Exporting Your Portfolio

[Martin Blais](mailto:blais@furius.ca), December 2015 (v2)

<http://furius.ca/beancount/doc/export>

# Overview

This document explains how to export your portfolio of holdings from Beancount to a Google Finance portfolio (and eventually to other portfolio tracking websites).

*Note: This is the second version of this document, rewritten in Dec 2015, after greatly simplifying the process of exporting portfolios and completely separating the specification of stock tickers for price downloads. This new, simplified version only uses a single metadata field name: “export”. The previous document can be found* [*here*](https://docs.google.com/document/d/1eZIDRmQZxR6cmDyOJf7U3BnCm4PDMah2twxYFfKPJtM/)*.*

# Portfolio Tracking Tools

There are multiple websites on the internet that allow someone to create a portfolio of investments (or upload a list of transactions to creates such a portfolio) and that reports on the changes in the portfolio due to price movements, shows you unrealized capital gains, etc. One such website is the [Google Finance](http://finance.google.com) portal. Another example is the [Yahoo Finance](http://finance.yahoo.com) one. These are convenient because they allow you to monitor the impact of price changes on your entire portfolio of assets, across all accounts, during the day or otherwise.

However, each of these sites expects their users to use their interfaces and workflows to painfully enter each of the positions one-by-one. A great advantage of using Beancount is that you should never have to enter this type of information manually; instead, you should be able to extract it and upload it to one of these sites. You can be independent of the particular portfolio tracking service you use and should be able to switch between them without losing any data; Beancount can serve as your pristine source for your list of holdings as your needs evolve.

Google Finance supports an “import” feature to create portfolio data which supports the Microsoft OFX financial interchange data format. In this document, we show how we built a Beancount report that exports the portfolio of holdings to OFX for creating a Google Finance portfolio.

# Exporting to Google Finance

## Exporting your Holdings to OFX

First, create an OFX file corresponding to your Beancount holdings. You can use this command to do this:

**bean-report** *file.beancount* **export\_portfolio** > *portfolio.ofx*

See the report’s own help for options:

**bean-report** *file.beancount* **export\_portfolio** --help

## Importing the OFX File in Google Finance

Then we have to import that OFX file in a web-based portfolio.

## Visit <http://finance.google.com> and click on “Portfolios” on the left (or simply visit <https://www.google.com/finance/portfolio>, this works as of Jan 2015)

1. If you have an existing, previously imported portfolio, click on “Delete Portfolio” to get rid of it.
2. Click on “Import Transactions”, then “Choose File” and select the *portfolio.ofx* file you exported to, then click on “Preview Import”.
3. You should see a list of imported lots, with familiar stock symbols and names, and Type “Buy” with realistic Shares and Price columns. If not, see the note below. Otherwise, scroll to the bottom of the page and click “Import”.
4. Your portfolio should now appear. You are done.

You should never bother updating this portfolio directly using the website… instead, update your Beancount ledger file, re-export to a new OFX file, **delete** the previous portfolio and re-import a brand new one over it. Your pristine source is always your Beancount file, ideally you should never have to be worried about corrupting or deleting the portfolio data in any external website.

# Controlling Exported Commodities

## Declaring Your Commodities

Generally, we recommend that you explicitly declare each of the commodities used in your input file. It is a neat place to attach information about those commodities, metadata that you should be able to use later on from bean-query or in scripts that you make. For example, you could declare a human-readable description of the commodity, and some other attributes, like this:

2001-09-06 commodity XIN  
 name: "iShares MSCI EAFE Index ETF (CAD-Hedged)"  
 class: "Stock"  
 type: "ETF"  
 ...

Beancount will work with or without these declarations (it automatically generates Commodity directives if you haven’t provided them). If you like to be strict and have a bit of discipline, you can *require* that each commodity be declared by using a plugin that will issue an error when an undeclared commodity appears:

plugin "**beancount.plugins.check\_commodity**"

You can use any date for that Commodity directive. I recommend using the date of the commodity’s inception, or perhaps when it was first introduced by the issuing country, if it is a currency. You can find a suitable date on Wikipedia or on the issuer’s websites. Google Finance may have the date itself.

## What Happens by Default

By default, all holdings are exported as positions with a ticker symbol named the same as the Beancount commodity that you used to define them. If you have a holding of “AAPL” units, it will create an export entry for “AAPL”. The export code attempts to export all holdings by default.

However, in any but the simplest unambiguous cases, this is probably not good enough to produce a working Google Finance portfolio. The name for each commodity that you use in your Beancount input file may or may not correspond to a financial instrument in the Google Finance database; due to the very large number of symbols supported in its database, just specifying the ticker symbol is often ambiguous. Google Finance attempts to resolve an ambiguous symbol string to the most likely instrument in its database. It is possible that it resolves it to a different financial instrument from the one you intended. So even if you use the same basic symbol that is used by the exchange, you often still need to disambiguate the symbol by specifying which exchange or symbology it lives in. Google provides a [list of these symbol spaces](http://www.google.com/googlefinance/disclaimer/).

