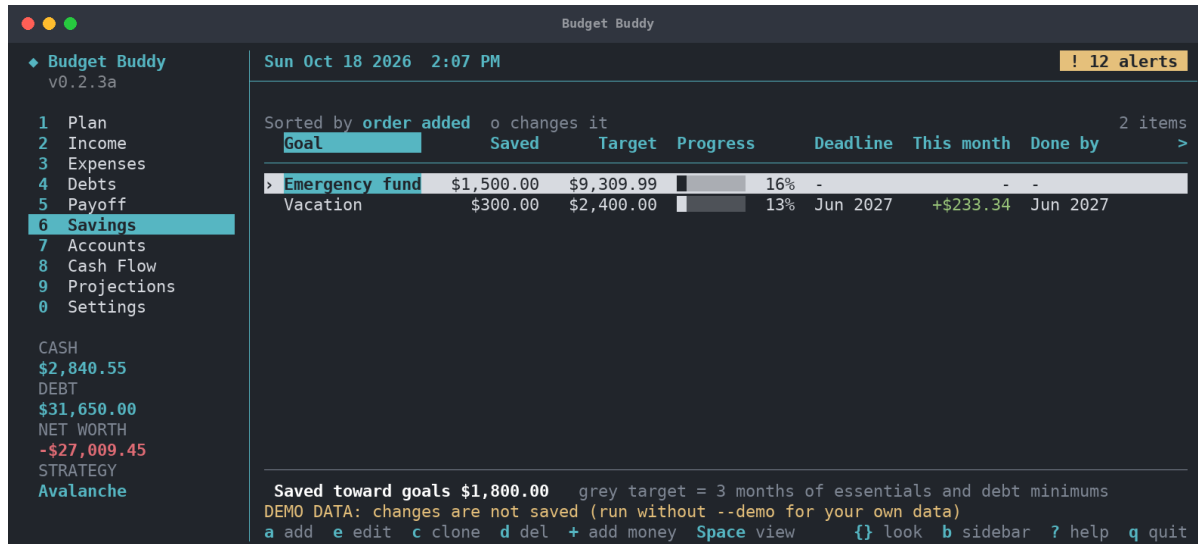


Figure 63. The Savings page.

Column	Meaning
Goal, Type	The name and kind of goal (Appendix B).
Saved	What you have so far.
Target	The amount you are aiming for. A target of 0 is open-ended; for an emergency fund it means <i>automatic</i> : the plan aims for a number of months of essentials and debt minimums (3 by default) and shows the result.
Progress	A bar and percentage. For an automatic emergency fund the grey part of the bar marks the target.
Deadline	When you want to be done, if you set one.
This month	What the plan contributes to the goal this month.
Done by	The estimated month you will reach the target at the current pace.
Held in	The account where the money is kept.

Below the table, **Saved toward goals** adds up what you have saved across all goals.

10.1 Adding and editing a goal

Press **a** to add a goal, **e** / **Enter** to edit the highlighted one, or **c** to start a new goal as a copy of it (the name is left empty). **Space** shows every field of a goal, including any column the window has no room for.

Field	What to enter
Type	Emergency fund, Vacation / Travel, Home down payment, Car / Vehicle, Retirement, Education, Wedding / Event, Holiday gifts, Medical / Dental, Home repairs, Taxes, Big purchase or Other. <i>Emergency fund</i> gets special treatment (below).
Name	Any name.
Target amount	How much you want to have. 0 means open-ended (automatic for an emergency fund).
Saved so far	What you already have.

Field	What to enter
Deadline (optional)	Leave blank for none. With a deadline the plan sets aside the amount needed each month to be on time, and counts it in step 3 of the plan (Chapter 5). Without one, the goal is funded from the leftover money.
Priority	1 to 9, where 1 is funded first when several goals compete (5 is normal).
Held in	The account where the money is kept, or (<i>not tracked</i>). A goal held in an account counts as earmarked money for that account on the Accounts page.

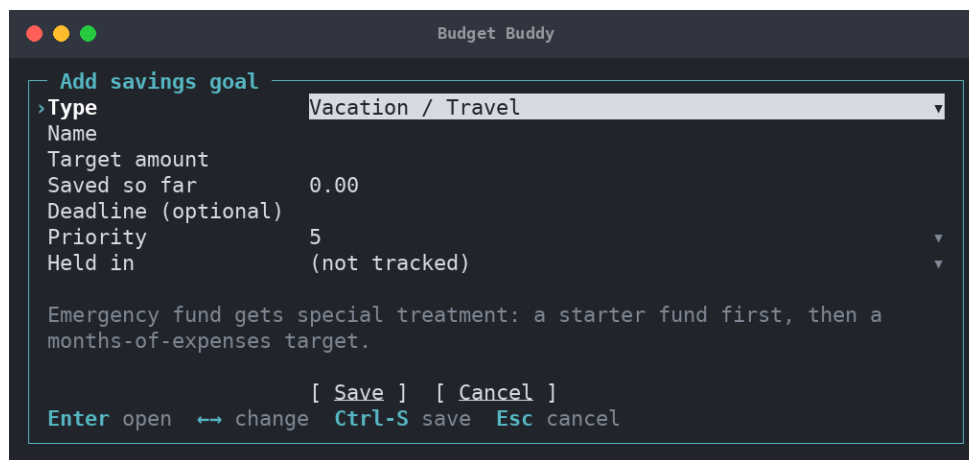


Figure 64. The goal form.

10.2 Adding money to a goal

When you actually move money into a goal, press **+** (or **=**, the same key without Shift). The dialog suggests this month's planned contribution. (In earlier versions this was **c**; **c** now clones a goal, as it does on every list.)

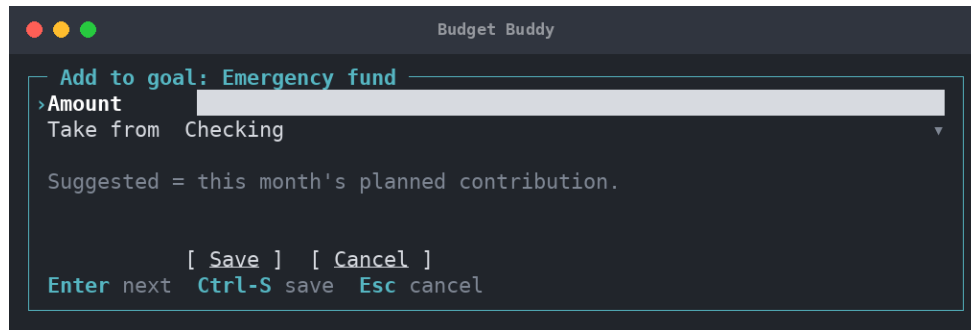


Figure 65. Adding money to the Emergency fund goal.

- **Take from** is the account the money comes out of.
- If the goal is **held** in another account, the money is **transferred** there, and the transfer is logged in both accounts' histories.
- If the goal is not held in an account, the money is simply withdrawn from the chosen account.
- If the goal is held in the same account the money comes out of, only the goal's balance changes.

10.3 The emergency fund

Budget Buddy treats an emergency fund in two stages. First, a **starter fund** (default \$1,000, see Settings) is built before any extra money goes to debt. After the high-interest debts are handled, the fund grows to your target of several months of essentials and minimums. If you have not created an emergency-fund goal, the Plan uses a virtual one and reminds you to add a real one so it can track your actual balance.

11. Accounts

The Accounts page (**7**) lists your bank accounts and is where you keep their balances right. It is also the doorway to each account's history (Chapter 12), to importing bank statements (Chapter 13) and to exporting (Chapter 14).

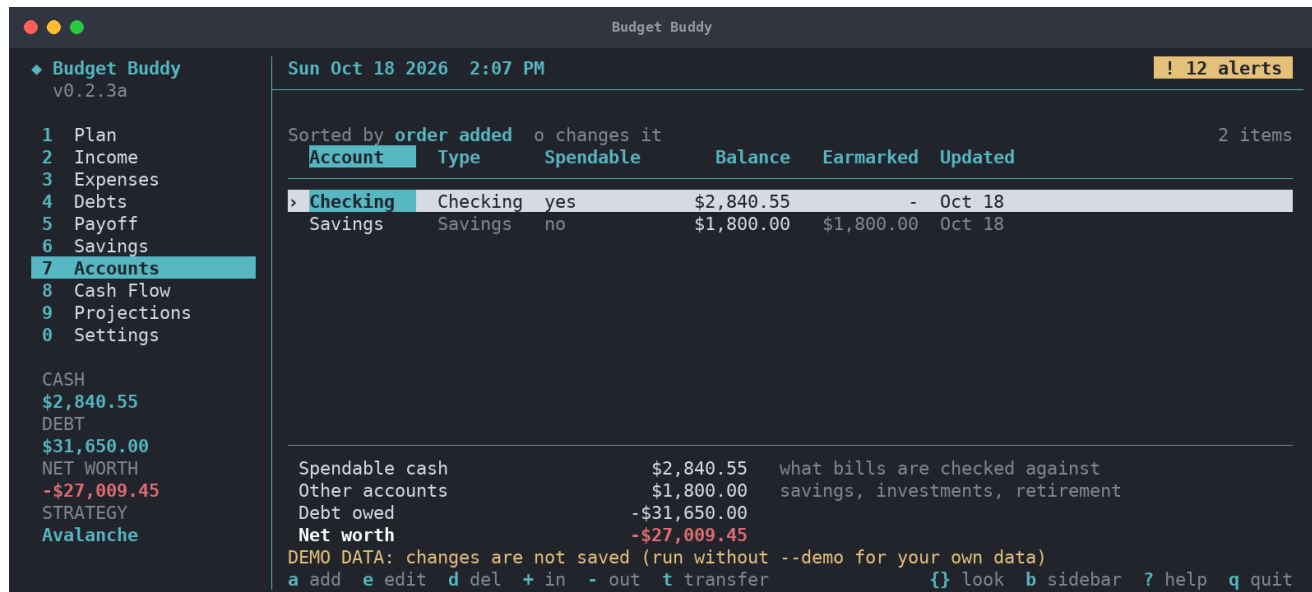


Figure 66. The Accounts page.

Column	Meaning
Account, Type	The name and kind (Checking, Savings, Cash on hand, Money market, Brokerage / Investments, Retirement account, HSA / FSA, Other).
Spendable	yes if the account counts as bill cash : its balance is the starting cash for the Cash Flow forecast and the Projections plan.
Balance	What the account holds.
Earmarked	How much of the balance is spoken for by savings goals that are <i>held in</i> this account.
Updated	The date the balance was last confirmed or changed.

The summary underneath shows **Spendable cash** (what bills are checked against), **Other accounts** (savings, investments and retirement), **Debt owed** and your **Net worth**, which is all account balances minus all debt.

Key	Action
a	Add an account.
e / Enter	Edit the account's name, type or bill-cash setting. (Not the balance; see below.)
c	Clone the account: a new account that starts with the same type and setting, with the name and balance left for you to fill in.
d / Delete	Delete the account and its history . You are asked first.
B	Set the balance to match your bank (section 11.3). Works on every page.

Key	Action
+ / -	Record a deposit / a withdrawal (section 11.4). = and _ are the same keys without Shift.
t	Transfer between two accounts (section 11.5).
Space or H	Open the account's history (Chapter 12).
u	Undo the last entry added (section 11.6).
i	Import a bank CSV (Chapter 13).
x	Export history to CSV (Chapter 14).
h / l / o / 0 / f	The current column, sorting and searching (Chapter 18).

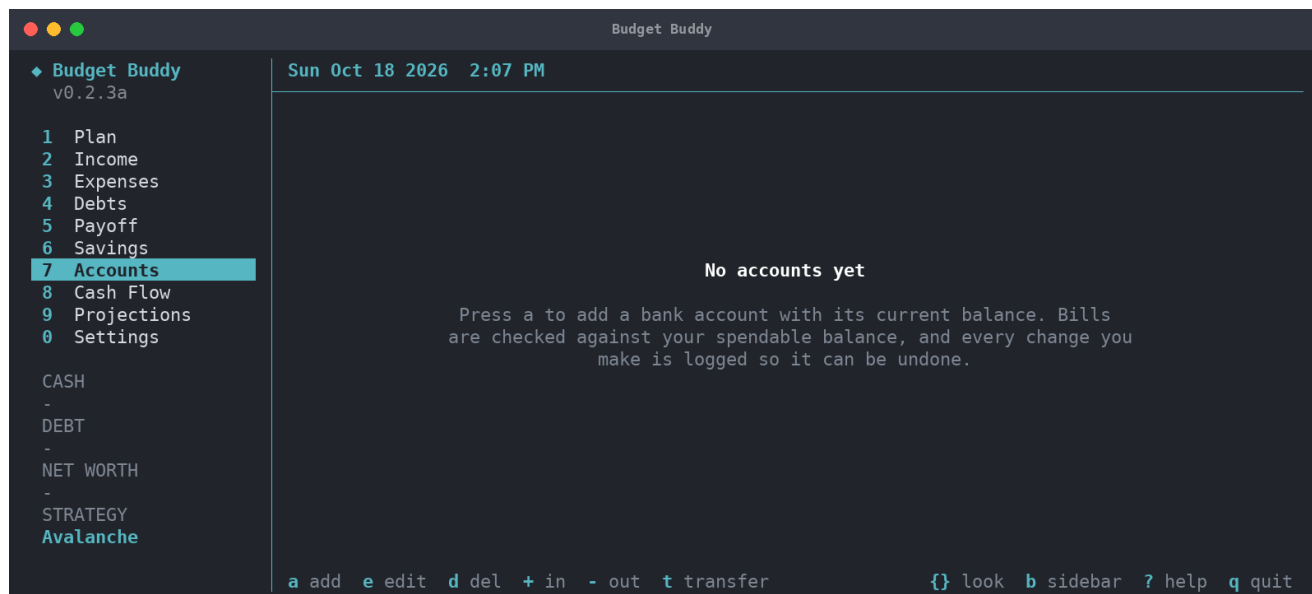


Figure 67. The Accounts page before any account exists.

Import, export and the history belong to this page: pressing **i** or **x** on another list tells you to go to Accounts.

11.1 Why balances change through entries

Budget Buddy never lets a balance change silently. When you add an account you type its opening balance, and after that every change is an **entry** in the account's history: a reconciliation, a deposit, a withdrawal, a transfer, a bill or debt payment, a daily purchase, or an imported bank line. That is why the edit form does not include the balance, and why the history always adds up.

11.2 Adding an account

Chapter 4 walks through the form: **Account type**, **Name**, **Current balance** and **Counts as bill cash**. Choose **Yes** for the money you pay bills from (checking, cash) and **No** for savings and investments, which you do not want the forecast to spend. The type sets a sensible default. Editing an account afterwards offers the same fields except the balance.

11.3 Setting a balance to match your bank

This is the most important routine in the program. Every so often, compare your account with your bank and tell Budget Buddy what the bank shows. Press **B** on the account, type the **Actual balance**, and add a note if you like. The difference is recorded in the history as an adjustment, so nothing is lost and the history still adds up.

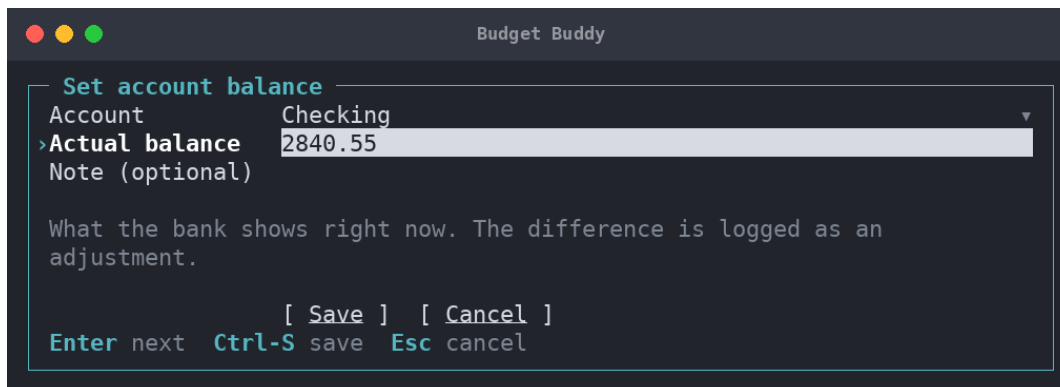


Figure 68. Setting an account's balance to what the bank shows.

You do not have to go to the Accounts page: press **B** on *any* page for the same dialog. On the Accounts page it starts on the highlighted account; elsewhere you pick the account in the dialog. (**b**, in lowercase, now turns the sidebar on and off.)

Tip. A spendable account that has not been updated for more than 14 days is flagged on the Plan (“Checking was last updated 16 days ago”). A quick reconcile before you read the Plan keeps the forecast trustworthy.

If there are no accounts yet, **B** opens the Add account form instead.

11.4 Deposits and withdrawals

Use **+** to record money that arrived and **-** for money that left, when neither is a bill, a purchase or a transfer: a refund, a cash withdrawal, a birthday check. Type the **Amount** and an optional **Note** (“Refund”, “Cash”).

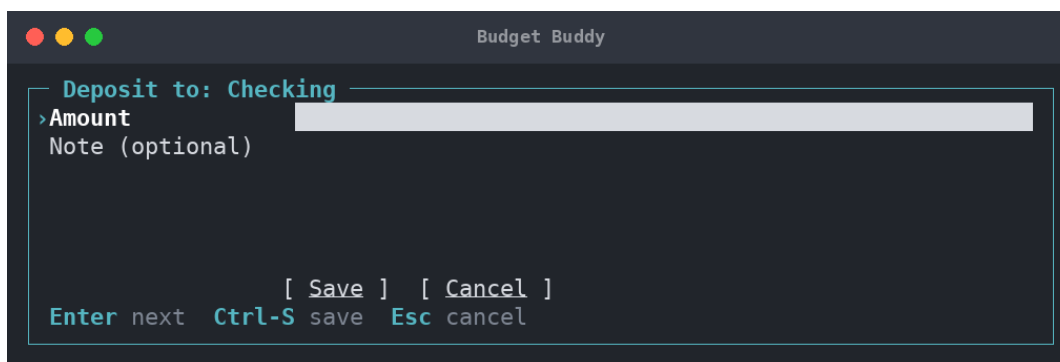


Figure 69. Recording a deposit.

11.5 Transfers

t moves money from one account to another, for example from Checking to Savings. It needs at least two accounts. Choose the **From** and **To** accounts, the **Amount** and an optional **Note**. Both sides are logged together, so the two histories agree.

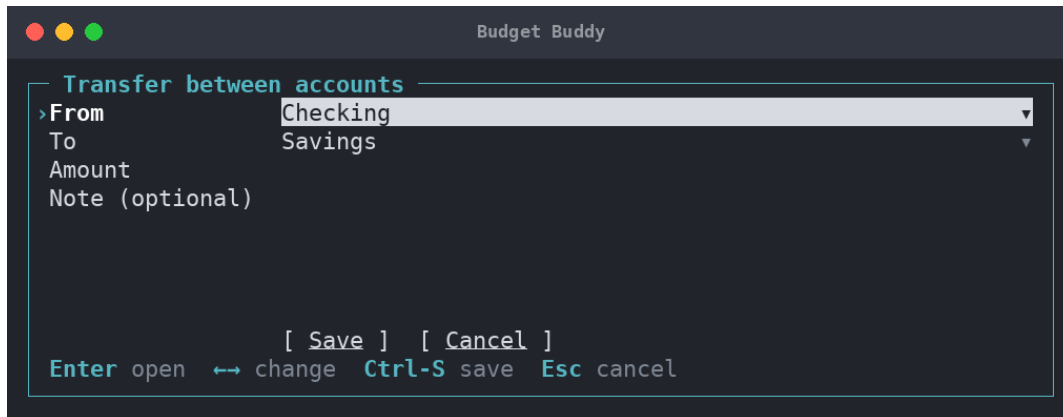


Figure 70. A transfer between accounts.

11.6 Undoing and deleting

- **Undo** (**u**) removes the entry most recently *added* to the account (a whole transfer at once), provided nothing was added after it. You are shown what will be undone and asked first.
- **Delete an entry** from the history (**d** there) removes any entry, for fixing a mistake from further back (section 12.5).
- **Delete the account** (**d** on this page) removes the account and its whole history. Bills and debts that named it as their *Pay from* account, and goals held in it, forget it, and daily purchases that came out of it stay in the ledger as cash / not tracked.

Careful. Deleting an account deletes its history for good. Export the history first (Chapter 14) if you may need it.

