

---

# **GnuCash Tutorial and Concepts Guide**

***Release 3.10***

**Apr 25, 2020**



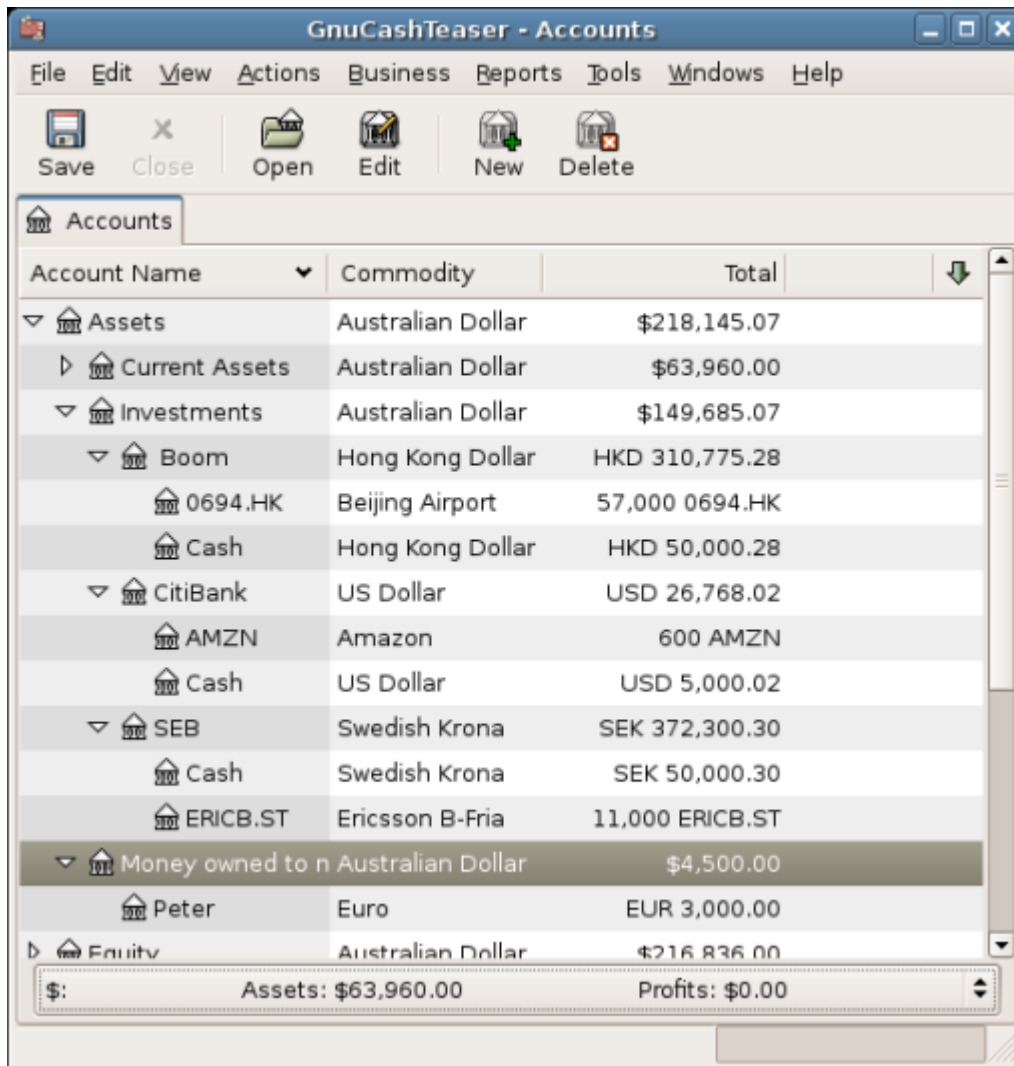
<b>1</b>	<b>Features</b>	<b>3</b>
1.1	Easy to Use . . . . .	3
1.2	Tracks Your Investments . . . . .	4
1.3	International Support . . . . .	4
1.4	Business Support . . . . .	4
1.5	Accounting Features . . . . .	5
<b>2</b>	<b>About this Book</b>	<b>7</b>
2.1	The Basics . . . . .	7
2.2	Accounts . . . . .	26
2.3	Transactions . . . . .	35
2.4	Importing Data into GnuCash . . . . .	59
2.5	Configuring GnuCash . . . . .	64
2.6	Checkbook . . . . .	65
2.7	Expense Accounts . . . . .	73
2.8	Credit Cards . . . . .	74
2.9	Loans . . . . .	85
2.10	Investments . . . . .	98
2.11	Reports . . . . .	138
2.12	Capital Gains . . . . .	151
2.13	Multiple Currencies . . . . .	158
2.14	Business Features . . . . .	171
2.15	Budgets . . . . .	196
2.16	Other Assets . . . . .	201
2.17	Depreciation . . . . .	206
2.18	Python Bindings . . . . .	212
2.19	Importing Business Data . . . . .	213
2.20	GnuCash Glossary . . . . .	220
2.21	Migration Guide . . . . .	223
2.22	Using Accounts vs. Categories . . . . .	223
2.23	Importing Data . . . . .	224
2.24	Converting XML GnuCash File . . . . .	224
2.25	Frequently Asked Questions . . . . .	225
2.26	Sources of Information . . . . .	225
2.27	General Information . . . . .	226
2.28	Using GnuCash . . . . .	227

2.29	Accounting . . . . .	228
2.30	Contributed Account Trees . . . . .	228
2.31	UK Vat . . . . .	228
2.32	Auxiliary File Formats . . . . .	230
2.33	Check Format Files (*.chk) . . . . .	230
2.34	GNU Free Documentation License . . . . .	236
2.35	0. PREAMBLE . . . . .	236
2.36	1. APPLICABILITY AND DEFINITIONS . . . . .	236
2.37	2. VERBATIM COPYING . . . . .	237
2.38	3. COPYING IN QUANTITY . . . . .	237
2.39	4. MODIFICATIONS . . . . .	238
2.40	5. COMBINING DOCUMENTS . . . . .	239
2.41	6. COLLECTIONS OF DOCUMENTS . . . . .	240
2.42	7. AGGREGATION WITH INDEPENDENT WORKS . . . . .	240
2.43	8. TRANSLATION . . . . .	240
2.44	9. TERMINATION . . . . .	240
2.45	10. FUTURE REVISIONS OF THIS LICENSE . . . . .	240
2.46	Addendum . . . . .	241
<b>3</b>	<b>Getting More Help</b>	<b>243</b>
3.1	Context Help . . . . .	243
3.2	GnuCash Website . . . . .	243
3.3	GnuCash Wiki . . . . .	243
3.4	GnuCash On-line Assistance . . . . .	243
3.5	Topic Search . . . . .	244
<b>4</b>	<b>Installation</b>	<b>245</b>

**Warning:** This version of the Guide on ReadTheDocs.org is an unofficial work in progress. Please go to <https://www.gnucash.org/viewdoc.phtml?rev=3&lang=C&doc=guide> for the official documentation.

GnuCash is the personal finance software package made for you. It is versatile enough to keep track of all your financial information, from the simple to the very complex. It is one of the few financial software packages that supports global currencies, and it is the only open-source program of its kind. Best of all, GnuCash is easy to learn and use!

So, what can GnuCash do for you? It can keep track of your personal finances in as much detail as you prefer. If you are just starting out, use GnuCash to keep track of your checkbook. You may then decide to track cash as well as credit card purchases to better determine where your money is being spent. When you start investing, you can use GnuCash to help monitor your portfolio. Buying a vehicle or a home? GnuCash will help you plan the investment and track loan payments. If your financial records span the globe, GnuCash provides all the multiple-currency support you need.



Account Name	Commodity	Total
Assets	Australian Dollar	\$218,145.07
Current Assets	Australian Dollar	\$63,960.00
Investments	Australian Dollar	\$149,685.07
Boom	Hong Kong Dollar	HKD 310,775.28
0694.HK	Beijing Airport	57,000 0694.HK
Cash	Hong Kong Dollar	HKD 50,000.28
CitiBank	US Dollar	USD 26,768.02
AMZN	Amazon	600 AMZN
Cash	US Dollar	USD 5,000.02
SEB	Swedish Krona	SEK 372,300.30
Cash	Swedish Krona	SEK 50,000.30
ERICB.ST	Ericsson B-Fria	11,000 ERICB.ST
Money owned to n	Australian Dollar	\$4,500.00
Peter	Euro	EUR 3,000.00
Equity	Australian Dollar	\$216,836.00
Assets: \$63,960.00		Profits: \$0.00

While GnuCash is well suited for personal finances, it is also powerful enough for business use. There are many business features, from integrated accounts receivable and payable systems, to tax table construction. You will find these and the many other business features surprisingly powerful and easy to use.



### 1.1 Easy to Use

Within a matter of minutes you will be able to enter your personal finance information and generate color graphs that represent your financial status. If you can use the register in the back of your checkbook - you can use GnuCash! Type directly into the register, tab between fields, and use quick-fill to automatically complete transactions. The interface is customizable from within the application itself.

- *Easy to Use Menus*: GnuCash menus conform to the GNOME Human Interface Guidelines. This means that they are simple and similar in appearance to many other GNOME applications.
- *Documentation*: GnuCash has built-in Help and extensive User's Guide documentation.
- *Import Methods*: GnuCash supports many ways to input transactions besides manual entry. If you can access your bank accounts on-line, this is especially useful, as most banks and credit card companies support one of the following import methods. You will spend less time entering data and more time analyzing results.
  - *Quicken Import File (QIF)*: Import Quicken QIF style files, a popular file format with many commercial personal finance software packages.
  - *Open Financial Exchange (OFX)*: GnuCash has been the first free software application to support the Open Financial Exchange protocol. Many financial institutions are moving towards this format.
  - *Home Banking Computer Interface (HBCI)*: GnuCash has been the first free software application to support the German Home Banking Computer Interface protocol. This protocol includes statement download, initiates bank transfers, and makes direct debits possible.
- *Reports*: GnuCash comes with over 30 prebuilt reports, including Account Summary, Income, Expenses, Transaction Report, Balance Sheet, Profit&Loss, Portfolio Valuation, and many others. Reports support graphical views of the data, including pie charts, bar charts, and scatter plots. The reports can be exported to HTML files, and are easily customized.
- *Scheduled Transactions*: GnuCash now has the ability to automatically create and enter transactions, or remind you when these transactions are due, giving you the choice of entering, postponing or removing the automated transaction.
- *Mortgage and Loan Repayment Assistant*: Used to set up a variable payment loan scheduled transaction.

- *Easy Account Reconciliation*: Integrated reconciliation makes reconciling your GnuCash accounts with statements simple and effective.
- *Multi-platform Compatibility*: GnuCash is supported on a variety of platforms and operating systems. The list of fully supported operating systems (and platforms) is for

**GnuCash 3.10** GNU/Linux (x86, x86\_64), FreeBSD (x86, x86\_64), OpenBSD (x86, x86\_64), Apple MacOS (64-bit Intel), and Microsoft Windows (Windows 7 and later).

**GnuCash 2.6.21** Apple MacOS (32-bit Intel, PPC) and Microsoft Windows (XP and Vista).

**GnuCash 2.2.9** Microsoft Windows (2000).

**Previous versions of GnuCash** have been known to work with SGI IRIX (MIPS), IBM AIX 4.1.5 (RS/6000), Unixware 7 (Intel), SCO OpenServer 5.0.4 (Intel), and Solaris (Sparc) but their current status is unknown.

## 1.2 Tracks Your Investments

GnuCash includes a number of investment features that allow you to track all your investments. With GnuCash you can track stocks individually (one per account) or in a portfolio of accounts (a group of accounts that can be displayed together).

GnuCash supports online stock and mutual fund quotes. This means you no longer need to look up stock prices one at a time. The process can be automated, and you can see the latest value of your stocks.

## 1.3 International Support

GnuCash is truly an application that works with and understands users from all around the world. There are many built-in features to facilitate interaction with the international world we live in today.

- *Native Languages*: GnuCash has been translated into 56 languages. The best ones, with over 90% translated, are Croatian, German, Hebrew, Portuguese, Spanish, and Ukrainian. Almost as good, with over 80% translated, are Catalan, Dutch, Japanese, Latvian, Russian, Serbian, and Turkish.<sup>1</sup>
- *International Format Handling*: GnuCash understands that different countries display the date and numbers differently. You are able to work with the date and number formats you are accustomed to.
- *Multiple Currencies and Currency Trading*: Multiple currencies are supported and can be bought and sold (traded). Currency movements between accounts remain fully balanced if “Trading Accounts” is enabled.
- *On-line exchange rates*: With GnuCash, you no longer need look up your exchange rates one at a time. The process can be automated, to always present you with the account values converted to your preferred currency using the latest exchange rates.

## 1.4 Business Support

GnuCash has many features to support the needs of the business accounting community.

- *Accounts Receivable/Payable*: GnuCash has an integrated Accounts Receivable and Accounts Payable system. You can track Customers, Vendors, Invoicing and Bill Payment, and use different Tax and Billing Terms in a small business.
- *Depreciation*: GnuCash can track depreciation of capital assets.

---

<sup>1</sup> If you want to create or improve the translation of your language see ‘<<https://wiki.gnucash.org/wiki/Translation>>’.



- *Reports*: GnuCash offers a wide variety of ready-to-use business reports.

## 1.5 Accounting Features

For those knowledgeable in accounting, here is a list of GnuCash's accounting features.

- *Double Entry*: Every transaction must debit one account and credit others by an equal amount. This ensures the "books balance" - that the difference between income and expense exactly equals the sum of all bank, cash, stock and other assets.
- *Split Transactions*: A single transaction can be split into several pieces to record taxes, fees, and other compound entries.
- *Chart of Accounts*: A master account can have a hierarchy of detail accounts underneath it. This allows similar account types such as Cash, Bank, or Stock to be grouped into a master account such as "Assets".
- *General Journal*: One register window can display multiple accounts at the same time. This eases the trouble of tracking down typing/entry errors. It also provides a convenient way of viewing a portfolio of many stocks, by showing all transactions in that portfolio.
- *Income/Expense Account Types (Categories)*: These categorize your cash flow and, when used properly with the double-entry feature, will provide an accurate Profit&Loss statement.



# CHAPTER 2

---

## About this Book

---

This manual's goal is to save you time. It will get you started using GnuCash as quickly as possible.

Each chapter follows a simple format. A chapter begins with a “Concepts” discussion which introduces general themes and terminology, addressed and used within that chapter. “How-To” sections, which address specific procedures follow. Finally, a “Putting It All Together” section ends the chapter by giving detailed, concrete examples.

Beginning users will find the “Concepts” sections very helpful. They provide a reference for good ways to track your finances, and serve as a general introduction to financial background and terminology. Experienced users can flip to the “How-To” sections to quickly scan procedures. These sections provide to-the-point steps for accomplishing specific tasks. The “Putting It All Together” sections present real-world examples in the form of a tutorial. Beginning with creation of a file in *The Basics*, each successive chapter builds on the previous chapter's tutorial.

This manual is organized into 3 main parts:

- *Getting Started*

## 2.1 The Basics

This chapter will introduce some of the basics of using GnuCash. It is recommended that you read through this chapter, before starting to do any real work with GnuCash. Next chapters will begin to show you hands on examples.

### 2.1.1 Accounting Concepts

GnuCash is easy enough to use that you do not need to have a complete understanding of accounting principles to find it useful. However, you will find that some basic accounting knowledge will prove to be invaluable as GnuCash was designed using these principles as a template. It is highly recommended that you understand this section of the guide before proceeding.

## The 5 Basic Accounts

Basic accounting rules group all finance related things into 5 fundamental types of “accounts”. That is, everything that accounting deals with can be placed into one of these 5 accounts:

**Assets** Things you own

**Liabilities** Things you owe

**Equity** Overall net worth

**Income** Increases the value of your accounts

**Expenses** Decreases the value of your accounts

It is clear that it is possible to categorize your financial world into these 5 groups. For example, the cash in your bank account is an asset, your mortgage is a liability, your paycheck is income, and the cost of dinner last night is an expense.

## The Accounting Equation

With the 5 basic accounts defined, what is the relationship between them? How does one type of account affect the others? Firstly, equity is defined by assets and liability. That is, your net worth is calculated by subtracting your liabilities from your assets:

$$\text{Assets} - \text{Liabilities} = \text{Equity}$$

Furthermore, you can increase your equity through income, and decrease equity through expenses. This makes sense of course, when you receive a paycheck you become “richer” and when you pay for dinner you become “poorer”. This is expressed mathematically in what is known as the Accounting Equation:

$$\text{Assets} - \text{Liabilities} = \text{Equity} + (\text{Income} - \text{Expenses})$$

This equation must always be balanced, a condition that can only be satisfied if you enter values to multiple accounts. For example: if you receive money in the form of income you must see an equal increase in your assets. As another example, you could have an increase in assets if you have a parallel increase in liabilities.

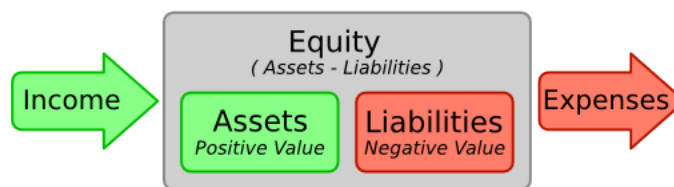


Fig. 1: The basic accounts relationships

## Debits and Credits

The use of debits and credits in accounting and their effect on accounts of different types is often confusing when first encountered in accounting. The accounting equation introduced above is the key to understanding which accounts types are debited or credited and when. First of all we need to rearrange the expanded form a little bit with Assets on the left hand side of the equal sign and transposing any account type with a negative sign to the other side to obtain:

$$\text{Assets} + \text{Expenses} = \text{Liabilities} + \text{Equity} + \text{Income}$$

With the accounting equation in this form, the accounts on the left hand side of the equal sign are known as *debit balance accounts* in accounting practice, that is the normal positive balance for these accounts is increased by *debit* entries to accounts of these types. Conversely credit entries to accounts of these types will decrease the balance of accounts of these types.

Similarly, the account types on the right hand side of the equal sign are known as *credit balance accounts*, that is the normal positive balance for these account types is increased by *credit* entries to the accounts of these types. Again debit entries to accounts of these types will reduce the balance in the account.

balance of accounts of the 5 account types

Account Type	Effect on Account Balance	
Assets	Increase	Decrease
Expenses		
Liabilities	Decrease	Increase
Equity		
Income		

## Double Entry

The accounting equation is the very heart of a double entry accounting system. For every change in value of one account in the Accounting Equation, there must be a balancing change in another. This concept is known as the Principle of Balance, and is of fundamental importance for understanding GnuCash and other double entry accounting systems. When you work with GnuCash, you will always be concerned with at least 2 accounts, to keep the accounting equation balanced.

