Dynamic Ledger

Generated by Doxygen 1.8.5

Sun Dec 22 2013 16:56:16

Contents

Chapter 1

Module Index

1.1 Modules

Here is a list of all modules:

Chapter 2

Class Index

2.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

Chapter 3

File Index

3.1 File List

Here is a list of all documented files with brief descriptions:

Chapter 4

Module Documentation

4.1 Totals Macros

Indices for credit and bank totals.

Macros

- #define I_NOT_THERE_YET 0
- #define I_PENDING 1
- #define I_CLEARED 2
- #define I_PENDING_BAL 3
- #define I_OVERALL_BAL 4
- #define N_TOTALS 5

4.1.1 Detailed Description

Ledger.bank_totals and Ledger.credit_totals are arrays of doubles that store various summaries of the ledger: for example, how much total money has cleared in a particular bank. These macros store indices that say which entry in Ledger.bank_totals and Ledger.credit_totals stores which total.

4.1.2 Macro Definition Documentation

4.1.2.1 #define I_CLEARED 2

In Ledger.bank_totals, etc., index of "available" balance.

4.1.2.2 #define I_NOT_THERE_YET 0

In Ledger.bank_totals, etc., index of money not yet arrived. This does not include pending money.

4.1.2.3 #define I_OVERALL_BAL 4

In Ledger.bank_totals, etc., index of overall balance. This is the true balance: "available" balance + pending money + money not yet arrived.

4.1.2.4 #define I_PENDING 1

In Ledger.bank_totals, etc., index of the money listed as "pending" on the bank's website.

4.1.2.5 #define I_PENDING_BAL 3

In Ledger.bank_totals, etc., index of pending balance (pending money + "available" balance).

4.1.2.6 #define N_TOTALS 5

Number of entries in Ledger.bank_totals, etc.

4.2 Return Value Macros

Return values from various functions.

Macros

- #define LSUCCESS 0

  Success return code.
- #define LFAILURE 1

  Failure return code.
- #define LNO 0

  "No" return code.
- #define LYES 1

  "Yes" return code.
- #define NO_INDEX -1

4.2.1 Detailed Description

Some functions have return types of err_t, bool_t, or index_t. These macros define the values that these types can take on.

4.2.2 Macro Definition Documentation

4.2.2.1 #define LFAILURE 1

Return value indicating a failure for functions with return type err_t.

4.2.2.2 #define LNO 0

Return value indicating a "yes" for functions with return type bool_t.

4.2.2.3 #define LSUCCESS 0

Return value indicating a success for functions with return type err_t.

4.2.2.4 #define LYES 1

Return value indicating a "no" for functions with return type bool_t.

4.2.2.5 #define NO_INDEX -1

For functions returning an index_t type (usually array lookup functions like which), this is the return value indicating that the index of a candidate entry was not found in the respective array.

4.3 Internal Macros

Internal macros.

Macros

- #define ENTRYSIZE 256

  Maximum entry size.
- #define EPS 0.000025

  "Epsilon"
- #define FILENAMESIZE 256

  File name size.
- #define LINESIZE 4096

  Line size.
- #define NFIELDS 6

  Number of fields (columns).
- #define NIL "\0"

  The empty string.

4.3.1 Detailed Description

These macros define internal parameters such the maximum size of filenames and the number of fields (columns) in a ledger file.

4.3.2 Macro Definition Documentation

4.3.2.1 #define ENTRYSIZE 256

Maximum number of characters for the entries in Ledger.entries.

4.3.2.2 #define EPS 0.000025

A small constant used to check if doubles and floats round to 0.00.

4.3.2.3 #define FILENAMESIZE 256

Maximum number of characters for filenames

4.3.2.4 #define LINESIZE 4096

Maximum number of characters for each line in an input ledger file.

4.3.2.5 #define NFIELDS 6

Number of columns in the ledger. Each column represents a particular feature of a transaction (row). The fields include transaction amount, transaction status code, credit account, bank account, bank partition, and description.

4.4 Return Types

Return types.

Typedefs

- typedef int err_t
  Error status type.
- typedef int bool_t
  Boolean type.
- typedef int index_t
  Array index type.
- typedef char * color_t
  Color code type.

4.4.1 Detailed Description

Defines the return types of most functions.

4.4.2 Typedef Documentation

4.4.2.1 typedef int bool_t

Can take on values LYES or LNO.

4.4.2.2 typedef char* color_t

Type definition for coloring output to the terminal window.

4.4.2.3 typedef int err_t

The error status can take on values LSUCCESS and LFAILURE.

4.4.2.4 typedef int index_t

Index in an array (used for lookup functions).

4.5 Utility Functions

Utility functions.

Functions

- bool_t col_delim_char (char c)

  Checks whether a character is a column separator.
- bool_t filled_partitions (Ledger *ledger, int bank)

  Tests if any of the partitions in a given bank have nonzero balance.
- bool_t input_file (char *filename)

  Tests if the given input file is usable.
- bool_t legal_double (char *s)

  Checks if a character string can be converted into a meaningful floating point number.
- bool_t legal_amounts (Ledger *ledger)

  Checks if a the AMOUNTS column in the ledger stores strings that can be converted into meaningful floating point numbers.
- bool_t legal_status_code (char *s)

  Checks if a character string is one of the legal status codes defined in the Status_Macros module.
- bool_t legal_status_codes (Ledger *ledger)

  Checks if all the character strings in the STATUS column of the ledger are legal status codes as defined in the Status_Macros module.
- bool_t locked (char *status)

  Checks if a status code "locks" a transaction.
- bool_t output_file (char *filename)

  Tests if the given output file is usable.
- bool_t row_delim_char (char c)

  Checks whether a character is a row separator.
- bool_t small_norm (double d)

  Checks whether a floating point number rounds to 0.00.
- bool_t space (char c)

  Checks whether a character is a whitespace character.
- bool_t str_equal (const char *s1, const char *s2)

  Checks whether two strings are equal.
- bool_t untotaled (Ledger *ledger)

  Checks if account totals have been calculated for a Ledger object.
- color_t color (double d, int usecolor)

  Finds the correct color code for an amount in summary output.
- index_t col_delim_str (char *s)

  Finds the first column delimiter in a string.
- index_t row_delim_str (char *s)

  Finds the first row delimiter in a string.
- index_t which (char **s, char *find, int n)

  Finds an occurrence of "find" in "s".
- index_t which_bank_total (char *status)

  Given a status code, finds the correct index in the bank_totals array member of the Ledger type.
- index_t which_credit_total (char *status)

  Given a status code, finds the correct index in the credit_totals array member of the Ledger type.
- int qcmp (const void *a, const void *b)

  Comparison function for character strings in qsort.
- err_t str_strip (char *s)

Strips leading and trailing whitespace from a character string.

- err_t unique (char **a, int n, char ***ret, int *nunique)

  Finds all the unique elements in an array of character strings.

- err_t usage ()

  Prints usage details.

4.5.1 Detailed Description

These are miscellaneous utility functions. Many are for checking the quality of the data.

4.5.2 Function Documentation

4.5.2.1 bool_t col_delim_char (char *c*)

Parameters

c	Character to check.

Returns

bool_t: LYES or LNO

Tests if the given character is one of the legal column separators given in the character string macro, COLUMN_SEPARATORS.

4.5.2.2 index_t col_delim_str (char * *s*)

Parameters

s	Character string to check.

Returns

index_t

Loops through the characters in the argument character string and returns the index of the first character that is a column separator. if no column separator is found, the function returns no_index.

4.5.2.3 color_t color (double *d,* int *usecolor*)

Parameters

d	A double precision number representing some amount of money.
usecolor	An int: 1 if color-coded printing is enabled and 0 otherwise.

Returns

color_t: One of the color codes defined in the Printing_Macros module.

Finds the correct color code for an amount in summary output. Printing amounts in color to the terminal window makes the output prettier than otherwise. Negative amounts are printed in one color, positive amounts in another, and zeroes in a third color. Colors are defined in the Printing_Macros module.

4.5.2.4 bool_t filled_partitions (Ledger * *ledger,* int *bank*)

Parameters

ledger	A pointer to a Ledger object.
bank	Index of the bank in ledger->banks.

Returns

bool_t: LYES or LNO

Loops through all the partitions of a given bank account in a Ledger object. Returns LYES if any partition (including the unnamed partition) is found to have a nonzero amount. Returns LNO otherwise.

4.5.2.5 bool_t input_file (char * *filename*)

Parameters

filename	A character array giving the full path fo the filename.

Returns

bool_t: LYES or LNO

Checks if the input file with the given full path name is usable. That is, it tries to open the file for reading and returns LYES on success and LNO on failure.

4.5.2.6 bool_t legal_amounts (Ledger * *ledger*)

Parameters

ledger	A pointer to a Ledger object.

Returns

bool_t: LYES or LNO

Loops through all the entries in the AMOUNTS column of the given Ledger object and checks that all transaction amounts (stored as human-readable character strings) can be converted into meaningful floating point numbers. Specifically, it calls legal_double on every entry. The empty string is legal and taken to be 0.00.

4.5.2.7 bool_t legal_double (char * *s*)

Parameters

s	A character string that looks like a number to humans.

Returns

bool_t: LYES or LNO

Checks if a character string can be converted into a meaningful floating point number. Uses errno.h to do so.

4.5.2.8 bool_t legal_status_code (char * *s*)

Parameters

s	A character string representing a status code.

Returns

bool_t: LYES or LNO

Checks if a character string is one of the legal status codes defined in the Status_Macros module. The empty string NIL is also acceptable.

4.5.2.9 bool_t legal_status_codes (Ledger * *ledger*)

Parameters

ledger	A pointer to a Ledger object.

Returns

bool_t: LYES or LNO

Checks if all the character strings in the STATUS column of the ledger are legal status codes as defined in the Status_Macros module. Specifically, it loops through ledger->entries[STATUS] and calls legal_status_code on every entry. The empty string, NIL, is also legal.

4.5.2.10 bool_t locked (char * *status*)

Parameters

status	A character string representing a status code.

Returns

bool_t: LYES or LNO

Checks if a status code "locks" a transaction. Any transaction with a legal status code not equal to NIL or REMOVE is "locked": that is, it will be ignored by the trim_ledger, condense, and clean functions.

4.5.2.11 bool_t output_file (char * *filename*)

Parameters

filename	A character array giving the full path fo the filename.

Returns

bool_t: LYES or LNO

Checks if the output file with the given full path name is usable. That is, it tries to open the file for writing and returns LYES on success and LNO on failure.

4.5.2.12 int qcmp (const void * *a*, const void * *b*)

Parameters

a	First character string.
b	Second character string.

Returns

index_t

Comparison function for character strings in qsort. qcmp is used to sort arrays of character strings as in the function, unique.

4.5.2.13 bool_t row_delim_char (char *c*)

Parameters

c	Character to check.

Returns

bool_t: LYES or LNO

Tests if the given character is one of the legal row separators given in the character string macro, ROW_SEPARATORS.

4.5.2.14 index_t row_delim_str (char * *s*)

Parameters

s	Character string to check.