12. Account History

Every change to an account's balance is kept in its **history**. Press **Space** (or **H**) on an account to open it. It is a window over the Accounts page with the newest entries first and, along the bottom, a **totals line for exactly what is shown**: how many entries, the money in, the money out and the net.

Budget Buddy				
History: Checking		balance \$2,840.55		
/ filter	o sort	a all accounts	f full screen	x export
Date	Change	Balance	From	Note
> Sun Oct 18 2026	-\$4.75	\$2,840.55	spend	Spent: Coffee (Dining Out)
Sun Oct 18 2026	-\$12.45	\$2,845.30	spend	Spent: Lunch burrito (Dining Out)
Sat Oct 17 2026	-\$8.42	\$2,857.75	spend	Spent: Milk & eggs (Groceries)
Sat Oct 17 2026	-\$38.60	\$2,866.17	spend	Spent: Gas (Transportation)
Fri Oct 16 2026	-\$4.75	\$2,904.77	spend	Spent: Coffee (Dining Out)
Thu Oct 15 2026	-\$12.99	\$2,909.52	spend	Spent: Pharmacy (Healthcare)
Thu Oct 15 2026	-\$67.12	\$2,922.51	spend	Spent: Groceries (Groceries)
Wed Oct 14 2026	-\$4.75	\$2,989.63	spend	Spent: Coffee (Dining Out)
Tue Oct 13 2026	-\$8.00	\$2,994.38	spend	Spent: Parking (Transportation)
Mon Oct 12 2026	-\$18.50	\$3,002.38	spend	Spent: Movie night (Entertainment)
Sat Oct 10 2026	-\$35.00	\$3,020.88	spend	Spent: Haircut (Personal Care)
Fri Oct 9 2026	-\$4.75	\$3,055.88	spend	Spent: Coffee (Dining Out)
Wed Oct 7 2026	-\$71.20	\$3,060.63	spend	Spent: Groceries (Groceries)
Mon Oct 5 2026	-\$28.99	\$3,131.83	spend	Spent: Pet food (Pets)
Sat Oct 3 2026	-\$4.75	\$3,160.82	spend	Spent: Coffee (Dining Out)
Sun Oct 18 2026	+\$3,165.57	\$3,165.57		Opening balance
16 entries money in +\$3,165.57 money out -\$325.02 net +\$2,840.55				

Figure 71. The history of the Checking account.

Column	Meaning
Date	The date of the entry (with the year by default; section 12.7).
Change	Money in (green, +) or out (red, -).
Balance	The account's balance after the entry.
From	Where the entry came from: <i>spend</i> (a daily-ledger purchase), <i>bank</i> (imported from a bank file), <i>transfer</i> (one side of a move between accounts), or blank for entries you typed in (deposits, withdrawals, reconciliations, payments, opening balance).
Note	What it was: "Spent: Coffee (Dining Out)", "Paid Rent", "Opening balance", or the bank's description.

Key	Action
Up Down j k PgUp PgDn Home End	Move through the entries. The mouse works too: click an entry to highlight it, click a column heading to sort by it (again to reverse), turn the wheel to scroll.
/	Filter (section 12.6).
o / o	Sort : open the sort dialog (section 12.4).
r	Flip the sort direction.
a	Show every account in one list; press again for just this one.

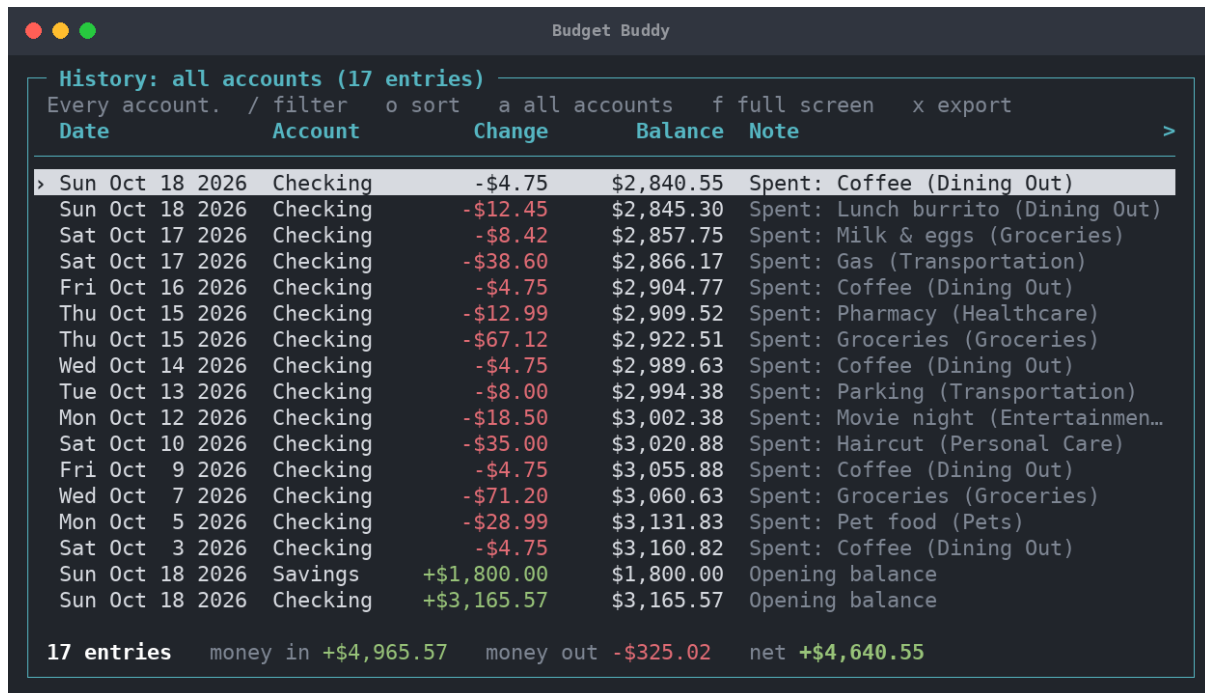
Key	Action
f	Full screen on or off. It stays the way you left it until you quit.
x	Export (Chapter 14).
d / Delete	Delete the highlighted entry.
D	Delete a whole bank import (Chapter 13).
u	Undo the last entry added.
y	Show or hide the year in dates.
Esc	Clear the filters; press again to close the history. (Enter , q and h also close it.)

12.1 Full screen and all accounts

f makes the history use the whole terminal, which is handy on a tall window with a long list. **a** adds an **Account** column and shows all your accounts in one list, newest first, so you can search everything at once.

History: Checking balance \$2,840.55 [full screen]				
/ filter	o sort	a all accounts	f full screen	x export
Date	Change	Balance	From	Note
> Sun Oct 18 2026	-\$4.75	\$2,840.55	spend	Spent: Coffee (Dining Out)
Sun Oct 18 2026	-\$12.45	\$2,845.30	spend	Spent: Lunch burrito (Dining Out)
Sat Oct 17 2026	-\$8.42	\$2,857.75	spend	Spent: Milk & eggs (Groceries)
Sat Oct 17 2026	-\$38.60	\$2,866.17	spend	Spent: Gas (Transportation)
Fri Oct 16 2026	-\$4.75	\$2,904.77	spend	Spent: Coffee (Dining Out)
Thu Oct 15 2026	-\$12.99	\$2,909.52	spend	Spent: Pharmacy (Healthcare)
Thu Oct 15 2026	-\$67.12	\$2,922.51	spend	Spent: Groceries (Groceries)
Wed Oct 14 2026	-\$4.75	\$2,989.63	spend	Spent: Coffee (Dining Out)
Tue Oct 13 2026	-\$8.00	\$2,994.38	spend	Spent: Parking (Transportation)
Mon Oct 12 2026	-\$18.50	\$3,002.38	spend	Spent: Movie night (Entertainment)
Sat Oct 10 2026	-\$35.00	\$3,020.88	spend	Spent: Haircut (Personal Care)
Fri Oct 9 2026	-\$4.75	\$3,055.88	spend	Spent: Coffee (Dining Out)
Wed Oct 7 2026	-\$71.20	\$3,060.63	spend	Spent: Groceries (Groceries)
Mon Oct 5 2026	-\$28.99	\$3,131.83	spend	Spent: Pet food (Pets)
Sat Oct 3 2026	-\$4.75	\$3,160.82	spend	Spent: Coffee (Dining Out)
Sun Oct 18 2026	+\$3,165.57	\$3,165.57		Opening balance
16 entries money in +\$3,165.57 money out -\$325.02 net +\$2,840.55				
Full screen (f to go back)				
/ filter o sort f full screen a all accounts x export d delete D delete import u undo				

Figure 72. Full-screen history.



History: all accounts (17 entries)				
Every account. / filter o sort a all accounts f full screen x export				
Date	Account	Change	Balance	Note
> Sun Oct 18 2026	Checking	-\$4.75	\$2,840.55	Spent: Coffee (Dining Out)
Sun Oct 18 2026	Checking	-\$12.45	\$2,845.30	Spent: Lunch burrito (Dining Out)
Sat Oct 17 2026	Checking	-\$8.42	\$2,857.75	Spent: Milk & eggs (Groceries)
Sat Oct 17 2026	Checking	-\$38.60	\$2,866.17	Spent: Gas (Transportation)
Fri Oct 16 2026	Checking	-\$4.75	\$2,904.77	Spent: Coffee (Dining Out)
Thu Oct 15 2026	Checking	-\$12.99	\$2,909.52	Spent: Pharmacy (Healthcare)
Thu Oct 15 2026	Checking	-\$67.12	\$2,922.51	Spent: Groceries (Groceries)
Wed Oct 14 2026	Checking	-\$4.75	\$2,989.63	Spent: Coffee (Dining Out)
Tue Oct 13 2026	Checking	-\$8.00	\$2,994.38	Spent: Parking (Transportation)
Mon Oct 12 2026	Checking	-\$18.50	\$3,002.38	Spent: Movie night (Entertainmen...
Sat Oct 10 2026	Checking	-\$35.00	\$3,020.88	Spent: Haircut (Personal Care)
Fri Oct 9 2026	Checking	-\$4.75	\$3,055.88	Spent: Coffee (Dining Out)
Wed Oct 7 2026	Checking	-\$71.20	\$3,060.63	Spent: Groceries (Groceries)
Mon Oct 5 2026	Checking	-\$28.99	\$3,131.83	Spent: Pet food (Pets)
Sat Oct 3 2026	Checking	-\$4.75	\$3,160.82	Spent: Coffee (Dining Out)
Sun Oct 18 2026	Savings	+\$1,800.00	\$1,800.00	Opening balance
Sun Oct 18 2026	Checking	+\$3,165.57	\$3,165.57	Opening balance
17 entries money in +\$4,965.57 money out -\$325.02 net +\$4,640.55				

Figure 73. Every account in one list, with an Account column.

12.2 What the totals line adds up

The line at the bottom always describes the entries *currently shown*: after a filter it counts only the matches. Filter to one year and one word and it becomes a quick answer to “how much did I spend on that?”.

12.3 Dates, balances and the order of entries

Entries are shown newest first by date; among entries on the same date, the one added most recently comes first. The **Balance** column is the true running balance in the order things happened, so it still makes sense after you sort.

12.4 Sorting

Press **o** (or **O**) and choose a column (Date, Account, Change, Balance, From or Note) and a direction; **r** then flips the direction, and clicking a column heading sorts by it. The column title shows ▲ or ▼, and a line above the list says what it is sorted by (“Sorted by Change ▼”). Choosing (*original order*) goes back to newest first.

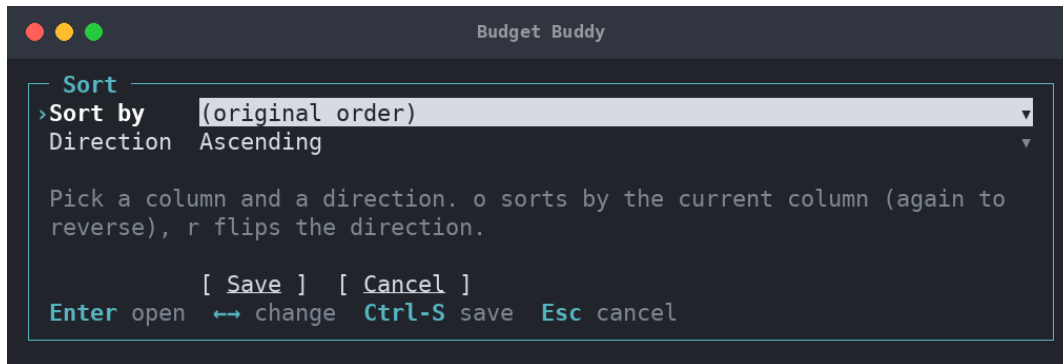


Figure 74. The sort dialog.

The screenshot shows a terminal window titled "Budget Buddy" with a "History: Checking" window open. The window shows a list of transactions sorted by "Change" in descending order. The total balance is \$2,840.55. The transactions are as follows:

Date	Change	Balance	From	Note
Sun Oct 18 2026	+\$3,165.57	\$3,165.57		Opening balance
Sun Oct 18 2026	-\$4.75	\$2,840.55	spend	Spent: Coffee (Dining Out)
Fri Oct 16 2026	-\$4.75	\$2,904.77	spend	Spent: Coffee (Dining Out)
Wed Oct 14 2026	-\$4.75	\$2,989.63	spend	Spent: Coffee (Dining Out)
Fri Oct 9 2026	-\$4.75	\$3,055.88	spend	Spent: Coffee (Dining Out)
Sat Oct 3 2026	-\$4.75	\$3,160.82	spend	Spent: Coffee (Dining Out)
Tue Oct 13 2026	-\$8.00	\$2,994.38	spend	Spent: Parking (Transportation)
Sat Oct 17 2026	-\$8.42	\$2,857.75	spend	Spent: Milk & eggs (Groceries)
Sun Oct 18 2026	-\$12.45	\$2,845.30	spend	Spent: Lunch burrito (Dining Out)
Thu Oct 15 2026	-\$12.99	\$2,909.52	spend	Spent: Pharmacy (Healthcare)
Mon Oct 12 2026	-\$18.50	\$3,002.38	spend	Spent: Movie night (Entertainment)
Mon Oct 5 2026	-\$28.99	\$3,131.83	spend	Spent: Pet food (Pets)
Sat Oct 10 2026	-\$35.00	\$3,020.88	spend	Spent: Haircut (Personal Care)
Sat Oct 17 2026	-\$38.60	\$2,866.17	spend	Spent: Gas (Transportation)
Thu Oct 15 2026	-\$67.12	\$2,922.51	spend	Spent: Groceries (Groceries)
Wed Oct 7 2026	-\$71.20	\$3,060.63	spend	Spent: Groceries (Groceries)

Summary: 16 entries, money in +\$3,165.57, money out -\$325.02, net +\$2,840.55

Figure 75. Sorted by Change, largest first.

12.5 Deleting entries

Press **d** on an entry. Budget Buddy tells you exactly what will happen before it does it:

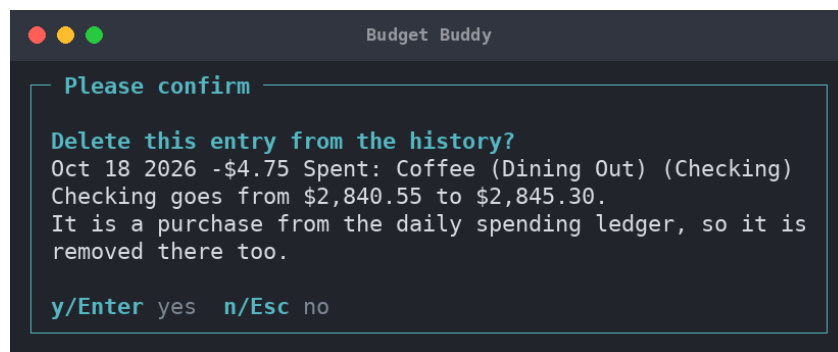


Figure 76. Deleting a purchase. The message says how the balance will change and that the daily-ledger purchase is removed as well.

- The account's balance moves back by the amount, and the later running balances are corrected.
- Deleting one side of a **transfer** removes the other side too.
- Deleting a **daily-ledger purchase** entry removes the purchase from the ledger.
- Deleting the entry for a recorded **bill or debt payment** only fixes the account; it does not move the bill's due date back or restore the debt's balance. Check those if it matters.
- **D** deletes a whole **bank import** at once (Chapter 13).

12.6 Filtering the history

Press **/** to open the filter form. It has five parts, and everything you fill in must match *at the same time*. Leave a part empty to ignore it.

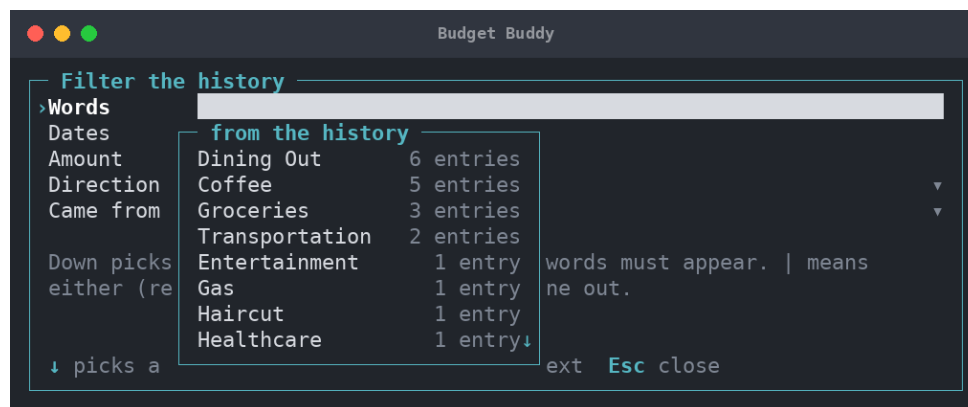


Figure 77. The filter form, with the Words box offering notes from the history.

Words

Every word you type must appear in the description. Case does not matter and neither does punctuation, so `7-eleven` finds `7-ELEVEN #35`.

You type	It finds
coffee shop	Rows containing both words.
"coffee shop"	Rows containing that exact phrase.
rent mortgage	Rows containing either word.
rent -car	Rows containing <i>rent</i> but not <i>car</i> .

With all accounts shown, the account name counts as a word too. The Words box is also a **dropdown of the notes in the history** you are looking at, most common first, with how many entries each has. It opens when the form does and narrows as you type.

- Nothing is chosen until you press **Down**, so **Enter** and **Tab** keep exactly what you typed. A filter of `rent` is wider than one particular `rent` note.
- **Down** then **Enter** picks a suggestion. A pick completes only the part after your last `|`, so `rent |` followed by a pick builds `rent | "Coffee shop"`.
- Store numbers and references are dropped, so `7-ELEVEN #35` and `7-ELEVEN #1204` are one suggestion, `7-ELEVEN`.

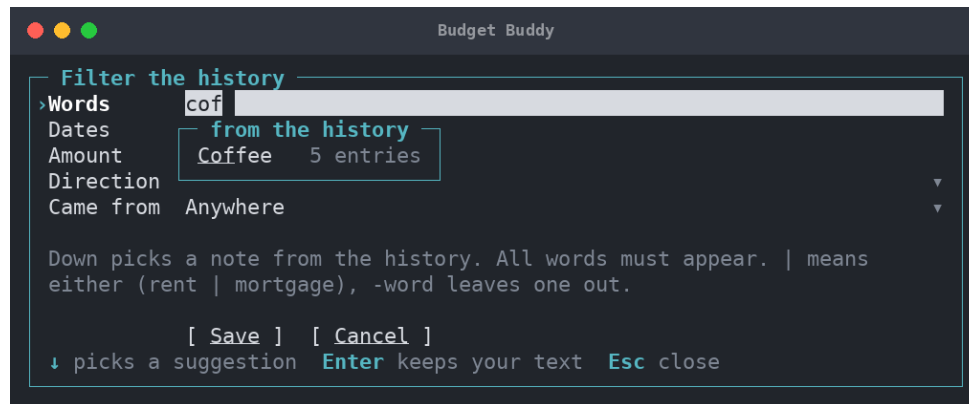


Figure 78. Typing “cof” narrows the suggestions.

- Daily-spending notes are split into the thing and its category: *Spent: Latte (Dining Out)* offers Latte and Dining Out.
- **Ctrl-N** brings the list back after you closed it with **Esc**.