Here is a real-life example. The symbol for the “[CAD-Hedged MSCI EAFE Index](http://www.blackrock.com/ca/individual/en/products/239624/ishares-msci-eafe-index-etf-cadhedged-fund)” ETF product issued by iShares/Blackrock is “XIN” on the Toronto Stock Exchange (TSE). If you just [looked up “XIN” on Google Finance](https://www.google.com/finance?q=xin), it would choose to resolve it by default to the more likely “NYSE:XIN” symbol ([Xinyuan Real Estate Co. on the New York Stock Exchange](https://www.google.com/finance?q=NYSE%3AXIN)). So you need to disambiguate it by specifying that the desired ETF ticker for this instrument is “TSE:XIN”.

## Explicitly Specifying Exported Symbols

You can specify which exchange-specific symbol is used to export a commodity by attaching an “export” metadata field to each of your Commodity directives, like this:

2001-09-06 commodity XIN  
 ...  
 **export: "TSE:XIN"**

## The “export” field is used to map your commodity name to the corresponding instrument in the Google Finance system. If a holding in that commodity needs to be exported, this code is used instead of the Beancount currency name.

## The symbology used by Google Finance appears to follow the following syntax:

## *Exchange:Symbol*

where *Exchange* is a code either for the exchange where the stock trades, or for another source of financial data, e.g. “MUTF” for “mutual funds in the US”, [and more](http://www.google.com/googlefinance/disclaimer/). *Symbol* is a name that is unique within that exchange. I recommend searching for each of your financial instruments in Google Finance, confirming that the instrument corresponds to your instrument (by inspecting the full name, description and price), and inserting the corresponding code like this.

## Exporting to a Cash Equivalent

To account for positions that aren’t supported in Google Finance, the export report can convert a holding to its cash-equivalent value. This is also useful for cash positions (e.g., cash sitting idle in a savings or checking account).

## For example, I hold units of an insurance policy investment vehicle (this is common in Canada, for example, with London Life). This is a financial instrument, but each particular policy issuance has its own associated value—there is no public source of data for each of those products, it’s rather opaque, I can obtain its value with my annual statement, but definitely not in Google Finance. But I’d still like for the asset’s value to be reflected in my portfolio.

## The way you tell the export code to make this conversion is to specify a special value of “CASH” for the “export” field, like this:

1878-01-01 commodity LDNLIFE  
  **export: "CASH"**

This would convert holdings in LDNLIFE commodities to their corresponding quoted value before exporting, using the price nearest to the date of exporting. Note that attempting to convert to cash a commodity that does not have a corresponding cost or price available for us to determine its value will generate an error. A price must be present to make the conversion.

Simple currencies should also be marked as cash in order to be exported:

1999-01-01 commodity EUR  
 name: "European Union Euro currency"  
 **export: "CASH"**

Finally, all converted holdings are agglomerated into a single cash position. There is no point in exporting these cash entries to separate OFX entries because the Google Finance code will agglomerate them to a single one anyhow.

## Declaring Money Instruments

There is a small hiccup in this cash conversion story: the Google Finance importer does not appear to correctly grok an OFX position in “cash” amounts in the importer; I think this is probably just a bug in Google Finance’s import code (or perhaps I haven’t found the correct OFX field values to make this work).

Instead, in order to insert a cash position the exporter uses a cash-equivalent commodity which always prices at 1 unit of the currency, e.g. $1.00 for US dollars. For example, for US dollars I I use [VMMXX](https://www.google.com/finance?q=MUTF:VMMXX) which is a Vanguard Prime Money Market Fund, and for Canadian dollars I use [IGI806](https://www.google.com/finance?q=MUTF_CA:IGI806). A good type of commodity for this is some sort of Money Market fund. It doesn’t matter so much which one you use, as long as it prices very close to 1. Find one.

If you want to include cash commodities, you need to find such a commodity for each of the cash currencies you have on your books and tell Beancount about them. Typically that will be only one or two currencies.

You declare them by append the special value “MONEY” for the “export” field, specifying which currency this commodity represents, like this:

1900-01-01 commodity VMMXX  
 **export: "MUTF:VMMXX (MONEY:USD)"**  
  
1900-01-01 commodity IGI806  
 **export: "MUTF\_CA:IGI806 (MONEY:CAD)"**

## Ignoring Commodities

Finally, some commodities held in a ledger should be ignored. This is the case for the imaginary commodities used in mirror accounting, for example, to track unvested shares of an employment stock plan, or commodities used to track amounts contributed to a retirement account, like this:

1996-01-01 commodity RSPCAD  
 name: "Canada Registered Savings Plan Contributions"

You tell the export code to ignore a commodity specifying the special value “IGNORE” for the “export” field, like this:

1996-01-01 commodity RSPCAD  
 name: "Canada Registered Savings Plan Contributions"  
 **export: "IGNORE"**

All holdings in units of RSPCAD will thus not be exported.

