Trading with Beancount

Martin Blais, July 2014

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

Introduction

This is a companion document for the Command-Line Accounting Cookbook that deals exclusively with the subject of trading and investments in Beancount. You probably should have read an introduction to the double-entry method before reading this document.

The subject of stock trading needs to be preceded by a discussion of "profit and loss," or P/L, for short (pronounce: "P and L"), also called capital gains or losses. The notion of P/L against multiple trades can be difficult for a novice to understand, and I've even seen professional traders lack sophistication in their understanding of P/L over varying time periods. It is worth spending a bit of time to explain this, and necessary to understand how to book your trades in a double-entry system.

This discussion will be weaved with detailed examples of how to book these trades in Beancount, wherever possible. There is a related, active proposal for improving the booking methods in Beancount that you might also be interested in. Discussions of basis for tax-deferred accounts will not be treated here, but in the more general cookbook.

What is Profit and Loss?

Let's imagine you have an account at the E*Trade discount broker and you buy some shares of a company, say IBM. If you buy 10 shares of IBM when its price is 160$/share, it will cost you 1600$.

That value is what we will call the "book value", or equivalently, "the cost." This is how much money you had to spend in order to acquire the shares, also called "the position." This is how you would enter this transaction in Beancount:

```
2014-02-16 * "Buying some IBM"
  Assets:US:ETrade:IBM                 10 IBM {160.00 USD}
  Assets:US:ETrade:Cash        -1600.00 USD
```

In practice you will probably pay some commission to E*Trade for this service, so let's put that in for completeness:

```
2014-02-16 * "Buying some IBM"
  Assets:US:ETrade:IBM                 10 IBM {160.00 USD}
  Assets:US:ETrade:Cash        -1609.95 USD
  Expenses:Financial:Commissions    9.95 USD
```

This is how you tell Beancount to deposit some units "at cost", in this case, units of "IBM at 160 USD/share" cost. This transaction balances because the sum of its legs is zero: 10 x 160 + -1609.95 + 9.95 = 0. Also note that we're choosing to use a subaccount dedicated to the shares of IBM; this is not strictly necessary but it is convenient for reporting in, for example, a balance sheet, because it will naturally aggregate all of your shares of each of your positions on their own line. Having a "cash" subaccount also emphasizes that uninvested funds you have there are not providing any return.

The next day, the market opens and IBM shares are going for 170$/share. In this context, we will call this "the price."[1] The "market value" of your position, your shares, is the number of them x the market price, that is, 10 shares x 170$/share = 1700$.

The difference between these two amounts is what we will call the P/L:

market value - book value = P/L

10 x 170$ - 10 x 160$ = 1700$ - 1600$ = 100$ (profit)

We will call a positive amount "a profit" and if the amount is negative, "a loss."

---

[1] This is a misleading notion, however. In reality, *there is no price*, there exist only markets where you get a "hint" of how much someone else might be willing to exchange your shares for (for different amounts to buy or to sell them, and for some limited number of them, we call this "a market"), but until you've actually completed selling your shares, you don't really know precisely how much you will be able to execute that trade at, only an estimate. If you're not intimately familiar with trading, this should give you pause, and hopefully a big "ah-ha! "moment about how the world works - there really does not exist a price for anything in the world - but in the context of this discussion, let's make abstraction of this and assume that you can buy or sell as many shares as you have instantly on the markets at the middle price, such as Google Finance or Yahoo Finance or your broker would report it. The process of deciding that we would be able to sell the shares at 170$ each is called "marking", that is, under reasonable assumptions, we believe that we would be able to actually sell those shares at that price (the term of art "marking to market" refers to the fact that we use the market price as the best indicator to our knowledge of our capability to realize the trade). For an individual buying and selling small amounts of shares that don't move the market and with an honest broker, this is mostly true in practice.

Realized and Unrealized P/L

The profit from the previous section is called an "unrealized profit." That is because the shares have not actually been sold yet - this is a hypothetical profit: *if* I can sell those shares at the market value, this is how much I *would* pocket. The 100$ I mentioned in the previous section is actually an "unrealized P/L."