Balancing changes (or transfers of money) among accounts are done by debiting one account and simultaneously crediting another. Accounting debits and credits do not mean “decrease” and “increase”. Debits and credits each increase certain types of accounts and decrease others as described in the previous section. In asset and expense accounts, debits increase the balance and credits decrease the balance. In liability, equity and income accounts, credits increase the balance and debits decrease the balance.

In traditional double-entry accounting, the left column in the register is used for debits, while the right column is used for credits. Accountants record increases in asset and expense accounts on the debit (left) side, and they record increases in liability, revenue, and equity accounts on the credit (right) side. GnuCash follows this convention in the register.

---

**Note:** This accounting terminology can be confusing to new users, which is why GnuCash allows you to use the common terms Deposit and Withdrawal. If you prefer the formal accounting terms, you can change the account register column headings to use them in the Accounts tab under Preferences (see the GnuCash Help Manual for more information on setting preferences).

---

**Warning:** Common use of the words *debit* and *credit* does not match how accountants use these words. In common use, *credit* generally has positive associations; in accounting, *credit* means *affecting the right column* of the ledger sheet of an account. This is associated with a *decrease* in asset and expense, but an *increase* of income, liability and equity accounts.

For more details see ‘<[https://en.wikipedia.org/wiki/Debits\\_and\\_credits](https://en.wikipedia.org/wiki/Debits_and_credits)>’\_\_.

## 2.1.2 Data Entry Concepts

When entering data in GnuCash, you should be aware of the 3 levels of organization in which GnuCash divides your data: files, accounts and transactions. These levels are presented in their order of complexity, one file contains many accounts and one account contains many transactions. This division is fundamental to understanding how to use GnuCash.

### Files

GnuCash stores information at the highest level in files. A file can be stored on your computer either as a single XML file (in all versions of GnuCash), or in a SQL database (in GnuCash version 2.4 and higher).

---

**Note:** SQL is pronounced “sequel”, so in spoken and written language we would say “a SQL database”.

---

With the XML file format, GnuCash stores your data in an XML data file, usually in compressed format (although this can be changed in the General tab of the GnuCash Preferences).

With SQL storage, GnuCash stores your data in a SQL database under the database application you select (SQLite3, MySQL or PostgreSQL).

You will need one main file or database for each set of accounts you are maintaining. To learn how to create and manage GnuCash files, see *Storing your financial data*.

---

**Note:** If you think you might need more than one set of accounts, you might want to consult a professional accountant or bookkeeper before proceeding. Most users will probably have only one data file.

---

Backup files and log files are automatically generated by GnuCash when appropriate. Backup and log files are described in *Backing Up and Recovering Data*.

### Accounts

An account keeps track of what you own, owe, spend or receive. Each GnuCash file can contain any number of accounts, and each account can contain many sub-accounts up to an arbitrary number of levels. This simple feature gives GnuCash much of its power in managing your finances, as you will see in later chapters.

Examples of accounts include: checking accounts, savings accounts, credit card accounts, mortgages, and loans. Each GnuCash account tracks the activity for that “real” account, and can inform you of its status.

In addition, accounts are also used to categorize the money you receive or spend. For example, you can create expense accounts to track the money you pay on utilities or groceries. Even though these are not accounts that receive statements, they allow you to determine how much money is being spent in each of these areas.

Accounts will be covered in more detail in *Accounts*.

### Transactions

A transaction represents the movement of money among accounts. Whenever you spend or receive money, or transfer money between accounts, that is a transaction.

Examples of transactions are: paying a phone bill, transferring money from savings to checking, buying a pizza, withdrawing money, and depositing a paycheck. *Transactions* goes more in depth on how to enter transactions.

In *double entry accounting*, transactions always involve at least two accounts—a source account and a destination account. GnuCash manages this by inserting a line into the transaction for every account that is affected, and recording the amounts involved in each line. A line within a transaction that records the account and amount of money involved is called a split. A transaction can contain an arbitrary number of splits.

---

**Note:** Splits in transactions will be covered in *Split Transaction Example*

---

### 2.1.3 Running GnuCash

GnuCash can be run from your desktop main menu by selecting the associated menu entry.

Alternatively it can be run from a command line prompt with the command `gnucash`.

During start up, GnuCash will display the Splash Screen, where some information about the program (version number, build, etc.) and the loading process are displayed.

#### Welcome to GnuCash dialog

The very first time you open GnuCash, you will see the Welcome to GnuCash! screen. This dialog includes three choices:

- Create a new set of accounts - Runs the New Account Hierarchy Setup assistant (see *New Account Hierarchy Setup*). Select this option if you want to be assisted in creating a set of accounts.
- Import my QIF files - Runs the Import QIF Files assistant (see *Import QIF*). Select this option if you already have Quicken files (.qif files) and wish to import them into GnuCash.
- Open the new user tutorial - Opens the GnuCash Tutorial and Concepts Guide. Select this option if you are completely new to GnuCash and accounting concepts.

---

**Note:** It is possible to access each of these items after you have left this screen, but the Welcome to GnuCash! screen will not reappear. To create a new set of accounts, see *New Account Hierarchy Setup*. To import QIF files, see *Import QIF*.

---

#### New Account Hierarchy Setup

The *New Account Hierarchy Setup* assistant helps you to create a set of GnuCash accounts. It will appear if you choose Create a new set of accounts in the Welcome to GnuCash! menu, or when you select File > New.

This assistant will create a new blank GnuCash file and guide you through the creation of a *Chart of Accounts*. There are several steps in the assistant, which are outlined below.

1. The first screen briefly describes what this assistant does.
2. New Book Options allows you to set different attributes for your file that affect the file as a whole. This screen has four tabs: Accounts, Budgeting, Business, and Counters. These items are explained elsewhere in the Guide, and can be changed at a later point.
3. Choose Currency sets the default currency for new accounts. This is based on the computer locale settings, and can be modified later in the Accounts tab under Preferences (see *Preferences: Accounts Tab*).
4. Choose accounts to create allows you to create an initial set of accounts. These can be edited as needed afterward. The screen is divided into three parts.

- The left upper portion has a list of Categories for commonly used hierarchies of accounts. Select from this list the types of accounts you wish to use. You can select as many of the categories of accounts as you wish.
  - The left lower section has a Category Description that displays a detailed description of the category currently highlighted.
  - The right side has a list of the Accounts that will be created from a selected category. Note that the accounts listed here are *only* the selected category; your final data file will include *all* of the accounts for all of the selected Categories.
5. Setup selected accounts lists all the accounts you selected on Choose accounts to create, and allows you to enter opening balances and to designate *Placeholder* accounts.

---

**Note:** Equity accounts do not have opening balances, so the opening balance value for this kind of account is locked and set to zero.

---

---

**Note:** *Placeholder* accounts are used to create a hierarchy of accounts and normally do not have transactions or opening balances.

---

- The left side of the screen has a list of Account Names. Select an account by “clicking” once in the Account Names column with the account highlighted. This will open the account name for changes.
  - The right side of the screen has a check-box to make an account a Placeholder and a box to add the Opening Balance for the selected account. Again a single click in the Opening Balance or Placeholder column will open the field for changes.
6. Finish account setup is the last screen and gives you a final option to cancel the process.

**Warning:** If you choose to cancel, any selections you have made up to this point will be lost.

### Tip of the Day

GnuCash provides a Tip of the Day screen to give helpful hints for using the program:

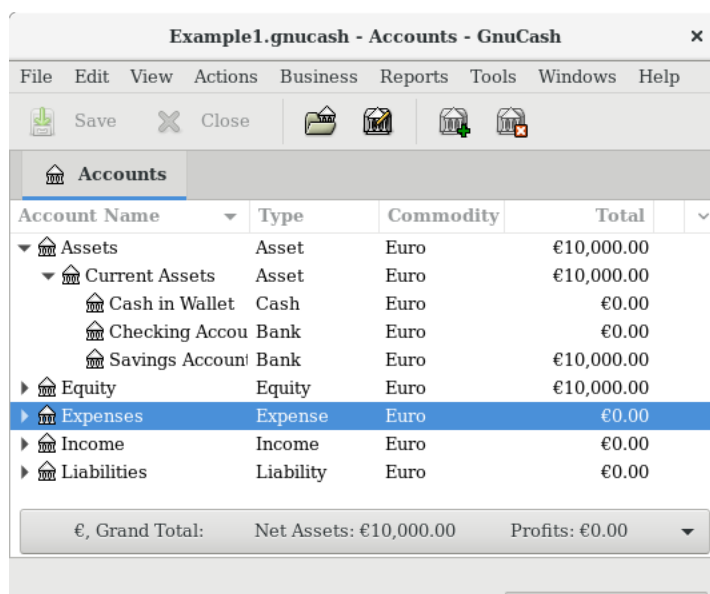




These tips provide useful information for beginning users. To view more of the tips, click Forward to continue. If you do not wish to see this screen box on start-up, deselect the box next to Show tips at startup. When you have finished viewing the helpful tips, click Close to close the Tip of the Day screen.

## Account Tree Window

You should now see the Accounts window, which appears as shown below. The exact layout of the account tree will depend on which default accounts you selected during the New Account Hierarchy Setup. In this example, the Common Accounts are shown.



The Account Tree window (also known as a Chart of Accounts, or CoA) provides an overview of the data contained in the current file. It contains a list of account names and their current balances.

From this window, you can open the register of any account either by double-clicking the account name, right clicking the account name and selecting Open Account from the menu, or by using the Open button on the toolbar. GnuCash allows you to have as many account registers open as you wish. For more information on using account registers, see [Account Register Window](#).

---

**Tip:** Clicking the small triangle to the left of an account that has children will expand the tree view showing child accounts.

---

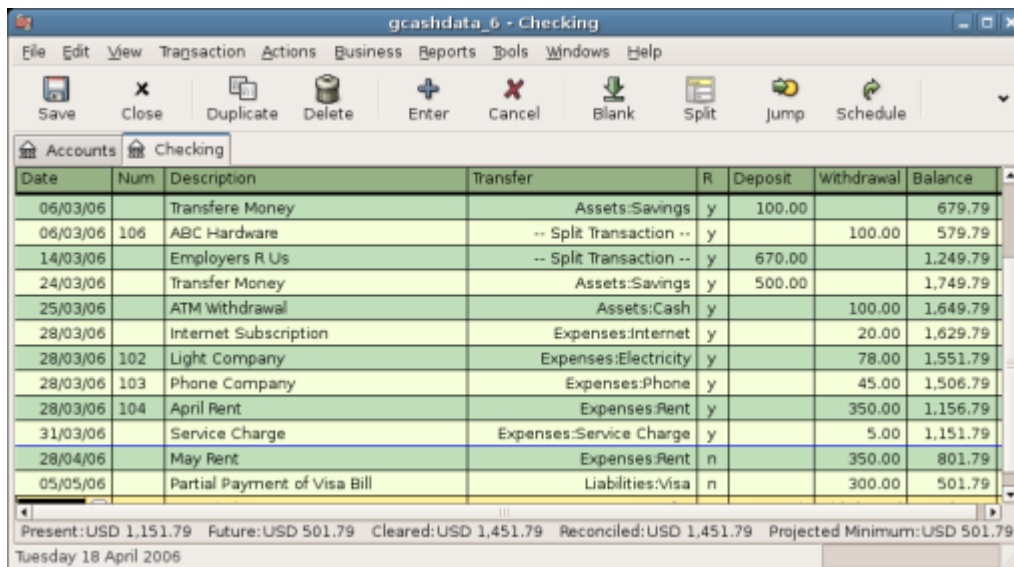
At the top of this window is the *Titlebar*, which displays the file name for this set of accounts (once you have saved the file.) Below that is the *Menubar*. You can access the menu options by either clicking on these menu headings or by using shortcuts and access keys (see [Menu Shortcuts](#)). Next is the *Toolbar*, which contains buttons for the most common functions.

The account tree appears below the *Toolbar*. Once you have started creating accounts, the account names will appear in the account tree. You can customize which headings show up by using the small down-arrow at the far right just above the account tree.

At the bottom is the *Statusbar*, which tells you information about what you own (Net Assets) and how much money you have made (Profits).

### Account Register Window

Account Register windows are used to enter and edit your account data. As the name suggests, they look similar to a checkbook register.



Date	Num	Description	Transfer	R	Deposit	Withdrawal	Balance
06/03/06		Transfer Money	Assets:Savings	y	100.00		679.79
06/03/06	106	ABC Hardware	-- Split Transaction --	y		100.00	579.79
14/03/06		Employers R Us	-- Split Transaction --	y	670.00		1,249.79
24/03/06		Transfer Money	Assets:Savings	y	500.00		1,749.79
25/03/06		ATM Withdrawal	Assets:Cash	y		100.00	1,649.79
28/03/06		Internet Subscription	Expenses:Internet	y		20.00	1,629.79
28/03/06	102	Light Company	Expenses:Electricity	y		78.00	1,551.79
28/03/06	103	Phone Company	Expenses:Phone	y		45.00	1,506.79
28/03/06	104	April Rent	Expenses:Rent	y		350.00	1,156.79
31/03/06		Service Charge	Expenses:Service Charge	y		5.00	1,151.79
28/04/06		May Rent	Expenses:Rent	n		350.00	801.79
05/05/06		Partial Payment of Visa Bill	Liabilities:Visa	n		300.00	501.79

Present:USD 1,151.79 Future:USD 501.79 Cleared:USD 1,451.79 Reconciled:USD 1,451.79 Projected Minimum:USD 501.79  
Tuesday 18 April 2006

[Transactions](#) explains more about account register windows and how to enter data into them. For now, note that the parts of an account register window are similar to the parts of the account tree window described earlier. The *Titlebar* at the top contains the account name. Below that, the *Menubar* contains menu options related to the account register. *Toolbar* buttons simplify common data entry functions. The *Statusbar* at the bottom of the window, displays some account balances covered in [Transactions](#). At the bottom of the account register window, information appears about the current location of the cursor.

---

**Note:** In the register windows, you can resize the various columns that GnuCash displays, *but keep in mind that the Description and Balance columns behave differently from other columns.*

The Description column is designed to expand automatically to fill all unused horizontal screen space. Therefore you should set the widths of all your other columns before setting the Description column width.

The Balance column must be resized by double-clicking on the column heading.

---

## Toolbar Buttons

Both the account tree window and the account register window contain *Toolbar* buttons. These buttons provide quick access to common functions such as Save and Open in the account tree window and Record and Delete in the account register window. If you are not sure what a button does, move the mouse pointer over that button, and you should see a description of the function appear.

Here is a summary of the account tree window buttons:

**Save** Save the current file to disk

**Close** Close the current notebook page

**Open, Edit, New and Delete** These are functions related to accounts. They are discussed in [Accounts](#).

Register-specific buttons are discussed in [Transactions](#).

## Tab Bar

GnuCash uses a tabbed model that allows you to open multiple account registers and reports simultaneously. Each open window (which can include account registers, reports, or Scheduled Transactions windows) is given a tab on this bar that you can click to view that window. Tabs can be configured in Preferences to appear along any side of the GnuCash window.

To see the full name for a tab, hover the mouse pointer over an account window tab.

If more screens are open than can be displayed across the screen, some tabs will not display. You can move through all tabs by clicking the arrows on either end of the tab bar. A complete list of tabs can be viewed by right-clicking the Tab Bar and any tab can be selected by clicking it.

## Menu Items

The account tree window and the account register window both contain menu headings in a *Menubar*. Clicking on a menu heading brings up the menu items for that heading.

You can click on the account tree menu headings and then move the mouse pointer over the menu items to see what they do. As the pointer moves over a menu item, a description of the item appears in the lower left-hand corner of the window (inside the *Statusbar*). To select a menu item, click on it.

You can also access the most common menu items in a window by right-clicking the mouse anywhere in that window. In the account tree window, this will bring up a list of account items. In the account register window, this will bring up a list of transaction items.

Other ways of accessing menu items are through keyboard shortcuts and access keys, described next.

## Menu Shortcuts

All of the menu items have access keys which are marked by underlined characters in the menu names. Pressing the Alt key with the underlined character in the menu heading will bring up the menu items for that heading. Once the menu items are displayed, type the underlined character in the menu item to activate it. For example, typing Alt+ +F in the main window brings up the File menu, then typing S will save the file. Access keys are fixed and cannot be changed by users.

Some of the more commonly used menu items also have shortcut keys that directly activate the command without having to traverse the menu structure. These shortcuts typically use the Ctrl key, although they can use any key combination. Menu shortcuts are displayed at the end of each menu item.

## 2.1.4 Storing your financial data

### Overview

GnuCash offers several formats for storing your financial data. The default file storage format is XML, while SQL storage is available in SQLite, MySQL, and PostgreSQL formats. Users can choose a file format for new files from File > Save and for existing files from File > Save As... dialogs.

The XML storage format is a text file that by default is compressed, which is a preference that is set at Edit > Preferences General Compress files. SQLite storage is also available, and stores your data in a single file on your system, like the XML format. However, internally, an SQLite file is managed as a database. The MySQL and PostgreSQL storage options require access to a MySQL or PostgreSQL database server and the installation of additional database drivers on your machine.

---

**Tip:** Users can change the format at any time by using File > Save As... This will create a copy of the data file in the selected format.

---

### Storage Comparison and Recommendations

Each storage format has benefits and shortcomings that users should consider for their needs and abilities. See the *#basics-storage-comparison-table* below for further details.

The XML format is the most stable and established, and for this reason, it is recommended for most users. SQL storage was added for the 2.4 release and has become an increasingly popular choice for users. The SQLite format allows users to realize the benefits of SQL storage without the overhead of installing or managing a full DBMS. MySQL and PostgreSQL require the installation of MySQL and PostgreSQL DBMS, respectively, and are best maintained only by experienced database administrators.

---

**Note:** Use of a SQL back end for storage implies to many that GnuCash has fully implemented DBMS features, including multi-user and incremental data manipulation. However, GnuCash does not currently implement these features, although it is a long term goal of the development team.

---

## Storage Comparison Table

Table 1: Storage Comparison

	XML	SQLite	MySQL	PostgreSQL
Availability	Built-in	Depends on packaging <sup>5</sup>		
File extension	gnucash	N/A <sup>6</sup>		
Additional software	None	MySQL	PostgreSQL	
Additional expertise	None	Database Administrator		
Compression	gzip	N/A		
File Save	On command	On commit		
Multi-user	No	No	No	No

## Creating a file

To create a new GnuCash file do the following:

1. From the GnuCash *Menubar*, choose File > New File. The New Account Hierarchy setup assistant will start.

---

**Note:** If you are running GnuCash for the first time, you will be presented with the Welcome to GnuCash! screen. This screen is described in detail in the GnuCash manual.