Returns

index_t

Loops through the characters in the argument character string and returns the index of the first character that is a row separator. if no row separator is found, the function returns no_index.

4.5.2.15 bool_t small_norm (double *d*)

Parameters

d	Double-precision floating point number to check.

Returns

bool_t: LYES or LNO

Checks whether a floating point number rounds to 0.00. Specifically, it checks if the square of the argument is less than EPS, returns LYES if so, and LNO otherwise.

4.5.2.16 bool_t space (char *c*)

Parameters

c	Character to check

Returns

bool_t: LYES or LNO

Checks whether a character is a whitespace character, returns LYES if it is, and LNO otherwise.

4.5.2.17 bool_t str_equal (const char * *s1*, const char * *s2*)

Parameters

s1	First string to check.
s2	Second string to check.

Returns

bool_t: LYES or LNO

Checks whether two strings are equal. It uses strcmp for nonnull strings, but unlike strcmp, it is safe to use with null char pointers.

4.5.2.18 err_t str_strip (char * *s*)

Parameters

s	Character string

Returns

err_t: LSUCCESS or LFAILURE

Strips leading and trailing whitespace from a character string.

4.5.2.19 err_t unique (char ** *a*, int *n*, char *** *ret*, int * *nunique*)

Parameters

a	Array of character strings.
n	Number of elements of s.
ret	Array of the unique elements of s.
nunique	Number elements of ret.

Returns

err_t: LSUCCESS or LFAILURE

Finds all the unique elements in an array of character strings. You can treat "a" and "n" as the arguments, and ret and nunique as the return values (passed by reference). ret and nunique should be unallocated and uninitialized when unique is called. IMPORTANT NOTE: unique adds a blank character string to the list of unique character strings if one is not already present. This is so that every ledger automatically has an unnamed bank account, an unnamed credit account, and an unnamed bank partition for every bank account.

4.5.2.20 bool_t untotaled (Ledger * *ledger*)

Parameters

ledger	A pointer to a Ledger object.

Returns

bool_t: LYES or LNO

Checks if account totals have been calculated for a Ledger object. If not calculated, pointers to these totals should have already been initialized to NULL by new_ledger. Returns LYES if totals have been calculated and LNO otherwise.

4.5.2.21 err_t usage ( )

Returns

err_t: LSUCCESS or LFAILURE

Prints usage details of the command line interface version of the program.

4.5.2.22 index_t which (char ** *s*, char * *find*, int *n*)

Parameters

s	Array of character strings.
find	Character string to find.
n	Int: how many elements in s.

Returns

index_t

Finds an occurrence of the character string, "find", in the array of character strings, "s", using binary search. The index returned need not be the index of the first occurrence. Returns NO_INDEX if "find" is not an element of s.

4.5.2.23 index_t which_bank_total (char * *status*)

Parameters

status	Character string status code.

Returns

index_t

Given a status code, this function finds the correct index in the bank_totals array member of the Ledger type. bank_totals is an array indexed by the transaction status given in the Status_Macros module. If no index is found for the given status code, NO_INDEX is returned.

4.5.2.24 index_t which_credit_total (char * *status*)

Parameters

status	Character string status code.

Returns

index_t

Given a status code, this function finds the correct index in the credit_totals array member of the Ledger type. credit_totals is an array indexed by the transaction status given in the Status_Macros module. If no index is found for the given status code, NO_INDEX is returned.

4.6 Ledger Memory Functions

Ledger memory functions.

Functions

- err_t alloc_entries (Ledger *ledger)

  Allocates the "entries" member of a Ledger object.
- err_t alloc_totals (Ledger *ledger)

  Allocates space for the numerical summaries of the ledger.
- err_t free_entries (Ledger *ledger)

  Frees the "entries" member of a Ledger object.
- err_t free_for_retotal (Ledger *ledger)

  Frees the account names and numerical summaries of a Ledger object.
- err_t free_ledger (Ledger **ledger)

  Frees a whole Ledger object.
- err_t get_names (Ledger *ledger)

  Gets the account names of a Ledger object.
- err_t get_totals (Ledger *ledger)

  Computes numerical summaries of a Ledger object.
- err_t new_ledger (Ledger **ledger)

  Creates a new Ledger object.
- err_t copy_ledger (Ledger **out_ledger, Ledger *in_ledger)

  Copies one Ledger object into another.

4.6.1 Detailed Description

These are functions for creating, initializing, copying, and destroying Ledger objects.

4.6.2 Function Documentation

4.6.2.1 err_t alloc_entries (Ledger * *ledger*)

Parameters

ledger	pointer to a Ledger object

Returns

err_t: LSUCCESS or LFAILURE

Allocate the "entries" member of a Ledger object. "entries" stores the entries of the Ledger spreadsheet with rows representing transactions and columns representing features of the transactions.

4.6.2.2 err_t alloc_totals (Ledger * *ledger*)

Parameters

ledger	pointer to a Ledger object

Returns

err_t: LSUCCESS or LFAILURE

Allocates space for the numerical summaries of the ledger. These include the bank_totals, credit_totals, and partition_totals members of a Ledger object.

4.6.2.3 err_t copy_ledger (Ledger ** *out_ledger*, Ledger * *in_ledger*)

Parameters

out_ledger	pointer to a pointer to the output Ledger object.
in_ledger	pointer to the input Ledger object.

Returns

err_t: LSUCCESS or LFAILURE

Copies one Ledger object into another. Specifically, out_ledger is freed, and then in_ledger is copies into it. out_ledger should point to NULL if it is empty.

4.6.2.4 err_t free_entries (Ledger * *ledger*)

Parameters

ledger	pointer to a Ledger object

Returns

err_t: LSUCCESS or LFAILURE

Frees the "entries" member of a Ledger object. "entries" stores the entries of the Ledger spreadsheet with rows representing transactions and columns representing features of the transactions.

4.6.2.5 err_t free_for_retotal (Ledger * *ledger*)

Parameters

ledger	pointer to a Ledger object.

Returns

err_t: LSUCCESS or LFAILURE

Frees the account names and numerical summaries of a Ledger object. bank_totals, credit_totals, partition_totals, banks, credits, etc. are freed. This is so that the numerical summaries and account names can be recomputed after a change to the Ledger object.

4.6.2.6 err_t free_ledger (Ledger ** *ledger*)

Parameters

ledger	pointer to a pointer to a Ledger object.

Returns

err_t: LSUCCESS or LFAILURE

Frees a whole ledger object and sets the pointer to NULL. so that the program knows that it is freed.

4.6.2.7 err_t get_names (Ledger * *ledger*)

Parameters

ledger	pointer to a Ledger object.

Returns

err_t: LSUCCESS or LFAILURE

Gets the account names of a ledger object. Specifically, fills the banks, credits, and partitions member arrays of the Ledger object.

4.6.2.8 err_t get_totals (Ledger * *ledger*)

Parameters

ledger	pointer to a Ledger object.

Returns

err_t: LSUCCESS or LFAILURE

Compute numerical summaries on a Ledger object. These summaries are stored in bank_totals, credit_totals, and partition_totals.

4.6.2.9 err_t new_ledger (Ledger ** *ledger*)

Parameters

ledger	pointer to a pointer to a Ledger object.

Returns

err_t: LSUCCESS or LFAILURE

Creates a new Ledger object. Pointers are initialized to NULL so that the program knows that they do not point to any meaningful memory. ledger should point to NULL if it is empty.

4.7 Ledger Modify Functions

Functions for modifying Ledger objects.

Functions

- err_t clean (Ledger *ledger, int sort_locked)

  Cleans a ledger object.
- err_t condense (Ledger *ledger)

  Condenses a ledger object.
- err_t copy_rows (Ledger *ledger, Ledger **clipboard, int *rows, int howmany)

  Copies selected rows (transactions).
- err_t cut_rows (Ledger *ledger, Ledger **clipboard, int *rows, int howmany)

  Cut selected rows (transactions).
- err_t edit_entry (Ledger *ledger, char *entry, int row, int field, int append)

  Modify a ledger entry and update the Ledger object accordingly.
- err_t edit_entry_noretotal (Ledger *ledger, char *entry, int row, int field, int append)

  Modify a ledger entry and DO NOT update the Ledger object accordingly.
- err_t edit_row (Ledger *ledger, char **entries, int row, int append)

  Modify a whole row (transaction) and update the Ledger object accordingly.
- err_t map_to_coords (Ledger *ledger, char *entry, int *rows, int *fields, int howmany, int append)

  Map a character string multiple entries.
- err_t map_to_coords_colmajor (Ledger *ledger, char *entry, int **coords, int howmany, int append)

  Map a character string multiple entries.
- err_t map_to_coords_rowmajor (Ledger *ledger, char *entry, int **coords, int howmany, int append)

  Map a character string multiple entries.
- err_t insert_rows (Ledger *ledger, int row, int howmany)

  Insert blank rows.
- err_t insert_filled_rows (Ledger *ledger, char **entries, int row, int howmany)

  Insert filled rows.
- err_t move_rows (Ledger *ledger, int *rows, int nrows, int moveto)

  Move rows to a specified location.
- err_t paste_rows (Ledger *ledger, Ledger *clipboard, int where)

  Paste rows into a specified location.
- err_t permute_rows (Ledger *ledger, int *order)

  Permute rows (transactions).
- err_t rename_bank (Ledger *ledger, char *from, char *to)

  Safely renames a bank account.
- err_t rename_credit (Ledger *ledger, char *from, char *to)

  Safely renames a credit account.
- err_t rename_partition (Ledger *ledger, char *bank, char *from, char *to)

  Safely renames a partition bank account.
- err_t remove_rows (Ledger *ledger)

  Removes rows (transactions) marked for removal.
- err_t retotal (Ledger *ledger)

  Recalculate names and totals.
- err_t trim_ledger (Ledger *ledger)

  Removes blank rows.
- err_t sort_by_status (Ledger *ledger, int sort_locked)

  Sorts rows (transactions) by transaction status code.

- err_t strip_ledger (Ledger *ledger)

  Removes whitespace from the entries of a Ledger object.
- err_t swap_rows (Ledger *ledger, int row1, int row2)

  Interchanges two rows of a ledger object.
- err_t repartition (Ledger *ledger, char *bank, char **partitions, double *amounts_arg, int npartitions, int as_percentages)

  Repartitions a bank account.
- err_t unlock (Ledger *ledger)

  Unlock all cleared transactions.

4.7.1 Detailed Description

These are functions for modifying Ledger objects. These include (but are not limited to) functions to edit entries and rows, copy and paste rows, and trim and condense entire Ledger objects.

4.7.2 Function Documentation

4.7.2.1 err_t clean (Ledger * *ledger,* int *sort_locked*)

Parameters

ledger	Pointer to a Ledger object.
sort_locked	Set to 1 to bring all locked rows to the top.

Returns

err_t: LSUCCESS or LFAILURE

Cleans a ledger object. Specifically, calls condense and sort_by_status.

4.7.2.2 err_t condense (Ledger * *ledger*)

Parameters

ledger	Pointer to a Ledger object.

Returns

err_t: LSUCCESS or LFAILURE

Condenses a ledger object. Specifically, this function condenses all the cleared and unlocked transactions (rows) to make a smaller ledger with the same account and partition totals.