So let's say you like this unrealized profit and you feel that it's temporary luck that IBM went up. You decide to sell 3 of these 10 shares to the market at 170$/share. The profit on these share will now be "realized":

> market value - book value = P/L

> 3 x 170$ - 3 x 160$ = 3 x (170 - 160) = 30$ (profit)

This 30$ is a "realized P/L." The remaining portion of your position is still showing an unrealized profit, that is, the price could fluctuate some more until you sell it:

> market value - book value = P/L

> 7 x 170$ - 7 x 160$ = 70$

This is how you would book this partial sale of your position in Beancount (again including a commission):

```
2014-02-17 * "Selling some IBM"
  Assets:US:ETrade:IBM                 -3 IBM {160.00 USD}
  Assets:US:ETrade:Cash          500.05 USD
  Expenses:Financial:Commissions   9.95 USD
```

Do you notice something funny going on here? -3 x 160 = -480, -480 + 500.05 + 9.95 = 30... This transaction does not balance to zero! The problem is that we received 510$ in cash in exchange for the 3 shares we sold. This is because the actual price we sold them at was 170$: 3 x 170 = 510$. This is where we need to account for the profit, by adding another leg which will absorb this profit, and conveniently enough, automatically calculate and track our profits for us:

```
2014-02-17 * "Selling some IBM"
  Assets:US:ETrade:IBM                 -3 IBM {160.00 USD}
  Assets:US:ETrade:Cash          500.05 USD
  Expenses:Financial:Commissions   9.95 USD
  Income:US:ETrade:PnL
```

The last leg will be automatically filled in by Beancount to `-30 USD`, as we're allowed one posting without an amount (and remember that in the double-entry system without credits and debits, a profit is a negative number for "Income" accounts). This is the number the government is interested in for your taxes.

In summary, you now have:

A position of 7 "shares at book value of 160$" = 1120$ (its book value)

A realized P/L of 30$

An unrealized P/L of 70$

Now at this point, some of you will jump up and down and say: "But wait, waiiit! I sold at 170$/share, not 160$/share, why do you put 160$ here?" The answer is that you did not have shares held at 170$ to sell. In order to explain this, I need to make a little detour to explain how we keep track of things in accounts...

So how do we keep track of these shares?

It's actually easy: when Beancount stores things in accounts, we use something called "an inventory." Imagine that an "inventory" is a bag with the name of that account on it. Each account has one such bag to hold the things in the account at a particular point in time, the "balance" of this account at that time. Imagine that the things it contains have a little label attached to each of them, with their cost, that is, the price that was paid to acquire them. Whenever you put a thing in the bag, you attach a new label to the thing. For things to work right, all things need to be labeled[2]. In our example, the bag contained 10 items of "shares of IBM bought at 160$/share". The syntax we used to put the IBM in the account can seem a little misleading; we wrote:

```
Assets:US:ETrade:IBM                10 IBM {160.00 USD}
```

but really, this is understood by Beancount closer to the following syntax:

```
Assets:US:ETrade:IBM                10 {IBM 160.00 USD}
```

But … it would be annoying to write this, so we use a syntax more intuitive to humans.

So the thing is, you can't subtract units of `{IBM at 170.00 USD}`… because there just aren't any in that bag. What you have in the bag are units of `{IBM at 160.00 USD}`. You can only take out these ones.

Now that being said, do you see how it's the amount that was exchanged to us for the shares that really helps us track the P/L? Nowhere did we actually need to indicate the price at which we sold the shares. It's the fact that we received a certain amount of cash that is different than the cost of the position we're selling that triggers the imbalance, which we book to a capital gain.

Hmmm… Beancount maintains a price database, wouldn't it be nice to at least record and attach that price to the transaction for documentation purposes? Indeed. Beancount allows you to also attach a price to that posting, but for the purpose of balancing the transaction, it ignores it completely. It is mainly there for documentation, and you can use it if you write scripts. And if you use the `beancount.plugins.implicit_prices` plugin, it will be used to automatically synthesize a `price` entry that will enrich our historical price database, which may be used in reporting the market value of the account contents (more details on this follow).