---

2. Set your preferences in the assistant and move through the screens with the Forward, Cancel and Previous buttons.

## Saving data

Follow these steps to save the file under your preferred name:

1. Choose File > Save As... from the *Menubar* or select the Save *Toolbar* button. GnuCash will bring up the save window.
2. Select the Data Format of the file you are saving from the drop down list. The default selection is XML but if you have set up a database back end you can change to that format.

Depending on the selected Data Format the window can change as described in the following.

3.
  - If you selected XML or sqlite3 you will see a screen like this:

Type your chosen filename in the Name field. It is not necessary to specify an extension when you write the file name. GnuCash will automatically add the extension `.gnucash` to the file.

---

**Note:** The `.gnucash` extension was introduced in the 2.3 series of GnuCash. For already existing files, the extension will never be changed. So if you open an existing file named `Myoldfile`, that name won't be changed if the file is saved. You might use the Save As... command and give the file a new name in order to have it saved with the extension `.gnucash`.

---

Select the path where the file will be saved by browsing the tree in the lower panes.

---

<sup>5</sup> SQLite relies on an additional package and driver (called libdbi and libdbd-sqlite3, respectively), which are installed by default on Mac OS and Windows. Linux users may need to manually install these for SQLite.

MySQL and PostgreSQL may require the installation of additional software drivers (libdbd-mysql and libdbd-pgsql).

<sup>6</sup> MySQL and PostgreSQL place data within their own storage system.

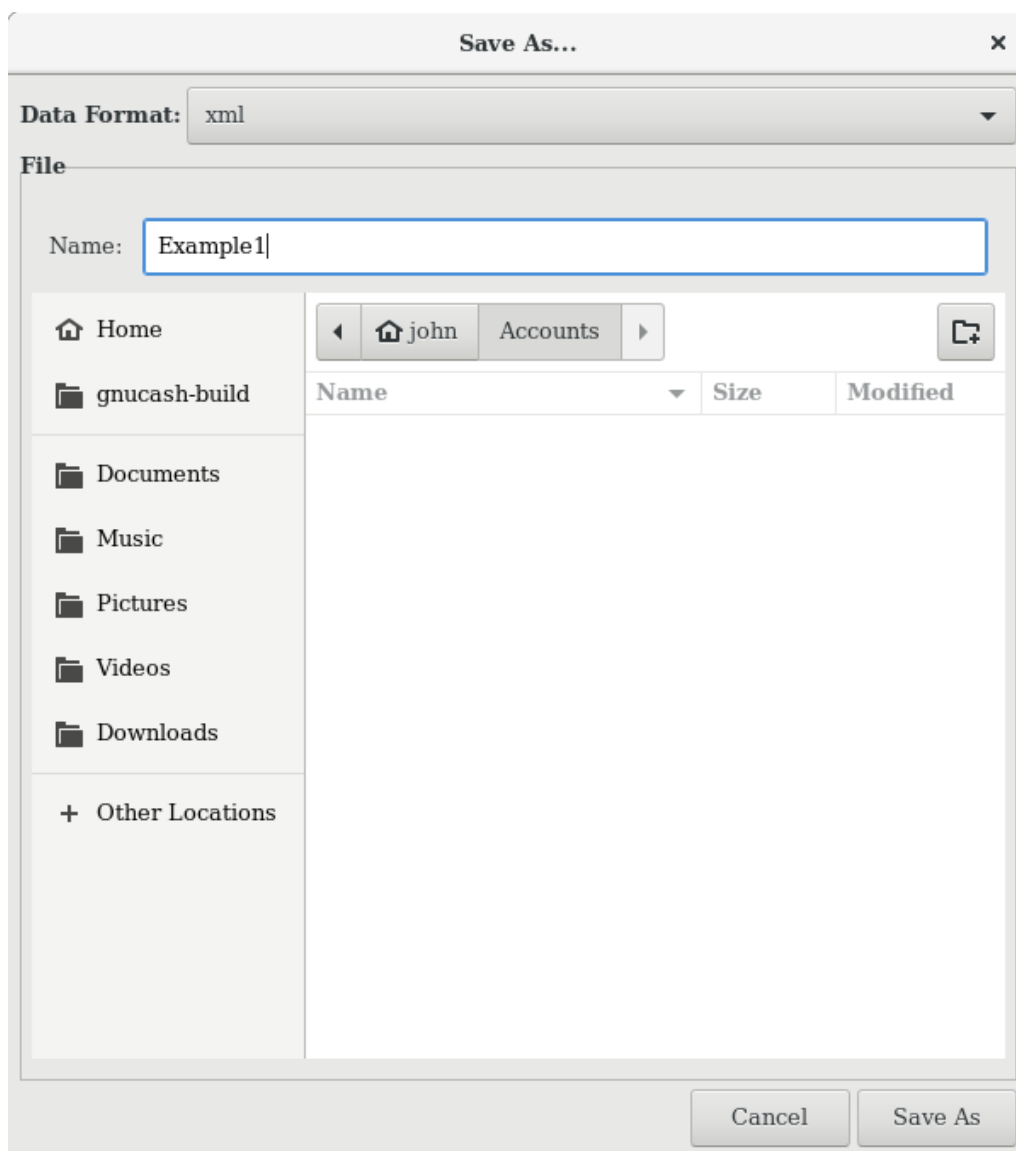


Fig. 2: Save screen when XML or sqlite3 is selected.

---

**Tip:** Click on the Create Folder button to create a new folder with a custom name in the selected path.

---

- If you selected mysql or postgres Data Format you will see a screen like this:

The screenshot shows a 'Save As...' dialog box. At the top, the title bar says 'Save As...' with a close button. Below the title bar, there is a 'Data Format:' label followed by a dropdown menu showing 'mysql'. Underneath this is a section titled 'Database Connection'. It contains four input fields: 'Host' with the value 'localhost', 'Database' with the value 'gnucash', 'Username' which is empty, and 'Password' which is empty. At the bottom right of the dialog are two buttons: 'Cancel' and 'Save As'.

Fig. 3: Save screen when mysql or postgres is selected.

Enter in this window the Database Connection information: Host, Database, Username and Password.

**Warning:** Saving to mysql or postgres requires the proper permissions in that database, that is you need to have the permissions to create a new database with the given database name, or you need to have write access to an existing database with the given database name.

4. Click the Save As button to save the file.

If you are keeping track of finances for a single household, you need only one file. But if you are also tracking business finances or want to keep data separate for some reason, then you will need more than one file.

Before ending each GnuCash session, be sure to save your data changes using File > Save or the Save *Toolbar* button.

---

**Note:** As it is very important to save your data frequently to avoid losing them for whatever reason, GnuCash is able to automatically save the opened file every a certain amount of time. This interval can be set in the General tab under Edit > Preferences (GnuCash > Preferences on MacOS). Keep in mind that this option is relevant only if you are saving in XML format. If you are working with a database, the Save button and the Save menu entry will be grayed out because changes are stored right away.

---

## Opening data

To open an existing file or database, select File > Open from the menu. In the window that will open, select the Data Format. If you selected File choose the file you want to open by browsing the folders in the lower panes. Else, enter the required Database Connection information.

---

**Tip:** GnuCash keeps a list of the recently opened files. Open the File menu and you will see listed the names of recently opened files. Click on the one you want to load to open it.

---

### Duplicating an Account Hierarchy

In some cases, it might be useful to duplicate the structure of an existing data file in a new file. For example, you might want to try out new accounting techniques without corrupting your actual accounting data, or you might need to follow accounting guidelines that require you to close your books at the end of the year and begin each year with a fresh set of books.

GnuCash allows you to create an empty copy of your Chart of Accounts simply by selecting File > Export > Export Accounts. When you select this command, you are asked to provide the name for the new empty file, and GnuCash creates a new data file that contains only your account hierarchy (that is, there is no transaction data). Once saved, the new file can be opened like any other GnuCash data file as described above.

### 2.1.5 Backing Up and Recovering Data

GnuCash creates several types of files to help ensure that your data is not lost. If you look in the folder where your saved file resides, you may see other files generated by GnuCash with the following extensions: `.gnucash`, `.log`, `.LCK`, `.LNK` in the same directory as your primary data file. What each of these files does is presented below.

---

**Note:** The following sections are relevant only if you are saving your financial data in the XML format

---

```
$ ls
myfile.gnucash
myfile.gnucash.20100414185747.gnucash
myfile.gnucash.20100414223248.log
myfile.gnucash.20100415114340.gnucash
myfile.gnucash.20100415154508.log
myfile.gnucash.20100415173322.gnucash
myfile.gnucash.20100415194251.log
myfile.gnucash.7f0982.12093.LNK
myfile.gnucash.LCK
```

#### Backup file (`.gnucash`)

Each time you save your data file, a backup copy will also be saved with the extension `.YYYYMMDDHHMMSS.gnucash`. This backup file is a complete copy of your previous data file, and the filename format refers to the data file, year, month, day and time of the backup. For example, the filename `myfile.gnucash.20100414185747.gnucash` indicates this is a backup copy of the file `myfile` saved in the year 2010, April 14, at 6:57:47 p.m.

To restore an old backup file, simply open the `.YYYYMMDDHHMMSS.gnucash` file with the date to which you wish to return. Be sure to save this file under a different name.

---

**Note:** `.YYYYMMDDHHMMSS.xac` instead of the actual extension `.YYYYMMDDHHMMSS.gnucash`. So if you upgrade from the 2.2 series to the 2.4 series, you may end up with both `.YYYYMMDDHHMMSS.xac` and `.YYYYMMDDHHMMSS.gnucash` backup files in your directory.

---



## Log file (.log)

Each time you open and edit a file in GnuCash, GnuCash creates a log file of changes you have made to your data file. The log file uses a similar naming format as the backup files: `.YYYYMMDDHHMMSS.log`. Log files are not a full backup of your data file - they simply record changes you have made to the data file in the current GnuCash session.

In case you exit GnuCash inadvertently, possibly due to a power outage or a system wide crash, it is possible to recover most of your work since the last time you saved your GnuCash file using this log file. This is the procedure:

1. Open the last saved GnuCash file.
2. Go to File > Import > Replay GnuCash .log file and select the one .log file with the same date as the saved file you just opened. Make sure that you picked the right .log file, or you will possibly wreak havoc in your accounts.

Log replaying will recover any transaction affecting the balance entered since the last save, including those created from scheduled transactions and business features (invoices, bills, etc.).

**Warning:** Changes to the scheduled transactions, invoices or bills themselves are NOT recovered, and their transactions that were recovered may not be properly associated with them, and should thus be double-checked. Especially for business transactions, you may have to delete and re-create some of them. If you do not, although the balance will be correct, some reports may not.

## Lock files (.LNK and .LCK)

You may occasionally see `.LNK` and `.LCK` files appear. These do not store any data, but they are created to prevent more than one user from opening the same file at the same time. These files are automatically created when you open the file, to lock it so no one else can access it. When you close your GnuCash session or open another file, GnuCash unlocks the first data file by deleting the `.LCK` and `.LNK` files.

If GnuCash crashes while you have a data file open, the `.LCK` and `.LNK` files are not deleted. The next time you try to open GnuCash, you will get a warning message that the file is locked. The warning message appears because the `.LNK` and `.LCK` files are still in your directory. It is safe to choose Yes to open the file, but you should delete the `.LNK` and `.LCK` files (using a terminal window or your file manager). Once those files are deleted, you will not get the warning message again unless GnuCash crashes.

## File Management

So which files should you keep around? Keep your main data file, of course. It's a good idea to keep some of the more recent `.YYYYMMDDHHMMSS.gnucash` backup files, but you can safely delete the `.log` files since they are not complete copies of your data.

---

**Note:** If you upgraded from a GnuCash version prior to 2.4, you may also have backup files in the old `.xac` format. For these files you can apply the same principle described above for `.YYYYMMDDHHMMSS.gnucash` backup files.

---

You should also delete any `.LCK` and `.LNK` files that you see after closing GnuCash. If you decide to back up your data file to another disk manually, it's enough to back up the main data file - not the `.YYYYMMDDHHMMSS.gnucash` backup files.

---

**Note:** By default GnuCash will automatically delete any `.log` and `.YYYYMMDDHHMMSS.gnucash` backup files that are older than 30 days. You can change this behavior in the GnuCash preferences in the General tab under Edit >

Preferences (GnuCash > Preferences on MacOS).

---

## 2.1.6 Migrating GnuCash data

Sometimes you may need to move your financial data and GnuCash settings to another machine. Typical use cases are when you buy a new computer or if you want to use the same settings over two different operating systems in a dual boot configuration.

### Migrating financial data

Migrating GnuCash financial data is as simple as copying `.gnucash` files with a file manager if you know where they are saved. If you can't remember where a file is stored but you can open it directly within GnuCash, save it in the desired path from within GnuCash.

All other files in the folder are either backups or log files. It won't do any harm to copy them too, but it's not likely to do any good, either.

### Migrating preferences data

Preferences are stored in three different locations: one for GnuCash preferences, one for reports, and one for online banking settings. Preferences are managed by `gsettings`, reports are managed by GnuCash itself, and online banking is managed by `aqbanking`. If you do not use online banking, then you will not have this folder on your machine.

Where the GnuCash preferences are stored varies depending on your operating system (see *table\_title*, *table\_title*, and *table\_title*). To back up and transfer your entire installation, you must copy these preferences as well.

Table 2: Application Settings Locations

Operating system	folder
Unix	GnuCash preferences are stored in <code>dconf</code> . You can use the commands <code>dconf dump /org/gnucash/</code> on the old machine and <code>dconf load /org/gnucash/</code> on the new machine to migrate your preferences.
Mac OSX	<code>~/Library/Preferences/gnucash.plist</code>
Windows	The preferences are stored in the Windows registry <code>HKEY_CURRENT_USER/software/GSettings</code>

Table 3: Saved Reports Locations

Operating system	folder
Unix	<code>~/.local/share/gnucash/</code> <sup>9</sup>
Mac OSX	<code>~/Library/Application Support/gnucash</code>
Windows	<code>Documents and Settings/Username/.gnucash</code> or <code>Users/Username/.gnucash</code>

---

<sup>9</sup> Up to GnuCash 2.6.21 it was `~/.gnucash/`

Table 4: Online Banking Settings Locations

Operating system	folder
Unix	<code>~/.aqbanking</code>
Mac OSX	<code>~/.aqbanking</code>
Windows	<code>Documents and Settings/Username/.aqbanking</code>

---

**Note:** On Unix and Mac OSX, these folders will generally not display in the file manager. You must set the file manager to show hidden files and folders to see them.

---

---

**Tip:** On Unix and Mac OSX, the `~` symbol means the home folder.

---

## 2.1.7 Putting It All Together

---

**Note:** This section begins a tutorial that will continue throughout this book. At the end of each chapter, you will see a Putting It All Together section that walks you through examples to illustrate concepts discussed in that section. Each Putting It All Together section builds on the previous one, so be sure to save your file for easy access.

---

Let's get started!

1. First, let's create a file to store your real data. Open GnuCash and select File > New File from the *Menubar*. This will start the New Account Hierarchy Setup assistant that allows you to create several accounts at once.

---

**Note:** If you are running GnuCash for the first time, you will be presented with the Cannot find default values screen which is described in details in the GnuCash manual.

---

The first screen of the assistant gives you a description of what the assistant does. Click the Forward button to proceed to the next screen.

2. In the second screen, set the New Book Options on the different tabs, then press the Forward button. You can also update these options later using File > Properties. For details of these options, see the GnuCash Help manual, chapter Customizing GnuCash, Book Options.
3. In the third screen, select the currency to use for the new accounts from the dropdown list. Then press the Forward button.

---

**Note:** The currency you select here, will be assigned to all the accounts created in this assistant.

---

4. In the fourth screen select the Common Accounts group in the Categories pane. Then press the Forward button to proceed.

---

**Note:** If you want, you can select one or more of the predefined account-groups here. For more information on account types, see [GnuCash Accounts](#).