4.7.2.3 err_t copy_rows (Ledger * *ledger,* Ledger ** *clipboard,* int * *rows,* int *howmany*)

Parameters

ledger	Pointer to a Ledger object.
clipboard	Pointer to a pointer to the Ledger object to copy the rows into.

rows	The indices of the rows in ledger to be copied.
howmany	Number of elements of rows.

Returns

err_t: LSUCCESS or LFAILURE

Copies selected rows (transactions). Specifically, the rows (transactions) whose indices are in "rows" are copied from "ledger" to "clipboard". The clipboard is overwritten in this function.

4.7.2.4 err_t cut_rows (Ledger * *ledger*, Ledger ** *clipboard*, int * *rows*, int *howmany*)

Parameters

ledger	Pointer to a Ledger object.
clipboard	Pointer to a pointer to the Ledger object to copy the rows into.
rows	The indices of the rows in ledger to be cut.
howmany	Number of elements of rows.

Returns

err_t: LSUCCESS or LFAILURE

Cuts the selected rows (transactions). Specifically, copy_rows is called, and then the copied rows are removed from the original Ledger object.

4.7.2.5 err_t edit_entry (Ledger * *ledger*, char * *entry*, int *row*, int *field*, int *append*)

Parameters

ledger	Pointer to a Ledger object.
entry	New entry (character string).
row	The row of the entry to overwrite.
field	The field (column) of the entry to overwrite.
append	Append option.

Returns

err_t: LSUCCESS or LFAILURE

Overwrite an entry of a Ledger object with a new entry. Specifically, ledger->entries[field][row] is replaced with "entry", and the other data in the Ledger object is updated with calls to get_names and get_totals. The append option works as in edit_entry_noretotal

4.7.2.6 err_t edit_entry_noretotal (Ledger * *ledger*, char * *entry*, int *row*, int *field*, int *append*)

Parameters

ledger	Pointer to a Ledger object.
entry	New entry (character string).
row	The row of the entry to overwrite.

field	The field (column) of the entry to overwrite.
append	Append option.

Returns

err_t: LSUCCESS or LFAILURE

Overwrite an entry of a Ledger object with a new entry. Specifically, ledger->entries[field][row] is replaced with "entry", and the other data in the Ledger object is NOT updated. Set "append" to 0 to overwrite each entry, 1 to append to the head of each entry, and 2 to append to the tail of each entry.

4.7.2.7 err_t edit_row (Ledger * *ledger,* char ** *entries,* int *row,* int *append*)

Parameters

ledger	Pointer to a Ledger object.
entries	New row (array of character string entries).
row	The row to overwrite.
append	Append option.

Returns

err_t: LSUCCESS or LFAILURE

Overwrite a row of a Ledger object with a new row. Specifically, ledger->entries[field][row] is replaced with entries[row] for row = 0, .., NFIELDS. The other data in the Ledger object is updated with calls to get_names and get_totals. APPEND works as in edit_entry_noretotal.

4.7.2.8 err_t insert_filled_rows (Ledger * *ledger,* char ** *entries,* int *row,* int *howmany*)

Parameters

ledger	Pointer to a Ledger object.
entries	Content of the new row to insert multiple times.
row	Where to insert the rows.
howmany	How many blank rows to insert.

Returns

err_t: LSUCCESS or LFAILURE

Insert a multiple copies of a new filled row at "row" in the ledger.

4.7.2.9 err_t insert_rows (Ledger * *ledger,* int *row,* int *howmany*)

Parameters

ledger	Pointer to a Ledger object.
row	Where to insert the rows.
howmany	How many blank rows to insert.

Returns

err_t: LSUCCESS or LFAILURE

Insert blank rows into the "entries" member array of a Ledger object.

4.7.2.10 **err_t map_to_coords (Ledger * *ledger,* char * *entry,* int * *rows,* int * *fields,* int *howmany,* int *append*)**

Parameters

ledger	Pointer to a Ledger object.
entry	Character string to map.
rows	Rows of entries to map to.
fields	Columns of entries to map to.
howmany	Number of elements of rows and fields.
append	Append option.

Returns

err_t: LSUCCESS or LFAILURE

Map a single character string to multiple entries of a Ledger object. Specifically, ledger->entries[fields[i]][rows[i]] is replaced with "entry", for i from 0 to length(rows) - 1. Set "append" to 0 to overwrite each entry, 1 to append to the head of each entry, and 2 to append to the tail of each entry.

4.7.2.11 err_t map_to_coords_colmajor (Ledger * *ledger*, char * *entry*, int ** *coords*, int *howmany*, int *append*)

Parameters

ledger	Pointer to a Ledger object.
entry	Character string to map.
coords	Coordinates of entries to map to in column major order.
howmany	Number of elements of rows and fields.
append	Append option.

Returns

err_t: LSUCCESS or LFAILURE

Map a single character string to multiple entries of a Ledger object. Specifically, ledger->entries[coords[0][i]][rows[1][i]] is replaced with "entry", for i from 0 to length(coords) - 1. Set "append" to 0 to overwrite each entry, 1 to append to the head of each entry, and 2 to append to the tail of each entry.

4.7.2.12 err_t map_to_coords_rowmajor (Ledger * *ledger*, char * *entry*, int ** *coords*, int *howmany*, int *append*)

Parameters

ledger	Pointer to a Ledger object.
entry	Character string to map.
coords	Coordinates of entries to map to in row major order.
howmany	Number of elements of rows and fields.
append	Append option.

Returns

err_t: LSUCCESS or LFAILURE

Map a single character string to multiple entries of a Ledger object. Specifically, ledger->entries[coords[i][0]][rows[i][1]] is replaced with "entry", for i from 0 to length(coords) - 1. Set "append" to 0 to overwrite each entry, 1 to append to the head of each entry, and 2 to append to the tail of each entry.

4.7.2.13 err_t move_rows (Ledger * *ledger*, int * *rows*, int *nrows*, int *moveto*)

Parameters

ledger	Pointer to a Ledger object.
rows	Array of the indices of the rows to move.
nrows	Number of rows to move.
moveto	Where to move the rows: i.e., the destination row index.

Returns

err_t: LSUCCESS or LFAILURE

Move the rows (transactions) of the "entries" member array of a Ledger object to another location (row) in the same array. The workhorse of this function is permute_rows.

4.7.2.14 err_t paste_rows (Ledger * *ledger,* Ledger * *clipboard,* int *where*)

Parameters

ledger	Pointer to a Ledger object.
clipboard	Pointer to a Ledger object containing the rows to paste.
where	Where to paste the rows: i.e., the destination row index.

Returns

err_t: LSUCCESS or LFAILURE

Pastes rows (transactions) in "clipboard" into "ledger" at row "where". insert_rows is called, and then the relevant rows are copied in.

4.7.2.15 err_t permute_rows (Ledger * *ledger,* int * *order*)

Parameters

ledger	Pointer to a Ledger object.
order	Array of indices to sort the rows by.

Returns

err_t: LSUCCESS or LFAILURE

Permute the rows (transactions) of a Ledger object. Specifically, bubble sort is applied to the int vector, "order". The same sorting operations on "order" are applied to the rows (transactions) of the Ledger object in ledger->entries.

4.7.2.16 err_t remove_rows (Ledger * *ledger*)

Parameters

ledger	Pointer to a Ledger object.

Returns

err_t: LSUCCESS or LFAILURE

Remove the rows in a Ledger object with transaction status code REMOVE. The rows with this status are sent to the bottom of the "entries" array of the Ledger object and then freed.

4.7.2.17 err_t rename_bank (Ledger * *ledger,* char * *from,* char * *to*)

Parameters

ledger	Pointer to a Ledger object.
from	Which bank account to rename.
to	New name for the bank account.

Returns

err_t: LSUCCESS or LFAILURE

Safely renames a bank account and updates the other data in the Ledger object to reflect the change.

4.7.2.18 err_t rename_credit (Ledger * *ledger*, char * *from*, char * *to*)

Parameters

ledger	Pointer to a Ledger object.
from	Which credit account to rename.
to	New name for the credit account.

Returns

err_t: LSUCCESS or LFAILURE

Safely renames a credit account and updates the other data in the Ledger object to reflect the change.

4.7.2.19 err_t rename_partition (Ledger * *ledger*, char * *bank*, char * *from*, char * *to*)

Parameters

ledger	Pointer to a Ledger object.
bank	Bank account of the partition to rename.
from	Which partition to rename.
to	New name for the partition.

Returns

err_t: LSUCCESS or LFAILURE

Safely renames a partition of a bank account and updates the other data in the Ledger object to reflect the change.

4.7.2.20 err_t repartition (Ledger * *ledger*, char * *bank*, char ** *partitions*, double * *amounts_arg*, int *npartitions*, int *as_percentages*)

Parameters

ledger	Pointer to a Ledger object.
bank	Bank account to repartition.
partitions	Array of character strings giving the names of the new partitions.
amounts_arg	Amount allocated to each partition
npartitions	Number of character strings in "partitions"

as_percentages	Whether to interpret the entries in amounts_arg as percentages.

Returns

err_t: LSUCCESS or LFAILURE

Repartition a bank account, allocating amounts_arg[i] of money to partition partitions[i]. Set as_percentages to 1 to interpret the elements of amounts_arg as percentages, in which case the sum of the entries of amounts_arg must equal 100. Set as_percentages to 0 otherwise, in which case the sum of the entries of amounts_arg must equal the overall true balance of the given bank account.

4.7.2.21 err_t retotal (Ledger * *ledger*)

Parameters

ledger	Pointer to a Ledger object.

Returns

err_t: LSUCCESS or LFAILURE

Call free_for_retotal, get_names, and get_totals to reflect any recent changes to ledger->entries.

4.7.2.22 err_t sort_by_status (Ledger * *ledger,* int *sort_locked*)

Parameters

ledger	Pointer to a Ledger object.
sort_locked	Set to 1 to send all locked transactions to the top and 0 otherwise.

Returns

err_t: LSUCCESS or LFAILURE

Sorts the rows (transactions) of a Ledger object such that the transactions that have not completely cleared rise to the top of the ledger. This is useful because delayed transactions have status codes that will eventually need to be changed. Calls permute_rows to do the job.

4.7.2.23 err_t strip_ledger (Ledger * *ledger*)

Parameters

ledger	Pointer to a Ledger object.

Returns

err_t: LSUCCESS or LFAILURE

Removes leading and trailing whitespace from every entry of a Ledger object. Specifically, str_strip is called on every character string in the "entries" member array of the Ledger object.

4.7.2.24 err_t swap_rows (Ledger * *ledger,* int *row1,* int *row2*)

Parameters

ledger	Pointer to a Ledger object.
row1	Index of the first row.
row2	Index of the second row.

Returns

err_t: LSUCCESS or LFAILURE

Interchanges two rows of a Ledger object. Rows row1 and row2 of the "entries" member array of the Ledger object are interchanged.

4.7.2.25 err_t trim_ledger (Ledger * *ledger*)

Parameters

ledger	Pointer to a Ledger object.