The question of whether some commodities should be exported or not sometimes presents interesting choices. Here is an example: I track my accumulated vacation hours in an asset account. The units are “VACHR”. I associate with this commodity a price that is roughly equivalent to my net hourly salary. This gives me a rough idea how much vacation time money is accumulated on the books, e.g. if I quit my job, how much I’d get paid. Do I want to them included in my total net worth? Should the value from those hours be reflected in the value of my exported portfolio? I think that largely depends on whether I plan to use up those vacations before I leave this job or not, whether I want to have this accumulated value show up on my balance sheet.

# Comparing with Net Worth

The end result is that the sum total of all your exported positions plus the cash position should approximate the value of all your assets, and the total value calculated by the Google Finance website should be very close to the one reported by this report:

bean-report *file.beancount* **networth**

As a point of comparison, the value of my own portfolio is usually close to within a few hundred US dollars.

# Details of the OFX Export

## Import Failures

Exporting a portfolio with symbols that Google Finance does not recognize **fatally** trips up Google’s import feature. Google Finance then proceeds to fail to recognize your **entire** file. I recommend that you use explicit exchange:symbol names on all commodities that get exported in order to avoid this problem, as is described further in this document.

Google Finance can also be a little bit finicky about the format of the particular OFX file you give it to import. The export\_portfolio command attempts to avoid OFX features that would break it but it’s fragile, and it’s possible that the particulars of your portfolio’s contents triggers output that fails to import. If this is the case, at step (4) above, instead of a list of stock symbols you would see a long list of positions that look like XML tags (this is how failure manifests itself). If that is the case, send email to the mailing-list (best if you can isolate the positions that trigger breakage and have the capability to diff files and do some troubleshooting).

## Mutual Funds vs. Stocks

The OFX format distinguishes between stocks and mutual funds. In practice, the Google Finance importer does not appear to distinguish between these two (at least it appears to behave the same way), so this is likely an irrelevant implementation detail. Nevertheless, the export code is able to honor the OFX convention of distinguishing between “BUYMF” vs. “BUYSTOCK” XML elements.

To this effect, the export code attempts to classify which commodities represent mutual funds by inspecting whether the ticker associated with the commodity begins with the letters “MUTF” and is followed by a colon. For example, “MUTF:RGAGX” and “MUTF\_CA:RBF1005" will both be detected as mutual funds, for example.

## Debugging the Export

In order to debug how each of your holdings gets exported, use the --debug flag, which will print a detailed account of how each holding is handled by the export script to **stderr**:

**bean-report** *file.beancount* **export\_portfolio** --debug 2>&1 >/dev/null | more

The script should print the list of exported positions and their corresponding holdings, then the list of converted positions and their corresponding holdings (usually many cash positions are aggregated together) and finally, the list of ignored holdings. This should be enough to explain the entire contents of the exported portfolio.

## Purchase Dates

Beancount does not currently have all the lot purchase dates, so the purchase dates are exported as if purchased the day before the export.

Eventually, when the purchase date is available in Beancount (pending the [inventory booking changes](http://furius.ca/beancount/doc/booking-proposal)) the actual lot purchase date will probably be used in the export format. However, it’s not yet clear that using the correct date is the right thing to do, because Google Finance might insist on inserting cash for dividends since the reported purchase date… but Beancount already takes care of inserting a special lot for cash that should already include this. We shall see when we get there.

## Disable Dividends

Under the “Edit Portfolio” option there is a checkbox that appears to disable the calculation of dividends offered. It would be nice to find a way to automatically disable this checkbox upon import.

## Automate Upload

It would be nice to automate the replacement of the portfolio with a Python script. Unfortunately, the Google Finance API has been deprecated. Maybe someone can write a screen-scraping routine to do this.

# Summary

Each holding’s export can be controlled by how its commodity is treated, in one of the following ways:

1. **Exported** to a portfolio position. This is the default, but you should specify the ticker symbol using the “ticker” or “export” metadata fields, in “*ExchangeCode:Symbol*” format.
2. **Converted** to cash and exported to a money market cash-equivalent position, by setting the value of the “export” metadata field to the special value “CASH”.
3. **Ignored** by specifying the “export” metadata field to the special value “IGNORE”.
4. Provided as **Money Instrument**, to be used for cash-equivalent value of each holding intended to be converted to cash and included in the portfolio. These are identified by a special value “(MONEY:<currency>)” in the “export” metadata field.