---

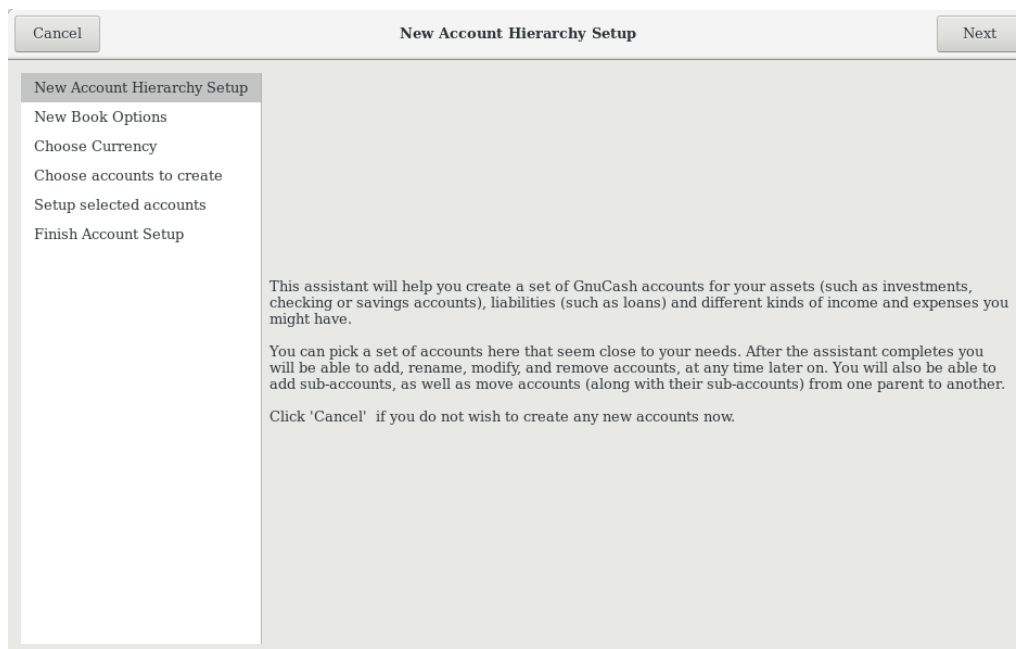


Fig. 4: New Account Hierarchy Setup: Introduction

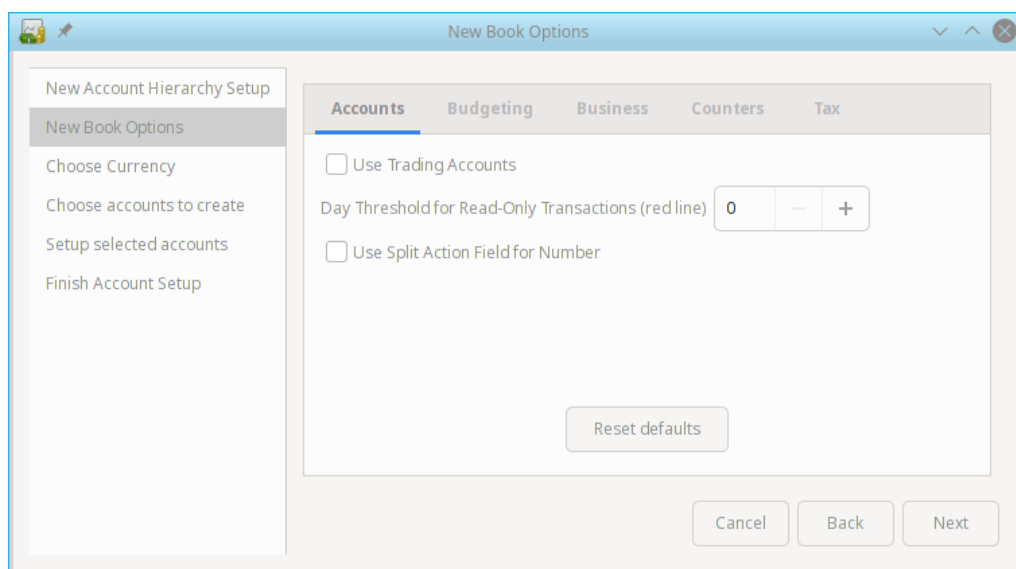


Fig. 5: New Account Hierarchy Setup: Book Options

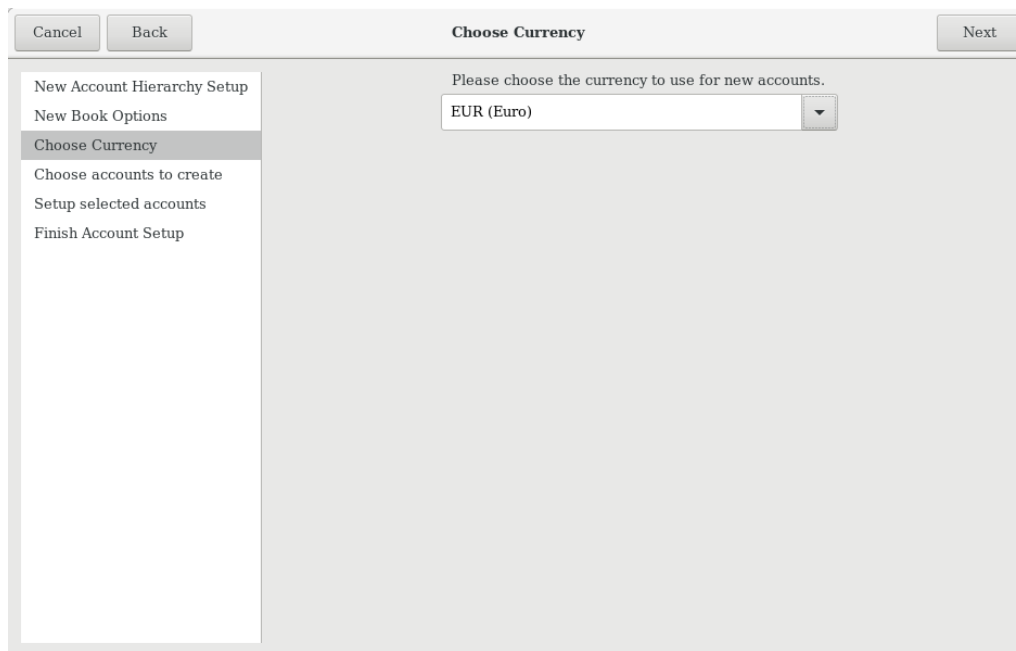


Fig. 6: New Account Hierarchy Setup: Currency Selection

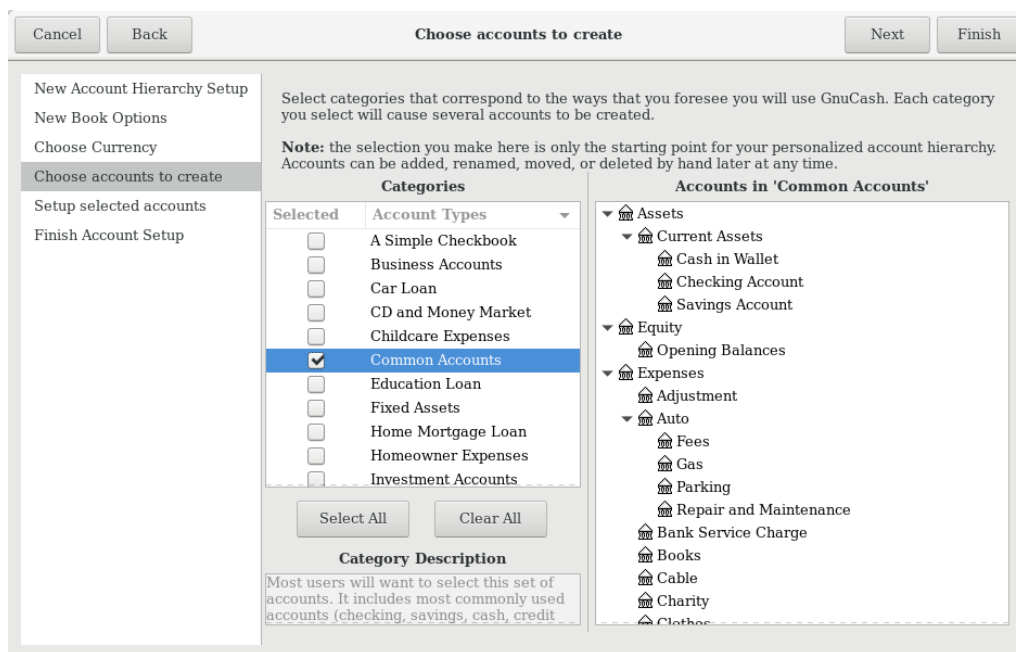


Fig. 7: New Account Hierarchy Setup: Account Selection

5. In the fifth screen you will be able to set an Opening Balance on each of the accounts, as well as indicate if the account should be a Placeholder. As these features will be described in next chapters, leave all as configured by GnuCash and click Forward to open the last screen of the assistant.

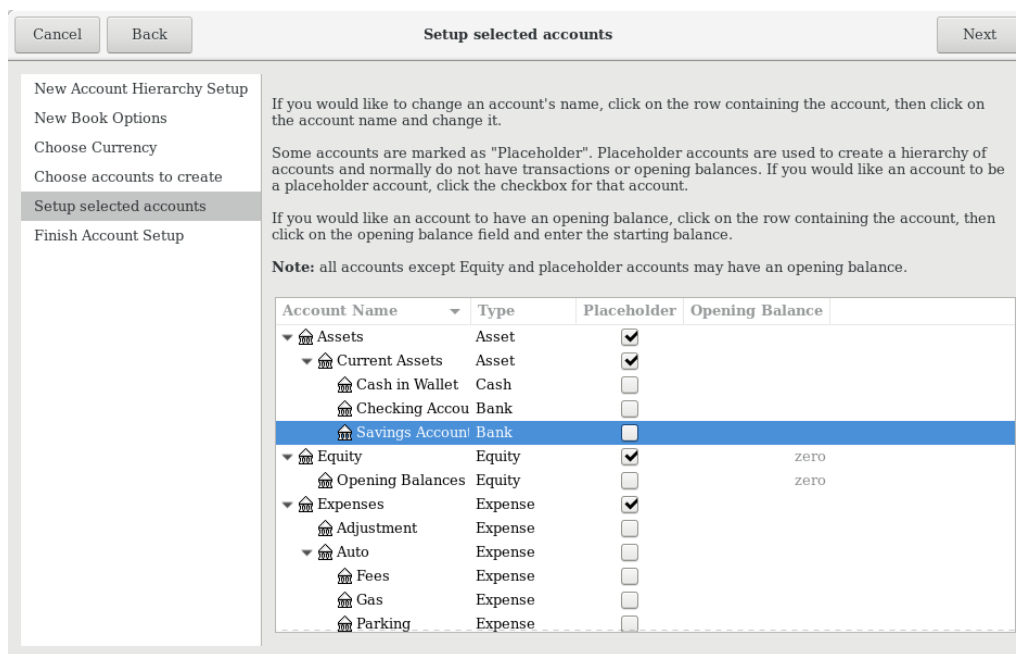


Fig. 8: New Account Hierarchy Setup: Account Setup

6. In the last screen of the assistant, click Apply to create all the accounts and leave the assistant.
7. After pressing Apply in the previous window, you will be presented with the save dialog. Select the XML Data Format, Name the file as `gcashdata_1`, select the folder where to save the file (remember it as the data file will be used in the tutorials throughout this manual), and finally press the Save as button.

Your main window should now look something like this:

## 2.2 Accounts

This section discusses some useful concepts for organizing your accounts. Since GnuCash does not impose any specific account tree layout, you are free to design your account structure in any manner you wish. However, there are a few basic accounting concepts which you will probably want to follow when designing your accounts to maximize their utility.

### 2.2.1 Basic Accounting Concepts

As we saw in the previous chapter, accounting is based on 5 basic account types: Assets, Liabilities, Equity, Income and Expenses. We will now expand on our understanding of these account types, and show how they are represented in GnuCash. But first, let's divide them into 2 groups, the balance sheet accounts and the income and expense accounts. As the name says the balance sheet accounts can be summarized in the balance of what you own and owe *at a point in the time*, while the income and expense accounts can be summarized in the Profit & Loss report, which shows the *change of values in a period of time* like the economic year

Let's have a quick look at the Accounting Equation ( $Assets - Liabilities = Equity + (Income - Expenses)$ ) again as a reminder, before we go deeper into each account type.

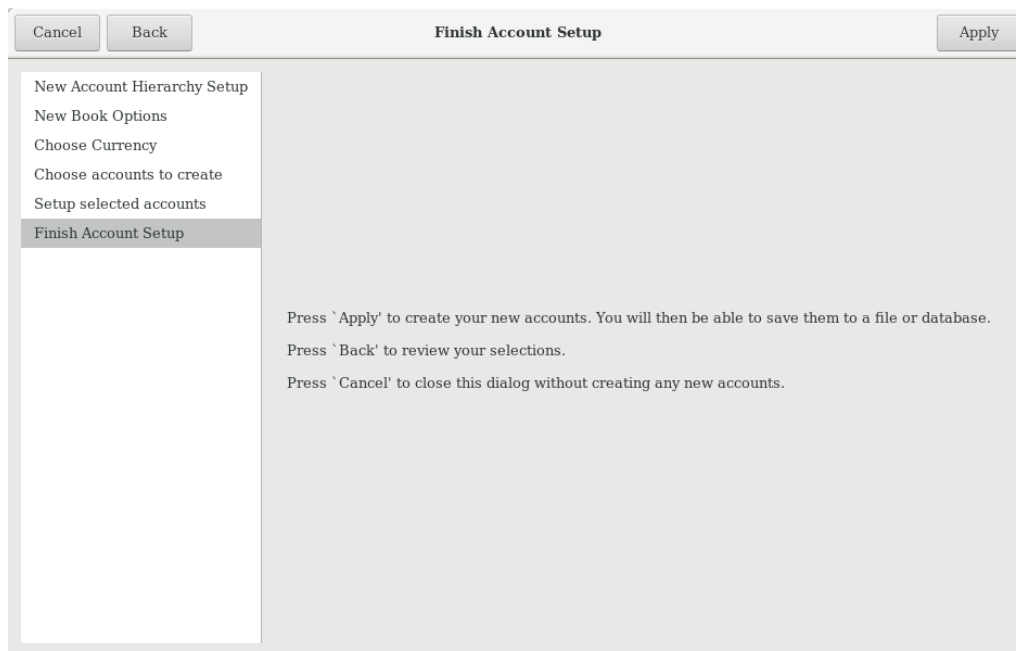


Fig. 9: New Account Hierarchy Setup: Finish

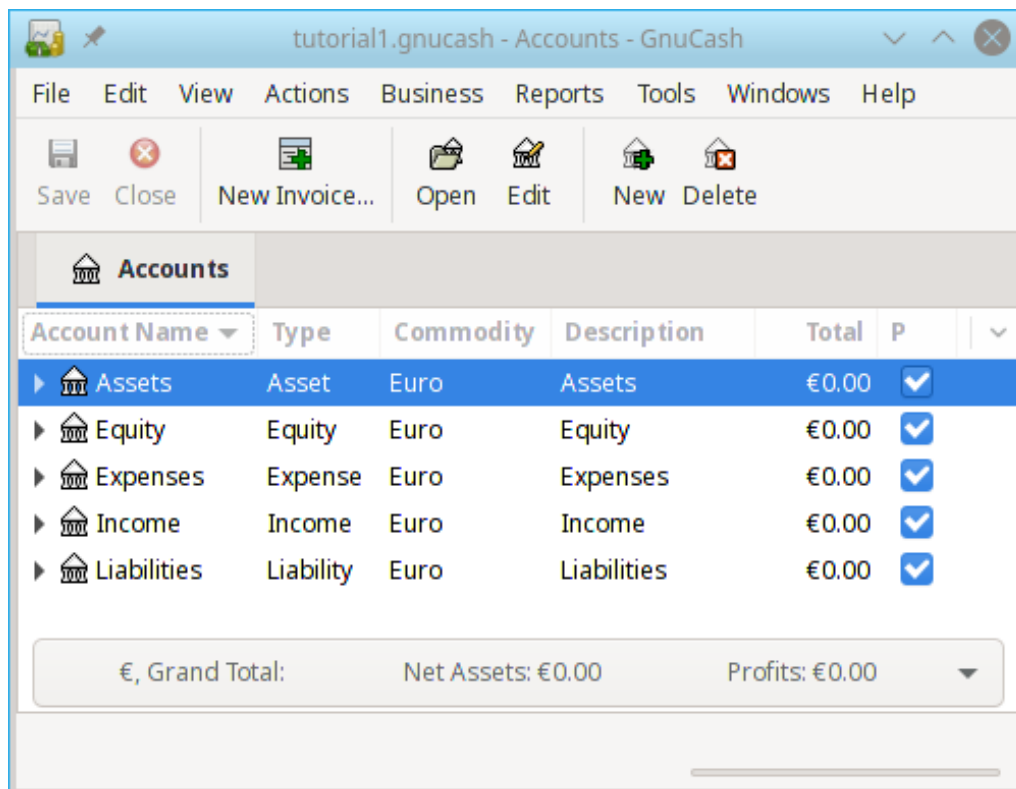
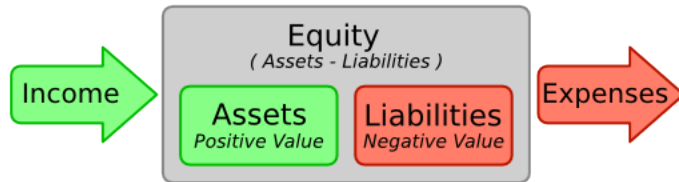


Fig. 10: Tutorial: Starting Account View of the Test File



## Balance Sheet Accounts

The three so-called *Balance Sheet Accounts* are Assets, Liabilities, and Equity. Balance Sheet Accounts are used to track the things you own or owe.

Assets is the group of things that you own. Your assets could include a car, cash, a house, stocks, or anything else that has convertible value. Convertible value means that theoretically you could sell the item for cash.

Liabilities is the group of things on which you owe money. Your liabilities could include a car loan, a student loan, a mortgage, your investment margin account, or anything else which you must pay back at some time.

Equity is the same as “net worth.” It represents what is left over after you subtract your liabilities from your assets. It can be thought of as the portion of your assets that you own outright, without any debt.

## Income and Expense Accounts

The two Income and Expense Accounts are used to increase or decrease the value of your accounts. Thus, while the balance sheet accounts simply *track* the value of the things you own or owe, income and expense accounts allow you to *change* the value of these accounts.

Income is the payment you receive for your time, services you provide, or the use of your money. When you receive a paycheck, for example, that check is a payment for labor you provided to an employer. Other examples of income include commissions, tips, dividend income from stocks, and interest income from bank accounts. Income will always increase the value of your Assets and thus your Equity.

Expense refers to money you spend to purchase goods or services provided by someone else *for early consumption*. Examples of expenses are a meal at a restaurant, rent, groceries, gas for your car, or tickets to see a play. Expenses will always decrease your Equity. If you pay for the expense immediately, you will decrease your Assets, whereas if you pay for the expense on credit you increase your Liabilities.

### 2.2.2 GnuCash Accounts

This section will show how the GnuCash definition of an account fits into the view of the 5 basic accounting types.

But first, let’s begin with a definition of an account in GnuCash. A GnuCash account is an entity which contains other sub-accounts, or that contains transactions. Since an account can contain other accounts, you often see account trees in GnuCash, in which logically associated accounts are grouped together within a common parent account.

A GnuCash account must have a unique name (that you assign) and one of the predefined GnuCash “account types”. There are a total of 12 account types in GnuCash. These 12 account types are based on the 5 basic accounting types; the reason there are more GnuCash account types than basic accounting types is that this allows GnuCash to perform specialized tracking and handling of certain accounts. There are 6 asset accounts (*Cash*, *Bank*, *Stock*, *Mutual Fund*, *Accounts Receivable*, and *Other Assets*), 3 liability accounts (*Credit Card*, *Accounts Payable*, and *Liability*), 1 equity account (*Equity*), 1 income account (*Income*), and 1 expense account (*Expenses*).

These GnuCash account types are presented in more detail below.



## Balance Sheet Accounts

The first balance sheet account we will examine is *Assets*, which, as you remember from the previous section, refers to things you own.

To help you organize your asset accounts and to simplify transaction entry, GnuCash supports several types of asset accounts:

1. **Cash** Use this account to track the money you have on hand, in your wallet, in your piggy bank, under your mattress, or wherever you choose to keep it handy. This is the most *liquid*, or easily traded, type of asset.
2. **Bank** This account is used to track your cash balance that you keep in institutions such as banks, credit unions, savings and loan, or brokerage firms - wherever someone else safeguards your money. This is the second most *liquid* type of account, because you can easily convert it to cash on hand.
3. **Stock** Track your individual stocks and bonds using this type of account. The stock account's register provides extra columns for entering number of shares and price of your investment. With these types of assets, you may not be able to easily convert them to cash unless you can find a buyer, and you are not guaranteed to get the same amount of cash you paid for them.
4. **Mutual Fund** This is similar to the stock account, except that it is used to track funds. Its account register provides the same extra columns for entering share and price information. Funds represent ownership shares of a variety of investments, and like stocks they do not offer any guaranteed cash value.