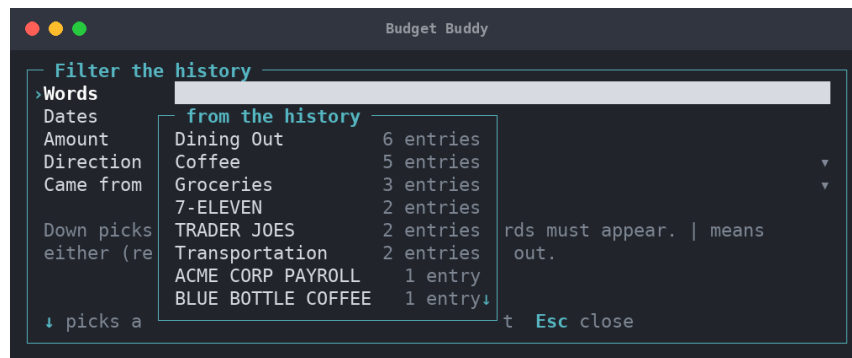


Figure 79. After importing a bank file, the suggestions include its merchants, with store numbers removed.

Dates

One or more periods, separated by commas. A row matches if its date is in **any** of them. Dates are month/day/year or the unambiguous 2024-06-15.

You type	It means
2025	The whole year.
Mar 2023 or 2023-03	That month.
2024-06-15	That day.
2024-01-01..2024-03-31 or Jan 2024..Mar 2024	A range (a month at the end means its last day).
since 2024, after 2023	From then on.
until Mar 2025, before 2022	Up to then.
this year, last year, this month, last month, ytd	Relative periods.
last 90 days	Also weeks, months and years.
Mar 2023, 2025	March 2023 plus all of 2025.

Amount

The size of the amount, whether in or out: 12.50, >50, <=20, 10..100 or 25.. (25 or more).

Direction

Money in and out, Money in only or Money out only.

Came from

Anywhere, Imported from a bank file, Daily spending ledger, Transfers between accounts or Typed in by hand.

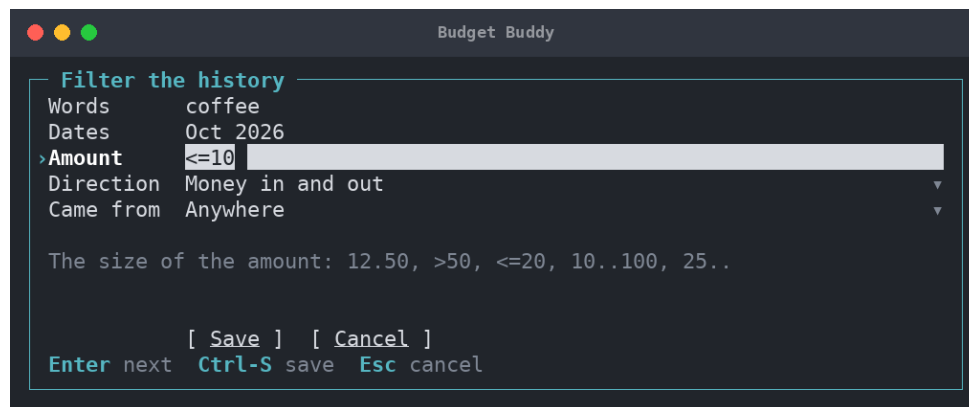


Figure 80. A filter for coffee purchases in October 2026 of \$10 or less.

Date	Change	Balance	From	Note
Thu Oct 22 2026	-\$5.25	\$4,804.64	bank	BLUE BOTTLE COFFEE
Sun Oct 18 2026	-\$4.75	\$2,840.55	spend	Spent: Coffee (Dining Out)
Fri Oct 16 2026	-\$4.75	\$2,904.77	spend	Spent: Coffee (Dining Out)
Wed Oct 14 2026	-\$4.75	\$2,989.63	spend	Spent: Coffee (Dining Out)
Fri Oct 9 2026	-\$4.75	\$3,055.88	spend	Spent: Coffee (Dining Out)
Sat Oct 3 2026	-\$4.75	\$3,160.82	spend	Spent: Coffee (Dining Out)

6 entries money in \$0.00 money out -\$29.00 net -\$29.00

Figure 81. The result. The line above the list says what is active; Esc clears it.

Two examples. “All the 7-Eleven purchases in March 2023 or all of 2025” is Words 7 eleven and Dates Mar 2023, 2025. “All the rent I paid in 2025” is Words rent, Dates 2025 and Direction *Money out only*; the totals line then shows the year’s rent. Press **x** and choose *What the history shows now* to save exactly those rows as a CSV for a spreadsheet or a tax preparer.

Note. Filters apply to the history window only, are forgotten when you close it and never change any data.

12.7 Showing the year

y shows or hides the year in dates in the history (and in the import preview). It is shown by default, and your choice is saved with your data.

13. Importing a Bank CSV

Instead of typing in every transaction, you can import the CSV file your bank offers (usually under “Download transactions”). Budget Buddy reads the file, shows you **exactly what it would do**, and changes nothing until you say so. Imported lines go into the account's history, where you can sort, filter and export them like any other entry.

13.1 Getting the file

Log in to your bank's website, choose the account and a date range, and download the transactions as **CSV** (not PDF or Excel). A browser normally saves it in your Downloads folder.

13.2 Starting an import

On the Accounts page, press **i** (on another page the program tells you to go there). Pick the account (if you have more than one) and the file.

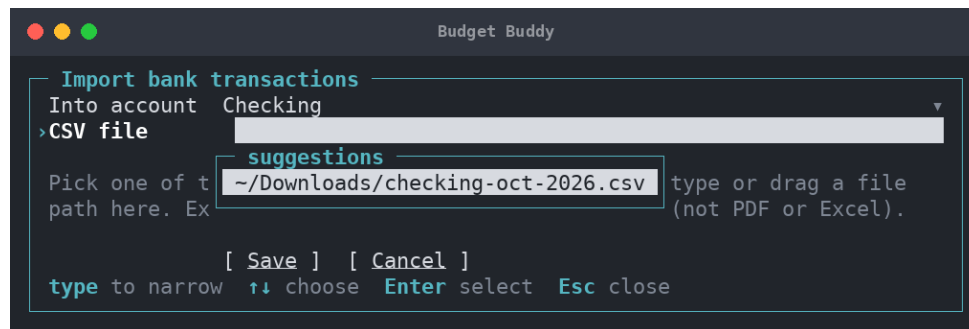


Figure 82. The import dialog. The menu lists CSV files found in Downloads, Desktop, Documents and the current folder, newest first.

- Pick a file from the list, or **type or paste a path**. Dragging a file from your file manager into the terminal window also works.
- Quotes, ~ and file:// prefixes are cleaned up for you (a file:// link has its %20 and similar codes turned back into characters).
- Press **ctrl-s** to read the file and see the preview.

13.3 The preview

Before anything changes you get a preview.

Line	What it tells you
File	The file name and how many transactions were read.
Dates	The range of dates in the file.
Columns	How the columns were understood: date, description, amount (or separate debit and credit columns), and a balance column if there is one.

Line	What it tells you
New	How many lines are new, how many are <i>already here</i> and how many <i>matched</i> something you entered by hand, plus the money in and out of the new ones.
Balance	The import mode in use (section 13.4), the balance before and after, and the bank's own closing balance if the file has one ("matches" when it agrees).
Table	The new transactions. Scroll with Up / Down / PgUp / PgDn .

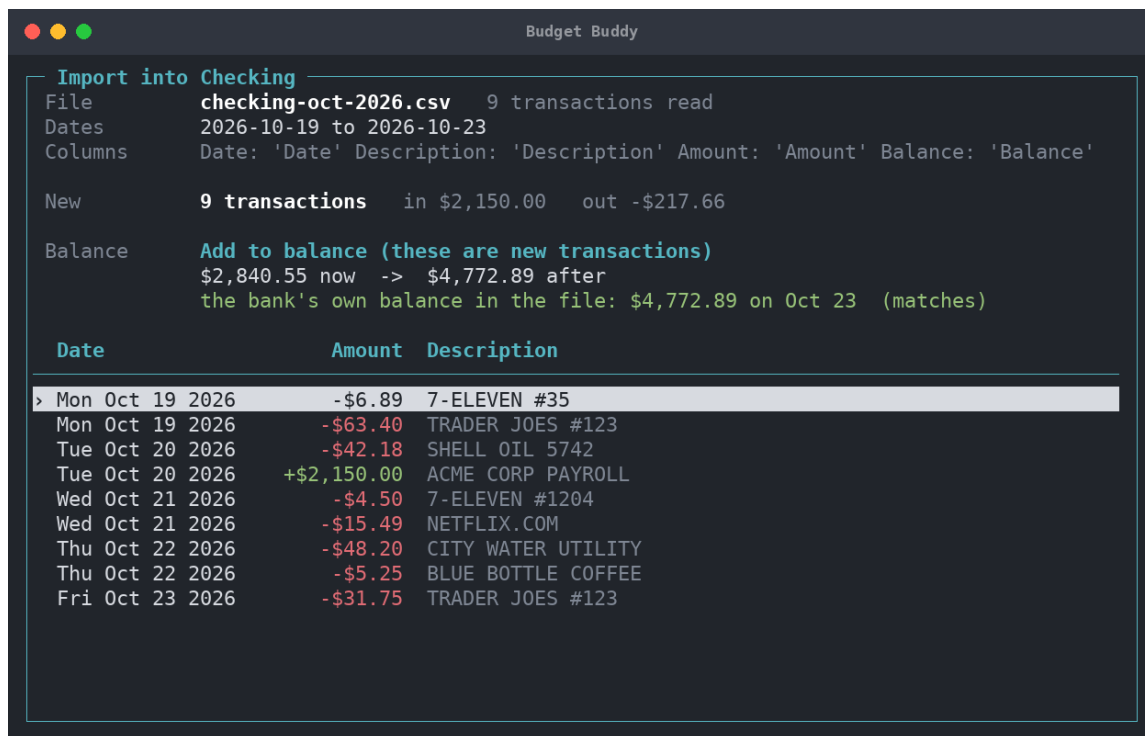


Figure 83. The import preview. Nothing has been changed yet.

Key	Action
Enter	Import.
Esc , n or q	Cancel; nothing changes.
s	Flip the signs, for banks that show spending as positive numbers. Budget Buddy guesses, and tells you what it guessed.
m or Tab	Choose how the balance is handled (below). M or Shift-Tab goes the other way.
y	Show or hide the year in dates.
Up / Down / PgUp / PgDn	Scroll the list of new transactions.

13.4 How the balance is handled

There are three modes. The preview starts with the one that fits the file and shows the result of each as you cycle through them.

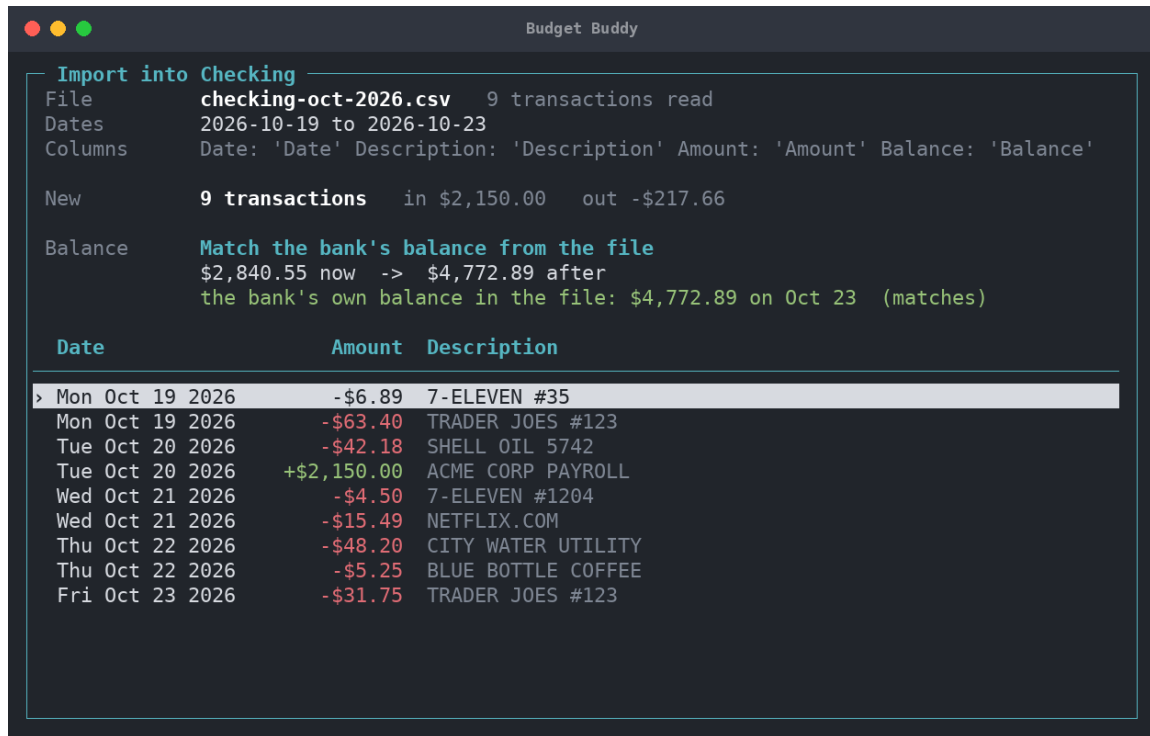


Figure 84. Cycling to Match the bank's balance from the file; the preview shows the balance that would result.

Mode	What happens to the balance
Keep balance (already counted, just fill in the history)	The balance stays as it is. New transactions are logged, and a “Balance before imported transactions” entry is added so the history still adds up. Use this when you have already reconciled to the bank.
Add to balance (these are new transactions)	The new transactions are added to the balance. Use this when the file only contains activity you have not recorded.
Match the bank's balance from the file	The balance is set to the bank's closing balance from the file. Only offered if the file has a balance column.

If the file has the bank's own balance, the default is the mode whose result agrees with it: **Add** when your balance plus the new lines equals it, **Keep** when your balance already equals it. If none agree, an empty new account takes the bank's figure and an account with history keeps its balance, and the preview shows the difference so you can choose. A file with no balance column adds to an empty account and keeps the balance otherwise.

13.5 After the import

Budget Buddy				
History: Checking		balance \$4,772.89		
/ filter	o sort	a all accounts	f full screen	x export
Date	Change	Balance	From	Note
> Fri Oct 23 2026	-\$31.75	\$4,772.89	bank	TRADER JOES #123
Thu Oct 22 2026	-\$5.25	\$4,804.64	bank	BLUE BOTTLE COFFEE
Thu Oct 22 2026	-\$48.20	\$4,809.89	bank	CITY WATER UTILITY
Wed Oct 21 2026	-\$15.49	\$4,858.09	bank	NETFLIX.COM
Wed Oct 21 2026	-\$4.50	\$4,873.58	bank	7-ELEVEN #1204
Tue Oct 20 2026	+\$2,150.00	\$4,878.08	bank	ACME CORP PAYROLL
Tue Oct 20 2026	-\$42.18	\$2,728.08	bank	SHELL OIL 5742
Mon Oct 19 2026	-\$63.40	\$2,770.26	bank	TRADER JOES #123
Mon Oct 19 2026	-\$6.89	\$2,833.66	bank	7-ELEVEN #35
Sun Oct 18 2026	-\$4.75	\$2,840.55	spend	Spent: Coffee (Dining Out)
Sun Oct 18 2026	-\$12.45	\$2,845.30	spend	Spent: Lunch burrito (Dining Out)
Sat Oct 17 2026	-\$8.42	\$2,857.75	spend	Spent: Milk & eggs (Groceries)
Sat Oct 17 2026	-\$38.60	\$2,866.17	spend	Spent: Gas (Transportation)
Fri Oct 16 2026	-\$4.75	\$2,904.77	spend	Spent: Coffee (Dining Out)
Thu Oct 15 2026	-\$12.99	\$2,909.52	spend	Spent: Pharmacy (Healthcare)
Thu Oct 15 2026	-\$67.12	\$2,922.51	spend	Spent: Groceries (Groceries)
Wed Oct 14 2026	-\$4.75	\$2,989.63	spend	Spent: Coffee (Dining Out)
Tue Oct 13 2026	-\$8.00	\$2,994.38	spend	Spent: Parking (Transportation)
Mon Oct 12 2026	-\$18.50	\$3,002.38	spend	Spent: Movie night (Entertainment)
Sat Oct 10 2026	-\$35.00	\$3,020.88	spend	Spent: Haircut (Personal Care)
Fri Oct 9 2026	-\$4.75	\$3,055.88	spend	Spent: Coffee (Dining Out)
25 entries	money in +\$5,315.57	money out -\$542.68	net	+\$4,772.89

Figure 85. The result: the history now includes the nine imported lines, marked bank.

The history opens with the new lines marked **bank** in the From column, in date order, and a message confirms how many were added and the new balance.

13.6 Importing the same file twice

Importing is safe to repeat. A line already imported into that account (same date, amount and description) is shown as **Already here** and skipped. Identical lines on one day are counted, so two \$4.75 coffees on the same day are two lines the first time and none the second; a third one in a later file would still come in. A line that matches something you entered by hand (same amount, within three days) is marked as **matched** and not added again.

The only way a repeat gets through is if your bank words the description differently in two exports; then it shows in the preview's list of new lines and you can cancel.

13.7 Backing out an import

If you imported the wrong file or with the wrong sign setting, open the account's history, highlight any line from that import, and press **D**. Budget Buddy shows how many entries and what net amount it will remove and asks first. **d** deletes a single line instead.

13.8 What files can be read

- Commas, semicolons or tabs as separators; quoted fields.

- Decimal commas and the usual ways of grouping thousands: 1,234.56, 1.234,56 and 1.234.567 all read correctly, and so does 12,50. In a file that uses decimal commas, 1.234 means one thousand two hundred thirty-four.
- Dates written 31/12/2025, 12/31/2025, 2025-12-31, 17.10.2026 or Dec 31, 2025. Dates with dots are always read day first. For slash dates the program works out day-first or month-first from the whole file; if no date in the file settles it (every day is 12 or less), your language and region decide (LC_ALL, LC_TIME, then LANG), and the United States, the Philippines or no region at all mean month first.
- A single signed Amount column, or separate Debit and Credit columns, with or without a Balance column.
- Title or summary lines above the header row, byte-order marks, UTF-16 exports, and files saved by Windows in Windows-1252 (accented letters and the euro sign come out right).
- Files with no header at all (it guesses the columns).
- Files up to 25 MB. Folders, devices and pipes are refused.

If Budget Buddy cannot make sense of a file it tells you why, and nothing is changed.

14. Exporting History to CSV

You can copy an account's history into a spreadsheet, or hand it to a tax preparer, with **x**. It works from the Accounts page and from inside an account's history.

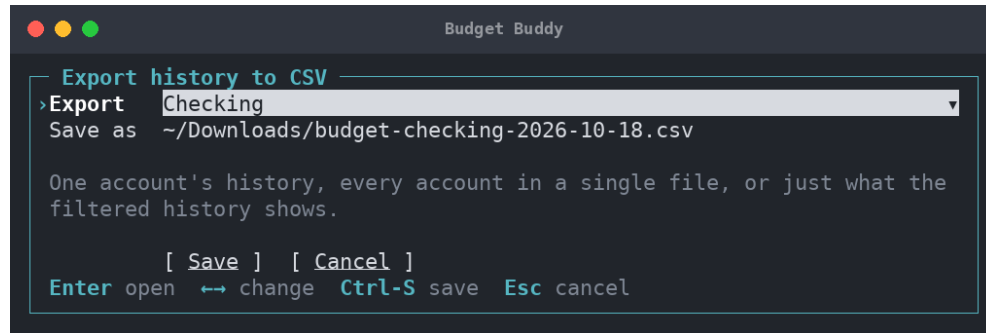


Figure 86. The export dialog.

Field	What to enter
Export	One account, All accounts (a single file with an Account column), or, when the history is filtered, sorted or showing every account, What the history shows now , which saves exactly the rows on screen in the order shown.
Save as	A file name, a folder, or a full path. The suggestion is in your Downloads folder (or your home folder if there is no Downloads) with a name such as budget-checking-2026-10-18.csv. A bare name is saved in the suggested folder, ~ works, and a name without an extension gets .csv.

An existing file is **never replaced**. The suggested name skips to -2, -3... and a name you type that is already taken is refused.