Returns

err_t: LSUCCESS or LFAILURE

Remove rows (transactions) with transaction amounts of zero. These empty transactions do not actually contribute to the content of the ledger. They are marked for removal and then removed with remove_rows.

4.7.2.26 err_t unlock (Ledger * *ledger*)

Parameters

ledger	Pointer to a Ledger object.

Returns

err_t: LSUCCESS or LFAILURE

Unlocks all cleared transactions. Specifically, this function changes the status of all transactions with status LOCKED to status NIL:

4.8 Ledger Input Functions

Functions for reading Ledger objects.

Functions

- err_t parse_char (Ledger *ledger, char c, int *char_index, int *field, int *row)
  Parse a character while reading a ledger from a file.
- err_t get_entries_from_filename (Ledger *ledger, char *filename)
  Get ledger entries from a file.
- err_t get_entries_from_stream (Ledger *ledger, FILE *fp)
  Get ledger entries from a file stream.
- err_t get_entries_from_string (Ledger *ledger, char *s)
  Get ledger entries from a string.
- err_t get_ledger (Ledger **ledger, char *filename, FILE *fp, char *str)
  Recommended way to read in a Ledger object.

4.8.1 Detailed Description

These are functions for reading Ledger objects from files, file streams, and character strings.

4.8.2 Function Documentation

4.8.2.1 err_t get_entries_from_filename (Ledger * *ledger,* char * *filename*)

Parameters

ledger	Pointer to a Ledger object.
filename	Full path of the file to read from.

Returns

err_t: LSUCCESS or LFAILURE

Read the entries of a ledger from a filename into a Ledger object. This is really a wrapper around get_entries_from_stream. get_entries_from_filename mostly just opens the file and calls get_entries_from_stream.

4.8.2.2 err_t get_entries_from_stream (Ledger * *ledger,* FILE * *fp*)

Parameters

ledger	Pointer to a Ledger object.
fp	FILE pointer to the file stream to read from.

Returns

err_t: LSUCCESS or LFAILURE

Get ledger entries from a file stream. This function first finds out how many rows there are in the input ledger and the iterates over the file stream and parses characters individually with parse_char.

4.8.2.3 err_t get_entries_from_string (Ledger * *ledger,* char * *s*)

Parameters

ledger	Pointer to a Ledger object.
s	Character string to read from.

Returns

err_t: LSUCCESS or LFAILURE

Get ledger entries from a string. This function first finds out how many rows there are in the input ledger and the iterates over the string and parses characters individually with parse_char.

4.8.2.4 err_t get_ledger (Ledger ** *ledger*, char * *filename*, FILE * *fp*, char * *str*)

Parameters

ledger	Pointer to a Ledger object.
filename	Full path of the file to read from.
fp	FILE pointer to the file stream to read from.
str	Character string to read from.

Returns

err_t: LSUCCESS or LFAILURE

This function is the recommended way to read in a Ledger object from some source. It creates a new Ledger object, reads in the entries from the specified source, and then calculates summary data on the ledger entries. To read from a filename, use the filename argument and set fp and str to NULL. To read from a file stream, use fp and set filename and str to NULL. To read from a string, use str and set filename and fp to NULL.

4.8.2.5 err_t parse_char (Ledger * *ledger*, char *c*, int * *char_index*, int * *field*, int * *row*)

Parameters

ledger	Pointer to a Ledger object.
c	The character to parse.
char_index	The index of the character in the current ledger entry.
field	The field (column) of the current ledger entry being read.
row	The row (transaction) of the current ledger entry being read.

Returns

err_t: LSUCCESS or LFAILURE

Parse a character while reading a ledger from a file. Non-separators will be read into the "entries" member array of the Ledger object: specifically, a non-separator character c will be concatenated to the tail of ledger->entries[*field]`*row`. If c is a separating character like a row separator or a column separator, then a new entry will be started: that is, *row and *field will be changed and *char_index will be reset to 0.

4.9 Ledger Output Functions

Functions to output Ledger objects.

Functions

- err_t print_ledger_to_filename (Ledger *ledger, char *filename)

  Print ledger entries to a file.
- err_t print_ledger_to_stream (Ledger *ledger, FILE *fp)

  Print ledger entries to a file stream.
- err_t print_ledger_to_string (Ledger *ledger, char **s)

  Print ledger entries to a string.
- err_t print_ledger_verbose (Ledger *ledger, FILE *fp)

  Print out all the information on a Ledger object to a file stream.

4.9.1 Detailed Description

These are functions to output Ledger objects to files, file streams, and character strings.

4.9.2 Function Documentation

4.9.2.1 err_t print_ledger_to_filename (Ledger * *ledger,* char * *filename*)

Parameters

ledger	Pointer to a Ledger object.
filename	Full path of the file to print to.

Returns

err_t: LSUCCESS or LFAILURE

Print the ledger entries of a Ledger object to a file. This function is a wrapper around print_ledger_to_stream. All it does is safely open the file and then call print_ledger_to_stream.

4.9.2.2 err_t print_ledger_to_stream (Ledger * *ledger,* FILE * *fp*)

Parameters

ledger	Pointer to a Ledger object.
fp	FILE pointer to the file stream to print to.

Returns

err_t: LSUCCESS or LFAILURE

Prints the ledger entries of a Ledger object to a file stream in a format that can be easily read back into a Ledger object by get_ledger.

4.9.2.3 err_t print_ledger_to_string (Ledger * *ledger,* char ** *s*)

Parameters

ledger	Pointer to a Ledger object.
s	Character string to write to.

Returns

err_t: LSUCCESS or LFAILURE

Prints the ledger entries of a Ledger object to a character string in a format that can be easily read back into a Ledger object by get_ledger.

4.9.2.4 err_t print_ledger_verbose (Ledger * *ledger,* FILE * *fp*)

Parameters

ledger	Pointer to a Ledger object.
fp	FILE pointer to the file stream to print to.

Returns

err_t: LSUCCESS or LFAILURE

Print all the information about a Ledger object to a file stream. This function is intended for debugging purposes only. The output is ugly.

4.10 Ledger Summary Functions

Functions to summarize Ledger objects.

Functions

- err_t print_summary_to_filename (Ledger *ledger, char *filename, int usecolor)

  Print out a summary of a Ledger object to a file.
- err_t print_summary_to_stream (Ledger *ledger, FILE *fp, int usecolor)

  Print out a summary of a Ledger object to a file.
- err_t print_summary_to_string (Ledger *ledger, char **s, int usecolor)

  Print out a summary of a Ledger object to a file.

4.10.1 Detailed Description

These are functions to output *summaries* of Ledger objects to files, file streams, and character strings.

4.10.2 Function Documentation

4.10.2.1 err_t print_summary_to_filename (Ledger * *ledger,* char * *filename,* int *usecolor*)

Parameters

ledger	Pointer to a Ledger object.
filename	Full path of the file to print to.
usecolor	Include command line interface color codes?

Returns

err_t: LSUCCESS or LFAILURE

Print a summary of a Ledger object to a file. This function is a wrapper around print_summary_to_stream. All it does is open the file and call print_summary_to_stream. Set usecolor to 1 to print with command line interface color codes defined in the Printing_Macros module. Set usecolor to 0 to not use these color codes.

4.10.2.2 err_t print_summary_to_stream (Ledger * *ledger,* FILE * *fp,* int *usecolor*)

Parameters

ledger	Pointer to a Ledger object.
fp	FILE pointer of file stream to print to.
usecolor	Include command line interface color codes?

Returns

err_t: LSUCCESS or LFAILURE

Print a summary of a Ledger object to a file stream. Set usecolor to 1 to print with command line interface color codes defined in the Printing_Macros module. Set usecolor to 0 to not use these color codes.

4.10.2.3 err_t print_summary_to_string (Ledger * *ledger,* char ** *s,* int *usecolor*)

Parameters

ledger	Pointer to a Ledger object.
s	Character string to print to.
usecolor	Include command line interface color codes?

Returns

err_t: LSUCCESS or LFAILURE

Print a summary of a Ledger object to a file stream. Set usecolor to 1 to print with command line interface color codes defined in the Printing_Macros module. Set usecolor to 0 to not use these color codes.

4.11 Top Level Functions

Top level functions.

Functions

- err_t standalone (int argc, char **argv)

  Top level function of the standalone command line interface version.

4.11.1 Detailed Description

These are functions that govern the program at the top leve. For example, standalone() is the main function of the command line interface version of the program.

4.11.2 Function Documentation

4.11.2.1 err_t standalone (int *argc,* char ** *argv*)

Parameters

argc	Number of arguments to int main.
argv	Arguments to int main.

Returns

err_t: LSUCCESS or LFAILURE

Top level function of the standalone command line interface version of this program. Receives arguments argc and argv directly from int main. If the user calls the program with no arguments, standalone will print the usage information and return. If the user gives one argument, standalone will take the argument as the full path of a ledger file, read it in, and print it in color to the Terminal window. If two arguments are given, standalone will read a ledger from the file whose full path is given by the first argument, condense the ledger, and then write the condensed ledger to a new file whose full path is the second argument.

4.12 Column Indices

Ordering of the columns in the ledger file.

Macros

- #define AMOUNT 0

  Amount index.
- #define STATUS 1

  Status index.
- #define CREDIT 2

  Credit index.
- #define BANK 3

  Bank index.
- #define PARTITION 4

  Partition index.
- #define DESCRIPTION 5

  Description index.

4.12.1 Detailed Description

These numbers define the ordering of the columns Each macro stores a column index. For example, if BANK is 3, then the ledger file should have bank account names in column 3. All column indices must be from 0 to 5 inclusive.

4.12.2 Macro Definition Documentation

4.12.2.1 #define AMOUNT 0

Column index for transaction amounts in the ledger file.

4.12.2.2 #define BANK 3

Column index for bank account names in the ledger file.

4.12.2.3 #define CREDIT 2

Column index for credit account names in the ledger file.

4.12.2.4 #define DESCRIPTION 5

Column index for transaction descriptions in the ledger file.

4.12.2.5 #define PARTITION 4

Column index for bank partition names in the ledger file.

4.12.2.6 #define STATUS 1

Column index for transaction status codes in the ledger file.

4.13 Separator Macros

Row and column separators.

Macros

- #define ROW_SEPARATORS "\n\r"

  Row separators.
- #define COLUMN_SEPARATORS "\t"

  Column separators.

4.13.1 Detailed Description

These macros define which characters are used to separate rows and columns of the ledger file. For example, if COLUMN_SEPARATORS is "," then the program will expect the ledger file to be stored in Comma Separated Values (CSV) format. The row separators should be different from the column separators.

4.13.2 Macro Definition Documentation

4.13.2.1 #define COLUMN_SEPARATORS "\t"

Characters that encode the end of a column in the ledger file

4.13.2.2 #define ROW_SEPARATORS "\n\r"

Characters that encode the end of a row in the ledger file

4.14 Status Macros

Transaction status codes.