GnuCash treats account types Stock and Mutual Fund the same.

5. **Accounts Receivable (A/Receivable)** This is typically a business use only account in which you place outstanding debts owed to you. It is considered an asset because you should be able to count on these funds arriving.

Transactions involving an Accounts Receivable account should not be added, changed or deleted in any way other than by using

- post/unpost bill/invoice/voucher or
- process payment

6. **Asset** No matter how diverse they are, GnuCash handles many other situations easily. The account type "Asset", covers all assets not listed above.

GnuCash treats account types Cash, Bank and Asset the same.

Accounts are repositories of information used to track or record the kinds of actions that occur related to the purpose for which the account is established.

For businesses, activities being tracked and reported are frequently subdivided more finely than what has been considered thus far. For a more developed treatment of the possibilities, please read the descriptions presented in *Other Assets* of this Guide.

For personal finances a person can follow the business groupings or not, as they seem useful to the activities the person is tracking and to the kind of reporting that person needs to have to manage their financial assets. For additional information, consult *Other Assets* of this Guide.

The second balance sheet account is *Liabilities*, which as you recall, refers to what you owe, money you have borrowed and are obligated to pay back some day. These represent the rights of your lenders to obtain repayment from you. Tracking the liability balances lets you know how much debt you have at a given point in time.

GnuCash offers three liability account types:

1. **Credit Card** Use this to track your credit card receipts and reconcile your credit card statements. Credit cards represent a short-term loan that you are obligated to repay to the credit card company. This type of account can also be used for other short-term loans such as a line of credit from your bank.

2. Accounts Payable (A/Payable) This is typically a business use only account in which you place bills you have yet to pay.

Transactions involving an Accounts Payable account should not be added, changed or deleted in any way other than by using

- post/unpost bill/invoice/voucher or
- process payment

3. Liability Use this type of account for all other loans, generally larger long-term loans such as a mortgage or vehicle loan. This account can help you keep track of how much you owe and how much you have already repaid.

GnuCash treats account types Credit Card and Liability the same.

---

**Tip:** Liabilities in accounting act in an opposite manner from assets: *credits* (right-column value entries) increase liability account balances and *debits* (left-column value entries) decrease them. (See note later in this chapter)

---

The final balance sheet account is *Equity*, which is synonymous with “net worth”. It represents what is left over after you subtract your liabilities from your assets, so it is the portion of your assets that you own outright, without any debt. In GnuCash, use this type of account as the source of your opening bank balances, because these balances represent your beginning net worth.

There is usually only a single GnuCash equity account, called naturally enough, Equity. For companies, cooperatives etc. you can create a subaccount for each partner.

---

**Tip:** In equity accounts, credits increase account balances and debits decrease them. (See note later in this chapter)

---

---

**Note:** The accounting equation that links balance-sheet accounts is  $\text{Assets} = \text{Liabilities} + \text{Equity}$  or rearranged  $\text{Assets} - \text{Liabilities} = \text{Equity}$ . So, in common terms, the *things you own* minus the *things you owe* equals your *net worth*.

---

## Income and Expense Accounts

*Income* is the payment you receive for your time, services you provide, or the use of your money. In GnuCash, use an Income type account to track these.

---

**Tip:** Credits increase income account balances and debits decrease them. As described in [Accounting Concepts](#), credits represent money transferred *from* an account. So in these special income accounts, when you transfer money *from* (credit) the income account to another account, the balance of the income account *increases*. For example, when you deposit a paycheck and record the transaction as a transfer from an income account to a bank account, the balances of both accounts increase.

---

*Expenses* refer to money you spend to purchase goods or services provided by someone else. In GnuCash, use an Expense type account to track your expenses.

---

**Tip:** Debits increase expense account balances and credits decrease them. (See note later in this chapter.)

---

**Note:** When you subtract total expenses from total income for a time period, you get net income. This net income is then added to the balance sheet as retained earnings, which is a type of Equity account.

Below are the standard Income and Expense accounts after selecting Common Accounts in the assistant for creating a new Account Hierarchy (Actions > New Account Hierarchy...).

Example1.gnucash - Accounts - GnuCash				
File Edit View Actions Business Reports Tools Windows Help				
Accounts				
Account Name	Type	Commodity	Total	
Income	Income	Euro	€0.00	
Bonus	Income	Euro	€0.00	
Gifts Received	Income	Euro	€0.00	
Interest Income	Income	Euro	€0.00	
Other Income	Income	Euro	€0.00	
Salary	Income	Euro	€0.00	
€, Grand Total:			Net Assets: €10,000.00	Profits: €0.00

Example1.gnucash - Accounts - GnuCash				
File Edit View Actions Business Reports Tools Windows Help				
Accounts				
Account Name	Type	Commodity	Total	
Expenses	Expense	Euro	€0.00	
Adjustment	Expense	Euro	€0.00	
Auto	Expense	Euro	€0.00	
Bank Service Char	Expense	Euro	€0.00	
Books	Expense	Euro	€0.00	
Cable	Expense	Euro	€0.00	
Charity	Expense	Euro	€0.00	
Clothes	Expense	Euro	€0.00	
Computer	Expense	Euro	€0.00	
Dining	Expense	Euro	€0.00	
Education	Expense	Euro	€0.00	
Entertainment	Expense	Euro	€0.00	
Gifts	Expense	Euro	€0.00	
€, Grand Total:			Net Assets: €10,000.00	Profits: €0.00

## Other Account Types

There are some special other account types.

*Trading* Multiple currency transactions have splits in “Trading” accounts to make the transaction balance in each currency as well as in total value. See [Multiple Currencies](#) for more information.

*Money Market* and *Credit Line* are used only in the OFX importer, apparently for completeness with the specification.

### 2.2.3 Putting It All Together

Let's go through the process of building a common personal finance *chart of accounts* using the information we have learned from this chapter. A chart of accounts is simply a new GnuCash file in which you group your accounts to track your finances. In building this chart of accounts, the first task is to divide the items you want to track into the basic account types of accounting. This is fairly simple, let's go through an example.

#### Simple Example

Let us assume you have a checking and a savings account at a bank, and are employed and thus receive a paycheck. You have a credit card (Visa), and you pay monthly utilities in the form of rent, phone, and electricity. Naturally, you also need to buy groceries. For now, we will not worry about how much money you have in the bank, how much you owe on the credit card, etc. We want to simply build the framework for this chart of accounts.

Your assets would be the bank savings and checking account. Your liabilities are the credit card. Your Equity would be the starting values of your bank accounts and credit card (we do not have those amounts yet, but we know they exist). You have income in the form of a salary, and expenses in the form of groceries, rent, electricity, phone, and taxes (Federal, Social Security, Medicare) on your salary.

#### The Basic Top Level Accounts

Now, you must decide how you want to group these accounts. Most likely, you want your *Assets* grouped together, your *Liabilities* grouped together, your *Equity* grouped together, your *Income* grouped together, and your *Expenses* grouped together. This is the most common way of building a GnuCash chart of accounts, and it is highly recommended that you always begin this way.

Start with a clean GnuCash file by selecting File > New File from the menu. The New Account Hierarchy Setup assistant will start. Press Cancel to close the assistant as we don't want to use one of the predefined accounts structure; instead we will build a basic starting account structure from scratch. In the empty GnuCash window select View > New Accounts Page from the menu: the Accounts tab will open. Finally select Actions > New Account. . .

Now you are ready to build this basic starting account structure

1. Account name Assets (account type Asset, parent account New top level account)

**New Account - Assets** ✕

---

**General**
**Opening Balance**

---

**Identification**

Account name:

Account code:

Description:

Security/currency:  Select...

Smallest fraction: Use Commodity Value ▼

Account Color:  Default

Notes:

☐ Tax related      ☐ Hidden

☐ Placeholder

**Account Type    Parent Account**

Bank  
 Cash  
Asset

New top level account

Help
Cancel
OK

2. Account name Liabilities (account type Liability, parent account New top level account)
3. Account name Equity (account type Equity, parent account New top level account)
4. Account name Income (account type Income, parent account New top level account)
5. Account name Expenses (account type Expenses, parent account New top level account)

When you have created the top-level accounts, the main Account page in GnuCash should look like below.

The screenshot shows the 'Accounts' window in GnuCash. It has a menu bar (File, Edit, View, Actions, Business, Reports, Tools, Windows, Help) and a toolbar with icons for Save, Close, and other functions. Below the toolbar is a table of accounts. The 'Equity' account is selected and highlighted in blue. At the bottom, there is a summary bar showing the Grand Total, Net Assets, and Profits.

Account Name	Type	Commodity	Total
Assets	Asset	Euro	€10,000.00
Equity	Equity	Euro	€10,000.00
Expenses	Expense	Euro	€0.00
Income	Income	Euro	€0.00
Liabilities	Liability	Euro	€0.00

Summary bar: € Grand Total: Net Assets: €10,000.00 Profits: €0.00

## Making Sub-Accounts

You can now add to this basic top-level tree structure by inserting some real transaction-holding sub-accounts. Notice that the tax accounts are placed within a sub-account named *Taxes*. You can make sub-accounts within sub-accounts. This is typically done with a group of related accounts (such as tax accounts in this example).

**Tip:** Instead of selecting Actions > New Account... from the menu, you can alternatively create a new sub-account of an account by right clicking on the main account's name and selecting the New Account... entry. This will open a dialog similar to the one depicted in *The Basic Top Level Accounts* where the new sub-account will be already set as a child of the main account.

1. Account name Cash (account type Cash, parent account *Assets*)
2. Account name Checking (account type Bank, parent account *Assets*)
3. Account name Savings (account type Bank, parent account *Assets*)
4. Account name Visa (account type Credit Card, parent account *Liabilities*)
5. Account name Salary (account type Income, parent account *Income*)
6. Account name Phone (account type Expense, parent account *Expenses*)
7. Account name Electricity (account type Expense, parent account *Expenses*)
8. Account name Rent (account type Expense, parent account *Expenses*)
9. Account name Groceries (account type Expense, parent account *Expenses*)
10. Account name Taxes (account type Expense, parent account *Expenses*)
11. Account name Federal (account type Expense, parent account *Expenses:Taxes*)
12. Account name Social Security (account type Expense, parent account *Expenses:Taxes*)
13. Account name Medicare (account type Expense, parent account *Expenses:Taxes*)
14. Account name Opening Balance (account type Equity, parent account *Equity*)

After you have created these additional sub-accounts, the end result should look like below

Account Name	Type	Commodity	Total
Assets	Asset	Euro	€10,000.00
Current Assets	Asset	Euro	€10,000.00
Cash in Wallet	Cash	Euro	€0.00
Checking Account	Bank	Euro	€0.00
Savings Account	Bank	Euro	€10,000.00
Equity	Equity	Euro	€10,000.00
Opening Balances	Equity	Euro	€10,000.00
Expenses	Expense	Euro	€0.00
Income	Income	Euro	€0.00
Bonus	Income	Euro	€0.00
Gifts Received	Income	Euro	€0.00
Interest Income	Income	Euro	€0.00
Checking Interest	Income	Euro	€0.00
Other Interest	Income	Euro	€0.00
Savings Interest	Income	Euro	€0.00
Other Income	Income	Euro	€0.00
Salary	Income	Euro	€0.00
Liabilities	Liability	Euro	€0.00
Credit Card	Credit Card	Euro	€0.00

€ Grand Total: Net Assets: €10,000.00 Profits: €0.00

Save this chart of accounts with the name `gcashdata_3`, as well as `gcashdata_3emptyAccts`, as we will continue to use them in the later chapters.

You have now created a chart of accounts to track a simple household budget. With this basic framework in place, we can now begin to populate the accounts with transactions. The next chapter will cover this subject in greater detail.

## 2.3 Transactions

This chapter will give you the basic information you need to understand and use transactions in GnuCash. Whereas accounts are the framework and structure of a chart of accounts, transactions are the data which fills each account.

### 2.3.1 Basic Concepts

A *transaction* in a double entry accounting system such as GnuCash is an exchange between at least 2 accounts. Thus, a single transaction must always consist of at least two parts, a *from* and a *to* account. The *from* account is transferring value to the *to* account. Accountants call these parts of a transaction *Ledger Entries*. In GnuCash, they are called *Splits*.

A split identifies the account to which it refers, the amount of money specifically moved to or from that account, and can contain a few other specific pieces of information if needed. GnuCash supports multiple splits in a single transaction, and the splits can move money into or out of the involved accounts arbitrarily.

For example, you receive a paycheck and deposit it into your savings account at the bank. The *transaction* that occurs is that your bank savings account (an asset) received money from your income account. Two accounts are affected, and in this case there is a net increase in your equity.

Working with transactions in GnuCash is performed using what is known as the *account register*. Every account you create has an account register. It will appear familiar to you as it looks very similar to the log used to track checkbooks.

The account register is explained in the upcoming section, *The Account Register*.

## 2.3.2 The Account Register

The *account register* is the window that allows you to view or edit preexisting transactions, or add new transactions for a particular account. To open an account register from the Account Tree, double-click the account name, right click the account name and select Open Account from the menu, or use the Open button on the toolbar. GnuCash will display the account register window.

### Features of the Account Register

The *Titlebar* of the account register displays the account name. Below the *Titlebar*, the *Menubar* displays the menu items available within the account register, and the *Toolbar* contains handy buttons that help you work with the account register.

At the bottom left of the register window, GnuCash displays helpful messages as you move about the register. To the right, you can see the current account balance and the total of cleared splits.

### Choosing a Register Style

GnuCash offers several options for viewing your registers. The default style is Basic Ledger mode, which displays only the summary of splits affecting the current account. This is the style that most closely resembles other popular personal financial packages. You can choose a different register style from the View menu. There are two other view modes:

- View > Auto-Split Ledger style expands the current transaction automatically. As you highlight a new transaction in the register, the transaction automatically expands to show all splits.
- View > Transaction Journal style shows all splits for all transactions in the register, which is more like an accounting journal.

All styles permit you to view your data in either single-line or double-line format. Select View > Double Line, and you will see your transaction line expand to two register lines. Double-line mode will also display the transaction-level Notes field.

Below are screenshots that demonstrate how the Basic Ledger and Transaction Journal views differ.

For this example, let's assume that you have purchased 3 pair of Jeans for \$1,000, and have recorded the purchase as a split transaction with each pair entered on a separate split.

The below screenshots illustrate the different view modes.

First let's view the Jeans transaction from your checking account:

Account	R	Deposit	Withdrawal
2006-01-15 Buying Clothes			1,000.00
Jeans R Us	n	500.00	
Jeans R Us	n	300.00	
Jeans R Us	n	200.00	
Assets:Current Assets:Checking	n		1,000.00

Present: USD 9,000.00 Future: USD 9,000.00 Cleared: USD 0.00 Reconciled: USD 0.00 Projected Minimum: USD 9,000.00

Now, let's open the *Expenses:Clothes* account, and look at it in Basic view.



Date	Nur	Description	Tot Exper	Tot Rebate	Balance
2006-01-15		Buying Clothes	1,000.00		1,000.00
		Jeans R Us	n 500.00		
		Jeans R Us	n 300.00		
		Jeans R Us	n 200.00		
		Assets:Current Assets:Checking Account	n 1,000.00		

Present:USD 1,000.00 Future:USD 1,000.00 Cleared:USD 0.00 Reconciled:USD 0.00 Projected Minimum:USD 1,000.00

Enter a description of the transaction

Three entries appear here, but there was only the single split entry in the checking account. Further examination shows that each row has a different amount, \$200, \$300, and \$500. This demonstrates that each row in this view reflects a single split from the original transaction.

Changing to Transaction Journal mode will display only the original split transaction.

Date	Nur	Description	Tot Exper	Tot Rebate	Balance
2006-01-15		Buying Clothes	1,000.00		1,000.00
		Jeans R Us	n 500.00		
		Jeans R Us	n 300.00		
		Jeans R Us	n 200.00		
		Assets:Current Assets:Checking Account	n 1,000.00		

Present:USD 1,000.00 Future:USD 1,000.00 Cleared:USD 0.00 Reconciled:USD 0.00 Projected Minimum:USD 1,000.00

Enter a description of the transaction

## Using Entry Shortcuts

GnuCash provides several time-saving shortcuts for entering your data. When you type the first few characters of a description that you have used before, the QuickFill feature automatically fills in the rest of the transaction as you last entered it. When you type in the first characters of an account name in either the Transfer field of the transaction line or the Account field of the split line, QuickFill will automatically complete the name from your account list. It also helps you with entering sub-accounts in these fields: simply type the first characters of the parent account name followed by a : (colon) and the first characters of the sub-account name. For example, to enter *Assets:Checking*, you might type A : C and let GnuCash fill in the rest.

**Warning:** Because : is the account separator symbol, you can not use it in your account names.

**Tip:** If you really need the colon in your account names, you can select another symbol by Edit > Preferences > Accounts.

Register keyboard shortcuts also save you time, and GnuCash provides several of them. In the date field, you can type:

- – or = to increment the date and - or \_ to decrement the date
- ] or } to increment the month and [ or { to decrement the month
- M or m to enter the first date of the month
- H or h to enter the last date of the month
- Y or y to enter the first date of the year
- R or r to enter the last date of the year
- T or t to enter today's date

---

**Tip:** These date shortcuts not only work in the account register but everywhere you can enter a date.

---

---

**Tip:** Specifically for dates there's another way you can save time: you can enter partial dates. For example if you only enter one number, GnuCash will interpret it as the day in the current month of the current year. If you only enter a day and month, GnuCash will automatically append the year. The default this to take such a date in a sliding window starting 11 months before the current month. This means that if you enter 1/5 (January 5th) as date while the current date is in December 2015, the date will be completed as 1/5/16. This default can be changed via Edit > Preferences > Date/Time. You can change it to either always evaluate a date in the current calendar year or change the start of the sliding window.

---

In the Num field of the transaction line, you can type + to increment the transaction number from the last one you typed in. Typing - will decrement the number. This will also work in the Action field of the split line, if you choose to enter split numbers there. The Action field also supports QuickFill - if you type the first characters of a common action (such as Deposit), GnuCash will fill in the rest.

