bal manual

Zach Flynn

This is the manual for bal, a personal account manager.

Table of Contents

1 Introduction

bal is an interactive, command-line program for managing personal finances. Its goal is to be a simple but easily-extensible tool for helping you keep track of your spending and an aid in making future plans. **bal** was conceived from the need to track my own personal finances leading up to my wedding and finding that most free software personal finance packages were large, complicated GUI programs that were difficult to hack to make work the way I wanted them to. **bal** is a command-line program, but it is possible to write a GUI on top of the program. It is possible because **bal** can be extended arbitrarily using the Guile Scheme programming language. If you make a neat extension like this, let me know and I will reference it in this manual.

No knowledge of Scheme or programming in general is required to use **bal**. I have included almost all the functions I use personally in the **bal** distribution itself. In other words, I use **bal** without any programming in Scheme on a day-to-day basis. The only commands I have written that are not included in the main **bal** distribution are some commands to fetch the price of various stocks I own and update their value automatically, but I will show how to write such a command later in this document as an example.

The other extreme in the accounting software world were programs that required the user to edit text files to add transactions (the best example being **ledger**). I found this to be difficult to use as a user and wanted something more interactive. Of course, you may have your own preference. Like **ledger**, **bal** stores data in human-readable text files, but **bal**'s save files are particularly easy to analyze with other tools. The **bal** save file is simply a **tar** archive of comma-separated data files with data about your various transactions. You can open these files in **R**, for example, and analyze your past spending that way.

All commands that ship with **bal** are small Guile Scheme commands which call a few functions exported by **bal** from C. These functions as well as the built-in interactive functions are documented in the reference manual to help you construct your own commands. The **bal** prompt, in fact, is a Scheme interpreter. Any Scheme expression can be written there.

This document has two main sections. In the first, I describe the basic workflow for using **bal**. This section is more in-depth than the manpage and gives some examples of how to add accounts and make transactions. It is a tutorial that will help get you up-and-running using **bal**. In the second section, I describe all the Scheme commands exported by **bal** which is more useful if you are trying to write your own commands.

2 Basic usage

This chapter will be a tutorial demonstrating basic usage of **bal**.

2.1 Adding a cash asset account

2.2 Paying for expenses

An example of adding an expense and income account and transferring money between them.

2.3 Earning income

2.4 Paying back debt

2.5 Adding new debt

2.6 Non-cash assets

3 Programming bal

This chapter will be more of a reference manual for the various Scheme functions available in **bal**.