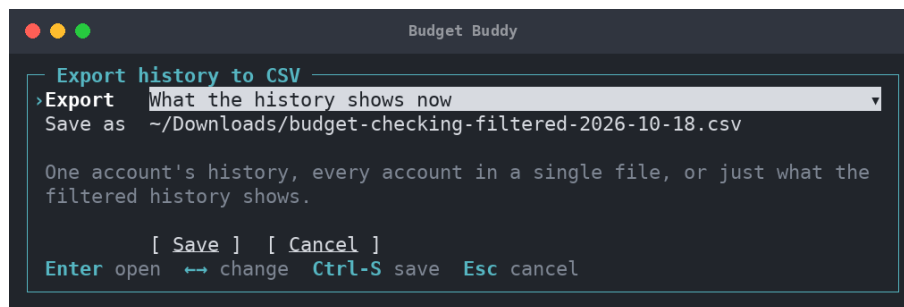


Figure 87. Exporting a filtered history. The suggested name ends in -filtered.

14.1 The file

```
Date,Account,Description,Amount,Balance,Source
2026-10-01,Main Checking,Opening balance,1000.00,1000.00,
2026-10-16,Main Checking,Spent: Coffee (Dining Out),-4.75,995.25,spend
2026-10-17,Main Checking,Transfer to Savings (Rainy day),-200.00,795.25,transfer
```

- One row per history entry, oldest first (or in the order you sorted a filtered view).

- **Amount** is a plain number: negative means money out, with no \$ and no thousands commas. **Date** is YYYY-MM-DD. A spreadsheet can sort and total both directly.
- **Balance** is that account's balance after the entry.
- **Source** says where the entry came from: bank (imported), spend (a daily-ledger purchase), transfer (one side of a move between accounts; both sides appear in the all-accounts file, and cancel out), or blank for entries you typed in.
- Text that a spreadsheet could read as a formula (it starts with =, +, - or @) gets a leading apostrophe.

A file exported for one account can be imported again into another account with **i**. Cash purchases in the daily ledger that did not come out of an account are not in any account's history, so they are not in the export either.

15. Cash Flow

The Cash Flow page (**8**) answers: *will I have enough money on the day each bill is due?* It starts from the money in your **spendable accounts** (the accounts marked “Counts as bill cash”) and replays your income and bills day by day for the forecast horizon (90 days by default).

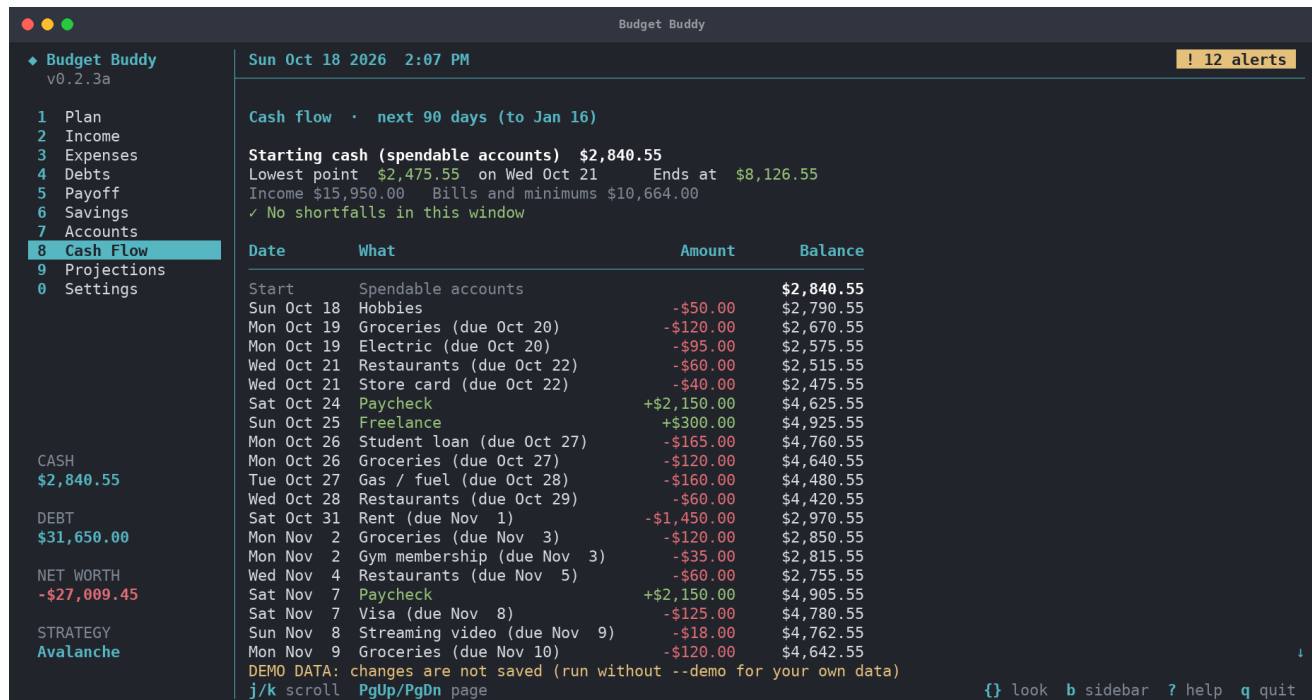


Figure 88. The Cash Flow page for the sample budget. No shortfalls.

Part	Meaning
Heading	The forecast window: “next 90 days (to Jan 16)”.
Starting cash	The total of your spendable accounts today.
Lowest point, Ends at	The lowest balance the forecast reaches and when, and the balance at the end of the window.
Income, Bills and minimums	Totals for the window.
Verdict	A green ✓ <i>No shortfalls in this window</i> , or a red warning for each gap (below).
Table	Every dated event in order, with the running balance after it.

Bills and debt payments appear on the day you should **pay** them, which is the due date moved earlier by your “pay this many days early” setting (one day by default), so “Groceries (due Oct 20)” shows on Oct 19. Income appears on its pay date.

15.1 When you will run short

If the balance falls below zero (or below your **minimum cash cushion** if you set one), the screen names the period and tells you how low it goes:

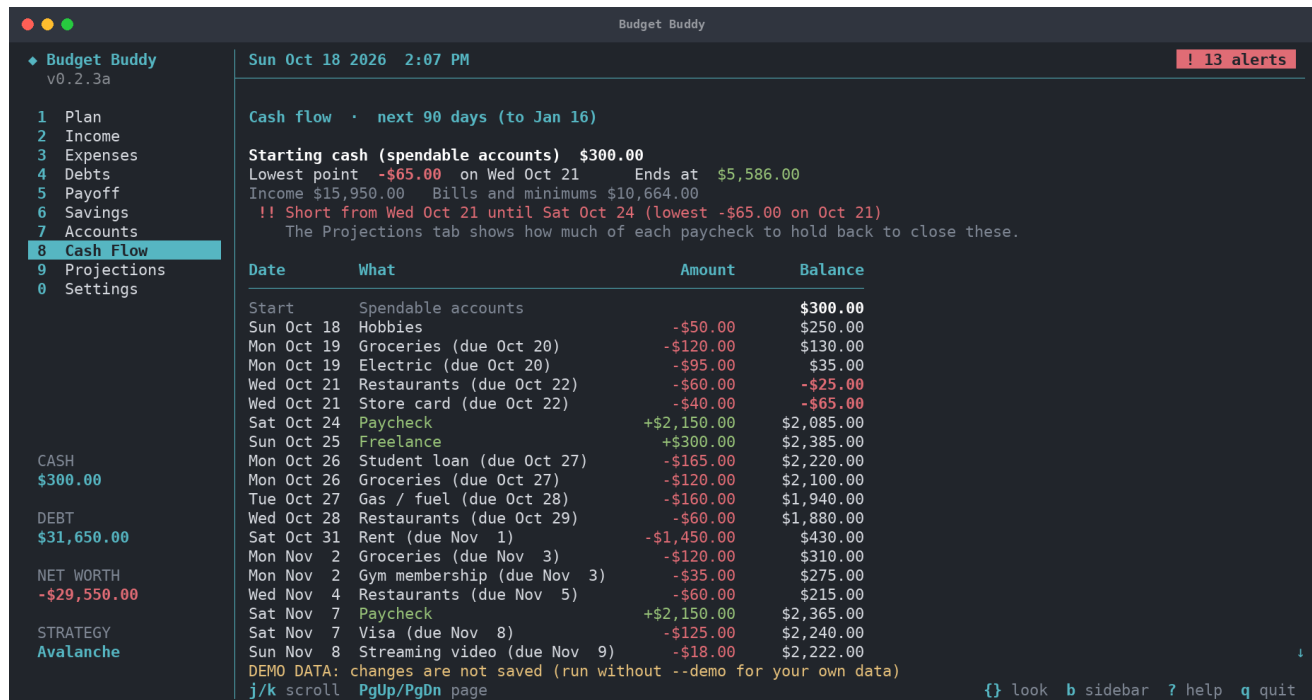


Figure 89. The same budget with only \$300 in checking. The forecast shows when it will be short and how much.

The same finding appears in the Plan's *Needs attention* list as a **Cash gap** alert, together with how much more cash you would need on hand today. Common fixes: move money from savings into checking, reduce flexible spending, or delay a bill if the biller allows it.

15.2 What it does and does not include

- It uses the **total** of your spendable accounts, not each account separately.
- It includes your income and your bills and debt minimums. It does **not** include the extra money the Plan sends to debts and savings, or purchases in the Daily spending ledger that have not happened yet.
- Recurring items are expanded into their actual dates, so a month with three paychecks looks different from one with two.
- Keys: **j** **k** **PgUp** **PgDn** **Home** **End** scroll, and **B** reconciles an account. The page is a document, so it has no rows to highlight and no views.

Tip. The forecast is only as good as the starting balance. Press **B** to reconcile a spendable account before you trust it, especially if the Plan says the account has not been updated recently.

16. Projections

The Projections page (**9**) is for looking ahead. It was called Set-Aside in version 0.1a, and the name changed to make room for more ways of looking ahead later, such as a calendar. Today it has one job, and does it in two views.

Knowing that you owe \$1,450 for rent on the 1st is one thing; knowing *which paycheck* should pay for it is what keeps you out of trouble. Projections works that out. It looks at every bill and debt payment due in the forecast window and decides how much of each paycheck to hold back for it.

For each bill, the paychecks that arrive **after the previous payment was due and before this one is due** (allowing for your “pay this many days early” setting) share its cost **in proportion to their size**, so a \$300 freelance payment is not asked to carry as much as a \$2,150 paycheck. Large occasional bills, such as car insurance or registration, are built up gradually over the paychecks before they are due. Anything that falls due **before any paycheck can fund it** is flagged: that one has to come from cash you already have.

There are two **views** of the same plan, named on the bar at the top of the page: *By paycheck* and *By bill*. Press **v** (or **v**) to switch, or click the name. (Earlier versions listed them as modes under Projections in the sidebar.)

16.1 By paycheck

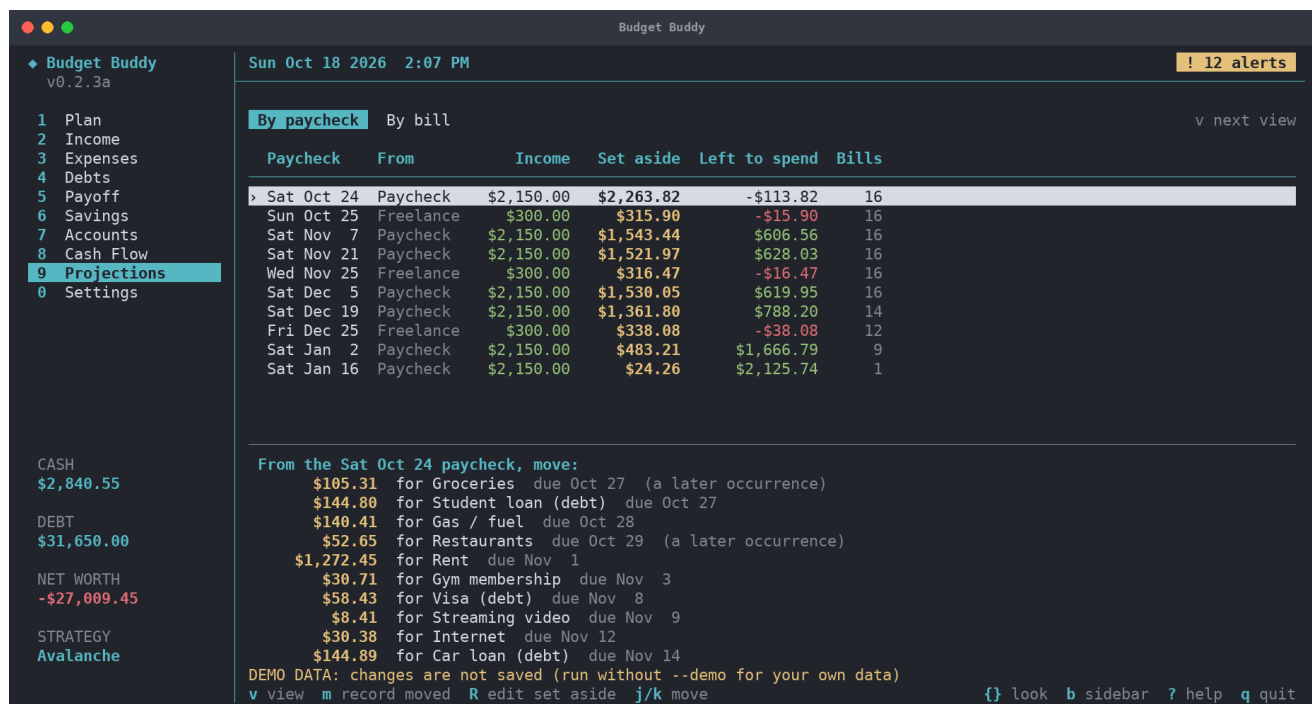


Figure 90. The by-paycheck view. The list below the table shows what to move from the highlighted paycheck.

Column	Meaning
Paycheck, From	The date and the income source (several sources paid on the same day are combined).
Income	What arrives that day.

Column	Meaning
Set aside	How much of it should be moved aside for bills and debt payments.
Left to spend	Income minus the set-aside: what is yours for extra debt payments, savings and spending. A negative figure means the bills that depend on this paycheck add up to more than it brings in, so the shortfall has to come from cash on hand.
Bills	How many items it is funding.

Under the table, the highlighted paycheck is spelled out: “From the Sat Oct 24 paycheck, move: \$1,272.45 for Rent due Nov 1”, and so on, soonest due first. Items marked *(a later occurrence)* are for a payment that is not the very next one for that bill.

When you have actually moved the money, press **m** on that paycheck to **record** it. Each item is added to that bill's **Set aside** amount, so tomorrow's plan does not ask for it again. (Items for later occurrences are not counted yet.)

Careful. **m** adds to the amounts already set aside each time you press it. Press it once per paycheck.

16.2 By bill

Budget Buddy v0.2.3a

Sun Oct 18 2026 2:07 PM ! 12 alerts

By paycheck **By bill** v next view

Bill	Due	Amount	Have	Still needed	Paychecks	Each	Start
Hobbies	Sun Oct 18	\$50.00	-	\$50.00	0	-	cash now
Electric	Tue Oct 20	\$95.00	-	\$95.00	0	-	cash now
Groceries	Tue Oct 20	\$120.00	-	\$120.00	0	-	cash now
Restaurants	Thu Oct 22	\$60.00	-	\$60.00	0	-	cash now
Store card	Thu Oct 22	\$40.00	-	\$40.00	0	-	cash now
Groceries	Tue Oct 27	\$120.00	-	\$120.00	2	\$105.31	Oct 24
Student loan	Tue Oct 27	\$165.00	-	\$165.00	2	\$144.80	Oct 24
Gas / fuel	Wed Oct 28	\$160.00	-	\$160.00	2	\$140.41	Oct 24
Restaurants	Thu Oct 29	\$60.00	-	\$60.00	2	\$52.65	Oct 24
Rent	Sun Nov 1	\$1,450.00	-	\$1,450.00	2	\$1,272.45	Oct 24
Groceries	Tue Nov 3	\$120.00	-	\$120.00	0	-	cash now
Gym membership	Tue Nov 3	\$35.00	-	\$35.00	2	\$30.71	Oct 24
Restaurants	Thu Nov 5	\$60.00	-	\$60.00	0	-	cash now

Hobbies \$50.00 due Sun Oct 18
No paycheck arrives before the due date (Oct 18 to Oct 18): this has to come from cash on hand.

DEMO DATA: changes are not saved (run without --demo for your own data)

v view m record moved R edit set aside j/k move {} look b sidebar ? help q quit

CASH \$2,840.55
DEBT \$31,650.00
NET WORTH -\$27,009.45
STRATEGY Avalanche

Figure 91. The by-bill view. The note underneath explains the highlighted bill.

Column	Meaning
Bill, Due, Amount	The bill, the date it is due, and how much.
Have	Money you have already set aside for it.
Still needed	The amount minus what you have.

Column	Meaning
Paychecks	How many paychecks share it.
Each	About how much comes from each paycheck.
Start	The first paycheck date to begin, or <i>cash now</i> when no paycheck arrives in time.

The text under the table explains the highlighted bill, for example “No paycheck arrives before the due date (Oct 18 to Oct 18): this has to come from cash on hand.” Press **R** here to record or correct how much you have already set aside for the highlighted bill.

Key	Action
v / v	Switch between the views: by paycheck and by bill.
m	Record that you moved a paycheck's set-asides (by paycheck view).
R	Edit how much is already set aside for the highlighted bill (by bill view).
j k	Move the highlight.
B	Reconcile an account.

Tip. A simple routine: when a paycheck arrives, open Projections, move that paycheck's list into a separate savings or “bills” account (a transfer, Chapter 11), then press **m** .

17. Settings

The Settings page (**0**) has two kinds of setting. The first six rows are about the **program**: how it looks and how it starts. They are the same six rows, in the same order, as in the other Electric Shepherd programs. The rest are the **rules the Plan follows**. Every value has a sensible default, so you may never need to change one. To change a setting, highlight it and press **Enter** , **e** or **Space** ; a menu or a small dialog opens with the current value and an explanation. The change takes effect immediately and is saved.

Note. The two kinds are kept in different places. The rules of the Plan are part of your **data file**, so they travel with it when you copy it to another computer. The six program settings are in a small **settings file** of their own (section 17.2), so choosing a look never touches your money data and a copied data file does not change how the program looks on another machine. The product name and version shown on every screen are not settings.

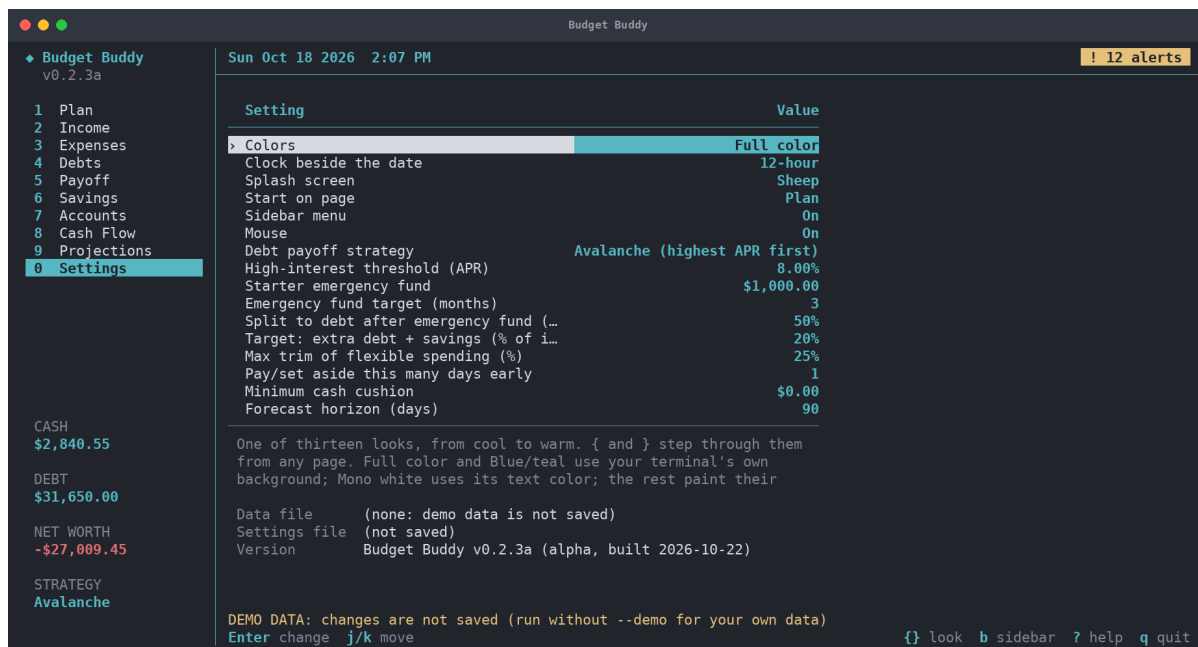


Figure 92. The Settings page. The first six rows control the program; the line under the table explains the highlighted setting, and the last lines show where your data file and settings file are.