The Transfer field supports QuickFill of account names. You can start typing an account name and GnuCash will fill in the remaining part of the name. Typing the separator character at any time will complete the current level of the account name, leaving the cursor positioned to start the next level of account name. For example, typing A:C the standard set of account names will complete to the *Assets:Checking* account. You can also type the Menu or Ctrl+Down keys in this field to pop up a list of all account names.

In any of the amount fields, you can use a built-in calculator. Simply type in the first value, followed by +, -, \*, or /, then type in the second value. GnuCash will perform the calculation and return the resulting value to the amount field when you press the Tab key.

All of the menu items have access keys defined, and these are marked by underlined characters in the menu names. Press Alt + [underlined character] to bring up the menu, then select an item by typing its underlined character. For example, typing +Alt+ +A+ brings up the Actions menu, then typing P will split the transaction. A few of the menu items also have shortcut keys that immediately invoke the command (typically using the Ctrl key). These shortcuts are listed next to the item.

To move around the register, use these keys to save time:

- Tab to move to the next field, +Shift+ +Tab+ to move to the previous field
- Home to move to the beginning of the field, End to move to the end of the field
- Enter or ↓ to move to the next transaction, ↑ to move to the previous transaction
- **Page Up to move up one screen, Page Down** to move down one screen
- +Shift+ +Page Up+ to go to the first transaction, +Shift+ +Page Down+ to go to the last transaction

In the Reconcile window you can use these keyboard shortcuts:

- Tab moves to the next box and Shift+Tab moves to the previous box
- Space bar toggles the status between reconciled and not reconciled
- ↑ and ↓ navigate through the entries within the current box

### 2.3.3 Simple vs. Split Transactions

Every transaction in GnuCash has at least two splits, but a transaction can have more than two splits. A transaction with only two splits is called a *simple transaction*, since it only involves the current account and a single remote account. A transaction with three or more accounts is called a *split transaction*.

When the register is in Basic view, you will see a summary of the splits affecting the current account. For a simple transaction, the Transfer column will display the other account from which money is *transferred*. For a split transaction, the Transfer column will display – Split Transaction –. You can see the individual splits of each transaction by clicking the Split button in the *Toolbar* while selecting the appropriate transaction.

For split transactions, the first line of the transaction is the *transaction line*. It contains a Date, optional Num (such as a check number), transaction Description, total amount affecting the current account (Tot Deposit here), and updated account Balance after the current transaction. Note that in the expanded view, the Transfer column heading disappears, and there is no account name listed in that field. This line shows you only a summary of the transaction's effect on the current account. For more detailed information, you need to look at the individual splits that make up the transaction.

**Note:** An account register displays a transaction when that transaction has a split assigned to the account. If a given transaction has more than one split assigned to the account, then in Basic and Auto-Split views, that transaction will appear in the register one time for each split assigned to that account. In Transaction Journal view, such a transaction will only appear once in the register.

The partial lines below the transaction line are the *split lines*, and they are separated by gray lines. As you highlight one of the split lines, the column headings change to show the split-related fields:

Date	Num	Description	Account	R	Deposit	Withdrawal	Balance
05/03/06	101	Grocery Store	Expenses:Groceries	n		45.21	954.79
14/03/06		Employers R Us			670.00		1,624.79
		After Tax	Assets:Checking	n	670.00		
		Federal Tx	Expenses:Taxes:Federal	n		180.00	
		Medicare	Expenses:Taxes:Medicare	n		90.00	
		Social Security	Expenses:Taxes:Social Secur	n		60.00	
		Before Tax	Income:Salary	n		1,000.00	

Present:USD 1,624.79 Future:USD 1,624.79 Cleared:USD 0.00 Reconciled:USD 0.00 Projected Minimum:USD 1,624.79  
Assets:Checking

Each split contains an optional Action, or type of split, which you can either type in or choose from a pull-down list. The split also contains an optional Memo which describes the split. Each split affects an Account, which can be selected from a pull-down list of your accounts. The R field indicates whether the split has been reconciled. The last two columns show the amount of the split and whether money is coming into or going out of the account.

As we discussed in *Accounting Concepts*, total debits (left-column entries) must equal total credits (right-column entries) for each transaction. In the example shown above, the total debits equal the total credits, so this transaction is balanced. If you notice, the transaction line contains the same debit amount as the Checking split line. Why is this shown twice? Because the transaction line is merely a *summary* of the transaction's effect on the current account. The same transaction in a different account will have a different transaction line, one that shows the effect on that particular account. You can see this by highlighting another split line and clicking the Jump button on the *Toolbar*.

In this example, if you jump to the *Income:Salary* account, GnuCash brings up the same transaction in the *Income:Salary* - Register:

Date	Num	Description	Tot Chang	Tot Income	Balance
14/03/06		Employers R Us	1,000.00	1,000.00	1,000.00
		After Tax	670.00		
		Federal Tx	180.00		
		Medicare	90.00		
		Social Security	60.00		
		Before Tax	1,000.00		
18/04/06					

Present:USD 1,000.00 Future:USD 1,000.00 Cleared:USD 0.00 Reconciled:USD 0.00 Projected Minimum:USD 1,000.00  
Employers R Us

Note that the transaction line total now summarizes the effect on the *Income:Salary* account instead of the *Checking Account*, because you are looking at the *Income:Salary* account register. The splits are exactly the same, but the transaction line now reflects the credit to the *Income:Salary* account.

## Simple Transaction Example

Starting with the chart of accounts we created in the previous chapter `gcashdata_3`, double click on the Checking asset account. Let's add a simple transaction to the checking account. When you first create your accounts in GnuCash, it is common to start them off with an initial balance.

In the first transaction row, enter a date (eg: March, 1, 2006), a description (eg: "Opening Balance"), click on the Transfer pop-up menu and select *Equity:Opening Balances*, add a deposit value of \$1000, and press the Enter key. The account register should now appear similar to this figure:

Date	Num	Description	Transfer	R	Deposit	Withdrawal	Balance
01/03/06		Opening Balance	Equity:Opening Balance	n	1,000.00		1,000.00
05/03/06	101	Grocery Store	Expenses:Groceries	n		45.21	954.79
05/03/06		Num Description	Transfer	n	Deposit	Withdrawal	Balance

Present:USD 954.79 Future:USD 954.79 Cleared:USD 0.00 Reconciled:USD 0.00 Projected Minimum:USD 954.79  
Sunday 05 March 2006

Setting the starting balances of an account is an example of a simple two account transaction. In this case, affecting the *Assets:Checking* and the *Equity:Opening Balances* accounts.

As another example of a simple 2 account transaction, add another transaction to describe the purchase of \$45.21 worth of groceries. From within the *Assets:Checking* account, you would set Transfer to *Expenses:Groceries*. The account register should now appear:

Date	Num	Description	Transfer	R	Deposit	Withdrawal	Balance
01/03/06		Opening Balance	Equity:Opening Balance	n	1,000.00		1,000.00
05/03/06	101	Grocery Store	Expenses:Groceries	n		45.21	954.79
05/03/06		Num Description	Transfer	n	Deposit	Withdrawal	Balance

Present:USD 954.79 Future:USD 954.79 Cleared:USD 0.00 Reconciled:USD 0.00 Projected Minimum:USD 954.79  
Sunday 05 March 2006

## Split Transaction Example

The need for 3 or more splits in a transaction occurs when you need to split either the "from" or the "to" account in a transaction into multiple accounts. The classic example of this is when you receive a paycheck. Your take home pay from a paycheck will typically be less than your net pay, with the difference being due to taxes, retirement account payments, and/or other items. Thus, the single transaction of you receiving a paycheck involves other accounts besides simply *Assets:Checking* and *Income:Salary*.

To add the paycheck split transaction from the *Assets:Checking* account register window, click on a new transaction line and click Split. Note that if you have set your register view to Auto-Split or Transaction Journal, the splits will be opened for you. Enter the description of this transaction on the first line (e.g. "Employers R Us"). In the split lines below this, enter the various splits that make up this transaction, one by one. To enter the splits, first choose the account, then enter the amount by which to change the account. Keep in mind that when in an asset account register, amounts entered in the left column increase the account balance, while amounts entered in the right column decrease the balance (for more about this, see [Transactions](#)). Tab or click the next split line and repeat the process. Note that if you are using the keyboard to navigate the transaction, use Tab to move from field to field, as using Enter will commit the transaction and create splits to an Imbalance account.

**Note:** When creating a transaction in GnuCash, splits can be entered in any order. However, when the transaction is closed (either when leaving the transaction, or when pressing the Enter key), all debit splits will jump ahead of all credit splits.

In this example, choose the deposit account (*Assets:Checking*) and then enter the amount that is being deposited into it (e.g. \$670). Follow this by entering the amounts for the various taxes: *Expenses:Taxes:Federal* account, \$180; *Expenses:Taxes:Medicare* account, \$90; and *Expenses:Taxes:Social Security* account, \$60. Finally, enter the gross total of your paycheck (\$1,000 in this example) as a withdrawal transfer from *Income:Salary*.

The final split should look like *figure\_title*. Remember to press Enter after finishing the entry. But you should also know that when you press Enter, the split view will be “folded” back into a simplified transaction view. The splits are still there; you just have to click Split to view them. See *Features of the Account Register* for details.

Date	Num	Description		Tot Depos	Tot Withdra	Balance
05/03/06	101	Grocery Store	Expenses:Groceries	n	45.21	954.79
14/03/06		Employers R Us		670.00	Withdrawal	1,624.79
		After Tax	Assets:Checking	n	670.00	
		Federal Tx	Expenses:Taxes:Federal	n	180.00	
		Medicare	Expenses:Taxes:Medicare	n	90.00	
		Social Security	Expenses:Taxes:Social Secur	n	60.00	
		Before Tax	Income:Salary	n	1,000.00	

Present: USD 1,624.79 Future: USD 1,624.79 Cleared: USD 0.00 Reconciled: USD 0.00 Projected Minimum: USD 1,624.79  
Employers R Us

Fig. 11: Entering a split transaction

### 2.3.4 Reconciliation

Once transactions have been entered into GnuCash, it is important to verify that they agree with the records of your financial institution. This verification process is known as **reconciliation**, and it is key to determine whether your records are accurate. Most commonly, you will check transactions against bank statements, although you can use this process to verify any transaction.

GnuCash keeps track of the reconciliation status of each transaction. The reconciliation status of a transaction is shown by the reconciliation R field in a register: *y* indicates that a transaction has been reconciled, *n* indicates that it has not, and *c* indicates that it has been cleared, but not reconciled. A split marked cleared signifies that you got some kind of confirmation that the institution accepted the order (e.g. you have the restaurant’s receipt for your credit card purchase). You can toggle the reconciliation status between *n* and *c* by clicking in the R field; you can set it to *y* by using *Reconcile windows*.

At the bottom of the account window, there are (among others) two running balances (the cleared and reconciled balance), and the total balance. The former balances should correspond to how much money the bank thinks you have in your account, while the latter includes outstanding transactions.

For example, when you write a check for something, you should enter the transaction into GnuCash. The reconciliation R field of the transaction will initially contain *n* (new). Your new entry will contribute to the total balance, but not to the cleared and reconciled balance. Later, if you got some confirmation that the check has been cashed, you might click on the transaction’s R field to change it to *c* (cleared). When you do this, the cleared balance will change to include this amount. When the bank statement arrives, you can then compare it to what you’ve recorded in GnuCash by opening the reconciliation window. There, you will be able to change the R field to *y* (reconciled).

**Note:** You cannot reconcile directly in a register window by clicking in the R field. You must use the reconciliation window. Once a transaction has been marked *reconciled*, it can no longer be easily changed without breaking the

*Starting Balance* of the next reconciliation.

---

**Warning:** It is important to understand that reconciliation is done for a given date, and when you reconcile an account based on a statement from a given date, you are reconciling **all transactions prior to that date**. Therefore, if you add or modify transactions that predate your last reconciliation, your **reconciled** balances will be thrown off.

## Reconcile windows

The reconciliation windows are used to reconcile a GnuCash account with a statement that a bank or other institution has sent you. Reconciliation is useful not only to double-check your records against those of your bank, but also to get a better idea of outstanding transactions, e.g. uncashed checks.

To use the reconciliation windows, select an account from the account tree and click on Actions > Reconcile. A window like the one below will appear in which you can enter the reconcile information.

Date	Num	Description	Amount	R
03/01/06		Opening Balance	1,000.00	✓
03/05/06		Employers R Us	670.00	

Total: \$1,000.00

Date	Num	Description	Amount	R
03/14/06	101	Grocery Store	45.21	✓

Total: \$45.21

Statement Date: 04/18/06  
Starting Balance: \$0.00  
Ending Balance: \$1,624.79  
Reconciled Balance: \$954.79  
Difference: \$670.00

In the initial reconcile window, some Reconcile Information need to be entered.

**Statement Date** The date of the statement you will be reconciling against.

---

**Tip:** Click on the down arrow in the right of this field to open a calendar

---

**Starting Balance** This is a non-editable item which displays the balance from the previous reconciliation. It should match the starting balance in your statement.

**Warning:** Sometimes, the opening balance in GnuCash does not match that found on your statement. This can happen the first time you reconcile your account or when a previously-reconciled transaction is de-reconciled or deleted.

---

**Note:** The first time you reconcile your account, the starting balance will be 0.00, thus probably not the *opening balance* of your account. When you reconcile the account, the *opening balance* for the account will be included in the reconciliation, and the result should balance.

---

---

**Tip:** In the case when a previously-reconciled transaction is accidentally de-reconciled, you can simply re-reconcile the transaction along with the transactions on the current statement, and the result should balance.

The case of accidentally deleting a previously-reconciled transaction presents more of a challenge; if you cannot determine what was deleted and restore it to the register, you will have to create a dummy transaction to get the reconciliation to finish.

---

**Caution:** While the latter case does not matter for your private accounting, you should have a really good explanation for your auditors, if you are the accountant of a club or a company.

**Ending Balance** This field should be filled with the ending balance as it appears in the statement.

---

**Note:** GnuCash automatically fills this field with the Present balance as shown in the lower part of the account's register.

---

**Include Sub-accounts** Check this option if you want to include in the reconciliation the transactions that belongs to the sub-accounts of the currently selected account. Note that all sub-accounts must be in the same commodity as the parent for the option to be enabled.

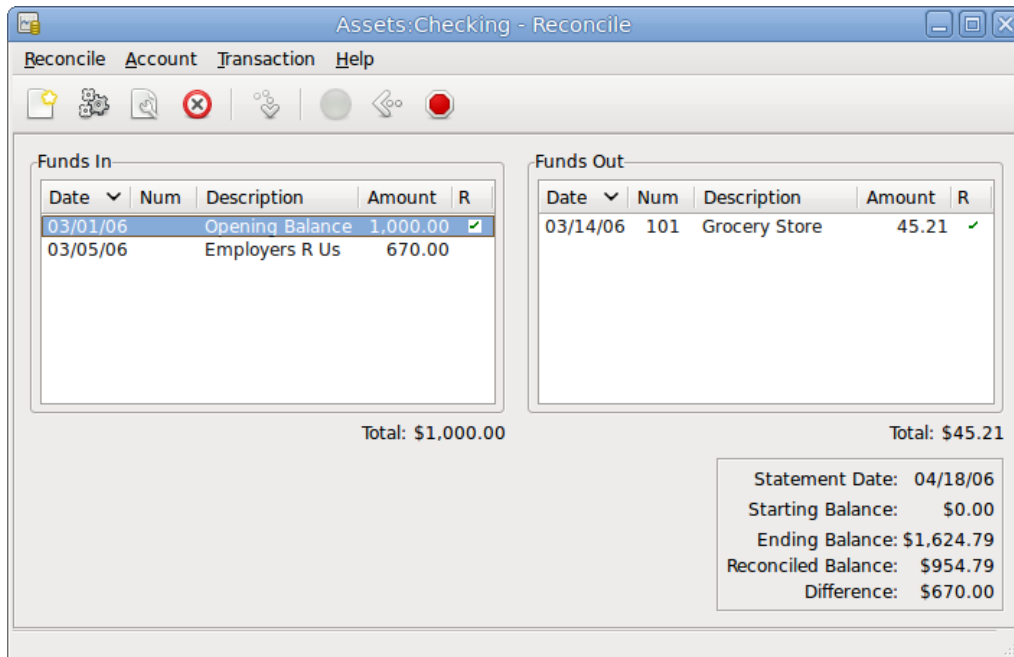
**Enter Interest Payment** Clicking this button opens a new window that allow you to enter an interest transaction to the account to be reconciled.

---

**Tip:** The Interest Payment window might be opened automatically when you start a reconciliation for an account of the type *Bank*, *Credit*, *Mutual*, *Asset*, *Receivable*, *Payable*, and *Liability*. If you want to disable this behavior for any of the previous accounts, go to the Register tab of the GnuCash Preferences and uncheck the Automatic interest transfer option. Alternatively, to disable this behavior only for the selected account, press the No Auto Interest Payments for this Account button in the Interest Payment window.

---

Then, click on the Ok button, and you will see the transactions listing reconcile window:



The two panes called Funds In and Funds Out, lists all the unreconciled transactions that belongs to the account that is going to be reconciled. The R columns show whether the transactions have been reconciled.

Now, examine each item on the bank statement, and look for the matching item in the Reconcile window.

If you cannot find a transaction, then perhaps you forgot to enter it, or did not know that the transaction had happened. You can use the New button on the *Toolbar*, or the Transaction > New menu item in the menu, to open a register window and enter the missing transaction. The new item will appear in the Reconcile window when you press the Enter button in the register after entering the transaction.

When you find the item in the Reconcile window, compare the amount in the item to the amount on the statement. If they disagree, you may have made an error when you entered the transaction in GnuCash. You can use the Edit button on the *Toolbar*, or the Transaction > Edit item, to open a register window and correct the transaction.

If the amounts agree, click on the item in the Reconcile window. A check mark will appear in the R column aside the selected transaction. GnuCash will automatically update the amounts in the lower right summary pane.

---

**Tip:** You can use the up/down arrow keys to scroll to the item, the space key to mark the item as reconciled and the Tab key to switch panes.

---

You then repeat this for each item that appears on the bank statement, verifying that the amounts match with the amounts in GnuCash, and marking off transactions in GnuCash as they are reconciled.

---

**Tip:** You can check or uncheck the reconcile status of all transactions in either pane by clicking on a transaction in the required pane then clicking Ctrl-A followed by space. This will toggle the reconcile status of all transactions in the pane to either checked or unchecked state. This procedure can be repeated to achieve the desired status for the transactions in the pane.

---

At the bottom of the Reconcile window there is a Difference field, which should show 0.00 when you are done reconciling. If it shows some other value, then either you have missed transactions, or some amounts may be incorrect in GnuCash. (Or, less likely, the bank may have made an error.)



