option "title" "Joe Smith's Personal Ledger"

The title of this ledger / input file. This shows up at the top of

every page.

option "name\_assets" "Assets"

option "name\_liabilities" "Liabilities"

option "name\_equity" "Equity"

option "name\_income" "Income"

option "name\_expenses" "Expenses"

Root names of every account. This can be used to customize your

category names, so that if you prefer "Revenue" over "Income" or

"Capital" over "Equity", you can set them here. The account names in

your input files must match, and the parser will validate these. You

should place these options at the beginning of your file, because

they affect how the parser recognizes account names.

option "account\_previous\_balances" "Opening-Balances"

Leaf name of the equity account used for summarizing previous

transactions into opening balances.

option "account\_previous\_earnings" "Earnings:Previous"

Leaf name of the equity account used for transferring previous

retained earnings from income and expenses accrued before the

beginning of the exercise into the balance sheet.

option "account\_previous\_conversions" "Conversions:Previous"

Leaf name of the equity account used for inserting conversions that

will zero out remaining amounts due to transfers before the opening

date. This will essentially "fixup" the basic accounting equation

due to the errors that priced conversions introduce.

option "account\_current\_earnings" "Earnings:Current"

Leaf name of the equity account used for transferring current

retained earnings from income and expenses accrued during the

current exercise into the balance sheet. This is most often called

"Net Income".

option "account\_current\_conversions" "Conversions:Current"

Leaf name of the equity account used for inserting conversions that

will zero out remaining amounts due to transfers during the exercise

period.

option "account\_rounding" "Rounding"

The name of an account to be used to post to and accumulate rounding

error. This is unset and this feature is disabled by default;

setting this value to an account name will automatically enable the

addition of postings on all transactions that have a residual

amount.

option "conversion\_currency" "NOTHING"

The imaginary currency used to convert all units for conversions at

a degenerate rate of zero. This can be any currency name that isn't

used in the rest of the ledger. Choose something unique that makes

sense in your language.

option "inferred\_tolerance\_default" "CHF:0.01"

option "default\_tolerance" "CHF:0.01"

THIS OPTION IS DEPRECATED: This option has been renamed to

'inferred\_tolerance\_default'

Mappings of currency to the tolerance used when it cannot be

inferred automatically. The tolerance at hand is the one used for

verifying (1) that transactions balance, (2) explicit balance checks

from 'balance' directives balance, and (3) in the precision used for

padding (from the 'pad' directive). The values must be strings in

the following format: <currency>:<tolerance> for example,

'USD:0.005'. By default, the tolerance used for currencies without

an inferred value is zero (which means infinite precision). As a

special case, this value, that is, the fallabck value used for all

currencies without an explicit default can be overridden using the

'\*' currency, like this: '\*:0.5'. Used by itself, this last example

sets the fallabck tolerance as '0.5' for all currencies. (Note: The

new value of this option is "inferred\_tolerance\_default"; it renames

the option which used to be called "default\_tolerance". The latter

name was confusing.) For detailed documentation about how precision

is handled, see this doc: http://furius.ca/beancount/doc/tolerances

(This option may be supplied multiple times.)

option "inferred\_tolerance\_multiplier" "1.1"

A multiplier for inferred tolerance values. When the tolerance

values aren't specified explicitly via the

'inferred\_tolerance\_default' option, the tolerance is inferred from

the numbers in the input file. For example, if a transaction has

posting with a value like '32.424 CAD', the tolerance for CAD will

be inferred to be 0.001 times some multiplier. This is the muliplier

value. We normally assume that the institution we're reproducing

this posting from applies rounding, and so the default value for the

multiplier is 0.5, that is, half of the smallest digit encountered.

You can customize this multiplier by changing this option, typically

expanding it to account for amounts slightly beyond the usual

tolerance, for example, if you deal with institutions with bad of

unexpected rounding behaviour. For detailed documentation about how

precision is handled, see this doc:

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

option "infer\_tolerance\_from\_cost" "True"

Enable a feature that expands the maximum tolerance inferred on

transactions to include values on cost currencies inferred by

postings held at-cost or converted at price. Those postings can

imply a tolerance value by multiplying the smallest digit of the

unit by the cost or price value and taking half of that value. For

example, if a posting has an amount of "2.345 RGAGX {45.00 USD}"

attached to it, it implies a tolerance of 0.001 x 45.00 \* M = 0.045

USD (where M is the inferred\_tolerance\_multiplier) and this is added

to the mix to enlarge the tolerance allowed for units of USD on that

transaction. All the normally inferred tolerances (see

http://furius.ca/beancount/doc/tolerances) are still taken into

account. Enabling this flag only makes the tolerances potentially

wider.

option "tolerance" "0.015"

THIS OPTION IS DEPRECATED: The 'tolerance' option has been

deprecated and has no effect.