17.1 The program's own settings

Setting	Default	Choices	What it does
Colors	Full color	Thirteen looks (below)	How the program is coloured (section 17.1).
Clock beside the date	12-hour	12-hour, 24-hour or Hidden	Whether and how the time of day is shown (section 17.1).
Splash screen	Sheep	Sheep, Classic or Off	What appears when the program starts (section 17.1).

Setting	Default	Choices	What it does
Start on page	Plan	Any of the ten pages	The page that is open when the program starts. It takes effect the next time you start the program; -p picks a page for one run (section 20).
Sidebar menu	On	On or Off	Whether the sidebar with the page list shows in windows 100 columns by 24 rows or larger. b switches it from any page (section 17.1).
Mouse	On	On or Off	Whether clicks and the wheel are used (section 17.1).

Each of these opens a **menu** in the middle of the page. **Up** and **Down** (or **j** and **k**) move through it, **Home** and **End** jump to its ends, **Enter** or **Space** applies the highlighted choice, and **Esc** puts the old choice back. Clicking an entry highlights it, and clicking it again applies it.

Colors

The **Colors** setting chooses how the program is drawn. There are thirteen looks, listed in the order **{** and **}** step through them, from cool to warm after the default. Figures 13 and 14 show them all.

Look	What it looks like
Full color	The default. Your terminal's own colours: cyan headings, and green, red and yellow carrying meaning as described in section 2.7. It paints no background, so your terminal's own background shows through.
Blue/teal	Blue and teal throughout, the look of earlier versions. Warnings stay yellow and errors red. No background is painted.
Winter	Cold slate with frost and ice blue: dark and quiet.
Harbor	Muted navy and sea blue.
Dusk	Muted violet and lavender after sunset.
Shepherd	The Electric Shepherd palette: beige text and sage headings on a dark slate background.
Spring	Dark green with new-leaf green and a blossom pink.
Autumn	Brown, rust and gold, like leaves in October.
Ember	Charcoal with brick, copper and coral, like a banked fire.
Samhain	Halloween: pumpkin orange and candle gold on near-black, with bone-white text and a deep violet highlight.
Mono white	Your terminal's own text colour, with bold, dim and reverse video doing the work colour did. Muted grey text becomes the dim attribute. This is also what NO_COLOR asks for.
Mono green	The whole screen in one green on black, like a phosphor monitor, in three brightness levels.
Mono amber	The same in amber.

The eight **painted** looks (Winter to Samhain) fill the whole screen with their own background and use exact 24-bit colours on a terminal that says it supports them (the COLORTERM variable says truecolor or 24bit), and the nearest colours in the 256-colour palette otherwise. Mono green and Mono amber use the 256-colour palette, which practically every modern terminal has. The splash screen uses the same look.

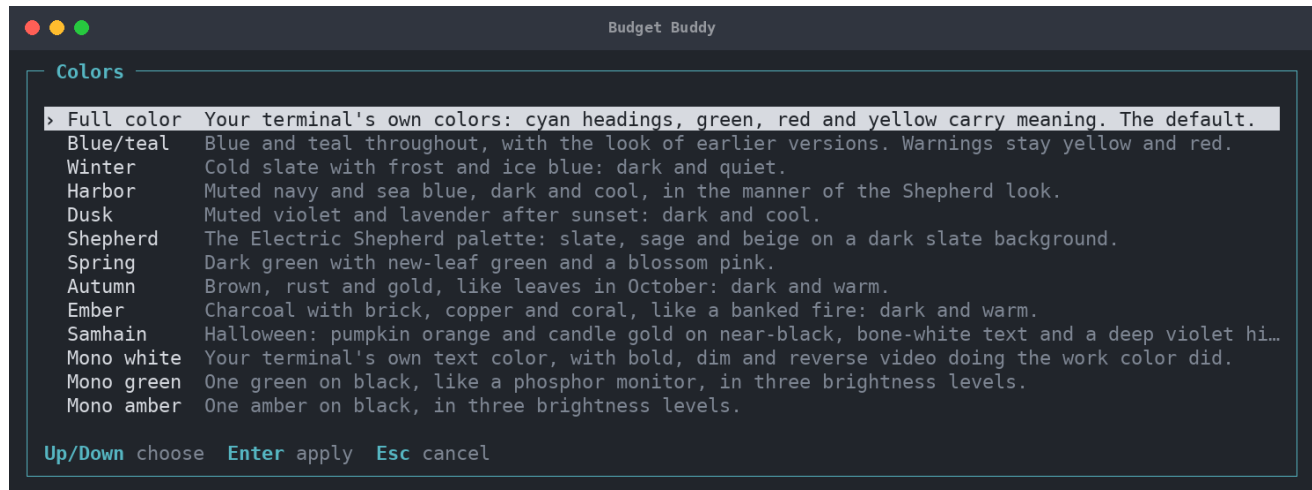


Figure 93. The Colors menu, with all thirteen looks and a line about each. While the menu is open, moving through it shows each look on the page behind; **Esc** puts the old one back.

Every signal that colour gives is also given by brightness or reverse video (alerts, the current page, the highlighted row), so every look stays readable.

You can also change the look without going to Settings: **{** and **}** step to the previous and next look from any page, and save the choice just as the menu does. For one run only, start the program with `-c LOOK` (for instance `budget -c amber`); the saved setting is not touched.

Note. When more than one thing asks for a look, `-c LOOK` wins, then the `NO_COLOR` environment variable (which asks for Mono white), then the saved Colors setting. Neither `-c` nor `NO_COLOR` is ever saved. Choosing a look in Settings, or stepping with **{** and **}**, replaces them for the rest of the run.

Clock

Clock beside the date puts the time of day next to today's date: in the sidebar layout above the page, or on the figures line in the top-bar layout (Figure 15).

Choice	Shows
12-hour	2:07 PM. The default.
24-hour	14:07.
Hidden	Only the date.

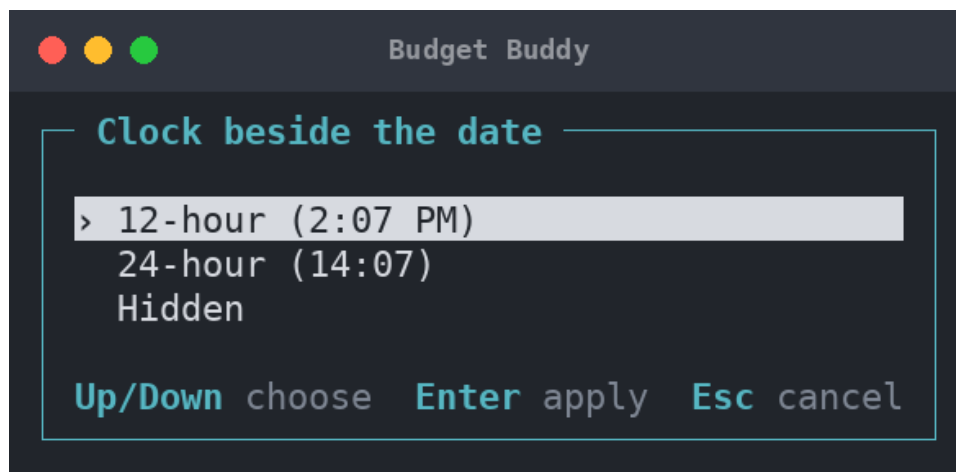


Figure 94. Choosing the clock format.

The time comes from the system clock (through the standard date command, because the Rust standard library has no time zones). It is read once a minute, just after the minute changes, and the program also notices when the day changes. If the system cannot say what time it is (for example there is no date command), the clock is simply hidden. If you set `BUDGET_TODAY` to pretend it is another date, the clock is hidden, because the time would be meaningless; add `BUDGET_TIME=14:07` to pretend a time as well (section 20.4).

Splash screen

The splash is what you see for a moment when the program starts. **Sheep** (the default) draws the Electric Shepherd sheep above the name, company, version and build date; **Classic** shows the same words in a framed title box; **Off** goes straight to the first page. Whichever you choose, the splash **waits for a key**: it shows “press any key to continue”, and the key you press is used up, so it does not also do something on the first page. A window too small for the sheep gets the framed box instead, and the splash is drawn again if you resize the window while it is showing. A terminal that cannot draw the sheep's block characters (in ASCII mode, for instance) also gets the box.

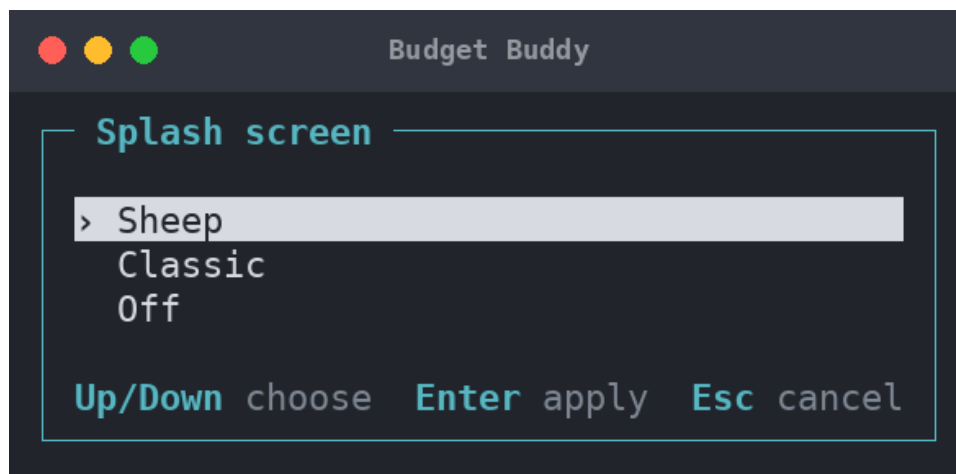


Figure 95. The Splash screen menu.

`-n` (or `--no-splash`) and `BUDGET_NO_SPLASH=1` skip the splash for one run, whatever the setting says.

Start on page

The page that is open when the program starts: any of the ten pages, with Plan the default. The choice takes effect the next time you start the program. -p PAGE opens a different page for one run.

Sidebar menu

The sidebar with the list of pages shows in windows at least 100 columns by 24 rows, as long as this setting is **On**. **b** switches it off and on from any page and saves the choice, the same as this menu. With the sidebar **Off** you get the top-bar layout at any size (section 2.1).

Mouse

While **Mouse** is **On** the program asks the terminal to report clicks and the wheel, and acts on them (section 3.5). The terminal's own text selection then usually needs **Shift** held down while you drag. Turn the setting **Off** to give the mouse back to the terminal. It takes effect at once.

17.2 The settings file

The six program settings are kept in a small text file of their own, apart from your data. By default it is ~/.config/budget-tui/settings (or \$XDG_CONFIG_HOME/budget-tui/settings if that is set to an absolute path); BUDGET_CONFIG names another place. The folder is created, private to you, the first time a setting is saved. The Settings page shows where the file is, on the **Settings file** line under the table.

```
# Budget Buddy settings. Edit here or in the program (Settings page).
colors=shepherd
clock=24h
splash=sheep
start=plan
sidebar=on
mouse=on
```

- The words are the ones in the menus, in lowercase: colors takes full, teal, winter, harbor, dusk, shepherd, spring, autumn, ember, samhain, white, green or amber; clock takes 12h, 24h or off; splash takes sheep, classic or off; start takes a page name or its number; sidebar and mouse take on or off.
- Blank lines, lines that start with #, and keys the program does not know are ignored. A value it does not understand leaves that one setting at its default and is reported once on the message line; the other lines still apply.
- A missing or unreadable file means the defaults, and the program starts anyway.
- **Coming from an earlier version.** Versions before 0.2.3a kept the colour mode and the clock in the data file. If there is no settings file yet, those two values are read from the data file once, so your choices carry over; after that the settings file is the one that counts. The data file no longer writes them.
- **Demo mode** (- -demo) neither reads nor writes the settings file: it always starts with the defaults, and changes to settings last only until you quit.

17.3 The rules of the Plan

The remaining rows are the rules the Plan follows. They are part of your data file.

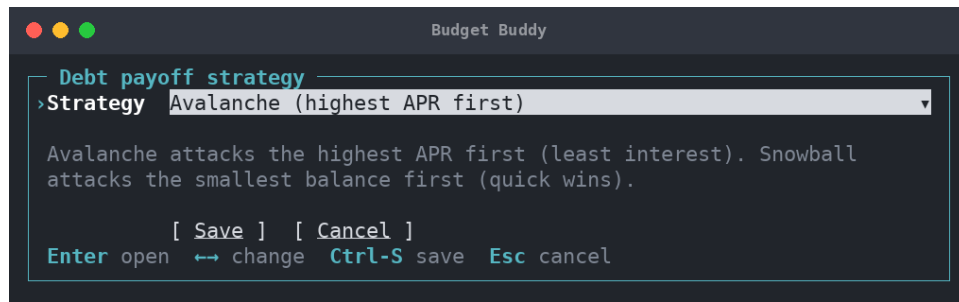


Figure 96. Changing the debt payoff strategy.

Setting	Default	Allowed	What it does
Debt payoff strategy	Avalanche	Avalanche or Snowball	Where extra debt money goes: highest APR first, or smallest balance first.
High-interest threshold (APR)	8.00%	any APR	Debts at or above this rate get extra payments before you build savings beyond the starter fund.
Starter emergency fund	\$1,000.00	\$0 or more	A small cushion funded first, before extra debt payments.
Emergency fund target (months)	3	0 to 36	How many months of essentials and debt minimums the emergency fund aims for (for goals with target 0).
Split to debt after emergency fund (%)	50%	0 to 100	After the emergency fund is full, this share of what is left goes to debt and the rest to savings goals.
Target: extra debt + savings (% of income)	20%	0 to 100	The plan tries to send at least this share of income to extra debt payments and savings, trimming flexible spending if needed.
Max trim of flexible spending (%)	25%	0 to 100	Flexible spending is never cut by more than this, unless income cannot cover it.
Pay/set aside this many days early	1	0 to 14	Bills are planned this many days before the due date to allow for processing time.
Minimum cash cushion	\$0.00	\$0 or more	The cash-flow forecast warns when projected cash drops below this.
Forecast horizon (days)	90	14 to 365	How far ahead Cash Flow and Projections look.

If you enter a value outside the allowed range, the dialog tells you what range is allowed and keeps the old value. In a narrow window the longer setting names are cut short on the page; the dialog shows the whole name.

18. Sorting and Searching

Income, Expenses (the Bills view, the Daily spending ledger and both Allotments views), Debts, Savings and Accounts all sort and search in the same way, so there is only one thing to learn. The keys are the same as in the other Electric Shepherd programs. The account history has its own, richer version of the same idea (Chapter 12).

18.1 The current column

A table always has one **current column**, shown with a highlighted heading and a highlighted cell in the row you are on (Figure 5). **h** and **l** move it to the left and right, and clicking a cell makes its column current. Sorting and column filters act on the current column. The current column is never hidden when the window is narrow: moving it to the right brings the columns that did not fit into view, and a **<** or **>** at the edge of the headings says that more are off to that side.

18.2 Sorting

Key	What it does
o or s	Sort by the current column , ascending. Press again to reverse (descending), and again to go back.
Alt-1 to Alt-9	Sort by the 1st to 9th column from the left, the same way.
r	Flip the sort direction. On a list that is not sorted yet, r sorts the current column from the top down.
o	Open the sort dialog: pick the column and the direction, and press Ctrl-S . Choosing (<i>original order</i>) puts the list back the way it was.
Click a heading	Sort by that column; click it again to reverse.

The sorted column's title shows ▲ (ascending) or ▼ (descending), and the line above the table says what the list is sorted by: “Sorted by Amount ▼ o changes it”. Until you sort, it says “Sorted by order added”. The highlight stays on the row it was on, wherever the sort moves it.

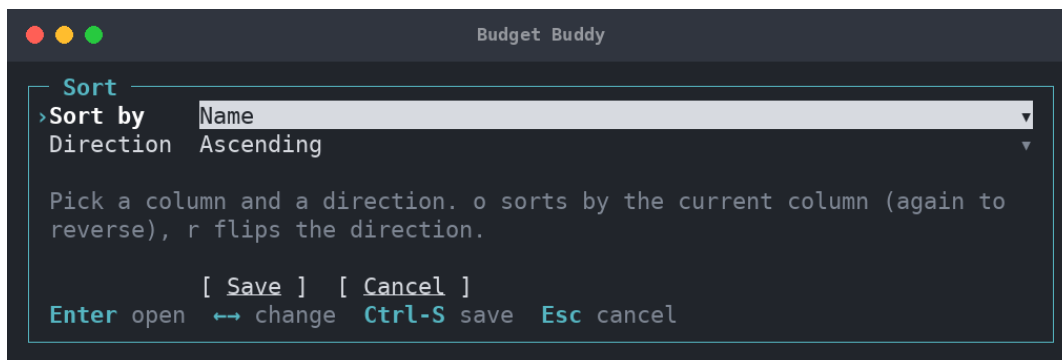


Figure 97. The sort dialog, opened with **o**.

Budget Buddy
v0.2.3a

Sun Oct 18 2026 2:07 PM ! 12 alerts

Bills Daily spending By category Allotments Suggested v next view
Sorted by **Amount** ▾ o changes it 12 items

Name	Need	Amount	Repeats	Next due	In	Monthly
> Rent	Essential	\$1,450.00	Monthly	Sun Nov 1	14d	\$1,450.00
Auto insurance	Essential	\$620.00	Every 6 months	Tue Dec 15	58d	\$103.33
Vehicle registration	Essential	\$180.00	Yearly	Thu Jan 21	95d	\$15.00
Gas / fuel	Essential	\$160.00	Monthly	Wed Oct 28	10d	\$160.00
Groceries	Essential	\$120.00	Weekly	Tue Oct 20	2d	\$520.00
Electric	Essential	\$95.00	Monthly	Tue Oct 20	2d	\$95.00
Internet	Essential	\$65.00	Monthly	Thu Nov 12	25d	\$65.00
Restaurants	Flexible	\$60.00	Weekly	Thu Oct 22	4d	\$260.00
Cell phone	Essential	\$55.00	Monthly	Sun Nov 15	28d	\$55.00
Hobbies	Flexible	\$50.00	Monthly	Sun Oct 18	today	\$50.00
Gym membership	Flexible	\$35.00	Monthly	Tue Nov 3	16d	\$35.00
Streaming video	Flexible	\$18.00	Monthly	Mon Nov 9	22d	\$18.00

Per month: essential \$2,463.33 flexible \$363.00 **total \$2,826.33**
 DEMO DATA: changes are not saved (run without --demo for your own data)
 a add e edit c clone d del p paid R set aside {} look b sidebar ? help q quit

Figure 98. The Bills page sorted by Amount, largest first. The arrow in the column title and the line above the table say so.

Text sorts without regard to case, and numbers inside text sort as numbers, so “Gym 2” comes before “Gym 10”. A cell with no value, such as a bill with no due date, always sorts last, whichever way you sort.

18.3 Searching

Key	What it does
/	Search: type some words and press Ctrl-S or Enter . Only rows that contain all your words are shown, in any column, ignoring case: rent car finds rows that mention both. Leave the box empty to show everything again.
f	Filter the current column only, with words the same way.
Esc	Clear the search or filter.

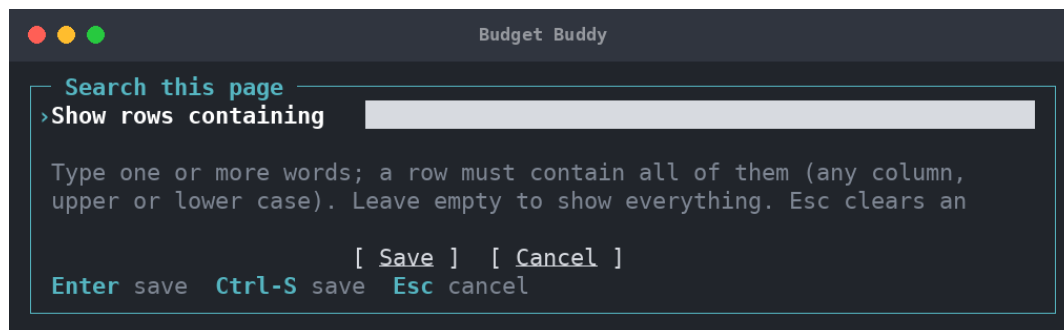


Figure 99. The search dialog.

