## NAME

penny-fit - Penny financial institution statements parser

## SYNOPSIS

**penny-fit** [global-options] COMMAND [local-options] ARGS

## DESCRIPTION

**penny-fit** works with data you have downloaded from your financial institution. It parses statements that you have downloaded and adds transactions from the statements to your ledger, skipping those that have already been added. It also helps you reconcile your ledger with financial institution statements.

First you will have to configure and compile a **penny-fit** binary. You can use the **penny-fit-sample.hs** file in the *doc* directory of the **penny-bin** package as an example. Alternatively, if you know your way around Haskell, see the Haddock documentation for the *Penny.Brenner* module. The comments in that file should help you get started.

Currently Penny includes a parser for Open Financial Exchange, or OFX, data. Many banks make information available in this format, as Quicken and the now-defunct Microsoft Money both support the format.

## IMPORTING

To use **penny-fit**, first you will download the appropriate data from your financial institution and place it in a file. *penny-fit info* will tell give you a little more information about the place to look on your institution's web site to download the data, if you have configured that information in your binary.

Then, run *penny-fit -f ACCOUNT import FILENAME*. The first time you run this command, you will have to add the *--new* option after the *import* command, which allows the creation of a new database. Without this option, if the database is not found, **penny-fit** quits with an error message. The ACCOUNT must be a financial institution account that you configured in your **penny-fit** binary (if you configured a default account and you want to use that, you can omit the *-f* option). The FILENAME is the location of the data that you just downloaded.

The *import* command will examine the data that you downloaded. Using the unique identifiers already assigned to each posting by your financial institution, the *import* command determines whether you have already downloaded each particular posting. If the posting is new, *import* assigns a different, unique number to the posting. This is called a *U-number*. The U-number allows you to uniquely identify each posting that you download. The data from the financial institution, along with the U-number, is added to a database at the location specified in your configuration. *import* automatically skips postings that have already been processed, so you do not have to worry about importing duplicate postings.

## MERGING

Next you will want to merge new postings into your ledger. Do this by running *penny-fit -f ACCOUNT merge LEDGER_FILE...* where LEDGER_FILE is one or more filenames for your ledger files. *merge* examines your ledgers to see if each of the postings in the database for this financial institution is represented in your ledger. To do this it looks at the postings in the *pennyAcct* specified in your configuration. For each U-number in the database, *merge* sees if there is a posting in the *pennyAcct* with a tag bearing the U-number (e.g. if the U-number is 5, it looks for the tag *U5*). If a posting has more than one U-number tag, only the first is used; the others are ignored. If such a posting is found, *merge* moves on to the next U-number in the database. If no matching posting is found for a U-number, *merge* sees if there is a matching posting that does *not* have a U-number tag. If there is a posting in the *pennyAcct* that has the same quantity and date as the financial institution posting, *merge* will then examine the debit or credit of the ledger posting.

This table describes whether *penny-fit* will find a match:

| If the financial institution posting is a | and translator is | and the ledger posting is a | then is there a match? |
|---|---|---|---|
| increase | IncreaseIsDebit | debit | Yes |
| increase | IncreaseIsDebit | credit | No |
| increase | IncreaseIsCredit | debit | No |

| | | | |
|---|---|---|---|
| increase | IncreaseIsCredit | credit | Yes |
| decrease | IncreaseIsDebit | debit | No |
| decrease | IncreaseIsDebit | credit | Yes |
| decrease | IncreaseIsCredit | debit | Yes |
| decrease | IncreaseIsCredit | credit | No |

If **penny-fit** finds a match for a financial institution posting in this way, then it will assign a new U-number tag to the posting. If **penny-fit** does not find a match, then it will create an entirely new transaction and append it to the end of your ledger.

If it is creating an entirely new transaction, **penny-fit** will attempt to give the new transaction the same account and payee information that you have used for similar transactions in the past. To do this, **penny-fit** will first search through the database to find the most recent financial institution posting that has the same payee as the one of the new transaction. If one is found, **penny-fit** then searches through the postings in your ledger file to find the one that has the same U-number and account as the old financial institution posting. If it is found, **penny-fit** will assign the payee name found on the posting in the ledger to the new posting. Also, if the posting found in the ledger has exactly one sibling posting, **penny-fit** will assign the same account name from that sibling to the new sibling. You can turn off this automatic assignment of information by using the *--no-auto* or *-n* option to the *merge* command.

The result of *merge* is printed to standard output, unless you use the *--output FILENAME* or *-o FILENAME* option, in which case the output is sent to *FILENAME*. You can use multiple *-o* options. To explicitly send output to standard output, use *-o -*. Use **diff**(1) or **penny-diff**(1) to see what changes *merge* made. Typically you will need to edit the output somewhat.

## RECONCILING

Next you may wish to reconcile your ledger with your financial institution data (that is, "balance the checkbook"). Typically the most time-consuming part of this process is finding the postings in your ledger that match the postings on your bank statement. **penny-fit clear** will help with this, dramatically speeding up the process. To do this, download data from your financial institution that corresponds to the data that is covered within the current statement period. Run **penny-fit import** and **penny-fit merge** as described above. Then run

```
penny-fit -f ACCOUNT clear FIT_FILE LEDGER_FILE...
```

where FIT_FILE is the data file you downloaded from your financial institution, and LEDGER_FILE contains your ledger data. The *clear* command will mark as cleared (that is, assign a *C* flag to) all postings in your LEDGER_FILEs that correspond to one of the postings in the FIT_FILE. It does this by matching the U-number tags on your postings to the U-numbers in the database. If a posting has more than one U-number tag, only the first is used; the others are ignored.

As with the *merge* command, the results are printed to standard output unless you use the *--output FILENAME* or *-o FILENAME* option. Once you have verified that things are as they should be, you can use **penny-reconcile**(1) to mark the cleared postings as reconciled. **penny-basics**(7) has more details on how to use **penny** when reconciling a financial institution statement.

## OTHER COMMANDS

The *database* command prints the database for a particular financial institution to standard output in human-readable form (the database unfortunately is not in plain human-readable text). For instance you might use this to see what U-number is assigned to a particular financial institution posting.

The *print* command parses a downloaded file of financial institution data and prints the result to standard output. This is useful for seeing the contents of a financial institution data file, or for testing new parsers.

Every **penny-fit** command has a *-h* and a *--help* option. There is also a global *--help* option, as in *penny-fit --help*.

## BUGS

To quote another man page: "Bugs? You must be kidding, there are no bugs in this software. But if we happen to be wrong, send us an email with as much detail as possible to" omari@smileystation.com.

## SEE ALSO

**penny-suite**(7)