The tolerance allowed for balance checks and padding directives. In

the real world, rounding occurs in various places, and we need to

allow a small (but very small) amount of tolerance in checking the

balance of transactions and in requiring padding entries to be auto-

inserted. This is the tolerance amount, which you can override.

option "use\_legacy\_fixed\_tolerances" "True"

Restore the legacy fixed handling of tolerances. Balance and Pad

directives have a fixed tolerance of 0.015 units, and Transactions

balance at 0.005 units. For any units. This is intended as a way for

people to revert the behavior of Beancount to ease the transition to

the new inferred tolerance logic. See

http://furius.ca/beancount/doc/tolerances for more details.

option "documents" "/path/to/your/documents/archive"

A list of directory roots, relative to the CWD, which should be

searched for document files. For the document files to be

automatically found they must have the following filename format:

YYYY-MM-DD.(.\*)

(This option may be supplied multiple times.)

option "operating\_currency" "USD"

A list of currencies that we single out during reporting and create

dedicated columns for. This is used to indicate the main currencies

that you work with in real life. (Refrain from listing all the

possible currencies here, this is not what it is made for; just list

the very principal currencies you use daily only.) Because our

system is agnostic to any unit definition that occurs in the input

file, we use this to display these values in table cells without

their associated unit strings. This allows you to import the numbers

in a spreadsheet (e.g, "101.00 USD" does not get parsed by a

spreadsheet import, but "101.00" does). If you need to enter a list

of operating currencies, you may input this option multiple times,

that is, you repeat the entire directive once for each desired

operating currency.

(This option may be supplied multiple times.)

option "render\_commas" "TRUE"

A boolean, true if the number formatting routines should output

commas as thousand separators in numbers.

option "plugin\_processing\_mode" "raw"

A string that defines which set of plugins is to be run by the

loader: if the mode is "default", a preset list of plugins are

automatically run before any user plugin. If the mode is "raw", no

preset plugins are run at all, only user plugins are run (the user

should explicitly load the desired list of plugins by using the

'plugin' option. This is useful in case the user wants full control

over the ordering in which the plugins are run).

option "plugin" "beancount.plugins.module\_name"

THIS OPTION IS DEPRECATED: The 'plugin' option is deprecated; it

should be replaced by the 'plugin' directive

A list of Python modules containing transformation functions to run

the entries through after parsing. The parser reads the entries as

they are, transforms them through a list of standard functions, such

as balance checks and inserting padding entries, and then hands the

entries over to those plugins to add more auto-generated goodies.

The list is a list of pairs/tuples, in the format (plugin-name,

plugin-configuration). The plugin-name should be the name of a

Python module to import, and within the module we expect a special

'\_\_plugins\_\_' attribute that should list the name of transform

functions to run the entries through. The plugin-configuration

argument is an optional string to be provided by the user. Each

function accepts a pair of (entries, options\_map) and should return

a pair of (new entries, error instances). If a plugin configuration

is provided, it is provided as an extra argument to the plugin

function. Errors should not be printed out the output, they will be

converted to strings by the loader and displayed as dictated by the

output medium.

(This option may be supplied multiple times.)

option "long\_string\_maxlines" "64"

The number of lines beyond which a multi-line string will trigger a

overly long line warning. This warning is meant to help detect a

dangling quote by warning users of unexpectedly long strings.

option "experiment\_explicit\_tolerances" "True"

Enable an EXPERIMENTAL feature that supports an explicit tolerance

value on Balance assertions. If enabled, the balance amount supports

a tolerance in the input, with this syntax: <number> ~ <tolerance>

<currency>, for example, "532.23 ~ 0.001 USD". See the document on

tolerances for more details:

http://furius.ca/beancount/doc/tolerances WARNING: This feature may

go away at any time. It is an exploration to see if it is truly

useful. We may be able to do without.

option "booking\_method" "SIMPLE"

The booking method to apply, for interpolation and for matching lot

specifications to the available lots in an inventory at the moment

of the transaction. Values may be 'SIMPLE' for the original method

used in Beancount, or 'FULL' for the newer method that does fuzzy

matching against the inventory and allows multiple amounts to be

interpolated (see http://furius.ca/beancount/doc/proposal-booking

for details).