So the complete and final transaction for selling those shares should be:

---

[2] As an aside, putting regular currencies in an account is just the degenerate case of a thing with an empty label. This is an implementation detail that works great in practice.

```
2014-02-17 * "Selling some IBM"
  Assets:US:ETrade:IBM            -3 IBM {160.00 USD} @ 170.00 USD
  Assets:US:ETrade:Cash           500.05 USD
  Expenses:Financial:Commissions   9.95 USD
  Income:US:ETrade:PnL
```

Trade Lots

In practice, the reality of trading gets a tiny bit more complicated than this. You might decide to buy some IBM multiple times, and each time, it is likely that you would buy them at a different price. Let's see how this works with another example trade. Given your previous position of 7 shares held at 160$ cost, the following day you see that the price went up some more, you change your mind on IBM and decide to "go long" and buy 5 more shares. The price you get is 180$/share this time:

```
2014-02-18 * "I put my chips on big blue!"
  Assets:US:ETrade:IBM                5 IBM {180.00 USD}
  Assets:US:ETrade:Cash            -909.95 USD
  Expenses:Financial:Commissions      9.95 USD
```

Now, what do we have in the bag for `Assets:US:ETrade:IBM`? We have two kinds of things:

- 7 shares of "IBM held at 160 USD/share", from the first trade
- 5 shares of "IBM held at 180 USD/share", from this last trade

We will call these "lots," or "trade lots."

In fact, if you were to sell this entire position, say, a month later, the way to legally sell it in Beancount (that is, without issuing an error), is by specifying both legs. Say the price is 172$/share at that moment:

```
2014-03-18 * "Selling all my blue chips."
  Assets:US:ETrade:IBM            -7 IBM {160.00 USD} @ 172.00 USD
  Assets:US:ETrade:IBM            -5 IBM {180.00 USD}
  Assets:US:ETrade:Cash           2054.05 USD
  Expenses:Financial:Commissions   9.95 USD
  Income:US:ETrade:PnL
```

Now your final position of IBM would be 0 shares.

Alternatively, since you're selling the entire position, Beancount should be able to unambiguously match all the lots against an unspecified cost. This is equivalent:

```
2014-03-18 * "Selling all my blue chips."
  Assets:US:ETrade:IBM            -12 IBM {} @ 172.00 USD
  Assets:US:ETrade:Cash           2054.05 USD
  Expenses:Financial:Commissions   9.95 USD
  Income:US:ETrade:PnL
```

Note that this won't work if the total amount of shares doesn't match all the lots (this would be ambiguous… which subset of the lots should be chosen isn't obvious).

Booking Methods

But what if you decided to sell only some of those shares? Say you need some cash to buy a gift to your loved one and you want to sell 4 shares this time. Say the price is now 175$/share.

Now you have a choice to make. You can choose to sell the older shares and realize a larger profit:

```
2014-03-18 * "Selling my older blue chips."
  Assets:US:ETrade:IBM           -4 IBM {160.00 USD} @ 175.00 USD
  Assets:US:ETrade:Cash           690.05 USD
  Expenses:Financial:Commissions   9.95 USD
  Income:US:ETrade:PnL         ;; -60.00 USD (profit)
```

Or you may choose to sell the most recently acquired ones and realize a loss:

```
2014-03-18 * "Selling my most recent blue chips."
  Assets:US:ETrade:IBM           -4 IBM {180.00 USD} @ 175.00 USD
  Assets:US:ETrade:Cash           690.05 USD
  Expenses:Financial:Commissions   9.95 USD
  Income:US:ETrade:PnL         ;;  20.00 USD (loss)
```

Or you can choose to sell a mix of both: just use two legs.

Note that in practice this choice will depend on a number of factors:

- The tax law of the jurisdiction where you trade the shares may have a defined method for how to book the shares and you may not actually have a choice. For example, they may state that you must trade the oldest lot you bought, a method called "first-in-first out."
- If you have a choice, the various lots you're holding may have different taxation characteristics because you've held them for a different period of time. In the USA, for example, positions held for more than one year benefit from a lower taxation rate (the "long-term" capital gains rate).
- You may have other gains or losses that you want to offset in order to minimize your cash flow requirements on your tax liability. This is sometimes called "tax loss harvesting."

There are more… but I'm not going to elaborate on them here. My goal is to show you how to book these things with the double-entry method.

Dated lots

We've almost completed the whole picture of how this works. There is one more rather technical detail to add and it begins with a question: What if I bought multiple lots of share at the same price?

As we alluded to in the previous section, the duration for which you held a position may have an impact on your taxation, even if the P/L ends up being the same. How do we differentiate between these lots?