While a search or filter is active the footer of the page, the totals line, is replaced by a line that says what is active: Search "essential" in every column 8 of 12 items Esc clears, Or Filter "ess" in Need ... for a filter on one column. The count at the right of the sorted-by line changes to “8 of 12 items” at the same time, and a message confirms “Showing 8 of 12 rows”.

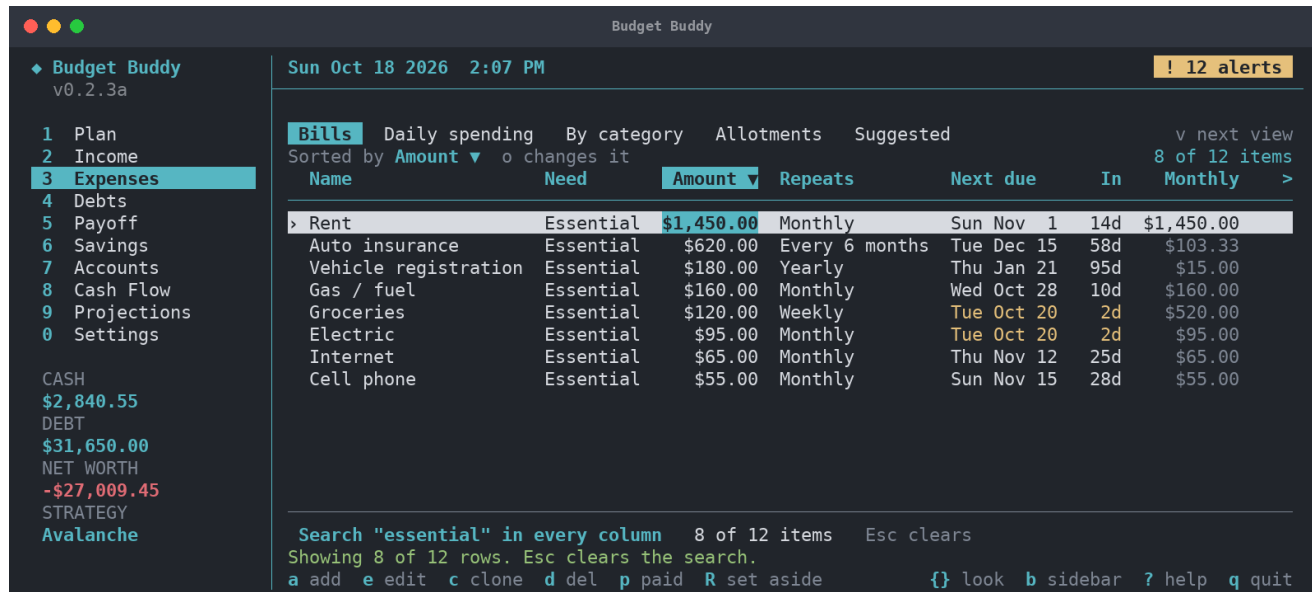


Figure 100. Searching the Bills page for “essential”. The footer line says what is active and how many rows match.

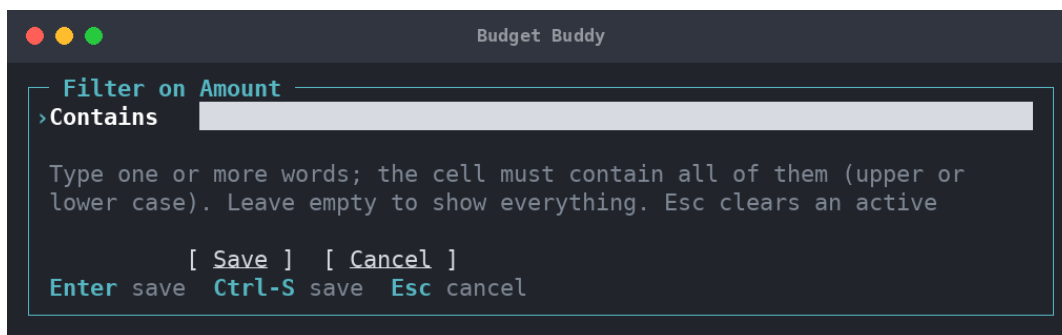


Figure 101. Filtering on the current column only, with **f**. The dialog is titled with the column's name.

18.4 Good to know

- **Actions go to the row you can see.** **a**, **e**, **d**, **p** and the rest always act on the highlighted row, however the list is sorted or searched.
- **New rows stay visible.** A row you just added is highlighted, and an active search is cleared if it would have hidden the new row.
- **Each view keeps its own.** Sort, search and current column are remembered per view while the program is open (Bills, Daily spending, Allotments and Suggested are separate). They are **not** saved to your data file.
- **Not everywhere.** Plan, Payoff, Cash Flow, Projections and Settings are documents or fixed lists, and so is the By category view of Expenses, so the sorting and searching keys just say so there.

19. How the Numbers Work

Budget Buddy is deliberately simple and predictable. This chapter explains the rules behind the numbers, so that when a figure surprises you, you can see why. None of it needs to be memorised.

19.1 Monthly equivalents

Anything that repeats is turned into an average monthly figure so every month of the Plan looks alike: weekly $\times 52 \div 12$, every 2 weeks $\times 26 \div 12$, twice a month $\times 2$, quarterly $\div 3$, every six months $\div 6$, yearly $\div 12$. One-time items have no monthly equivalent; they count only in the month they fall in. Amounts are whole cents throughout.

19.2 The allocation order

The Plan spends each month's income in a fixed order, and each step receives only what is left. The numbered steps are listed in section 5.2. Three details are worth knowing:

- **Essentials, debt minimums and deadline goals are never trimmed.** If they exceed your income, the Plan says so and cuts flexible spending to zero; it does not hide the problem.
- **Flexible spending can be trimmed**, but only as far as needed to reach your *target* for extra debt and savings (20% of income by default), and never by more than the *max trim* (25%).
- **Whatever cannot be used is not lost.** If the debt side of the final split has nothing left to pay, the savings side receives the money, and the other way round. Anything still unplaced is shown as *Unassigned*.

19.3 Debt interest and payoff

- Interest for one month is the balance $\times$ APR $\div 12$, rounded to the cent. It is a good estimate, not a statement: real lenders compute interest daily and on their own schedule, so expect small differences.
- The payoff projection pays every debt's minimum each month, then puts the extra money on the first debt in the order described in section 8.4. When a debt reaches zero, its minimum **rolls over** to the next one.
- The projection stops after 100 years, or earlier if a balance can never be paid off because the minimum does not cover the interest.
- When you record a payment on the Debts page, one month of interest is deducted first and the remainder reduces the balance.

19.4 Dates and schedules

- A bill due on the 29th, 30th or 31st falls on the **last day** of shorter months and goes back to its day afterwards.
- Recording a payment moves a bill or debt to its **next occurrence**. Income needs no recording; its next date is always worked out from its schedule.
- A *Twice a month* schedule uses two days of the month (for example the 1st and 15th, or the 15th and the last day).

- “Today” comes from your computer's clock and is refreshed while the program is running, so leaving it open overnight is fine.

19.5 Set-asides and the forecast

- Payments are planned **buffer days** early (one by default). The Cash Flow and Projections pages use that earlier date.
- A bill is funded by the paychecks that arrive after the previous payment was due and before this one is due, sharing the cost in proportion to paycheck size. A bill due before any paycheck can fund it is flagged as coming from cash on hand.
- Money you have already set aside for a bill counts toward its next payment only. The Cash Flow forecast still deducts the *full* payment, because that money is still sitting in your account.
- The forecast starts from the **total** of your spendable accounts.

19.6 Allotments

Allotments are standing monthly amounts. For any month, *Spent* is the purchases dated in that month plus the bills *due* in that month that are marked paid, counted in the month due. Suggestions are the larger of what your bills add up to and your three-month average spending, trimmed in the same way the Plan trims flexible spending when the plan has to. Section [7.6](#) describes the screen.

20. Command Line and Environment

Run `budget` with no options to start the program. This chapter lists the options, the environment variables and the exit statuses. `budget --help` prints the same summary:

```
Budget Buddy by Electric Shepherd: plan your money in the terminal

USAGE
  budget [-Dn] [-c LOOK] [-p PAGE] [-f FILE]
  budget -d [-n] [-f FILE | -D] [-s WxH] [-k KEYS] [PAGE | -p PAGE]

OPTIONS
  -f, --file FILE      use this data file (default: $BUDGET_FILE, or
                        $XDG_DATA_HOME/budget-tui/budget.dat, or
                        ~/.local/share/budget-tui/budget.dat)
  -D, --demo           start with sample data; nothing is saved
  -c, --colors LOOK    use this color look for this run only; the saved setting is
                        not changed. One of: full, teal, winter, harbor, dusk,
                        shepherd, spring, autumn, ember, samhain, white, green, amber
  -p, --page PAGE      open on this page (default: the Start on page setting)
  -d, --dump           print a page as plain text and exit (no terminal needed)
  -s, --size WxH       screen size for --dump, 1 to 1000 each (default 100x32)
  -k, --keys KEYS      with --dump: press these keys first, e.g. "a Tab Down Enter"
  -n, --no-splash      skip the start-up splash screen
  -h, --help           show this help
  -V, --version        show the version

Short options can be grouped (-dn) and take their value attached or
separate (-ffile, -f file). -- ends the options.

PAGE: plan income expenses debts payoff savings accounts cashflow projections
settings, or the number of its key (1 to 9, 0). The default is plan with --dump.

ENVIRONMENT
  BUDGET_FILE          data file location
  BUDGET_CONFIG        settings file location (default: $XDG_CONFIG_HOME/budget-tui/settings,
                        or ~/.config/budget-tui/settings)
  BUDGET_TODAY         pretend today is this date (YYYY-MM-DD)
  BUDGET_ASCII=1       draw with plain ASCII instead of box characters
  BUDGET_UNICODE=1     draw with box characters even if the locale isn't UTF-8
  BUDGET_NO_SPLASH=1   skip the start-up splash screen
  BUDGET_TIME=HH:MM    with BUDGET_TODAY: pretend it is this time of day (otherwise the clock is hidden)
  COLUMNS, LINES      override the terminal's width and height
  NO_COLOR             start in Mono white (-c wins over it; neither is saved)

EXIT STATUS
  0 finished normally          1 an error (file, terminal, save)
  2 the command line or an environment setting was wrong

Inside the program, press ? for the key reference.
```

20.1 Options

Option	What it does
<code>-f FILE, --file FILE</code>	Use this data file instead of the default. <code>--file=FILE</code> and <code>-fFILE</code> mean the same. Handy for keeping separate budgets, or practising on a copy. The file is created on the first save that changes something. The settings file (section 17.2) is separate and is not affected.
<code>-D, --demo</code>	Load the built-in sample budget. Nothing is saved (neither the data nor the settings), and the message line says so. It cannot be combined with <code>--file</code> .

Option	What it does
-c LOOK, --colors LOOK	Use this colour look for this run only ; the saved Colors setting is not changed. LOOK is one of full, teal, winter, harbor, dusk, shepherd, spring, autumn, ember, samhain, white, green or amber (section 17.1). An unknown name is reported and nothing is started.
-p PAGE, --page PAGE	Open on this page instead of the page the Start on page setting names, for this run only. PAGE is a name (plan, income, expenses, debts, payoff, savings, accounts, cashflow, projections, settings) or the number of its key, 1 to 9 or 0. With --dump, the page to print.
-d, --dump	Print one screen as plain text and exit; it does not need a real terminal and never writes the data file or the settings file. Name the screen with a PAGE after the options (or with -p). The old names setaside and aside still work for projections. The default is plan. It reads your settings file for the things that change the layout (the sidebar and the clock), except with --demo, which always uses the defaults.
-s WxH, --size WxH	The size of the screen for --dump, such as 120x40, each number from 1 to 1000. The default is 100x32, which gives the sidebar layout; try 80x24 to see the top bar.
-k KEYS, --keys KEYS	With --dump, press these keys first. See below.
-n, --no-splash	Skip the start-up splash screen.
-h, --help	Show a summary of the options.
-V, --version	Print the product name, version, stage and build date on one line, for example Budget Buddy v0.2.3a (alpha, built 2026-10-22).

20.2 Writing the options

Budget Buddy follows the usual Unix conventions for options:

- Short options can be **grouped**: budget -dn means budget --dump --no-splash.
- A value can be written **attached or separate**: -ffile, -f file and --file=file are the same, and so are -s80x24 and -s 80x24.
- -- ends the options. Everything after it is a PAGE, even if it starts with a dash.
- The page may come before or after the options (budget --dump income and budget income --dump). If the environment variable POSIXLY_CORRECT is set, the first page name ends the options, as the strictest reading of the standard has it.
- A mistake in how you called the program is **reported**, never ignored: --size or --keys without --dump, an unknown page or colour look, --page together with a page name, --demo together with --file, or a size outside 1 to 1000. Nothing is started, and the exit status is 2.
- -p and -c work with or without --dump; a bare page name (budget --dump income) only with --dump.

20.3 Exit status

Status	Meaning
0	The program ended normally, or printed what you asked for.

Status	Meaning
1	A problem while running: the data file could not be read or saved, the terminal went away, or the output could not be written (for example a full disk or a closed pipe). If the last save failed when you quit, the reason is printed after the screen is restored.
2	The command line, or an environment setting such as BUDGET_TODAY, was wrong. A message says what, and nothing was started.

20.4 Environment variables

Variable	Effect
BUDGET_FILE	Path of the data file (the --file option wins).
BUDGET_CONFIG	Path of the settings file (section 17.2). Without it the file is \$XDG_CONFIG_HOME/budget-tui/settings (when XDG_CONFIG_HOME is an absolute path) or ~/.config/budget-tui/settings.
XDG_CONFIG_HOME	See above. A relative path is ignored.
XDG_DATA_HOME	If set to an absolute path (and no BUDGET_FILE), the data file is \$XDG_DATA_HOME/budget-tui/budget.dat. A relative path is ignored, as the XDG standard says. Otherwise the file is under ~/.local/share.
BUDGET_NO_SPLASH=1	Skip the splash screen (any value except empty or 0).
NO_COLOR	Set to anything except empty to turn colours off (see no-color.org). The program starts in the Mono white look. It wins over the saved Colors setting, and -c wins over it. It is never saved.
BUDGET_ASCII=1	Draw boxes and bars with plain ASCII characters instead of Unicode.
BUDGET_UNICODE=1	Draw with box characters even if your locale does not announce UTF-8. (BUDGET_ASCII wins if both are set.)
COLUMNS, LINES	When set to positive whole numbers they override the width and height of the terminal, the usual rule for terminal programs. If only one is set, the other still comes from the terminal. With --dump, --size decides instead.
BUDGET_TODAY=YYYY-MM-DD	Pretend today is this date (a year from 1900 to 2200). Useful for testing and for screenshots; do not set it in your everyday environment. The clock is hidden while it is set, unless you also set BUDGET_TIME. A value that is not a date is reported and the program does not start.
BUDGET_TIME=HH:MM	With BUDGET_TODAY: pretend it is this time of day (for example 14:07), so the clock shows. Without BUDGET_TODAY it has no effect.
POSIXLY_CORRECT	If set, the first page name ends the options (see above).
LANG, LC_ALL, LC_CTYPE	If none of them names a UTF-8 locale, Budget Buddy draws in ASCII automatically.
LC_ALL, LC_TIME, LANG	When an imported bank file's dates cannot be told apart as day-first or month-first, your language and region decide (section 13.8).

20.5 Printing a screen with --dump

--dump is convenient for pasting a screen into an email or a bug report, and for scripts:

```
budget --dump cashflow --size 100x30
budget --file ~/budget-copy.dat --dump expenses --size 120x30 --keys "a Enter Enter text:Rent"
budget --demo --dump projections --size 120x40
budget -D -d -s 100x24 -k "v v" expenses
budget -Dd -s 80x24 plan
```

--keys takes a space-separated list of key names, pressed in order before the screen is printed. Single characters are themselves; text:abc types a word. The names are Enter, Esc, Tab, BackTab (also S-Tab), Backspace, Delete, Insert, Up, Down, Left, Right, Home, End, PgUp, PgDn, Space and F1 to F12; C-s or Ctrl-s is Ctrl-S and M-1 or Alt-1 is Alt-1. Keys are named as the keyboard sends them, so a capital letter is the letter itself (B, O), not S-b.

For example, to print the Bills page sorted by Amount, largest first (three columns to the right, sort, flip):

```
budget -D -d -s 120x24 -k "l l l o r" expenses
```

Note. --dump works on a copy in memory; whatever the keys do is never written to your data file.

21. Your Data File and Backups

21.1 Where it is and how it is saved

Your whole budget is one plain-text file. By default it is `~/.local/share/budget-tui/budget.dat` (or under `$XDG_DATA_HOME`, if that is set to an absolute path); `--file` and `BUDGET_FILE` choose another. Budget Buddy saves after **every** change. The program's own settings (colour look, clock, splash, start page, sidebar and mouse) are not in this file; they are in a small settings file of their own (section 21.3).

- **Atomic saves.** The new version is written to a temporary file beside the data file (its name ends in `.tmp` and includes the program's process number), flushed to disk, and then renamed over the old file. A crash or power cut leaves either the old file or the new one, never half of each. If the data file is a symbolic link, the file it points to is updated and the link stays.
- **Checked before it is written.** Before saving, the program reads back what it is about to write. If it could not read that text again, it refuses to save and says so, and the file on disk is left as it was.
- **An automatic backup.** The first save of each session first copies the existing file to `budget.dat.bak`. If the backup cannot be written, the save is stopped instead of going ahead without it.
- **Nothing is written without a reason.** A session in which nothing changed writes nothing, and creates no file.
- **Private to you.** The data file, the backup, the lock file and `crash.log` are created readable and writable by their owner only (permissions `600`), and any folder the program has to create gets `700`. A folder or file that already exists keeps the permissions it has. If your files come from an older version, tighten them once: `chmod 600 budget.dat budget.dat.bak` and `chmod 700 ~/.local/share/budget-tui`.
- **One copy at a time.** While it runs, Budget Buddy keeps a small lock file beside the data file (`budget.dat.lock`) holding the number of the running process. A second copy aimed at the same file refuses to start and says which process has it. A lock left behind by a copy that was killed, or by a power cut, is recognised as stale and taken over automatically. A lock held by a running copy is never broken. If the message says the file is in use but you are sure no Budget Buddy is running, delete the lock file the message names.
- **A save that fails is never silent.** If a save fails (a full disk, a folder that is no longer writable), the message line says `SAVE FAILED` and a red `NOT SAVED:` line stays at the bottom of the screen until a save works; quitting asks before throwing the changes away (section 3.7).
- **A file it does not understand is never overwritten.** The program stops with a message that names the file and the line, so you can look at it.

21.2 What is in it

A small header line, a note, then one record per line, tab-separated. Amounts are in cents and rates in hundredths of a percent ("basis points"), so \$1,450.00 is 145000 and 24.99% is 2499.

Record	What it holds
setting	One of the rules of the Plan (Settings page), and the show-year choice. Since 0.2.3a the colour look and the clock are no longer written here (section 21.3).
account	An account: id, name, type, whether it is spendable, balance and the date it was last confirmed.

Record	What it holds
adj	A history entry on an account.
bank	Marks which history entries came from which imported file, so a whole import can be removed.
spend	A daily-ledger purchase.
allot	A category's monthly allotment, for example allot<TAB>dining<TAB>20000.
income, expense, debt, goal	The rows of the Income, Bills, Debts and Savings pages.

The first line is `budget -tui v1`. It is called that for historical reasons and is kept unchanged so that data from earlier builds keeps loading.

When the file is loaded it is **checked**, not just read. Settings must lie within their allowed ranges, amounts cannot be negative, dates must fall between 1900 and 2200, account numbers must be unique, and every record that refers to an account, a history entry or a purchase must refer to one that exists. The first problem found is reported with its line number and a plain description, and nothing is changed.

21.3 The settings file

How the program looks and starts is kept apart from your money, in `~/.config/budget-tui/settings` (or `$XDG_CONFIG_HOME/budget-tui/settings`, or the file named by `BUDGET_CONFIG`). It is a few lines of `name=value` text, written the same careful way as the data file (a temporary file, then a rename) and readable by you only (600, in a 700 folder). Section 17.2 describes the lines. Nothing in it is financial, but if you move to another computer and want your look and start page too, copy it with the data file. Demo mode neither reads nor writes it, and `--dump` reads it but never writes it.

