Beancount User's Manual

This is the top-level page for all documentation related to Beancount.

http://furius.ca/beancount/doc/index

Documentation for Users

Command-line Accounting in Context: A motivational document that explains what command-line accounting is, *why* I do it, with a simple example of *how* I do it. Read this as an introduction.

The Double-Entry Counting Method: A gentle introduction to the double-entry method, in terms compatible with the assumptions and simplifications that command-line accounting systems make.

Installing Beancount: Instructions for download and installation on your computer, and software release information.

Running Beancount and Generating Reports: How to run the Beancount executables and generate reports with it. This contains technical information for Beancount users.

Getting Started with Beancount: A text that explains the basics of how to create, initialize and organize your input file, and the process of declaring accounts and adding padding directives.

Beancount Language Syntax: A full description of the language syntax with examples. This is the main reference for the Beancount language.

Beancount Options Reference: A description and explanation of all the possible option values.

Precision & Tolerances: Transactions and Balance assertions tolerance small amounts of imprecision which are inferred from the context. This document explains how this works.

Beancount Query Language: A high-level overview of the bean-query command-line client tool that allows you to extract various tables of data from your Beancount ledger file.

Beancount Cheat Sheet: A single page “cheat sheet” that summarizes the Beancount syntax.

How Inventories Work: An explanation of inventories, their representation, and the detail of how lots are matched to positions held in an inventory. This is preliminary reading for the trading doc.

Exporting Your Portfolio: How to export your portfolio to external portfolio tracking websites.

Tutorial & Example: A realistic example ledger file that contains a few years of a hypothetical Beancount user's financial life. This can be used to kick the tires on Beancount and see what reports or its web interface look like. This can give a quick idea of what you can get out of using Beancount.

Beancount Mailing-list: Questions and discussions specific to Beancount occur there. Of related interest is also the Ledger-CLI Forum which contains lots of more general discussions and acts as a meta-list for command-line accounting topics.

Beancount History and Credits: A description of the development history of Beancount and a shout out to its contributors.

A Comparison of Beancount and Ledger & HLedger: A qualitative feature comparison between CLI accounting systems.

Fetching Prices in Beancount: How to fetch and maintain a price database for your assets using Beancount's tools.

Importing External Data: A description of the process of importing data from external files that can be downloaded from financial institutions and the tools and libraries that Beancount provides to automate some of this.

Cookbook & Examples

These documents are examples of using the double-entry method to carry out specific tasks. Use these documents to develop an intuition for how to structure your accounts.

Command-line Accounting Cookbook: Various examples of how to book many different kinds of financial transactions. This is undoubtedly the best way to build an intuition for how to best use the double-entry method.

Trading with Beancount: An explanation of trading P/L and worked examples of how to deal with various investing and trading scenarios with Beancount. This is a complement to the cookbook.

Stock Vesting in Beancount: An example that shows how to deal with restricted stock units and vesting events typical of those offered by technology companies & startups.

Sharing Expenses with Beancount: A realistic and detailed example of using the double-entry method for sharing expenses for a trip or project with other people.

How We Share Expenses: A more involved description of a continuous system for (a) sharing expenses between partners when both are contributing and (b) sharing expenses to a specific project (our son) who has a ledger of his own. This describes our real system.

Documentation for Developers

Beancount Scripting & Plugins: A guide to writing scripts that load your ledger contents in memory and process the directives, and how to write plugins that transform them.

Beancount Design Doc: Information about the program's architecture and design choices, code conventions, invariants and methodology. Read this if you want to get a deeper understanding.

LedgerHub Design Doc: The design and architecture of the importing tools and library implemented in `beancount.ingest`. This used to be a separate project called LedgerHub (now defunct), whose useful parts have been eventually folded into Beancount. This is the somewhat dated original design doc.

Source Code: The official repository of the Beancount source code lives at BitBucket.

External Contributions: A list of plugins, importers and other codes that build on Beancount's libraries that other people have made and shared.

Enhancement Proposals & Discussions

I occasionally write proposals for enhancements in order to organize and summarize my thoughts before moving forward, and solicit feedback from other users. This is good material to find out what features I'm planning to add, how things work in detail, or compare the differences with other similar software such as Ledger or HLedger.

A Proposal for an Improvement on Inventory Booking: A proposal in which I outline the current state of affairs in Ledger and Beancount regarding inventory booking, and a better method for doing it that will support average cost booking and calculating capital gains without commissions.

Settlement Dates in Beancount: A discussion of how settlement or auxiliary dates could be used in Beancount, and how they are used in Ledger.

Balance Assertions in Beancount: A summary of the different semantics for making balance assertions in Beancount and Ledger and a proposal to extend Beancount's syntax to support more complex assertions.

Fund Accounting with Beancount: A discussion of fund accounting and how best to go about using Beancount to track multiple funds in a single ledger.

Rounding & Precision in Beancount: A discussion of important rounding and precision issues for balance checks and a proposal for a better method to infer required precision. (Implemented)

External Links

Documents, links, blog entries, writings, etc. about Beancount written by other authors.]

Beancount Source Code Documentation (Dominik Aumayr): Sphinx-generated source code documentation of the Beancount codebase. The code to produce this is located here.

Beancount ou la comptabilité pour les hackers (Cyril Deguet): An overview blog entry (in french).

About this Documentation

You may have noticed that I'm using Google Docs. I realize that this is unusual for an open source project. If you take offense to this, so you know, I do like text formats too: in graduate school I used LaTeX extensively, and for the last decade I was in love with the reStructuredText format, I even wrote Emacs' support for it. But something happened around 2013: Google Docs became good enough to write solid technical documentation and I've begun enjoying its revision and commenting facilities extensively: I am hooked, I love it. For users to be able to suggest a correction or place a comment in context is an incredibly useful feature that helps improve the quality of my writing a lot more than patches or comments on a mailing-list. It also gives users a chance to point out passages that need improvement. I have already received orders of magnitude more feedback on documentation than on any of my other projects; it works.

It also looks really *good*, and this is helping motivate me to write more and better. I think maybe I'm falling in love again with WYSIWYG... Don't take me wrong: LaTeX and no-markup formats are great, but the world is changing and it is more important to me to have community collaboration and a living document than a crufty TeX file slowly aging in my repository. My aim is to produce quality text that is easy to read, that prints nicely, and most especially these days, that can render well on a mobile device or a tablet so you can read it anywhere. I'm also able to edit it while I'm on the go, which is fun for jotting down ideas or making corrections. Finally, Google Docs has an API that provides access to the doc, so should it be needed, I could eventually download the meta-data and convert it all to another format. I don't like everything about it, but what it offers, I really do like.

If you want to leave comments, I'd appreciate if you can **log into your Google account** while you read, so that your name appears next to your comment. Thank you!

—Martin Blais