Well… I had simplified things a tiny bit earlier, just to make it simpler to understand. When we put positions in an inventory, on the label that we attach to the things we put in it, we also mark down the date that lot was acquired if you supply it. This is how you would book entering the position this way:

```
2014-05-20 * "First trade"
  Assets:US:ETrade:IBM            5 IBM {180.00 USD, 2014-05-20}
  Assets:US:ETrade:Cash          -909.95 USD
  Expenses:Financial:Commissions    9.95 USD

2014-05-21 * "Second trade"
  Assets:US:ETrade:IBM            3 IBM {180.00 USD, 2014-05-21}
  Assets:US:ETrade:Cash          -549.95 USD
  Expenses:Financial:Commissions    9.95 USD
```

Now when you sell, you can do the same thing to disambiguate which lot's position you want to reduce:

```
2014-08-04 * "Selling off first trade"
  Assets:US:ETrade:IBM           -5 IBM {180.00 USD, 2014-05-20}
  Assets:US:ETrade:Cash          815.05 USD
  Expenses:Financial:Commissions    9.95 USD
  Income:US:ETrade:PnL
```

Note that it's really unlikely that your broker will provide the information in the downloadable CSV or OFX files from their website… you probably won't be able to automate the lot detail of this transaction, you might have to pick up the PDF trade confirmations your broker provides to enter this manually, if it ever happens. But how often does it happen that you buy two lots at the same price? I trade relatively frequently - about every two weeks - and in 8 years worth of data I don't have a single occurrence of it. In practice, unless you do thousands of trades per day- and Beancount isn't really designed to handle that kind of activity, at least not in the most efficient way - it just won't happen very much.

(*Technical Detail*: that we're working on bettering the mechanism for lot selection so that you never have to insert the lot-date yourself, and so that you could disambiguate lot selection by supplying a name instead. See upcoming changes.)

Reporting Unrealized P/L

Okay, so our account balances are holding the cost of each unit, and that provides us with the book value of these positions. Nice. But what about viewing the market value?

The market value of the positions is simply the number of units of these instruments x the market price at the time we're interested in. This price fluctuates. So we need the price.

Beancount supports a type of entry called a `price` entry that allows you to tell it what the price of an instrument was at a particular point in time, e.g.

```
2014-05-25 price IBM    182.27 USD
```

In order to keep Beancount simple and with few dependencies, the software does not automatically fetch these prices (you can check out LedgerHub for this purpose, or write your own script that will insert the latest prices in your input file if so desired… there are many libraries to fetch prices from the internet online). It only knows about market prices from all these price entries. Using these, it builds an in-memory historical database of prices over time and can query it to obtain the most current values.

Instead of supporting different reporting modes with options, you can trigger the insertion of unrealized gains by enabling a plugin:

```
plugin "beancount.plugins.unrealized" "Unrealized"
```

This will create a synthetic transaction at the date of the last of directives, that reflects the unrealized P/L. It books one side as Income and the other side as a change in Asset:

```
2014-05-25 U "Unrealized gain for 7 units of IBM (price:
              182.2700 USD as of 2014-05-25,
              average cost: 160.0000 USD)"
   Assets:US:ETrade:IBM:Unrealized          155.89 USD
   Income:US:ETrade:IBM:Unrealized         -155.89 USD
```

Note that I used an option in this example to specify a sub-account to book the unrealized gains to. The unrealized P/L shows up on a separate line in the balance sheet and the parent account should show the market value on its balance (which includes that of its sub-accounts).

Commissions

So far we have not discussed trading commissions. Depending on the tax law that applies to you, the costs associated with trading may be deductible from the raw capital gain as we've calculated it in the previous examples. These are considered expenses by the government, and it is often the case that you can deduct those trading commissions (it's entirely reasonable from their part, you did not pocket that money after all).

In the examples above, the capital gains and commission expenses get tracked into two separate accounts. For example, you could end up with reported balances that look like this:

```
   Income:US:ETrade:PnL                  -645.02 USD
   Expenses:Financial:Commissions          39.80 USD
```

(Just to be clear, this is to be interpreted as a *profit* of $645.02 and an expense of $39.80.) You could subtract these numbers to obtain an *approximation* of the P/L without costs: 645.02 - 39.80 =

$605.22. However, this is only an approximation of the correct P/L value. To understand why, we need to look at an example where a partial number of shares are sold across a reporting period.