**Note:** Under some circumstances, it may be difficult or impossible to determine why an account will not reconcile. If you are unable to correct the discrepancy between your books and a statement, GnuCash includes a Balance button on the *Toolbar* that will automatically create a balancing entry for you in the amount that cannot be reconciled.

To use this, carry out the full reconciliation (marking all transactions that you can identify) and then click this button. GnuCash will create a balancing entry for the remaining discrepancy that uses the *Special Accounts:Orphan-XXX* account (where “XXX” represents your currency). The Reconcile window will close; re-opening it will allow you to check the newly-created balancing entry and finish the process.

When you have marked off all the items on the bank statement and the difference is 0.00, press the Finish button on the *Toolbar* or select Reconcile > Finish from the menu. The Reconcile window will close. In the register window, the R field of the reconciled transactions will change to y.

In this case, we have not received all the information yet, so we simply press the Postpone button, so we can continue at a later stage. Observe that the R column indicates we cleared (c) two transactions. They have not been reconciled yet, but we have verified these two transactions so they have been marked as cleared. If you look at the *Statusbar* at the bottom of the account register, you will see a summary of what has been reconciled and what has been cleared (Cleared:USD 954.79 Reconciled:USD 0.00)

Date	Num	Description	Transfer	R	Deposit	Withdrawal	Balance
01/03/06		Opening Balance	Equity:Opening Balance	c	1,000.00		1,000.00
05/03/06	101	Grocery Shop	Expenses:Groceries	c		45.21	954.79
14/03/06		Employers R Us	Income:Salary	n	670.00		1,624.79
05/03/06				n			

Present:USD 1,624.79 Future:USD 1,624.79 Cleared:USD 954.79 Reconciled:USD 0.00 Projected Minimum:USD 1,624.79

## 2.3.5 Scheduled Transactions

Scheduled transactions are made to help entering repetitive money operations, like subscriptions, insurances or taxes. By using scheduled transactions, you only have to enter the concerned transaction once, set a few parameters like start date, frequency and a little description, and then GnuCash will tell you whenever a scheduled transaction is ready to be created, and create it for you.

In this howto, we'll take a monthly Internet subscription of 20 USD as example, which is taken on the 28th of each month.

In GnuCash, there are two ways of creating scheduled transactions, *from the ledger* or from the *Scheduled Transactions Editor*.

### Creating from the Ledger

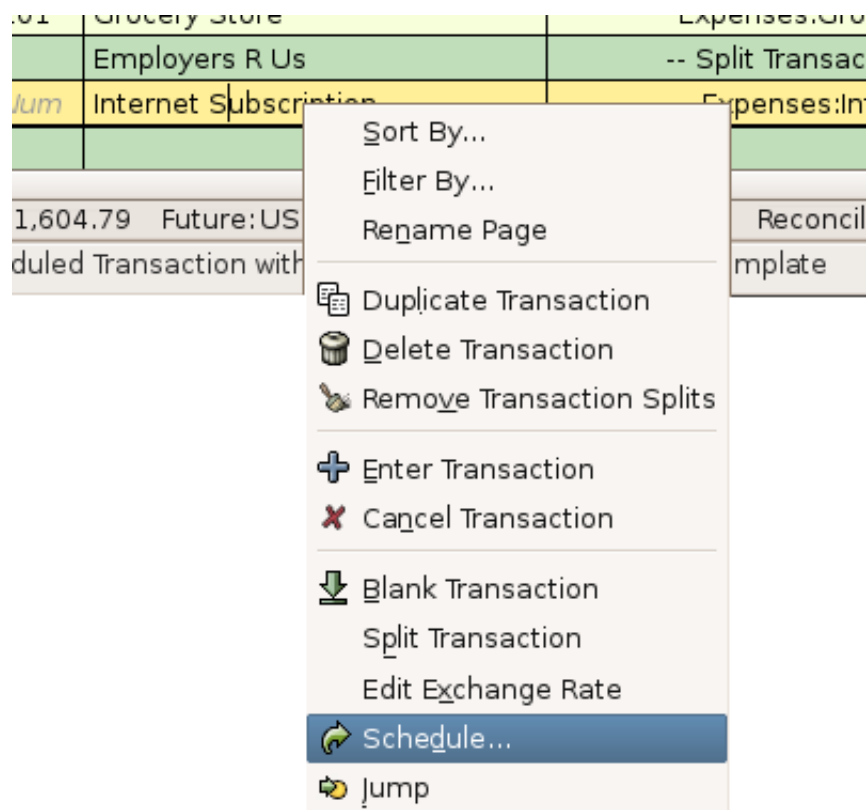
Enter the first occurrence of your to-schedule transaction in the ledger. In the Transfer field for this transaction, type *Expenses:Internet* as shown in the next screenshot.

Date	Num	Description	Transfer	R	Deposit	Withdrawal	Balance
01/03/06		Opening Balance	Equity:Opening Balance	n	1,000.00		1,000.00
05/03/06	101	Grocery Store	Expenses:Groceries	n		45.21	954.79
14/03/06		Employers R Us	-- Split Transaction --	n	670.00		1,624.79
28/03/06		Internet Subscription	Expenses:Internet	n		20.00	1,604.79
28/03/06			Transfer	n			

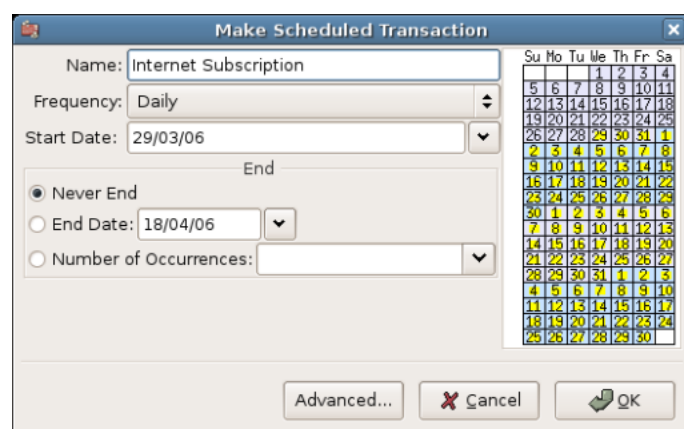
Present:USD 1,604.79 Future:USD 1,604.79 Cleared:USD 0.00 Reconciled:USD 0.00 Projected Minimum:USD 1,604.79

**Note:** Since we did not create the *Expenses:Internet* account, GnuCash will prompt us to create it.

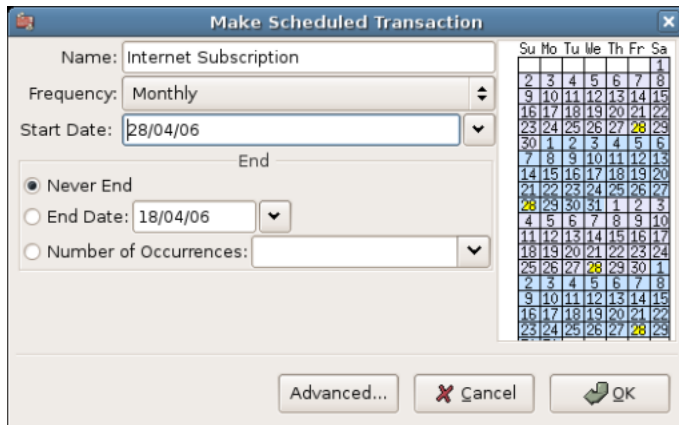
Then you right click on your transaction and select Schedule...



A window like this will appear:



Let's fill the values, we know that the subscription is taken on the 28th each month, and the next one is for next month (since we entered the one for this month manually) :

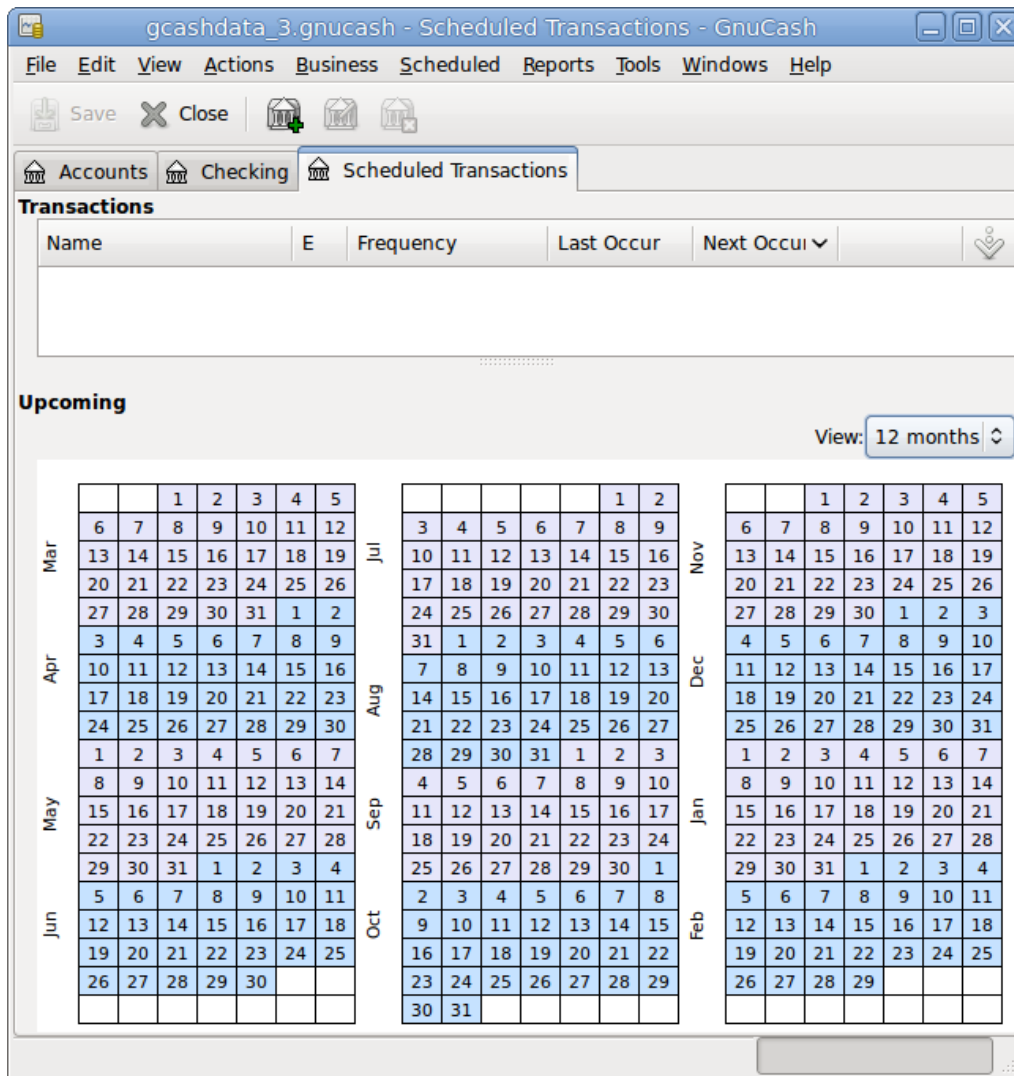


Click the OK button, and the transaction will be scheduled. GnuCash now has memorized this scheduled transaction and on the 28th of next month, it will pop up a window asking if it should create it (see far below for a screenshot of this window).

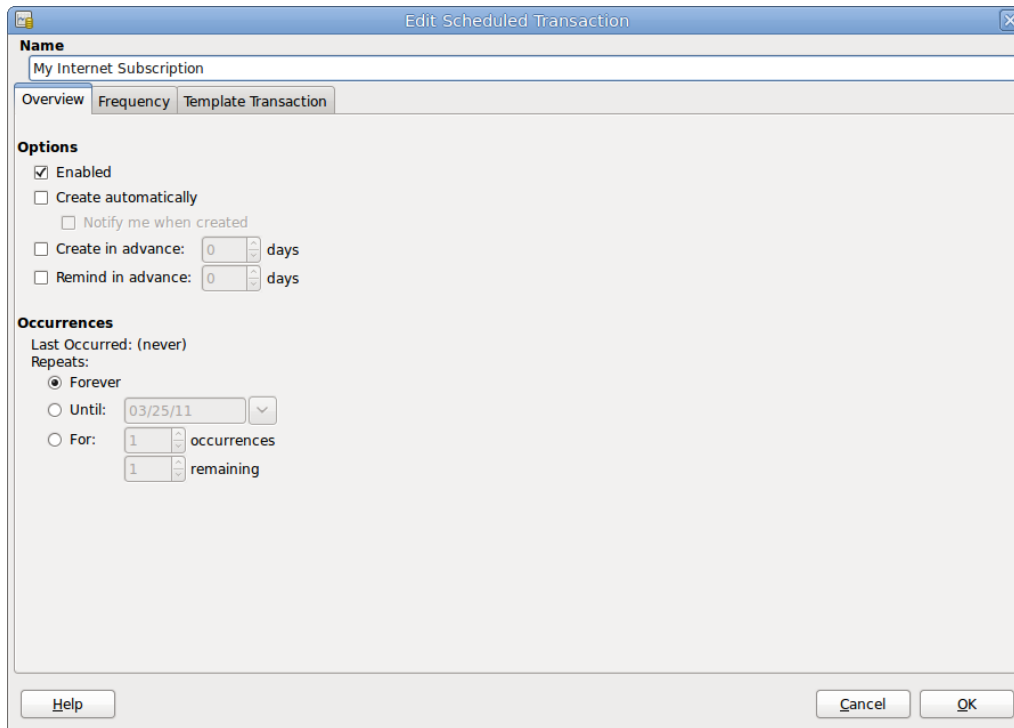
### Creating from the Editor

Another way of entering a scheduled transaction is from the Scheduled Transaction Editor, it may be faster if we have several scheduled transactions to create at once.

From the main accounts windows, select Actions > Scheduled Transactions > Scheduled Transaction Editor from the *Menubar*. A new Scheduled Transactions tab will be opened in the current GnuCash window as shown above:



This tab contains a list, now empty, of all the scheduled transactions. Let's create a new one by clicking on the New button in the *Toolbar*. A window like the one below will pop up:



First, let's enter a name for this new scheduled transaction in the top of the window.

---

**Note:** This name will only identify the transaction in the Edit Scheduled Transaction window, it will never appear in the ledger.

---

In the Options pane of the Overview tab you have four options:

**Enable** Sets the status of the scheduled transaction.

**Create automatically** If enabled, will insert this transaction in the ledger without asking you before (see below). If needed, you can be advised when the transaction is entered by checking the Notify me when created option.

**Create in advance** Sets how many days in advance the transaction will be created.

**Remind in advance** Sets how many days in advance a reminder is presented. This can be used, for example, when you have to pay something by check, and a reminder one week before allows you to send your check before the deadline.

The Occurences pane allows you to tell GnuCash that this scheduled transaction won't last for ever. For example if you are repaying a loan, you can enter the loan end date or the number of occurences left.

Select now the Frequency tab in the Edit Scheduled Transaction window. Here you can set the time-related options of the transaction.

**Edit Scheduled Transaction**

**Name**  
My Internet Subscription

**Overview** **Frequency** **Template Transaction**

Frequency: Monthly

Start Date: 04/28/06

Every 1 months.

On the 28th except on weekends: No change

View: 12 months

Calendar view showing months from April to March. The 28th of each month is highlighted in yellow.

Buttons: Help, Cancel, OK

**Frequency** Sets the basic frequency of the transaction, with options for once, daily, weekly, semi-monthly and monthly. In this example, this is set to monthly.

**Note:** It is possible to set a transaction to occur at intervals *other* than those listed in the Frequency list, by changing the setting in the Every control (see below).

**Start Date** Sets when the transaction will begin. In the example, this would be set to the start of the next month.

**Every** This option allows you to schedule transactions by multiplies of the value in Frequency. For example, to create a transaction that runs every 3 weeks, set Frequency to Weeks and Every to 3.

**On the and Except on weekends** Sets the day of the month that the transaction is scheduled, and controls what GnuCash will do when the day occurs on a weekend day.

**Note:** We know that the subscription is taken on the 28th each month, so the Start date will be November 1, 2003 (assuming November is the next month), the Frequency will be Monthly, it will be taken every month on the 28th. Since in our example the internet subscription is automatically taken from the account, we have no need to create it in advance, nor give an end date.

When the elements on this tab are filled in, GnuCash will highlights the calendar below to indicate when future transactions will be run.

Finally select the Template Transaction tab and enter your transaction in the lower part as you would do in the ledger, with the only difference of having no date.

Now, you should have a window like this:

**Edit Scheduled Transaction**

Name: My Internet Subscription

Overview Frequency **Template Transaction**

Edit Transaction View Actions

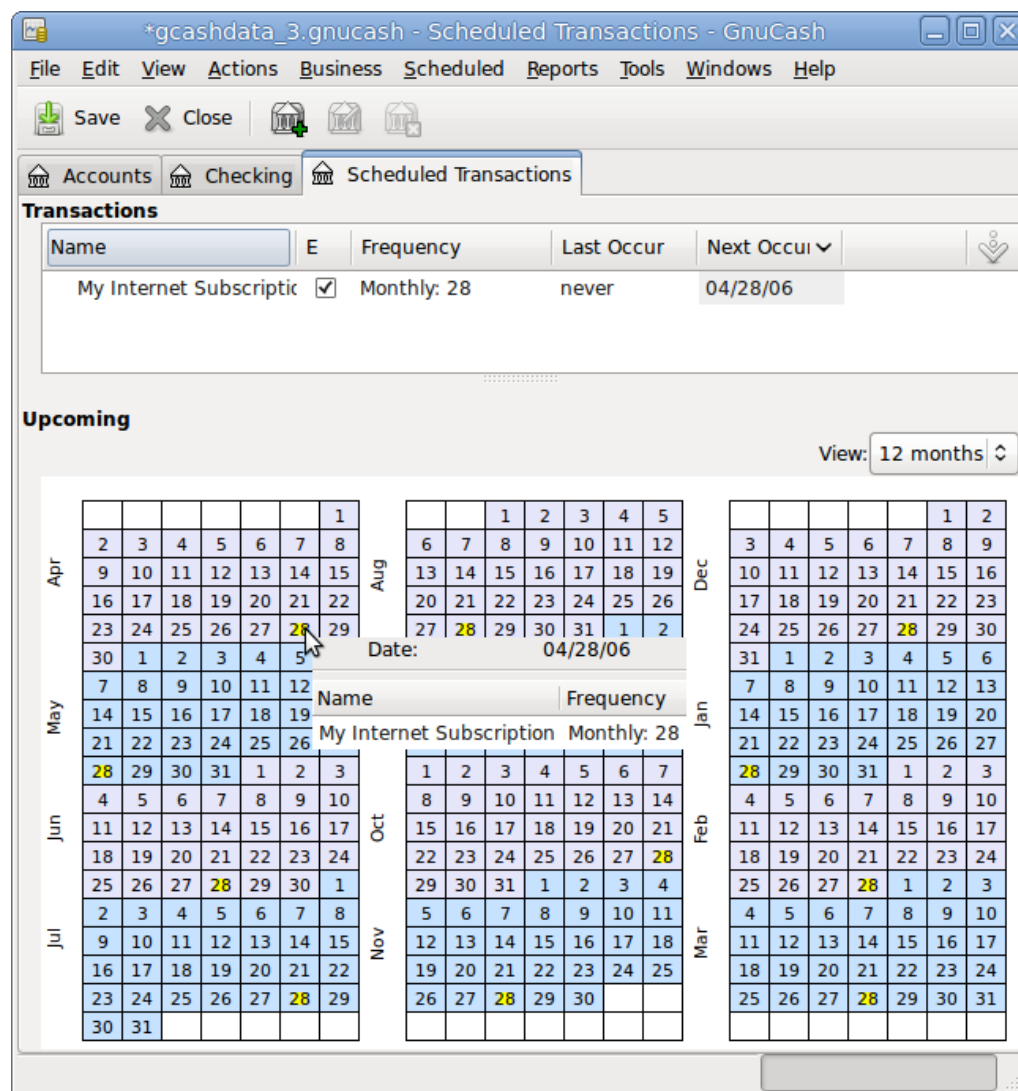
Date	Num	Description	Tot Funds In	Tot Funds Out
<b>Scheduled</b>				
		Internet Subscription		
		Assets:Checking	n	20
		Expenses:Internet	n 20	
<b>Scheduled</b>		Description		Tot Funds In
		Notes		Tot Funds Out
			n	

Enter the transaction number, such as the check number

Help Cancel OK

Remember to click on the Enter icon, to validate and enter the transaction.