Macros

- #define CREDIT_NOT_THERE_YET "cn"

  Made with a credit account, but not arrived online.
- #define CREDIT_PENDING "cp"

  Pending in a credit account.
- #define CREDIT_CHARGED "c"

  Charged to a credit account.
- #define NOT_THERE_YET "n"

  Not yet arrived at the bank.
- #define PENDING "p"

  Pending in a bank.
- #define LOCKED "l"

  Locked status.
- #define REMOVE "REMOVE"

  Pending removal from the ledger.

4.14.1 Detailed Description

Transaction status codes tell the program where each transaction is in time: i.e., whether it's pending in the bank or it has not shown up in the credit account, etc. These macros are the character strings that the user manually enters in the STATUS column of the ledger file for each transaction.

4.14.2 Macro Definition Documentation

4.14.2.1 #define CREDIT_CHARGED "c"

Transactions that show up as "cleared" in a credit account, but for which no payment from a bank has been made.

4.14.2.2 #define CREDIT_NOT_THERE_YET "cn"

Use this status if you make a transaction with a credit account and the transaction hasn't shown up yet on the credit account's website.

4.14.2.3 #define CREDIT_PENDING "cp"

Transactions that show up as "pending" on credit accounts

4.14.2.4 #define LOCKED "l"

Cleared, but protected: that is, if a transaction is locked and has a nonzero amount, then the trim_ledger, condense, and clean functions will not remove it.

4.14.2.5 #define NOT_THERE_YET "n"

The transaction has cleared the credit account (if applicable), but the transaction or associated credit card payment hasn't shown up yet on the bank account's website.

4.14.2.6 #define PENDING "p"

Transactions that show up as "pending" in bank accounts.

4.14.2.7 #define REMOVE "REMOVE"

About to be removed from the ledger by remove_rows.

4.15 Printing Macros

Print formatting macros.

Macros

- #define PRINT_EMPTY_ACCOUNTS 0

  Option to print empty accounts in summaries.
- #define USE_COLOR 1

  Option to use color-coded printing in summaries.
- #define NORMAL_COLOR "\x1B[0m"

  Normal text color code.
- #define NEGATIVE_COLOR "\x1B[31m"

  Negative text color code.
- #define POSITIVE_COLOR "\x1B[32m"

  Positive text color code.
- #define ZERO_COLOR "\x1B[34m"

  Zero text color code.

4.15.1 Detailed Description

These macros define settings for printing summaries of ledgers.

4.15.2 Macro Definition Documentation

4.15.2.1 #define NEGATIVE_COLOR "\x1B[31m"

Terminal color code for negative totals.

4.15.2.2 #define NORMAL_COLOR "\x1B[0m"

Terminal color code for regular text.

4.15.2.3 #define POSITIVE_COLOR "\x1B[32m"

Terminal color code for positive totals.

4.15.2.4 #define PRINT_EMPTY_ACCOUNTS 0

Set to 1 to include empty named accounts in summaries. Set to 0 to ignore accounts with balances of $0.00.

4.15.2.5 #define USE_COLOR 1

Set to 1 to enable color-coded printing to the terminal in summaries. Set to 0 for no coloring.

4.15.2.6 #define ZERO_COLOR "\x1B[34m"

Terminal color code for empty totals.

Chapter 5

Class Documentation

5.1 Ledger Struct Reference

Stores an individual ledger.

`#include <ledger.h>`

Public Attributes

- char * filename
- char ** banks
- char ** credits
- char *** partitions
- char *** entries
- int nrows
- int nbanks
- int ncredits
- int * npartitions
- double ** bank_totals
- double ** credit_totals
- double ** partition_totals

5.1.1 Detailed Description

Stores the spreadsheet associated with the ledger along with important summaries. This is the core data type of the program.

5.1.2 Member Data Documentation

5.1.2.1 double** Ledger::bank_totals

Stores how much money is in each bank account. For each bank, this includes the amount of available money, pending money, etc.

5.1.2.2 char** Ledger::banks

Array of names of all the bank accounts in the ledger.

5.1.2.3 double** Ledger::credit_totals

Stores how much money is in each credit account. For each account, this includes the amount of available money, pending money, etc.

5.1.2.4 char** Ledger::credits

Array of names of all the credit accounts in the ledger.

5.1.2.5 char*** Ledger::entries

The matrix of actual entries in the ledger. Rows are individual transactions, and columns are fields like the amount, status, and bank of the transaction.

5.1.2.6 char* Ledger::filename

Name of the file that the ledger came from.

5.1.2.7 int Ledger::nbanks

Number of bank accounts (including an automatic unnamed bank account).

5.1.2.8 int Ledger::ncredits

Number of credit accounts (including an automatic unnamed credit account).

5.1.2.9 int* Ledger::npartitions

Number of credit accounts (including an automatic unnamed partition).

5.1.2.10 int Ledger::nrows

Number of rows in the ledger: i.e., number of transactions.

5.1.2.11 double** Ledger::partition_totals

Stores how much money will be in each partition of each bank account after all charges clear

5.1.2.12 char*** Ledger::partitions

Arrays of names of all the partitions of all the bank accounts in the ledger.

The documentation for this struct was generated from the following file:

- src/ledger.h

Chapter 6

File Documentation

6.1 src/alloc_entries.c File Reference

```
#include <errno.h>
#include "ledger.h"
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include "user_settings.h"
```

Functions

- err_t alloc_entries (Ledger *ledger)

  Allocates the "entries" member of a Ledger object.

6.1.1 Detailed Description

Author

Will Landau (http://www.will-landau.com/)

Date

2013-2014

Copyright

GNU General Public License 3.0

6.2 src/alloc_totals.c File Reference

```
#include <errno.h>
#include "ledger.h"
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include "user_settings.h"
```

Functions

- err_t alloc_totals (Ledger *ledger)

  Allocates space for the numerical summaries of the ledger.

6.2.1 Detailed Description

Author

Will Landau (http://www.will-landau.com/)

Date

2013-2014

Copyright

GNU General Public License 3.0

6.3 src/clean.c File Reference

```
#include <errno.h>
#include "ledger.h"
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include "user_settings.h"
```

Functions

- err_t clean (Ledger *ledger, int sort_locked)

  Cleans a ledger object.

6.3.1 Detailed Description

Author

Will Landau (http://www.will-landau.com/)

Date

2013-2014

Copyright

GNU General Public License 3.0

6.4 src/col_delim_char.c File Reference

```
#include <errno.h>
#include "ledger.h"
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include "user_settings.h"
```

Functions

- bool_t col_delim_char (char c)

  Checks whether a character is a column separator.

6.4.1 Detailed Description

Author

Will Landau (http://www.will-landau.com/)

Date

2013-2014

Copyright

GNU General Public License 3.0

6.5 src/col_delim_str.c File Reference

```
#include <errno.h>
#include "ledger.h"
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include "user_settings.h"
```

Functions

- index_t col_delim_str (char *s)

  Finds the first column delimiter in a string.

6.5.1 Detailed Description

Author

Will Landau (http://www.will-landau.com/)

Date

2013-2014

Copyright

GNU General Public License 3.0

6.6 src/color.c File Reference

```
#include <errno.h>
#include "ledger.h"
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include "user_settings.h"
```

Functions

- color_t color (double d, int usecolor)

  Finds the correct color code for an amount in summary output.

6.6.1 Detailed Description

Author

Will Landau (http://www.will-landau.com/)

Date

2013-2014

Copyright

GNU General Public License 3.0

6.7 src/condense.c File Reference

```
#include <errno.h>
#include "ledger.h"
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include "user_settings.h"
```

Functions

- err_t condense (Ledger *ledger)

  Condenses a ledger object.

6.7.1 Detailed Description

Author

Will Landau (http://www.will-landau.com/)

Date

2013-2014

Copyright

GNU General Public License 3.0

6.8 src/copy_ledger.c File Reference

```
#include <errno.h>
#include "ledger.h"
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include "user_settings.h"
```

Functions

- err_t copy_ledger (Ledger **out_ledger, Ledger *in_ledger)

  Copies one Ledger object into another.

6.8.1 Detailed Description

Author

Will Landau (http://www.will-landau.com/)

Date

2013-2014

Copyright

GNU General Public License 3.0

6.9 src/copy_rows.c File Reference

```
#include <errno.h>
#include "ledger.h"
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include "user_settings.h"
```

Functions

- err_t copy_rows (Ledger *ledger, Ledger **clipboard, int *rows, int howmany)

  Copies selected rows (transactions).

6.9.1 Detailed Description

Author

Will Landau (http://www.will-landau.com/)

Date

2013-2014

Copyright

GNU General Public License 3.0

6.10 src/cut_rows.c File Reference

```
#include <errno.h>
#include "ledger.h"
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include "user_settings.h"
```

Functions

- err_t cut_rows (Ledger *ledger, Ledger **clipboard, int *rows, int howmany)

  Cut selected rows (transactions).

6.10.1 Detailed Description

Author

Will Landau (http://www.will-landau.com/)

Date

2013-2014

Copyright

GNU General Public License 3.0

6.11 src/edit_entry.c File Reference

```
#include <errno.h>
#include "ledger.h"
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include "user_settings.h"
```

Functions

- err_t edit_entry (Ledger *ledger, char *entry, int row, int field, int append)
  Modify a ledger entry and update the Ledger object accordingly.

6.11.1 Detailed Description

Author

Will Landau (http://www.will-landau.com/)

Date

2013-2014

Copyright

GNU General Public License 3.0

6.12 src/edit_entry_noretotal.c File Reference

```
#include <errno.h>
#include "ledger.h"
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include "user_settings.h"
```

Functions

- err_t edit_entry_noretotal (Ledger *ledger, char *entry, int row, int field, int append)
  Modify a ledger entry and DO NOT update the Ledger object accordingly.

6.12.1 Detailed Description

Author

Will Landau (http://www.will-landau.com/)

Date

2013-2014

Copyright

GNU General Public License 3.0

6.13 src/edit_row.c File Reference

```
#include <errno.h>
#include "ledger.h"
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include "user_settings.h"
```

Functions

- err_t edit_row (Ledger *ledger, char **entries, int row, int append)

  Modify a whole row (transaction) and update the Ledger object accordingly.

6.13.1 Detailed Description

Author

Will Landau (http://www.will-landau.com/)

Date

2013-2014

Copyright

GNU General Public License 3.0

6.14 src/filled_partitions.c File Reference

```
#include <errno.h>
#include "ledger.h"
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include "user_settings.h"
```

Functions

- bool_t filled_partitions (Ledger *ledger, int bank)

  Tests if any of the partitions in a given bank have nonzero balance.

6.14.1 Detailed Description

Author

Will Landau (http://www.will-landau.com/)

Date

2013-2014

Copyright

GNU General Public License 3.0

6.15 src/free_entries.c File Reference

```
#include <errno.h>
#include "ledger.h"
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include "user_settings.h"
```

Functions

- err_t free_entries (Ledger *ledger)

  Frees the "entries" member of a Ledger object.

6.15.1 Detailed Description

Author

Will Landau (http://www.will-landau.com/)

Date

2013-2014

Copyright

GNU General Public License 3.0

6.16 src/free_for_retotal.c File Reference

```
#include <errno.h>
#include "ledger.h"
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include "user_settings.h"
```

Functions

- err_t free_for_retotal (Ledger *ledger)

  Frees the account names and numerical summaries of a Ledger object.