Imagine that we have an account with a commission rate of $10 per trade, 100 shares of ITOT were bought in 2013, 40 of those shares were later sold in that same year, and the remaining 60 were sold the year after, a scenario that looks like this:

2013-09-01 Buy 100 ITOT at $80, **commission = 10$**

2013-11-01 Sell 40 ITOT at $82, commission = 10$

2014-02-01 Sell 60 ITOT at $84, commission = 10$

If you computed the sum of commissions paid at the end of 2013, you would have $20, and using the approximate method outlined previously, for so 2013 and 2014 you would declare

2013: P/L of 40 x ($82 - $80) - ($10 + $10) = $60

2014: P/L of 60 x ($84 - $80) - $10 = $230

However, strictly speaking, this is incorrect. The $10 commission paid on *acquiring* the 100 shares has to be pro-rated with respect to the number of shares sold. This means that on that first sale of 40 shares only 4$ of the commission is deductible: $10 x (40 shares / 100 shares), and so we obtain:

2013: P/L of 40 x ($82 - $80) - $(4 + 10) = $66

2014: P/L of 60 x ($84 - $80) - $(6 + 10) = $224

As you can see, the P/L declared for each year differs, even if the sum of the P/L for both years is the same ($290).

A convenient method to automatically allocate the acquisition costs to the pro-rata value of the number of shares sold is to add the acquisition trading cost to the total book value of the position. In this example, you would say that the position of 100 shares has a book value $8010 instead of $8000: 100 share x $80/share + $10, or equivalently, that the individual shares have a book value of $80.10 each. This would result in the following calculation:

2013: P/L of 40 x ($82 - $80.10) - $10 = $66

2014: P/L of 60 x ($84 - $80.10) - $10 = $224

You could even go one step further and fold the commission on *sale* into the price of each share sold as well:

2013: P/L of 40 x ($81.75 - $80.10) = $66

2014: P/L of 60 x ($83.8333 - $80.10) = $224

This may seem overkill, but imagine that those costs were much higher, as is the case on large commercial transactions; the details do begin to matter to the tax man. Accurate accounting is important, and we need to develop a method to do this more precisely.

We don't currently have a good method of doing this with our input syntax. A suitable method is currently being developed and a [proposal](#) is on the table. Also see mailing-list for details. [June 2014]

Stock Splits

Stock splits are currently dealt with by emptying an account's positions and recreating the positions at a different price:

```
2004-12-21 * "Autodesk stock splits"
   Assets:US:MSSB:ADSK          -100 ADSK {66.30 USD}
   Assets:US:MSSB:ADSK           200 ADSK {33.15 USD}
```

The postings balance each other, so the rule is respected. As you can see, this requires no special syntax feature. It also handles more general scenarios, such as the odd split of the Google company that occurred on the NASDAQ exchange in April 2014, into two different classes of stock (voting and non-voting shares, at 50.08% and 49.92%, respectively):

```
2014-04-07 * "Stock splits into voting and non-voting shares"
   Assets:US:MSSB:GOOG         -25 GOOG {1212.51   USD} ; Old GOOG
   Assets:US:MSSB:GOOG          25 GOOG { 605.2850 USD} ; New GOOG
   Assets:US:MSSB:GOOGL         25 GOOG { 607.2250 USD}
```

Ultimately, maybe a plug-in module should be provided to more easily create such stock split transactions, as there is some amount of redundancy involved. We need to figure out the most general way to do this. But the above will work for now.

One problem with this approach is that the *continuity* of the trade lots is lost, that is, the purchase date of each lot has now been reset as a result of the transaction above, and it becomes impossible to automatically figure out the duration of the trade and its associated impact on taxation, i.e. long-term *vs.* short-term trade. Even without this the profit is still calculated correctly, but it is an annoying detail nonetheless.

One way to handle this is by using the Dated Lots (see the appropriate section of this doc). That way, the original trade date can be preserved on the new lots. This provides accurate timing information in addition to the capital gain/loss based on the price.

Another method for solving this and for easily propagating the lot trade date [has been proposed](#) and will be implemented in Beancount later on.

A more important problem with the current implementation is that the meaning of a unit of ADSK before and after the stock split is different. The price graph for this commodity unit will show a radical discontinuity! This is a more general problem that has yet to be addressed in both Beancount and Ledger. The [Commodity Definition Changes](#) document has a discussion to address this topic.

Cost Basis Adjustment and Return of Capital

Readjustment in cost basis may occur in managed funds, due to the fund's internal trading activities. This will typically occur in tax-sheltered accounts where the gain that occurs from such an adjustment has no impact on taxes, and where the cost basis is held at the average cost of all shares in each position.