21.4 Compatibility

- Files written by earlier builds (0.1a to 0.2.2a) load unchanged in Budget Buddy v0.2.3a, and a file saved by this version still opens in the earlier ones.
- Files that contain `allot` lines (allotments) are **refused by older builds**, which would otherwise silently drop them. If you go back to an older build, keep a copy of the file from before you used allotments.
- Unknown setting keys are ignored, so files stay readable across versions. Versions 0.2a to 0.2.2a kept two program settings in the data file, `color_mode` and `clock`. Version 0.2.3a still **reads** them, but only once: when there is no settings file yet, those two values become your starting look and clock, and from then on the settings file decides. A 0.2.3a save does not write them, so if you go back to an older build it starts with its default look. A value that is not recognised falls back to the default.
- A hand-edited file is checked when it loads; a bad line is reported by line number and nothing is changed.

21.5 Backing up and moving

- Quit Budget Buddy (not strictly required, since every change is already saved).
- Copy `budget.dat` to wherever you keep backups; you may copy `budget.dat.bak` too. There is no need to copy the `.lock` file. Copy the settings file as well if you want the same look and start page.

3. To use it on another computer, put it in that computer's data folder, or run `budget --file /path/to/budget.dat`.

The data file contains your balances and the descriptions of the transactions you imported. Treat it, and any CSV you export, like any other financial document. The file is plain text and is not encrypted; the 600 permissions keep other accounts on the same computer out, but not anyone who can read your files.

21.6 Starting over

To start a completely new budget, move the data file aside (do not delete it until you are sure) and start the program; it begins empty. To practise without risk, use `budget --file /tmp/practice.dat` or `budget --demo`.

22. Troubleshooting

Problem	What to do
"Terminal too small"	Make the window at least 50 columns by 12 rows. A standard 80 by 24 window works well; 100 columns by 24 rows or more gives the sidebar, and about 160 by 34 shows every column of the widest table.
Boxes and bars look like ? or odd characters	Your terminal or locale is not using UTF-8. Set <code>BUDGET_ASCII=1</code> , or fix <code>LANG</code> (for example <code>en_US.UTF-8</code>) and use a font with box-drawing characters.
Colours are hard to read	Press <code>]</code> to step through the thirteen looks until one suits you, or open Settings (<code>o</code>) and choose Colors (section 17.1). Mono white, Mono green and Mono amber use no colours at all; <code>NO_COLOR=1</code> or <code>-c white</code> gets Mono white for one run. Everything stays readable in every look: the highlighted row and the current page are also shown by brightness or reverse video.
The terminal looks broken after the program ends unexpectedly	Type <code>reset</code> and press Enter (or <code>stty sane</code>). Budget Buddy restores the terminal when it is closed, interrupted or crashes, and a helper started with it puts the screen back even if the program is stopped with <code>kill</code> or <code>kill -9</code> , so this is rare.
The splash screen waits for me	It waits for a key, by design, and the key is used up. To skip it, use <code>--no-splash</code> , set <code>BUDGET_NO_SPLASH=1</code> in your shell profile, or set the Splash screen setting to Off (section 17.1).
The backgrounds are the wrong shade, or the colours look rough	The painted looks (Winter to Samhain) use exact colours when your terminal says it supports them (<code>COLORTERM=truecolor</code>) and the closest of the 256 standard colours otherwise. If your terminal supports 24-bit colour but does not say so, set <code>COLORTERM=truecolor</code> in your shell profile. Full color and Blue/teal paint no background, so they take your terminal's own.
I do not see "Budget Buddy v0.2.3a"	In the sidebar layout it is at the top of the sidebar. In the top-bar layout it is at the bottom right, on the line just above the key hints, and steps aside while a long message is showing, returning on the next key. In a terminal that is too small to draw the normal screen it is not shown either.
The sidebar is missing	It needs a window of at least 100 columns by 24 rows, and the Sidebar menu setting must be On. Press <code>b</code> to turn it on or off (section 2.1).
A column is missing from a table	The window is too narrow for all of it. A <code>></code> at the right end of the headings says that columns are hidden; press <code>l</code> to bring them into view, or make the window wider. Chapter 2 says how wide each table needs to be.
A key I know does not do what it used to	Version 0.2.3a changed several keys to match the other Electric Shepherd programs. Appendix F lists the ones that moved. <code>?</code> always shows the keys of the page you are on.
The date is wrong, or the clock is missing	Budget Buddy uses your computer's clock. Check the system date and time zone. (Is <code>BUDGET_TODAY</code> set in your environment? While it is, the clock is hidden unless <code>BUDGET_TIME</code> is set too.) The clock can also be turned off in Settings (section 17.1).
My balance is not what my bank says	Press <code>B</code> (on any page) and enter the bank's figure (section 11.3). The difference is recorded in the history. If it is large, the history shows what has been entered since the last time it matched.
Cash Flow warns of a gap but I have money elsewhere	Only accounts with <i>Counts as bill cash</i> = Yes are used. Edit the account (<code>e</code>) if it should count, or transfer money into your checking account.

Problem	What to do
An imported file shows spending as positive (or income as negative)	In the preview press S to flip the signs. If you already imported, press D on an imported line in the history to remove the whole import, and import again.
An import says transactions are “already here”	They were imported before, or match entries you made by hand. That is intended: nothing is doubled. If your bank words a description differently in two exports, the line shows as new in the preview.
The program says it cannot read my data file	The message names the file and line. The file is untouched. Besides damaged lines, it also reports values that are out of range (such as a date outside 1900 to 2200) and records that refer to an account, history entry or purchase that does not exist. You can fix the line with a text editor, or restore <code>budget.dat.bak</code> (copy it over <code>budget.dat</code> while the program is closed).
A red NOT SAVED: line is at the bottom of the screen	The last save failed, for instance because the disk is full, the folder is no longer writable or a drive was unplugged. Your changes are still in the program but not on disk. Fix the cause, then make any change: the next save that works writes everything and clears the line. If you quit first, the program tries once more and asks before it throws the changes away.
The program says the data file is “already open in another Budget Buddy”	Only one copy can use a data file at a time, so that two copies cannot overwrite each other's changes. The message gives the other copy's process number: close that copy first. If you are certain nothing is running (the message is left over from something that was stopped in an unusual way), delete the <code>.lock</code> file the message names and start again. Normally a leftover lock is taken over without any message.
An amount is refused with “uses a comma where the cents go”	Type the cents after a point: <code>12.50</code> , not <code>12,50</code> . A comma in an amount is only for thousands, in groups of three (<code>1,250</code>).
A date is refused as out of range	Dates must lie between the years 1900 and 2200, whether typed in a form or found in a file. Check for a mistyped year or a very large + offset.
An older build refuses my file	The file contains allotments (allot lines) that only newer builds understand. Use Budget Buddy v0.1a or newer, or restore a backup from before you used allotments.
<i>Internal error, sorry. Your saved data is intact.</i>	A bug was caught. The screen recovered, the form closed and your last saved data was reloaded. Details were appended to <code>crash.log</code> next to the data file (section 22.1).
Arrow keys or Tab misbehave inside tmux, screen or an SSH session	Use a standard xterm-compatible terminal type. Meanwhile most movement has a letter alternative: j / k move up and down, h / l move between columns, and 1 to 9 , 0 ,] and [jump between pages.
Clicking or scrolling does nothing	Check that the Mouse setting is On (section 17.1). Some terminals, terminal multiplexers and remote sessions do not pass the mouse on; everything can still be done from the keyboard.
I cannot select text with the mouse	While the Mouse setting is On the terminal gives clicks to the program, so hold Shift while dragging to select text (most terminals), or turn the Mouse setting Off.
The program says “Settings: settings line 3: bad value ...”	A line of the settings file was not understood and that one setting is on its default; the others apply. Fix or delete the line (section 17.2).
The program says “SETTING NOT SAVED”	The settings file could not be written (a folder that is not writable, a full disk). The setting works until you quit. Fix the cause, then choose the setting again. The data file is not affected.

22.1 The crash log

If Budget Buddy ever hits a bug it did not expect, it does not close. It shows a red message, closes any open form, reloads your last saved data, and appends a report to a file called `crash.log` in the same folder as your data file. Each report records:

- the date and operating system, and the **product version and build** (Budget Buddy v0.2.3a (alpha, built 2026-10-22));
- what the program was doing, and where the problem occurred;
- the page it was on and the terminal size;
- **your last keystrokes**, which let a developer replay the problem exactly.

Because it records keystrokes, it can include text you typed. Like your data file, it can be read only by you (permissions 600). Look it over before you share it. A crash can be replayed on a copy of your data without touching the original:

```
budget --file COPY_OF_YOUR_DATA --dump expenses --size 100x30 --keys "a Enter g r o c e r i e s Enter"
```

Tip. When you report a problem, include the version line from `budget --version` and the matching entry from `crash.log`.

23. Limits and Known Issues

Budget Buddy v0.2.3a is an **alpha** release: complete enough to use every day, but young. This chapter lists what it does not do, and the rough edges we know about.

23.1 By design

- **One currency**, shown as \$.
- **Debt interest is an estimate**: modelled monthly from the APR. Your lender's statement is the final word.
- **The cash-flow forecast uses the total** of your spendable accounts, not each account separately.
- **Plain-text data**. Your data file is readable (and editable) text. It is not encrypted; protect it as you would any financial document.
- **One budget at a time**. To keep several, use separate files with `--file`.
- **Bank import is CSV only**. PDF, Excel and other statement formats are not read.
- **Unix-style terminals only**. This release is tested on Linux (x86_64). Other Unix-like systems such as macOS are expected to work but have not been tested, and native Windows is not supported (use WSL). The supplied Linux program needs glibc 2.39 or newer, and no ARM (arm64) version is supplied.
- **One copy per data file**. A second copy on the same file is refused (section 21.1). The check protects against another Budget Buddy, not against other programs editing the file.
- **Projections has two views so far** (by paycheck and by bill). Graphs and a calendar are planned for it; they are not in this release.
- **The clock needs the system's date command**. Without it the clock is hidden and the date falls back to UTC.
- **Program settings belong to the computer**. The look, clock, splash, start page, sidebar and mouse are kept in the settings file, not in the data file, so copying a budget to another computer does not bring them (section 21.3).
- **Sorting and searching are not saved**. Each view starts in its natural order every time the program starts (Chapter 18).
- **The Alt keys need the terminal's help**. `Alt-1` to `Alt-9` work in terminals that send Alt as an Escape prefix (nearly all do, but a few terminals need a setting turned on). Where they do not, moving to the column with `h` and `l` and pressing `o`, or choosing it in the `o` dialog, does the same.
- **No general undo**. `u` undoes the last entry added to an account; other changes are undone by editing or deleting them.
- **Deleting an account-history entry for a bill or debt payment** corrects the account only. It does not move the bill's due date back or restore the debt's balance.

23.2 Known rough edges

- **Wide characters**. Characters that take two columns on screen (Chinese, Japanese and Korean text, many emoji) are counted as one, so lines that contain them can be misaligned. Imported bank descriptions in those scripts are stored correctly.

- **Pasting.** Pasted text arrives as typed keys, so a line break in it can act as **Enter**. Paste one line at a time into forms.
- **No suspending.** While Budget Buddy runs, **ctrl-z** is an ordinary key and does not suspend the program to the background; quit with **q** and start it again.
- **Resizing.** The window size is checked once a second, so a resized window redraws within about a second.
- **Mouse reporting.** The mouse works in terminals that report it (nearly all current ones). Holding **Shift** while dragging selects text in most of them; the **Mouse** setting turns the whole thing off (section 17.1).
- **The terminal type.** The `TERM` setting is not consulted: Budget Buddy assumes an ANSI (xterm-compatible) terminal.
- **Leftover locks.** A lock file from a copy that died is taken over automatically. If the process number in an old lock has since been given to an unrelated running program, the lock looks live, and you are told to delete it (section 22).
- On a very narrow terminal some key hints are omitted; press **?** for the complete list.
- In the top-bar layout, a long message on the message line makes the “Budget Buddy v0.2.3a” label step aside until the next key. (In the sidebar layout the label is in the sidebar and is never covered.)

23.3 Reporting a problem

Please include the line printed by `budget --version`, what you were doing, and the matching report from `crash.log` if there is one (look it over first, because it contains the keys you pressed). For a calculation that looks wrong, a screen printed with `--dump` from a copy of your data file shows exactly what you saw.

Appendix A. Key Reference

Every key, by screen. It matches the key reference inside the program (**?** or **F1**). Letters are case-sensitive: **o** is Shift-O. Keys that changed in version 0.2.3a are listed in Appendix F.

A.1 Everywhere

Key	Action
1 to 9 , 0	Go to a page: Plan, Income, Expenses, Debts, Payoff, Savings, Accounts, Cash Flow, Projections, Settings (0).
Tab , Right ,]	Next page.
Shift-Tab , Left , [	Previous page.
Up Down j k	Move the highlighted row, or scroll.
PgUp PgDn Home End	Page up / down; first / last (g and G too).
v / V	The next / previous view of the page (named on the line at the top of the page).
b	Sidebar on / off (it shows in windows 100 columns by 24 rows or larger).
{ / }	The previous / next colour look, saved as if chosen in Settings.
B	Reconcile: set an account's balance to match your bank.
w	Save now (every change is saved anyway).
Ctrl-L	Redraw the screen.
? , F1	The key reference.
q , Ctrl-C	Quit.
Mouse	Click a page, a view, a row (again to open it), a heading (sorts), a key hint, a field, a list entry or a button. The wheel scrolls.

A.2 Income, Expenses, Debts, Savings, Accounts

The same keys on every table.

Key	Action
a	Add a row.
e , Enter	Edit the highlighted row.
c	Clone the highlighted row; the name is left empty for you to fill in.
d , Delete	Delete the highlighted row (asks first).
Space	View every field of the row in full. On Accounts: its history.
h / l	Move the current column left / right.
o , s	Sort by the current column; again, to reverse.

Key	Action
O	Choose a sort column and direction.
Alt-1 to Alt-9	Sort by the 1st to 9th column from the left; again, to reverse.
r	Flip the sort direction.
/	Search every column for some words.
f	Filter on the current column only.
Esc	Clear the search or filter.

A.3 Plan

Key	Action
v / v	Detail or Steps.
, / .	Previous / next month (< and > too).
t	This month.
s	Switch between Avalanche and Snowball.

A.4 Expenses

Key	Where	Action
v / v	Expenses	Bills, Daily spending, By category, Allotments, Suggested.
p	Bills	Record a payment (it can come out of an account).
R	Bills, Debts	Edit the amount set aside.
a e d	Daily spending	Add, edit or delete a purchase; it comes out of an account and shows in its history.
, . t	Daily spending, Allotments	Previous month / next month / this month.
a e d	Allotments	Assign, edit or clear a category's monthly amount.
u	Allotments	Use the suggested amount for the highlighted category.
U	Allotments	Use the suggestion for every category that has none.

A.5 Debts, Payoff

Key	Where	Action
p	Debts	Record a payment.
R	Debts	Edit the amount set aside.
s	Debts, Payoff	Switch between Avalanche and Snowball.

Key	Where	Action
W	Payoff	What-if: extra money per month.
m	Payoff	Monthly / yearly rows.
v / v , , / .	Payoff	All debts together, or one debt at a time.

A.6 Savings and Accounts

Key	Where	Action
+ , =	Savings	Add money to a goal (it can come out of an account).
+ , =	Accounts	Deposit.
- , _	Accounts	Withdrawal.
t	Accounts	Transfer between accounts.
Space , H	Accounts	Open the history.
u	Accounts	Undo the last entry added.
i	Accounts	Import a bank CSV (you check a preview first).
x	Accounts	Export history to a CSV file.
B	Everywhere	Set the balance of an account to match the bank.

A.7 Account history

Key	Action
Up , Down , j , k , PgUp , PgDn , Home , End	Move through entries (a page is ten entries).
/	Filter by words, dates, amount, direction, source.
o , O	Choose a column and direction to sort by.
r	Flip the sort direction.
a	Every account / one account.
f	Full screen on / off.
x	Export what is shown.
d , Delete	Delete the entry.
D	Delete the whole bank import the entry came from.
u	Undo the last entry added.
y	Show or hide the year.
Esc	Clear filters; then close.
Enter , q , h	Close.

A.8 Import preview

Key	Action
Enter	Import.
Esc , q , n	Cancel.
s	Flip the signs.
m , Tab	Next balance mode (M , Shift-Tab : previous).
y	Show or hide the year.
Up Down PgUp PgDn Home End	Scroll the list of new lines.

A.9 Cash Flow, Projections, Settings

Key	Where	Action
j k PgUp PgDn	Cash Flow	Scroll.
v / V	Projections	By paycheck / by bill.
m	Projections (by paycheck)	Record that you moved a paycheck's set-asides.
R	Projections (by bill)	Edit the amount set aside for the bill.
Enter , e , Space	Settings	Open the menu or form for the highlighted setting.

A.10 Forms, menus and confirmations

Key	Action
Tab Down / Shift-Tab Up	Next / previous field; after the last, the Save and Cancel buttons.
Enter	Text field: next field, and save from the last. Menu field: open the menu. Button: press it.
Typing	In a menu: narrows the list. In a field with a value: replaces it.
Left Right	Menu field: previous / next choice. Text field: move the cursor.
Home Ctrl-A / End Ctrl-E	Start / end of the text.
Ctrl-U / Ctrl-K	Delete everything before / after the cursor.
Ctrl-N / Ctrl-P	On a field with suggestions: show the list, then step through it.
Ctrl-S , F2	Save from any field.
Esc	Close an open menu; then cancel the form.
y , Y , Enter / n , N , Esc , q	Confirmation boxes: yes / no.
In a Settings menu	Up Down move, Home End jump, Enter or Space applies, Esc puts the old choice back.

A.11 Typing dates and money

You type	Meaning
2026-11-01, 11/1, Nov 1	A date.
15	The next 15th.
+7, +2w	Seven days / two weeks from today.
today, tomorrow	What they say.
1500, 1,500.50, \$42	Money. A comma is only a thousands separator.
24.99, 24.99%	A percentage.

Appendix B. Categories and Types

These are the choices on the Type and Category menus. Picking one fills in sensible defaults; you can change them, and you can type any name you like.

B.1 Expense categories

Category	Usually	Usual frequency	Typical names
Housing	Essential	Monthly	Rent, Mortgage, HOA dues, Property tax, Home repairs
Utilities	Essential	Monthly	Electric, Gas / heating, Water & sewer, Trash & recycling
Phone & Internet	Essential	Monthly	Cell phone, Internet, Cable TV
Groceries	Essential	Weekly	Groceries, Household supplies
Transportation	Essential	Monthly	Gas / fuel, Public transit, Parking, Car maintenance, Tolls
Insurance	Essential	Monthly	Auto, Health, Renters, Home and Life insurance
Healthcare	Essential	Monthly	Prescriptions, Doctor / dental, Copays, Therapy
Childcare & Kids	Essential	Monthly	Daycare, School fees, Kids activities, Child support
Education	Essential	Monthly	Tuition, Books & supplies, Courses
Taxes & Fees	Essential	Yearly	Vehicle registration, Income tax payment, License renewal, Bank fees
Pets	Essential	Monthly	Pet food, Vet, Grooming
Subscriptions	Flexible	Monthly	Streaming video, Music, Software, Gym membership, News & magazines
Dining Out	Flexible	Weekly	Restaurants, Coffee, Takeout & delivery
Entertainment	Flexible	Monthly	Movies & events, Games, Hobbies
Shopping	Flexible	Monthly	Clothing, Electronics, Home goods
Personal Care	Flexible	Monthly	Haircut, Salon / spa, Toiletries
Travel	Flexible	Monthly	Vacation, Flights, Lodging
Gifts & Donations	Flexible	Monthly	Gifts, Charity, Holidays
Other	Flexible	Monthly	

B.2 Income types

Type	Usual frequency	Typical names
Paycheck / Salary	Every 2 weeks	Paycheck, Salary
Hourly / Weekly pay	Weekly	Weekly pay
Side gig / Freelance	Monthly	Freelance, Side gig, Gig work

Type	Usual frequency	Typical names
Bonus	One time	Bonus, Commission
Tax refund	One time	Tax refund
Rental income	Monthly	Rent received
Benefits / Social Security	Monthly	Social Security, Disability, Unemployment
Pension / Retirement	Monthly	Pension, Retirement withdrawal
Alimony / Child support	Monthly	Child support received, Alimony received
Interest & Dividends	Quarterly	Dividends, Interest
Gift	One time	Gift
Other	Monthly	

B.3 Debt types

Credit card, Store card / Buy-now-pay-later, Auto loan, Student loan, Mortgage, Personal loan, Home equity / HELOC, Medical bill, Tax debt, Family / friend loan, Payday loan, Other.