6.16.1 Detailed Description

Author

Will Landau (http://www.will-landau.com/)

Date

2013-2014

Copyright

GNU General Public License 3.0

6.17 src/free_ledger.c File Reference

```
#include <errno.h>
#include "ledger.h"
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include "user_settings.h"
```

Functions

- err_t free_ledger (Ledger **ledger)

  Frees a whole Ledger object.

6.17.1 Detailed Description

Author

Will Landau (http://www.will-landau.com/)

Date

2013-2014

Copyright

GNU General Public License 3.0

6.18 src/get_entries_from_filename.c File Reference

```
#include <errno.h>
#include "ledger.h"
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include "user_settings.h"
```

Functions

- err_t get_entries_from_filename (Ledger *ledger, char *filename)
  Get ledger entries from a file.

6.18.1 Detailed Description

Author

Will Landau (http://www.will-landau.com/)

Date

2013-2014

Copyright

GNU General Public License 3.0

6.19 src/get_entries_from_stream.c File Reference

```
#include <errno.h>
#include "ledger.h"
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include "user_settings.h"
```

Functions

- err_t get_entries_from_stream (Ledger *ledger, FILE *fp)
  Get ledger entries from a file stream.

6.19.1 Detailed Description

Author

Will Landau (http://www.will-landau.com/)

Date

2013-2014

Copyright

GNU General Public License 3.0

6.20 src/get_entries_from_string.c File Reference

```
#include <errno.h>
#include "ledger.h"
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include "user_settings.h"
```

Functions

- err_t get_entries_from_string (Ledger *ledger, char *s)

  Get ledger entries from a string.

6.20.1 Detailed Description

Author

Will Landau (http://www.will-landau.com/)

Date

2013-2014

Copyright

GNU General Public License 3.0

6.21 src/get_ledger.c File Reference

```
#include <errno.h>
#include "ledger.h"
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include "user_settings.h"
```

Functions

- err_t get_ledger (Ledger **ledger, char *filename, FILE *fp, char *str)

  Recommended way to read in a Ledger object.

6.21.1 Detailed Description

Author

Will Landau (http://www.will-landau.com/)

Date

2013-2014

Copyright

GNU General Public License 3.0

6.22 src/get_names.c File Reference

```
#include <errno.h>
#include "ledger.h"
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include "user_settings.h"
```

Functions

- err_t get_names (Ledger *ledger)

  Gets the account names of a Ledger object.

6.22.1 Detailed Description

Author

Will Landau (http://www.will-landau.com/)

Date

2013-2014

Copyright

GNU General Public License 3.0

6.23 src/get_totals.c File Reference

```
#include <errno.h>
#include "ledger.h"
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include "user_settings.h"
```

Functions

- err_t get_totals (Ledger *ledger)

  Computes numerical summaries of a Ledger object.

6.23.1 Detailed Description

Author

Will Landau (http://www.will-landau.com/)

Date

2013-2014

Copyright

GNU General Public License 3.0

6.24 src/input_file.c File Reference

```
#include <errno.h>
#include "ledger.h"
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include "user_settings.h"
```

Functions

- bool_t input_file (char *filename)

  Tests if the given input file is usable.

6.24.1 Detailed Description

Author

Will Landau (http://www.will-landau.com/)

Date

2013-2014

Copyright

GNU General Public License 3.0

6.25 src/insert_filled_rows.c File Reference

```
#include <errno.h>
#include "ledger.h"
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include "user_settings.h"
```

Functions

- err_t insert_filled_rows (Ledger *ledger, char **entries, int row, int howmany)
  Insert filled rows.

6.25.1 Detailed Description

Author

Will Landau (http://www.will-landau.com/)

Date

2013-2014

Copyright

GNU General Public License 3.0

6.26 src/insert_rows.c File Reference

```
#include <errno.h>
#include "ledger.h"
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include "user_settings.h"
```

Functions

- err_t insert_rows (Ledger *ledger, int row, int howmany)
  Insert blank rows.

6.26.1 Detailed Description

Author

Will Landau (http://www.will-landau.com/)

Date

2013-2014

Copyright

GNU General Public License 3.0

6.27 src/ledger.h File Reference

Main header file.

```
#include <errno.h>
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include "user_settings.h"
```

Classes

- struct Ledger

  Stores an individual ledger.

Macros

- #define I_NOT_THERE_YET 0
- #define I_PENDING 1
- #define I_CLEARED 2
- #define I_PENDING_BAL 3
- #define I_OVERALL_BAL 4
- #define N_TOTALS 5
- #define LSUCCESS 0

  Success return code.
- #define LFAILURE 1

  Failure return code.
- #define LNO 0

  "No" return code.
- #define LYES 1

  "Yes" return code.
- #define NO_INDEX -1
- #define ENTRYSIZE 256

  Maximum entry size.
- #define EPS 0.000025

  "Epsilon"
- #define FILENAMESIZE 256

  File name size.
- #define LINESIZE 4096

  Line size.
- #define NFIELDS 6

  Number of fields (columns).
- #define NIL "\0"

  The empty string.

Typedefs

- typedef int err_t

  Error status type.
- typedef int bool_t

  Boolean type.
- typedef int index_t

  Array index type.
- typedef char * color_t

  Color code type.

Functions

- bool_t col_delim_char (char c)

  Checks whether a character is a column separator.
- bool_t filled_partitions (Ledger *ledger, int bank)

  Tests if any of the partitions in a given bank have nonzero balance.
- bool_t input_file (char *filename)

  Tests if the given input file is usable.
- bool_t legal_double (char *s)

  Checks if a character string can be converted into a meaningful floating point number.
- bool_t legal_amounts (Ledger *ledger)

  Checks if a the AMOUNTS column in the ledger stores strings that can be converted into meaningful floating point numbers.
- bool_t legal_status_code (char *s)

  Checks if a character string is one of the legal status codes defined in the Status_Macros module.
- bool_t legal_status_codes (Ledger *ledger)

  Checks if all the character strings in the STATUS column of the ledger are legal status codes as defined in the Status_Macros module.
- bool_t locked (char *status)

  Checks if a status code "locks" a transaction.
- bool_t output_file (char *filename)

  Tests if the given output file is usable.
- bool_t row_delim_char (char c)

  Checks whether a character is a row separator.
- bool_t small_norm (double d)

  Checks whether a floating point number rounds to 0.00.
- bool_t space (char c)

  Checks whether a character is a whitespace character.
- bool_t str_equal (const char *s1, const char *s2)

  Checks whether two strings are equal.
- bool_t untotaled (Ledger *ledger)

  Checks if account totals have been calculated for a Ledger object.
- color_t color (double d, int usecolor)

  Finds the correct color code for an amount in summary output.
- index_t col_delim_str (char *s)

  Finds the first column delimiter in a string.
- index_t row_delim_str (char *s)

  Finds the first row delimiter in a string.
- index_t which (char **s, char *find, int n)

  Finds an occurrence of "find" in "s".

- index_t which_bank_total (char *status)

  Given a status code, finds the correct index in the bank_totals array member of the Ledger type.
- index_t which_credit_total (char *status)

  Given a status code, finds the correct index in the credit_totals array member of the Ledger type.
- int qcmp (const void *a, const void *b)

  Comparison function for character strings in qsort.
- err_t str_strip (char *s)

  Strips leading and trailing whitespace from a character string.
- err_t unique (char **a, int n, char ***ret, int *nunique)

  Finds all the unique elements in an array of character strings.
- err_t usage ()

  Prints usage details.
- err_t alloc_entries (Ledger *ledger)

  Allocates the "entries" member of a Ledger object.
- err_t alloc_totals (Ledger *ledger)

  Allocates space for the numerical summaries of the ledger.
- err_t free_entries (Ledger *ledger)

  Frees the "entries" member of a Ledger object.
- err_t free_for_retotal (Ledger *ledger)

  Frees the account names and numerical summaries of a Ledger object.
- err_t free_ledger (Ledger **ledger)

  Frees a whole Ledger object.
- err_t get_names (Ledger *ledger)

  Gets the account names of a Ledger object.
- err_t get_totals (Ledger *ledger)

  Computes numerical summaries of a Ledger object.
- err_t new_ledger (Ledger **ledger)

  Creates a new Ledger object.
- err_t copy_ledger (Ledger **out_ledger, Ledger *in_ledger)

  Copies one Ledger object into another.
- err_t clean (Ledger *ledger, int sort_locked)

  Cleans a ledger object.
- err_t condense (Ledger *ledger)

  Condenses a ledger object.
- err_t copy_rows (Ledger *ledger, Ledger **clipboard, int *rows, int howmany)

  Copies selected rows (transactions).
- err_t cut_rows (Ledger *ledger, Ledger **clipboard, int *rows, int howmany)

  Cut selected rows (transactions).
- err_t edit_entry (Ledger *ledger, char *entry, int row, int field, int append)

  Modify a ledger entry and update the Ledger object accordingly.
- err_t edit_entry_noretotal (Ledger *ledger, char *entry, int row, int field, int append)

  Modify a ledger entry and DO NOT update the Ledger object accordingly.
- err_t edit_row (Ledger *ledger, char **entries, int row, int append)

  Modify a whole row (transaction) and update the Ledger object accordingly.
- err_t map_to_coords (Ledger *ledger, char *entry, int *rows, int *fields, int howmany, int append)

  Map a character string multiple entries.
- err_t map_to_coords_colmajor (Ledger *ledger, char *entry, int **coords, int howmany, int append)

  Map a character string multiple entries.
- err_t map_to_coords_rowmajor (Ledger *ledger, char *entry, int **coords, int howmany, int append)

  Map a character string multiple entries.
- err_t insert_rows (Ledger *ledger, int row, int howmany)

Insert blank rows.

- err_t insert_filled_rows (Ledger *ledger, char **entries, int row, int howmany)

  Insert filled rows.
- err_t move_rows (Ledger *ledger, int *rows, int nrows, int moveto)

  Move rows to a specified location.
- err_t paste_rows (Ledger *ledger, Ledger *clipboard, int where)

  Paste rows into a specified location.
- err_t permute_rows (Ledger *ledger, int *order)

  Permute rows (transactions).
- err_t rename_bank (Ledger *ledger, char *from, char *to)

  Safely renames a bank account.
- err_t rename_credit (Ledger *ledger, char *from, char *to)

  Safely renames a credit account.
- err_t rename_partition (Ledger *ledger, char *bank, char *from, char *to)

  Safely renames a partition bank account.
- err_t remove_rows (Ledger *ledger)

  Removes rows (transactions) marked for removal.
- err_t retotal (Ledger *ledger)

  Recalculate names and totals.
- err_t trim_ledger (Ledger *ledger)

  Removes blank rows.
- err_t sort_by_status (Ledger *ledger, int sort_locked)

  Sorts rows (transactions) by transaction status code.
- err_t strip_ledger (Ledger *ledger)

  Removes whitespace from the entries of a Ledger object.
- err_t swap_rows (Ledger *ledger, int row1, int row2)

  Interchanges two rows of a ledger object.
- err_t repartition (Ledger *ledger, char *bank, char **partitions, double *amounts_arg, int npartitions, int as_percentages)

  Repartitions a bank account.
- err_t unlock (Ledger *ledger)

  Unlock all cleared transactions.
- err_t parse_char (Ledger *ledger, char c, int *char_index, int *field, int *row)

  Parse a character while reading a ledger from a file.
- err_t get_entries_from_filename (Ledger *ledger, char *filename)

  Get ledger entries from a file.
- err_t get_entries_from_stream (Ledger *ledger, FILE *fp)

  Get ledger entries from a file stream.
- err_t get_entries_from_string (Ledger *ledger, char *s)

  Get ledger entries from a string.
- err_t get_ledger (Ledger **ledger, char *filename, FILE *fp, char *str)

  Recommended way to read in a Ledger object.
- err_t print_ledger_to_filename (Ledger *ledger, char *filename)

  Print ledger entries to a file.
- err_t print_ledger_to_stream (Ledger *ledger, FILE *fp)

  Print ledger entries to a file stream.
- err_t print_ledger_to_string (Ledger *ledger, char **s)

  Print ledger entries to a string.
- err_t print_ledger_verbose (Ledger *ledger, FILE *fp)

  Print out all the information on a Ledger object to a file stream.
- err_t print_summary_to_filename (Ledger *ledger, char *filename, int usecolor)

Print out a summary of a Ledger object to a file.