If we have the specific lot prices being adjusted, it is doable to book these in the same manner as we dealt with stock splits:

```
2014-04-07 * "Cost basis adjustment for XSP"
  Assets:CA:RRSP:XSP          -100 ADSK {21.10 CAD}
  Assets:CA:RRSP:XSP           100 ADSK {23.40 CAD}
  Income:CA:RRSP:Gains      -230.00 CAD
```

However, this is really uncommon. The more common case of this is of an account using the average cost booking method, we don't currently have a way to deal with this. There is an active proposal in place to make this possible.

The cost basis adjustment is commonly found in Return of Capital events. These happen, for example, when funds are returning capital to the shareholders. This can be caused by winding down the operation. From the taxation point of view, these are non-taxable events and affect the cost basis of the equity in the fund. The number of shares might stay the same, but their cost basis needs to be adjusted for potential Gain/Loss calculation at the point of sale in the future.

Dividends

Dividends don't pose a particular problem. They are just income. They can be received as cash:

```
2014-02-01 * "Cash dividends received from mutual fund RBF1005"
  Assets:Investments:Cash          171.02 CAD
  Income:Investments:Dividends
```

Or they can be received as stock itself:

```
2014-02-01 * "Stock dividends received in shares"
  Assets:Investments:RBF1005       7.234 RBF1005 {23.64 CAD}
  Income:Investments:Dividends
```

In the case of dividends received as stock, as for stock purchases, you provide the cost basis at which the dividend was received (this should be available in your statements). If the account is held at average cost, this posting will simply merge with the other legs at the time an average cost booking is needed to be performed.

Average Cost Booking

At the moment, the only way to perform booking at average cost is painful: you would have to use the method outlined in the Stock Split section in order to revalue your inventory. This is impractical, however. There is an [active proposal](#) with an associated syntax to fully solve this problem.

Once the proposal is implemented, it will look like this:

```
2014-02-01 * "Selling 5 shares at market price 550 USD"
  Assets:Investments:Stock              -5 GOOG {*}
  Assets:Investments:Cash          2740.05 USD
  Expenses:Commissions                9.95 USD
  Income:Investments:CapitalGains
```

Any posting with a cost of "*" acting on an inventory will select all the shares of that currency (GOOG), merge them into a single one at the average cost, and then reduce that position at this new average cost.

Future Topics

I'll be handling the following topics later on:

- **Mark-to-Market**: Handling end-of-year mark-to-market for Section 1256 instruments (i.e., futures and options), by re-evaluating the cost basis. This is similar to a cost basis readjustment applied at the end of each year for all of these types instruments.

- **Short Sales**: these require little changes. We just have to allow negative numbers of units held at cost. At the moment we spit a warning when units held at cost go negative in order to detect data entry errors, but it would be easy to extend the Open directive syntax to allow this to occur on specific accounts which can hold short sales, which should just show as negative shares. All the arithmetic should otherwise just work naturally. Interest payments on margins would show up as distinct transactions. Also, when you short the stock, you don't receive dividends for those positions, but rather you have to pay them out. You would an expense account for this, e.g., `Expenses:StockLoans:Dividends`.

- **Trading Options**: I have no idea how to do this at the moment, but I imagine these could be held like shares of stock, with no distinctions. I don't foresee any difficulty.

- **Currency Trading**: At the moment, I'm not accounting for the positions in my FOREX accounts, just their P/L and interest payments. This poses interesting problems:
  - Positions held in a FOREX account aren't just long or short the way that stocks are: they are actually offsetting two commodities at the same time. For example, a long position in USD/CAD should increase the exposure of USD and decrease the

exposure in CAD, it can be seen as holding a long asset of USD and a short asset in CAD, at the same time. While it is possible to hold these positions as if they were distinct instruments (e.g., units of "USDCAD" with disregard for its components) but for large positions, especially if held over long periods of time for hedging purposes, it is important to deal with this and somehow allow the user to reflect the net currency exposures of multiple currency positions against the rest of their assets and liabilities.

○ We also need to deal with the gains generated by the closing of these positions: those generate a gain in the currency of the account, after conversion to this currency. For example, if you hold a currency account denominated in USD, and you go long EUR/JPY, when you close the position you will obtain a gain in EUR, and after conversion of the P/L from the EUR into the equivalent number of USD (via EUR/USD) the USD gain will be deposited in your account. This means that two rates are being used to estimate the current market value of any position: the differential between the current rate and the rate at the time of buying, and the rate of the base currency (e.g., EUR) in the account currency (e.g., USD).

Some of these involve new features in Beancount, but some not. Ideas welcome.