B.4 Savings goal types

Emergency fund, Vacation / Travel, Home down payment, Car / Vehicle, Retirement, Education, Wedding / Event, Holiday gifts, Medical / Dental, Home repairs, Taxes, Big purchase, Other.

B.5 Account types

Type	Counts as bill cash by default
Checking	Yes
Savings	No
Cash on hand	Yes
Money market	No
Brokerage / Investments	No
Retirement account	No
HSA / FSA	No
Other	No

B.6 Repeat frequencies

One time, Weekly, Every 2 weeks, Twice a month, Monthly, Quarterly, Every 6 months, Yearly.

Appendix C. Input Syntax

C.1 Dates (in forms)

You type	Meaning
2026-11-01	Full date.
11/1, 11/01	Month / day.
Nov 1	Month name and day.
15	The next 15th.
+7	Seven days from today.
+2w	Two weeks from today.
today, tomorrow	What they say.

Dates must fall between the years 1900 and 2200.

C.2 Money and percentages

Field	Accepts
Money	1500, 1,500.50, \$42. A comma only separates thousands, in groups of three: 12,50 is refused, with a hint to type 12.50. Most money fields must be zero or more; a purchase must be more than zero.
Percent (APR)	24.99 or 24.99%. Use 0 for none.
Whole numbers	Months (0 to 36), percentages (0 to 100), buffer days (0 to 14), forecast horizon (14 to 365).

C.3 History filter

Part	Examples
Words	rent; coffee shop (both words); "coffee shop" (phrase); rent mortgage (either); rent -car (not car).
Dates	2025; Mar 2023; 2023-03; 2024-06-15; 2024-01-01..2024-03-31; Jan 2024..Mar 2024; since 2024; until Mar 2025; before 2022; after 2023; this year; last year; this month; last month; ytd; last 90 days; several, separated by commas: Mar 2023, 2025.
Amount	12.50; >50; <=20; 10..100; 25...
Direction	Money in and out; Money in only; Money out only.
Came from	Anywhere; Imported from a bank file; Daily spending ledger; Transfers between accounts; Typed in by hand.

Appendix D. Glossary

Term	Meaning
Allotment	The most you plan to spend in one category in a month.
APR	Annual percentage rate: the yearly interest rate on a debt.
Avalanche	Paying extra on the debt with the highest interest rate first.
Bill cash	The money in accounts marked “Counts as bill cash”: what bills are checked against.
Cash gap	A period when the forecast shows your spendable cash below zero (or below your cushion).
Clock	The time of day shown beside the date; 12-hour, 24-hour or hidden (Settings).
Clone	A new row that starts as a copy of the highlighted one (c).
Colour look	One of thirteen ways the whole program is coloured, from Full color to Mono amber ({ and } , or Settings).
Current column	The column of a table that o , s and f act on; its heading is highlighted. h and l move it.
Earmarked	The part of an account's balance that belongs to savings goals held in it.
Essential / Flexible	Whether a bill must be paid, or can be trimmed to free money for debt and savings.
Entry	One line in an account's history.
Horizon	How many days ahead the forecast and set-aside plan look.
Lock file	A small file beside your data file (ending in . Lock) that shows a copy of Budget Buddy is using it, so that two copies cannot overwrite each other.
Minimum cash cushion	A balance you do not want your cash to fall below.
Reconcile	Set an account's balance to what the bank shows, recording the difference (B).
Set aside	Money held back from a paycheck for a coming bill (what the Projections page works out).
Settings file	The small file where the program's own settings (look, clock, splash, start page, sidebar, mouse) are kept, apart from your data.
Sidebar	The list of pages down the left of the window, shown when the terminal is at least 100 columns by 24 rows and the Sidebar menu setting is On (b). Otherwise the top bar is used.
Splash	The start-up screen with the sheep, the name and the version. It waits for a key.
Snowball	Paying extra on the debt with the smallest balance first.
Spendable	An account that counts as bill cash.
Starter fund	A first emergency cushion, built before extra debt payments.
Top bar	The row of page names along the top of the window, used when there is no sidebar.
View	One way of looking at a page. The views are named on the bar at the top of the page and v / v step through them: Bills, Daily spending, By category, Allotments and Suggested on Expenses; By paycheck and By bill on Projections; Detail and Steps on the Plan.

Appendix E. Release Notes

E.1 Version 0.2.3a (alpha)

Budget Buddy v0.2.3a, built 2026-10-22, is a release about how the program is used. It now works like Inventory Central 0.1.10a, the other Electric Shepherd program: the same keys, the same way of moving between pages and views, the same sort and search line above every list, the same dialogs, the same mouse support and the same colour looks. The keyboard and the screen changed a great deal. The plan, the bank-file reader, the data file and the command line (apart from two new options) did not. **Several keys moved:** Appendix F lists every one.

Navigation and keys

- **Pages and views.** **1** to **9** and **0** go to a page; **Tab**, **Right** and **]** go to the next one, **Shift-Tab**, **Left** and **[** to the previous. A page with more than one way of being seen names the views on a bar at the top, and **v** / **v** step through them. Expenses now has the views Bills, Daily spending, By category, Allotments and Suggested (the old **l** and **m** sub-pages); Payoff has All debts and one view per debt; Projections has By paycheck and By bill; the Plan has Detail and Steps (Chapter 2).
- **Lists.** Income, Expenses, Debts, Savings and Accounts, and the daily ledger, answer to the same keys: **a** add, **e** or **Enter** edit, **c** clone (new), **d** delete, **Space** view the whole row (new), **h** and **l** move the current column (new), **o** or **s** sort by it, **o** the Sort dialog, **r** flip the direction, **Alt-1** to **Alt-9** sort by column, **/** search every column, **f** filter the current column (new), **Esc** clear. Sorting is natural (“Card 2” before “Card 10”) and blank values sort last both ways (Chapter 18).
- **Everywhere.** **?** and **F1** open the key reference, **b** turns the sidebar on and off, **{** and **}** step through the looks, **w** saves now, **B** reconciles an account, **Ctrl-L** redraws, **S** switches the debt strategy, **R** edits a set-aside, **w** opens the Payoff what-if, **+** and **-** put money in and out (Chapter 3).
- **Forms and dialogs.** Forms end in **[Save]** and **[Cancel]** buttons; **Ctrl-S** or **F2** saves, **Esc** cancels. Deletions and the quit-with-unsaved-changes question use one “Please confirm” dialog.
- **The mouse** (new, on by default): click a page, a view, a row (again to open it), a column heading (sorts), a key hint at the bottom, a form field, a list entry or a button, and scroll with the wheel. The **Mouse** setting turns it off, and the terminal is told to stop reporting the mouse when the program ends, even if it is killed (section 3.5).

The screen

- **A new layout.** A 24-column sidebar menu in windows of 100 by 24 or more, otherwise the top bar, and nothing smaller than 50 by 12. The date and time line with the alert badge, a rule under the header, a margin at the right, the > marker and bar on the highlighted row, a highlighted current column, and < and > where columns are out of sight.
- **Above and below every list:** “Sorted by Name ▲ o changes it” and “12 items” (“3 of 12 items” while a search is on); under the table a rule and the page’s figures; while searching, a line that names the search and says Esc clears it. The key hints at the bottom list the keys of the page, most useful first, leave out what does not fit, and every one of them can be clicked.

- **Thirteen colour looks:** Full color, Blue/teal, Winter, Harbor, Dusk, Shepherd, Spring, Autumn, Ember, Samhain, Mono white, Mono green and Mono amber, chosen in a menu that previews each look behind it, stepped with `{` and `}`, or picked for one run with the new `-c LOOK` option (section 17.1). The painted looks use exact 24-bit colour on a terminal that says it can show it, and the nearest of the 256 colours otherwise. Every signal that colour gives is also given by brightness or reverse video.
- **The splash screen** has the Electric Shepherd sheep above the name, company, version and build date, or a framed title, or nothing (the **Splash screen** setting). It now **waits for a key** and does not pass the key on. Before, it left by itself after about two seconds.
- **Settings.** The first six rows are about the program (Colors, Clock beside the date, Splash screen, Start on page, Sidebar menu, Mouse); the rules of the Plan follow, unchanged.

Files and the command line

- **A settings file** for the six program settings: `~/.config/budget-tui/settings` (or under `XDG_CONFIG_HOME`, or the file named by `BUDGET_CONFIG`), one name=value per line, private to you and written only when a setting changes. The colour look and the clock used to be kept in the data file; the first start without a settings file carries them over, and they are no longer written to the data file (section 17.2).
- New options: `-c LOOK` (a look for this run only) and `-p PAGE` (open on this page). `-c` wins over `NO_COLOR`, and neither is saved. `COLORTERM` is read, and `--dump` reads the settings file but never writes it (Chapter 20).
- The company name is spelled **Electric Shepherd**, with no “Software”, on the splash, the cover, in `--help` and in this manual.

Where it differs from Inventory Central

- The rows of a list start in the order they were added (“Sorted by order added”), where Inventory Central starts every list sorted by its main column. Press `o` to sort.
- `x` (export) and `i` (import) work on Accounts and in the history only; on other pages they say where to go.
- Mono green and Mono amber keep the 256-colour tints of earlier versions.
- `Alt-1` to `Alt-9` count all the columns of a list from the left, including any that the window is too narrow to show.

About the colours and the sheep

The colour values of the looks and the drawing of the sheep are a **reconstruction**. Inventory Central's manual shows its looks and its sheep only in pictures, so the values were measured from those pictures (flat areas to within about ten levels per channel, thin text and rules less exactly) and adjusted just far enough that text, headings, warnings, rules and dim text keep the contrast the Inventory Central manual promises. They are not copied from Inventory Central's program and may differ from its colours by a few levels. The colours that Budget Buddy has and the pictures never show (the field background, good and bad) keep Budget Buddy's own.

Compatibility

- The program is still called `budget`, and the data file keeps its name, location and format. A 0.2.2a file loads as it is, and a file saved by 0.2.3a opens in the older versions (which start in their own default look and clock).

- Known limits that remain are listed in Chapter [23](#).
- This manual was rewritten for the new screens and keys, and every screenshot was captured again from the 0.2.3a program.

E.2 Version 0.2.2a (alpha)

Budget Buddy v0.2.2a, built 2026-10-20, is a release about safety, correctness and standards. Every screen and key is as in 0.2a, apart from the messages and prompts described here, and the data file format is unchanged. The program was audited against the current POSIX standard (POSIX.1-2024); this release fixes what the audit found.

Your data

- **A far-future date can no longer ruin the data file.** Typing +99999999 as a date used to be saved as the year 275817, and the next start refused the file. Dates must now lie between 1900 and 2200 wherever they come from, and a save is refused unless the program can read back what it is about to write.
- **A comma in an amount no longer moves the decimal point.** 12,50 used to become \$1,250.00. It is now an error that tells you to type 12.50; a comma is only a thousands separator, in groups of three (1,234).
- **Your files are private.** The data file, its backup, the lock file and `crash.log` can be read only by you (permissions 600; folders created by the program 700). Files from older versions keep their permissions until you change them (section [21.1](#)).
- **Only one copy per data file.** A second Budget Buddy on the same file refuses to start and says which process has it. A lock left by a copy that died is taken over automatically.
- **The data file is checked when it is loaded.** Out-of-range settings, negative amounts, duplicate accounts and records that point at something that does not exist are refused with a line number, and nothing is overwritten.
- **A failed save cannot be missed.** A lasting red NOT SAVED: line shows until a save works; quitting asks first; and if the final save fails the program ends with exit status 1 and says why.
- **Saves are more careful.** The backup is written as carefully as the data and a backup that fails now stops the save; the folder is synced after the rename; saving through a symbolic link updates the file it points to; and a session in which nothing changed writes nothing.

The terminal

- **A killed program leaves the terminal usable.** `kill`, `kill -9` and similar no longer leave the terminal in raw mode on the alternate screen: a small helper started with the program puts the screen back and removes the lock.
- A terminal that goes away without hanging up now saves and ends the program with exit status 1 and a message.
- `COLUMNS` and `LINES` override the terminal's size when they are positive numbers, as the standard specifies.

Command line

- Options follow the Unix conventions: grouped short options (-dn), values attached or separate, --, and short forms -f, -D, -d, -s, -k and -n beside the long ones. The page for --dump is now written after it (budget --dump income, as before).
- Exit status 2** means a mistake in how the program was called; 1 means a problem while running (Chapter 20).
- size must be 1 to 1000 each way; before, an enormous size could make the program run out of memory. --help, --version and --dump report a full disk or a closed pipe instead of failing with a panic. Options that do nothing without --dump are now reported.

Bank import

- Dates written with dots (17.10.2026) are read day first; a file whose slash dates cannot be told apart follows your language and region.
- Decimal commas, 1.234,56 and 1.234.567 are read correctly. Windows-1252 files show accents and euro signs correctly. A file:// link is decoded. Folders, devices and pipes are refused, and a file is never read past 25 MB.
- The list of files in your Downloads folder looks at up to 100,000 entries instead of the first 2,000.

Other fixes

- Negative percentages print as -0.5%; a savings goal that would take billions of months no longer wraps around; an invalid BUDGET_TODAY is reported instead of freezing the date; NO_COLOR= (empty) no longer turns colour off; a relative XDG_DATA_HOME is ignored; a data file that cannot be examined for lack of permission is reported rather than treated as missing.
- The company name is now spelled **Electric Shepherd** everywhere.

Compatibility

- The program is still called budget, and the data file keeps its name, location and format. Files from every earlier version load unchanged, and files saved by this one open in them.
- Known limits that remain are listed in Chapter 23.

E.3 Version 0.2.1a (alpha)

A version-label release: no change to any screen, key or file format.

E.4 Version 0.2a (alpha)

Budget Buddy v0.2a, built 2026-10-19, is a release about how the program looks and is laid out. Plans, calculations, keys and data are as in 0.1.1a.

New in this release

- A new screen layout** (Chapter 2). At 100 columns by 24 rows or more there is a **sidebar** listing all ten tabs with their number keys and full names; on the Plan it also keeps cash, debt, net worth and the strategy in view. Above the page are the date, the time and an **alert badge** (“!! 2 overdue”, “! 12 alerts”, “i 2 notes” or “✓ all clear”).

- **The top bar stays** for smaller windows, with a rail that marks the current tab and shows each tab's number key. A standard **80 by 24** terminal keeps its full-width page.
- **Sub menus.** The pages inside a tab are shown as a menu: Bills, Daily spending and Allotments under Expenses; Assigned and Suggested under Allotments; and the modes of Projections listed in the sidebar.
- **Projections replaces Set-Aside** as tab 9 (Chapter 16), to make room for a calendar and other ways of looking ahead. Its two modes, by paycheck and by bill, work as before.
- **Colour modes** (section 17.1): Full color, Mono white, Mono green and Mono amber, chosen in Settings and saved with your data. `NO_COLOR` still turns colour off, and now overrides the setting.
- **A clock** beside the date (section 17.1): 12-hour, 24-hour or hidden. It updates once a minute. `BUDGET_TIME=HH:MM` pins a time for screenshots and tests when `BUDGET_TODAY` is set.
- Dialogs and menus are centred in the area beside the sidebar; the full-screen history covers the whole window.
- Page titles and mode tags no longer collide on Projections, and the summary lines under the tables (Income, Expenses, Debts and the rest) now stay inside the page instead of running into the margin. The Debts summary drops its interest figure first when space is short.
- `--dump` accepts projections; the old names `setaside` and `aside` still work.
- This manual was rebuilt for the new layout: every screenshot was captured again from the 0.2a program.

Compatibility

- The program is still called `budget`, and the data file keeps its name, location and format. Files from 0.1a and 0.1.1a load unchanged.
- Version 0.2a writes two new setting lines, `color_mode` and `clock`. Older builds ignore unknown settings, so the file stays readable in both directions.
- Data files that use allotments still cannot be opened by builds older than the allotments release.

E.5 Version 0.1.1a (alpha)

A portability release with no change to what you see: it made the program usable with the older compilers that some Linux distributions still ship. That aim was dropped in 0.2.2a, which moved to a current toolchain.

E.6 Version 0.1a (alpha)

Budget Buddy v0.1a, built 2026-10-18, was the first named alpha release from **Electric Shepherd Software**.

What was in it

- Plan, Income, Expenses, Debts, Payoff, Savings, Accounts, Cash Flow, Set-Aside and Settings tabs.
- Expenses with three pages: Bills, Daily spending ledger, and Allotments (Assigned and Suggested).
- Account histories with sorting, filtering and live totals; bank CSV import with a preview; CSV export.
- Avalanche and Snowball debt strategies, a payoff schedule, and a what-if.
- Crash-safe saving with an automatic backup; recovery from internal errors with a replayable crash log.
- A built-in demo budget (`--demo`) and a plain-text screen dump (`--dump`).

New in that release

- **Budget Buddy** name and **Electric Shepherd Software** branding throughout.
- A **splash screen** at start-up showing the name, company, version and build date. Skip it with any key, `--no-splash` or `BUDGET_NO_SPLASH=1`.
- The product name and version on every screen, and in the output of `--version` and in the crash log.
- Version numbering `v0.1a`, kept in one place in the source, with the build date stamped in at compile time.
- The first edition of this manual.

E.7 Known issues

See section [23.2](#).

Appendix F. Keys That Moved in 0.2.3a

To work like the other Electric Shepherd programs, some keys of version 0.2.2a and earlier now mean something else. Nothing was taken away: every action still has a key, listed here. If a key does not do what you expect, look for it in the first column. The full list of keys is in Appendix A, and **?** shows it inside the program.

F.1 Keys that changed

Key before	Where	Now	Chapter
b	Everywhere	B reconciles (sets an account's balance to match the bank). b turns the sidebar on and off.	11
s	Accounts	B sets the balance. s sorts, like o .	11
s	Plan, Debts, Payoff	S switches between Avalanche and Snowball.	8
h	Everywhere	? or F1 opens the key reference. h moves the current column left (l right).	3
[]	Plan, Daily spending, Allotments	, . change the month. [] change the page, like Tab and Shift-Tab .	3
l , m	Expenses	v / v step through the views: Bills, Daily spending, By category, Allotments, Suggested.	7
v	Plan	v / v switch between the Detail and Steps views.	5
v , , , .	Payoff	v / v pick All debts or one debt (, . still do). m switches the schedule between monthly and yearly.	9
v	Projections	v / v switch between By paycheck and By bill.	16
r	Expenses, Debts	R edits what is set aside.	7
r	Projections (by bill)	R edits what is set aside.	16
w	Payoff	w opens the what-if. w now saves at once, everywhere.	9
c	Savings	+ (or =) adds money to the goal. c now clones the goal.	10
Enter	Accounts	Space or H opens the history. Enter (or e) edits the account.	11
o	Every list	Sorts by the current column , and again to reverse. o opens the Sort dialog that o used to open.	18
o	Every list	Opens the Sort dialog. It used to flip the direction; that is r now.	18
ctrl-c	In a form	Does nothing in a form; Esc cancels it. From a page, ctrl-c quits as before.	3
Any key	Start-up	The splash screen waits for a key, instead of leaving by itself after about two seconds. The key is used up.	17

F.2 What did not change

- **1** to **9** and **0** still go to the pages, and **Tab**, **Shift-Tab**, **Left** and **Right** still step through them.
- **a**, **e**, **d**, **Delete**, **p**, **/**, **+**, **-**, **t**, **i**, **x**, **u**, **y** and **D** keep their jobs. **/** still searches, and **Esc** clears the search.
- Everything you type in a form is as before: the date shorthands, the money rules, **ctrl-s** to save and **Esc** to cancel.
- The data file is the same, apart from the two program settings that moved out of it (section 21.3).

F.3 New in this release

Key	What it does
c	Clone the highlighted row of a list.
Space	View every field of the highlighted row.
h / l	Move the current column; the keys o , s and f act on it.
f	Filter on the current column only.
Alt-1 to Alt-9	Sort by the 1st to 9th column.
v / V	Step through the views of a page.
b	Sidebar on and off.
{ / }	Step through the thirteen colour looks.
F1 , F2	The key reference; save a form.
Mouse	Click and scroll (section 3.5).