- err_t print_summary_to_stream (Ledger *ledger, FILE *fp, int usecolor)

  Print out a summary of a Ledger object to a file.

- err_t print_summary_to_string (Ledger *ledger, char **s, int usecolor)

  Print out a summary of a Ledger object to a file.

- err_t standalone (int argc, char **argv)

  Top level function of the standalone command line interface version.

6.27.1 Detailed Description

Author

Will Landau (http://www.will-landau.com/)

Date

2013-2014

Copyright

GNU General Public License 3.0

Includes all typedefs and function declarations, along with some of the macros.

6.28 src/legal_amounts.c File Reference

```
#include <errno.h>
#include "ledger.h"
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include "user_settings.h"
```

Functions

- bool_t legal_amounts (Ledger *ledger)

  Checks if a the AMOUNTS column in the ledger stores strings that can be converted into meaningful floating point numbers.

6.28.1 Detailed Description

Author

Will Landau (http://www.will-landau.com/)

Date

2013-2014

Copyright

GNU General Public License 3.0

6.29 src/legal_double.c File Reference

```
#include <errno.h>
#include "ledger.h"
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include "user_settings.h"
```

Functions

- bool_t legal_double (char *s)

  Checks if a character string can be converted into a meaningful floating point number.

6.29.1 Detailed Description

Author

Will Landau (http://www.will-landau.com/)

Date

2013-2014

Copyright

GNU General Public License 3.0

6.30 src/legal_status_code.c File Reference

```
#include <errno.h>
#include "ledger.h"
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include "user_settings.h"
```

Functions

- bool_t legal_status_code (char *s)

  Checks if a character string is one of the legal status codes defined in the Status_Macros module.

6.30.1 Detailed Description

Author

Will Landau (http://www.will-landau.com/)

Date

2013-2014

Copyright

GNU General Public License 3.0

6.31 src/legal_status_codes.c File Reference

```
#include <errno.h>
#include "ledger.h"
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include "user_settings.h"
```

Functions

- bool_t legal_status_codes (Ledger *ledger)

  Checks if all the character strings in the STATUS column of the ledger are legal status codes as defined in the Status_Macros module.

6.31.1 Detailed Description

Author

Will Landau (http://www.will-landau.com/)

Date

2013-2014

Copyright

GNU General Public License 3.0

6.32 src/locked.c File Reference

```
#include <errno.h>
#include "ledger.h"
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include "user_settings.h"
```

Functions

- bool_t locked (char *status)

  Checks if a status code "locks" a transaction.

6.32.1 Detailed Description

Author

Will Landau (http://www.will-landau.com/)

Date

2013-2014

Copyright

GNU General Public License 3.0

6.33 src/main.c File Reference

```
#include <errno.h>
#include "ledger.h"
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include "user_settings.h"
```

Functions

- int main (int argc, char **argv)

6.33.1 Detailed Description

Author

Will Landau (http://www.will-landau.com/)

Date

2013-2014

Copyright

GNU General Public License 3.0

6.33.2 Function Documentation

6.33.2.1 int main (int *argc*, char ** *argv*)

Main function of the standalone command line interface version. Just calls standalone and returns.

6.34 src/map_to_coords.c File Reference

```
#include <errno.h>
#include "ledger.h"
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include "user_settings.h"
```

Functions

- err_t map_to_coords (Ledger *ledger, char *entry, int *rows, int *fields, int howmany, int append)
  Map a character string multiple entries.

6.34.1 Detailed Description

Author

Will Landau (http://www.will-landau.com/)

Date

2013-2014

Copyright

GNU General Public License 3.0

6.35 src/map_to_coords_colmajor.c File Reference

```
#include <errno.h>
#include "ledger.h"
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include "user_settings.h"
```

Functions

- err_t map_to_coords_colmajor (Ledger *ledger, char *entry, int **coords, int howmany, int append)
  Map a character string multiple entries.

6.35.1 Detailed Description

Author

Will Landau (http://www.will-landau.com/)

Date

2013-2014

Copyright

GNU General Public License 3.0

6.36 src/map_to_coords_rowmajor.c File Reference

```
#include <errno.h>
#include "ledger.h"
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include "user_settings.h"
```

Functions

- err_t map_to_coords_rowmajor (Ledger *ledger, char *entry, int **coords, int howmany, int append)

  Map a character string multiple entries.

6.36.1 Detailed Description

Author

Will Landau (http://www.will-landau.com/)

Date

2013-2014

Copyright

GNU General Public License 3.0

6.37 src/move_rows.c File Reference

```
#include <errno.h>
#include "ledger.h"
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include "user_settings.h"
```

Functions

- err_t move_rows (Ledger *ledger, int *rows, int nrows, int moveto)

  Move rows to a specified location.

6.37.1 Detailed Description

Author

Will Landau (http://www.will-landau.com/)

Date

2013-2014

Copyright

GNU General Public License 3.0

6.38 src/new_ledger.c File Reference

```
#include <errno.h>
#include "ledger.h"
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include "user_settings.h"
```

Functions

- err_t new_ledger (Ledger **ledger)

  Creates a new Ledger object.

6.38.1 Detailed Description

Author

Will Landau (http://www.will-landau.com/)

Date

2013-2014

Copyright

GNU General Public License 3.0

6.39 src/output_file.c File Reference

```
#include <errno.h>
#include "ledger.h"
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include "user_settings.h"
```

Functions

- bool_t output_file (char *filename)

  Tests if the given output file is usable.

6.39.1 Detailed Description

Author

Will Landau (http://www.will-landau.com/)

Date

2013-2014

Copyright

GNU General Public License 3.0

6.40 src/parse_char.c File Reference

```
#include <errno.h>
#include "ledger.h"
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include "user_settings.h"
```

Functions

- err_t parse_char (Ledger *ledger, char c, int *char_index, int *field, int *row)

  Parse a character while reading a ledger from a file.

6.40.1 Detailed Description

Author

Will Landau (http://www.will-landau.com/)

Date

2013-2014

Copyright

GNU General Public License 3.0

6.41 src/paste_rows.c File Reference

```
#include <errno.h>
#include "ledger.h"
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include "user_settings.h"
```

Functions

- err_t paste_rows (Ledger *ledger, Ledger *clipboard, int where)
  Paste rows into a specified location.

6.41.1 Detailed Description

Author

Will Landau (http://www.will-landau.com/)

Date

2013-2014

Copyright

GNU General Public License 3.0

6.42 src/permute_rows.c File Reference

```
#include <errno.h>
#include "ledger.h"
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include "user_settings.h"
```

Functions

- err_t permute_rows (Ledger *ledger, int *order)
  Permute rows (transactions).

6.42.1 Detailed Description

Author

Will Landau (http://www.will-landau.com/)

Date

2013-2014

Copyright

GNU General Public License 3.0

6.43 src/print_ledger_to_filename.c File Reference

```
#include <errno.h>
#include "ledger.h"
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include "user_settings.h"
```

Functions

- err_t print_ledger_to_filename (Ledger *ledger, char *filename)

  Print ledger entries to a file.

6.43.1 Detailed Description

Author

Will Landau (http://www.will-landau.com/)

Date

2013-2014

Copyright

GNU General Public License 3.0

6.44 src/print_ledger_to_stream.c File Reference

```
#include <errno.h>
#include "ledger.h"
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include "user_settings.h"
```

Functions

- err_t print_ledger_to_stream (Ledger *ledger, FILE *fp)

  Print ledger entries to a file stream.

6.44.1 Detailed Description

Author

Will Landau (http://www.will-landau.com/)

Date

2013-2014

Copyright

GNU General Public License 3.0

6.45 src/print_ledger_to_string.c File Reference

```
#include <errno.h>
#include "ledger.h"
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include "user_settings.h"
```

Functions

- err_t print_ledger_to_string (Ledger *ledger, char **s)

  Print ledger entries to a string.

6.45.1 Detailed Description

Author

Will Landau (http://www.will-landau.com/)

Date

2013-2014

Copyright

GNU General Public License 3.0

6.46 src/print_ledger_verbose.c File Reference

```
#include <errno.h>
#include "ledger.h"
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include "user_settings.h"
```

Functions

- err_t print_ledger_verbose (Ledger *ledger, FILE *fp)

  Print out all the information on a Ledger object to a file stream.

6.46.1 Detailed Description

Author

Will Landau (http://www.will-landau.com/)

Date

2013-2014

Copyright

GNU General Public License 3.0

6.47 src/print_summary_to_filename.c File Reference

```
#include <errno.h>
#include "ledger.h"
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include "user_settings.h"
```

Functions

- err_t print_summary_to_filename (Ledger *ledger, char *filename, int usecolor)

  Print out a summary of a Ledger object to a file.

6.47.1 Detailed Description

Author

Will Landau (http://www.will-landau.com/)

Date

2013-2014

Copyright

GNU General Public License 3.0

6.48 src/print_summary_to_stream.c File Reference

```
#include <errno.h>
#include "ledger.h"
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include "user_settings.h"
```

Functions

- err_t print_summary_to_stream (Ledger *ledger, FILE *fp, int usecolor)
  Print out a summary of a Ledger object to a file.

6.48.1 Detailed Description

Author

Will Landau (http://www.will-landau.com/)

Date

2013-2014

Copyright

GNU General Public License 3.0

6.49 src/print_summary_to_string.c File Reference

```
#include <errno.h>
#include "ledger.h"
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include "user_settings.h"
```

Functions

- err_t print_summary_to_string (Ledger *ledger, char **s, int usecolor)
  Print out a summary of a Ledger object to a file.

6.49.1 Detailed Description

Author

Will Landau (http://www.will-landau.com/)

Date

2013-2014

Copyright

GNU General Public License 3.0

6.50 src/qcmp.c File Reference

```
#include <errno.h>
#include "ledger.h"
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include "user_settings.h"
```

Functions

- int qcmp (const void *a, const void *b)

  Comparison function for character strings in qsort.

6.50.1 Detailed Description

Author

Will Landau (http://www.will-landau.com/)

Date

2013-2014

Copyright

GNU General Public License >= 3.0 (See COPYING.txt)

6.51 src/remove_rows.c File Reference

```
#include <errno.h>
#include "ledger.h"
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include "user_settings.h"
```

Functions

- err_t remove_rows (Ledger *ledger)

  Removes rows (transactions) marked for removal.

6.51.1 Detailed Description

Author

Will Landau (http://www.will-landau.com/)

Date

2013-2014

Copyright

GNU General Public License 3.0

6.52 src/rename_bank.c File Reference

```
#include <errno.h>
#include "ledger.h"
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include "user_settings.h"
```

Functions

- err_t rename_bank (Ledger *ledger, char *from, char *to)

  Safely renames a bank account.

6.52.1 Detailed Description

Author

Will Landau (http://www.will-landau.com/)

Date

2013-2014

Copyright

GNU General Public License 3.0

6.53 src/rename_credit.c File Reference

```
#include <errno.h>
#include "ledger.h"
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include "user_settings.h"
```

Functions

- err_t rename_credit (Ledger *ledger, char *from, char *to)

  Safely renames a credit account.

6.53.1 Detailed Description

Author

Will Landau (http://www.will-landau.com/)

Date

2013-2014

Copyright

GNU General Public License 3.0

6.54 src/rename_partition.c File Reference

```
#include <errno.h>
#include "ledger.h"
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include "user_settings.h"
```

Functions

- err_t rename_partition (Ledger *ledger, char *bank, char *from, char *to)

  Safely renames a partition bank account.

6.54.1 Detailed Description

Author

Will Landau (http://www.will-landau.com/)

Date

2013-2014

Copyright

GNU General Public License 3.0

6.55 src/repartition.c File Reference

```
#include <errno.h>
#include "ledger.h"
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include "user_settings.h"
```

Functions

- err_t repartition (Ledger *ledger, char *bank, char **partitions, double *amounts_arg, int npartitions, int as_percentages)

  Repartitions a bank account.

6.55.1 Detailed Description

Author

Will Landau (http://www.will-landau.com/)

Date

2013-2014

Copyright

GNU General Public License 3.0

6.56 src/retotal.c File Reference

```
#include <errno.h>
#include "ledger.h"
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include "user_settings.h"
```

Functions

- err_t retotal (Ledger *ledger)

  Recalculate names and totals.

6.56.1 Detailed Description

Author

Will Landau (http://www.will-landau.com/)

Date

2013-2014

Copyright

GNU General Public License 3.0

6.57 src/row_delim_char.c File Reference

```
#include <errno.h>
#include "ledger.h"
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include "user_settings.h"
```

Functions

- bool_t row_delim_char (char c)

  Checks whether a character is a row separator.

6.57.1 Detailed Description

Author

Will Landau (http://www.will-landau.com/)

Date

2013-2014

Copyright

GNU General Public License 3.0

6.58 src/row_delim_str.c File Reference

```
#include <errno.h>
#include "ledger.h"
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include "user_settings.h"
```

Functions

- index_t row_delim_str (char *s)

  Finds the first row delimiter in a string.

6.58.1 Detailed Description

Author

Will Landau (http://www.will-landau.com/)

Date

2013-2014

Copyright

GNU General Public License 3.0

6.59 src/small_norm.c File Reference

```
#include <errno.h>
#include "ledger.h"
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include "user_settings.h"
```

Functions

- bool_t small_norm (double d)

  Checks whether a floating point number rounds to 0.00.

6.59.1 Detailed Description

Author

Will Landau (http://www.will-landau.com/)

Date

2013-2014

Copyright

GNU General Public License 3.0

6.60 src/sort_by_status.c File Reference

```
#include <errno.h>
#include "ledger.h"
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include "user_settings.h"
```

Functions

- err_t sort_by_status (Ledger *ledger, int sort_locked)

  Sorts rows (transactions) by transaction status code.

6.60.1 Detailed Description

Author

Will Landau (http://www.will-landau.com/)

Date

2013-2014

Copyright

GNU General Public License 3.0

6.61 src/space.c File Reference

```
#include <errno.h>
#include "ledger.h"
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include "user_settings.h"
```

Functions

- bool_t space (char c)

  Checks whether a character is a whitespace character.

6.61.1 Detailed Description

Author

Will Landau (http://www.will-landau.com/)

Date

2013-2014

Copyright

GNU General Public License 3.0

6.62 src/standalone.c File Reference

```
#include <errno.h>
#include "ledger.h"
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include "user_settings.h"
```

Functions

- err_t standalone (int argc, char **argv)

  Top level function of the standalone command line interface version.

6.62.1 Detailed Description

Author

Will Landau (http://www.will-landau.com/)

Date

2013-2014

Copyright

GNU General Public License 3.0

6.63 src/str_equal.c File Reference

```
#include <errno.h>
#include "ledger.h"
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include "user_settings.h"
```

Functions

- bool_t str_equal (const char *s1, const char *s2)

  Checks whether two strings are equal.

6.63.1 Detailed Description

Author

Will Landau (http://www.will-landau.com/)

Date

2013-2014

Copyright

GNU General Public License 3.0

6.64 src/str_strip.c File Reference

```
#include <errno.h>
#include "ledger.h"
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include "user_settings.h"
```

Functions

- err_t str_strip (char *s)

  Strips leading and trailing whitespace from a character string.

6.64.1 Detailed Description

Author

Will Landau (http://www.will-landau.com/)

Date

2013-2014

Copyright

GNU General Public License 3.0

6.65 src/strip_ledger.c File Reference

```
#include <errno.h>
#include "ledger.h"
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include "user_settings.h"
```

Functions

- err_t strip_ledger (Ledger *ledger)

  Removes whitespace from the entries of a Ledger object.

6.65.1 Detailed Description

Author

Will Landau (http://www.will-landau.com/)

Date

2013-2014

Copyright

GNU General Public License 3.0

6.66 src/swap_rows.c File Reference

```
#include <errno.h>
#include "ledger.h"
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include "user_settings.h"
```

Functions

- err_t swap_rows (Ledger *ledger, int row1, int row2)

  Interchanges two rows of a ledger object.

6.66.1 Detailed Description

Author

Will Landau (http://www.will-landau.com/)

Date

2013-2014

Copyright

GNU General Public License 3.0

6.67 src/trim_ledger.c File Reference

```
#include <errno.h>
#include "ledger.h"
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include "user_settings.h"
```

Functions

- err_t trim_ledger (Ledger *ledger)

  Removes blank rows.

6.67.1 Detailed Description

Author

Will Landau (http://www.will-landau.com/)

Date

2013-2014

Copyright

GNU General Public License 3.0

6.68 src/unique.c File Reference

```
#include <errno.h>
#include "ledger.h"
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include "user_settings.h"
```

Functions

- err_t unique (char **a, int n, char ***ret, int *nunique)

  Finds all the unique elements in an array of character strings.

6.68.1 Detailed Description

Author

Will Landau (http://www.will-landau.com/)

Date

2013-2014

Copyright

GNU General Public License 3.0

6.69 src/unlock.c File Reference

```
#include <errno.h>
#include "ledger.h"
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include "user_settings.h"
```

Functions

- err_t unlock (Ledger *ledger)
  Unlock all cleared transactions.

6.69.1 Detailed Description

Author

Will Landau (http://www.will-landau.com/)

Date

2013-2014

Copyright

GNU General Public License 3.0

6.70 src/untotaled.c File Reference

```
#include <errno.h>
#include "ledger.h"
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include "user_settings.h"
```

Functions

- bool_t untotaled (Ledger *ledger)
  Checks if account totals have been calculated for a Ledger object.

6.70.1 Detailed Description

Author

Will Landau (http://www.will-landau.com/)

Date

2013-2014

Copyright

GNU General Public License 3.0

6.71 src/usage.c File Reference

```
#include <errno.h>
#include "ledger.h"
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include "user_settings.h"
```

Functions

- err_t usage ()
  Prints usage details.

6.71.1 Detailed Description

Author

Will Landau (http://www.will-landau.com/)

Date

2013-2014

Copyright

GNU General Public License 3.0

6.72 src/user_settings.h File Reference

Header file for user settings.

Macros

- #define AMOUNT 0
  Amount index.
- #define STATUS 1
  Status index.
- #define CREDIT 2
  Credit index.
- #define BANK 3
  Bank index.
- #define PARTITION 4
  Partition index.
- #define DESCRIPTION 5
  Description index.

- #define ROW_SEPARATORS "\n\r"

  Row separators.
- #define COLUMN_SEPARATORS "\t"

  Column separators.
- #define CREDIT_NOT_THERE_YET "cn"

  Made with a credit account, but not arrived online.
- #define CREDIT_PENDING "cp"

  Pending in a credit account.
- #define CREDIT_CHARGED "c"

  Charged to a credit account.
- #define NOT_THERE_YET "n"

  Not yet arrived at the bank.
- #define PENDING "p"

  Pending in a bank.
- #define LOCKED "l"

  Locked status.
- #define REMOVE "REMOVE"

  Pending removal from the ledger.
- #define PRINT_EMPTY_ACCOUNTS 0

  Option to print empty accounts in summaries.
- #define USE_COLOR 1

  Option to use color-coded printing in summaries.
- #define NORMAL_COLOR "\x1B[0m"

  Normal text color code.
- #define NEGATIVE_COLOR "\x1B[31m"

  Negative text color code.
- #define POSITIVE_COLOR "\x1B[32m"

  Positive text color code.
- #define ZERO_COLOR "\x1B[34m"

  Zero text color code.

6.72.1 Detailed Description

Author

Will Landau (http://www.will-landau.com/)

Date

2013-2014

Copyright

GNU General Public License >= 3.0 (See COPYING.txt)

Users can redefine the macros in this file to set program preferences at compile time.

6.73 src/which.c File Reference

```
#include <errno.h>
#include "ledger.h"
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include "user_settings.h"
```

Functions

- index_t which (char **s, char *find, int n)

  Finds an occurrence of "find" in "s".

6.73.1 Detailed Description

Author

Will Landau (http://www.will-landau.com/)

Date

2013-2014

Copyright

GNU General Public License 3.0

6.74 src/which_bank_total.c File Reference

```
#include <errno.h>
#include "ledger.h"
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include "user_settings.h"
```

Functions

- index_t which_bank_total (char *status)

  Given a status code, finds the correct index in the bank_totals array member of the Ledger type.

6.74.1 Detailed Description

Author

Will Landau (http://www.will-landau.com/)

Date

2013-2014

Copyright

GNU General Public License 3.0

6.75 src/which_credit_total.c File Reference

```
#include <errno.h>
#include "ledger.h"
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include "user_settings.h"
```

Functions

- index_t which_credit_total (char *status)

  Given a status code, finds the correct index in the credit_totals array member of the Ledger type.

6.75.1 Detailed Description

Author

Will Landau (http://www.will-landau.com/)

Date

2013-2014

Copyright

GNU General Public License 3.0

Index