Now click OK, it takes you to the Scheduled Transactions tab, now showing one item in the Transactions list:



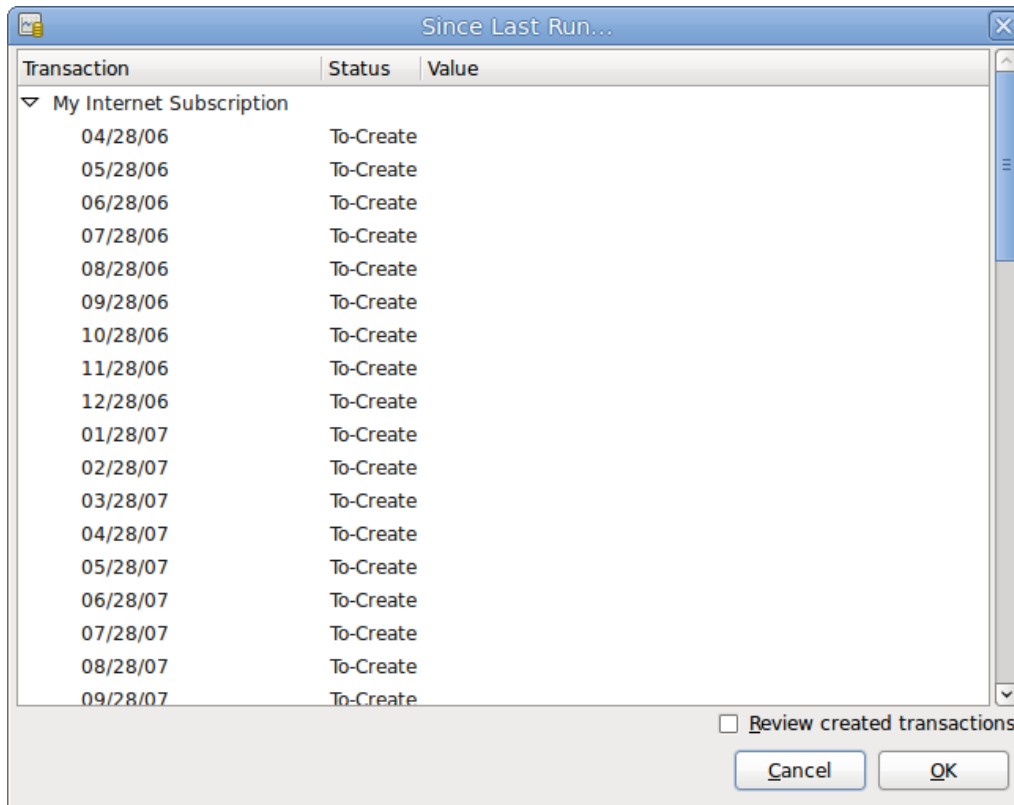
**Tip:** If you click in the calendar part on the first day of one month, a small window, following your mouse, will show you what is planned for this day. To make the small window disappear again, just click in it one more time.

You can now close the Scheduled Transaction tab, and save your work.

**Note:** What comes below is just an illustration, and is not meant to be entered into the GnuCash database at this stage. As per this example, the below dialogs will appear when the scheduled transaction is supposed to run.

From now on, when GnuCash is launched and a scheduled transaction is scheduled or need to be entered, you may see a Since Last Run... window summarizing the scheduled transactions operations (Reminder, To-Create etc... a better description of each option can be found in the GnuCash manual):





You can click on the Status column to change the scheduled transaction status if needed. Then just click on Ok to close the Since Last Run... window and to apply the pending operations. Select Cancel to skip entering in the ledger the pending operations.

If the Status field was set to To-Create, then if you take a look in your bank account register, you'll see the transaction has been created:

**Tip:** If you enable the Review created transactions option in the lower right of the Since Last Run... window, GnuCash will open, after pressing Ok, the register tab of each account where the scheduled transactions were entered automatically.

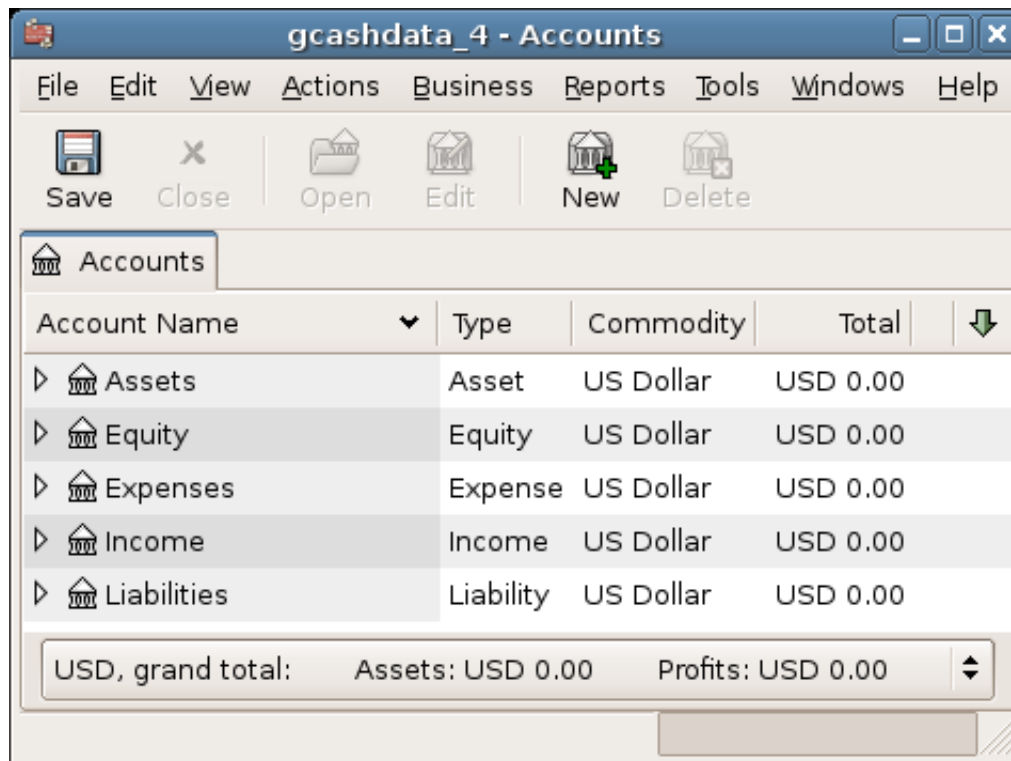
**Note:** If you entered the transaction for 28 of April at this stage, then please *revert back* to the last saved GnuCash file by selecting File > Open and choosing the previously saved `gcashdata_3` GnuCash file.

### 2.3.6 Putting It All Together

In the previous sections of this chapter the concepts and mechanics of working with transactions in GnuCash have been discussed. This section will expand upon the chart of accounts initially built in the previous chapter, by setting some opening balances, adding transactions and a scheduled transaction.

#### Open GnuCash file

Start with opening the previous datafile we stored, `gcashdata_3emptyAccts`, and store it as `gcashdata_4` directly. The main window should look something like this:

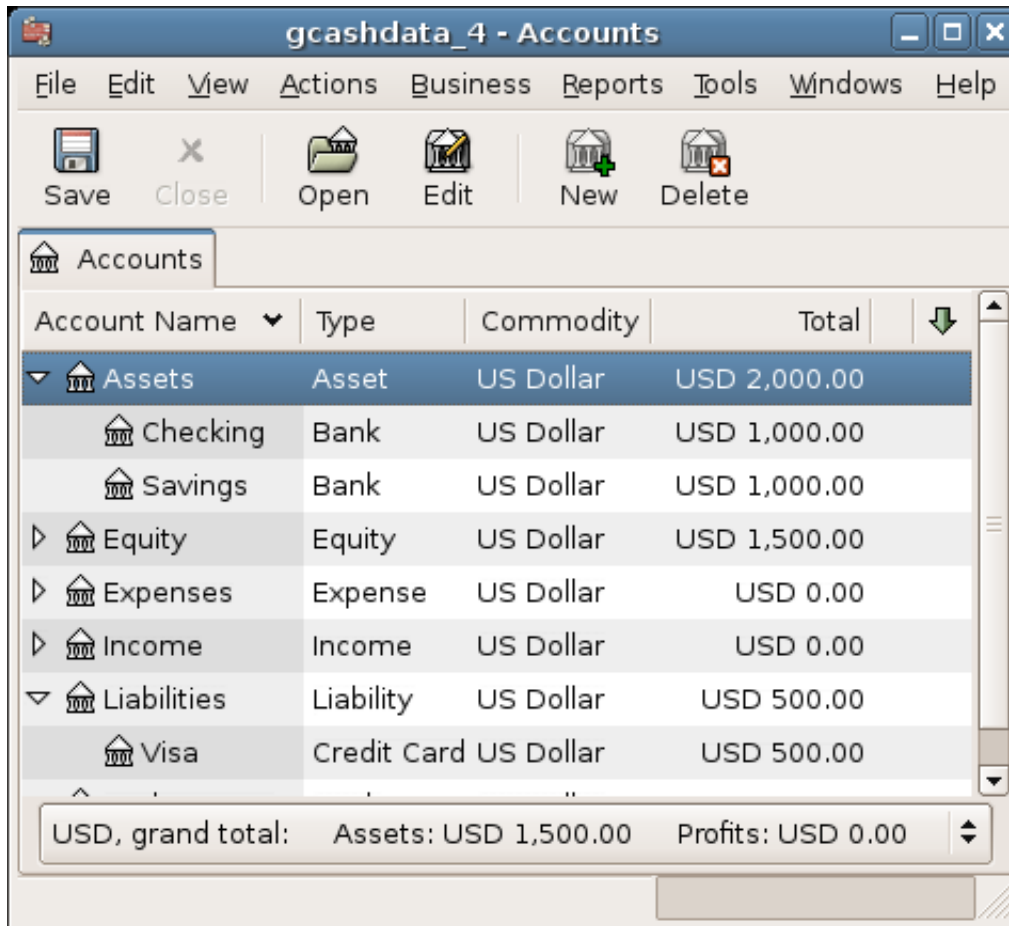


## Opening Balances

As shown earlier in *Simple Transaction Example* with the *Assets:Checking* account, the starting balances in an account are typically assigned to a special account called *Equity:Opening Balance*. To start filling in this chart of account, begin by setting the starting balances for the accounts. Assume that there is \$1000 in the savings account and \$500 charged on the credit card.

1. Open the *Assets:Savings* account register. Select View from the menu and check to make sure you are in Basic Ledger style. You will view your transactions in the other modes later, but for now let's enter a basic transaction using the basic default style.
2. From the *Assets:Savings* account register window, enter a basic 2 account transaction to set your starting balance to \$1000, transferred from *Equity:Opening Balance*. Remember, basic transactions transfer money from a source account to a destination account. Record the transaction (press the Enter key, or click on the Enter icon).
3. From the *Assets:Checking* account register window, enter a basic 2 account transaction to set your starting balance to \$1000, transferred from *Equity:Opening Balance*.
4. From the *Liabilities:Visa* account register window, enter a basic 2 account transaction to set your starting balance to \$500, transferred from *Equity:Opening Balance*. This is done by entering the \$500 as a charge in the *Visa* account (or decrease in the *Opening Balance* account), since it is money you borrowed. Record the transaction (press the Enter key, or click on the Enter icon).

You should now have 3 accounts with opening balances set. *Assets:Checking*, *Assets:Savings*, and *Liabilities:Visa*.



The screenshot shows the 'gcashdata\_4 - Accounts' window. It features a menu bar (File, Edit, View, Actions, Business, Reports, Tools, Windows, Help) and a toolbar with icons for Save, Close, Open, Edit, New, and Delete. Below the toolbar is a tab labeled 'Accounts'. The main area displays a table of accounts with columns: Account Name, Type, Commodity, and Total. The accounts are organized into a tree structure with expandable/collapsible icons. The 'Assets' account is expanded, showing 'Checking' and 'Savings' sub-accounts. The 'Liabilities' account is also expanded, showing a 'Visa' credit card account. At the bottom, a summary bar shows the grand total for USD, assets, and profits.

Account Name	Type	Commodity	Total
Assets	Asset	US Dollar	USD 2,000.00
Checking	Bank	US Dollar	USD 1,000.00
Savings	Bank	US Dollar	USD 1,000.00
Equity	Equity	US Dollar	USD 1,500.00
Expenses	Expense	US Dollar	USD 0.00
Income	Income	US Dollar	USD 0.00
Liabilities	Liability	US Dollar	USD 500.00
Visa	Credit Card	US Dollar	USD 500.00
USD, grand total:    Assets: USD 1,500.00    Profits: USD 0.00			

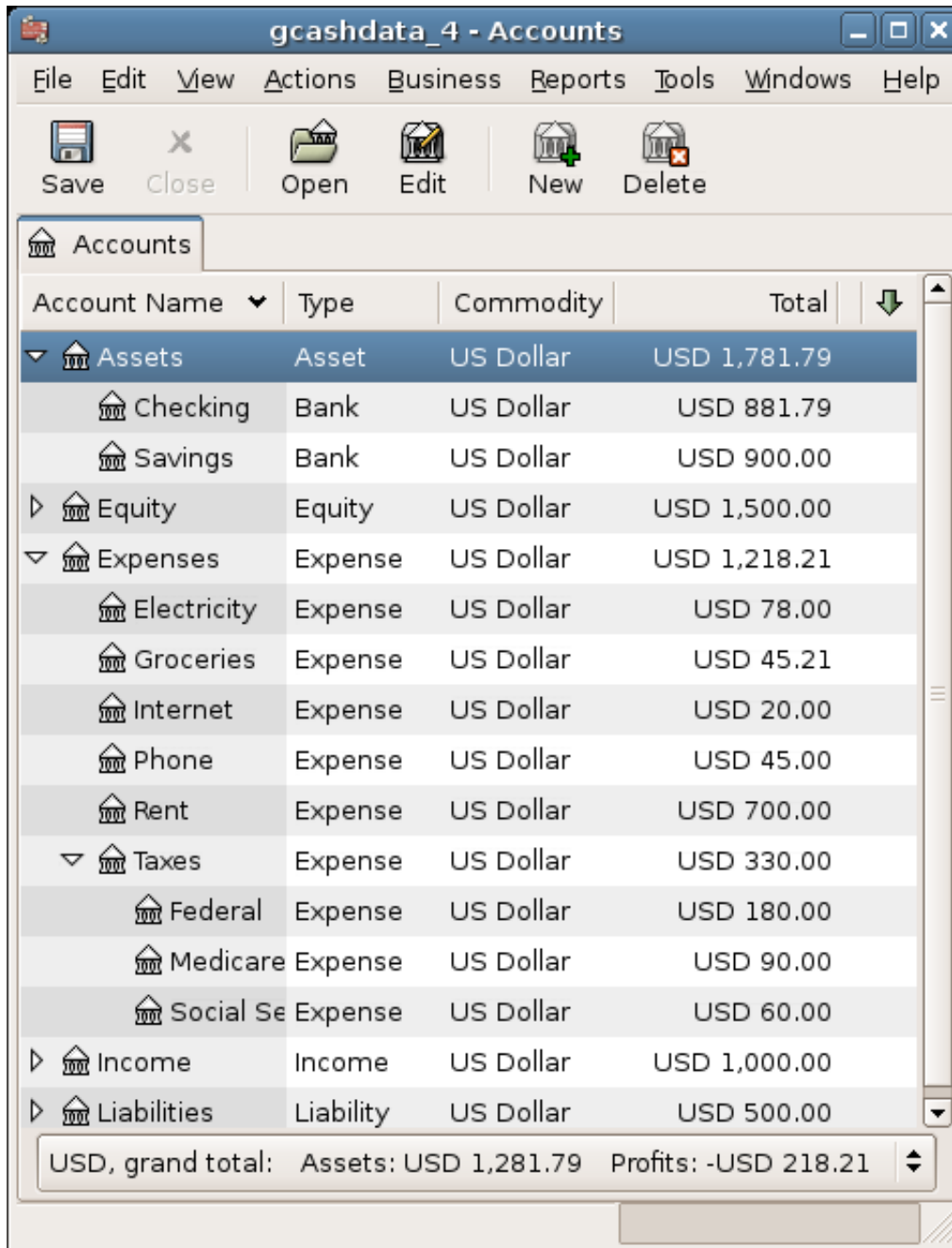
### Additional Transaction Examples

Now add some more transactions to simulate a month's expenses. During the month, \$78 is spent on electricity, \$45 on phone, and \$350 on rent. All paid by check. We also spent \$45.21 on groceries, received \$670 as salary, and paid our Internet bill this month. Finally, let's move \$100 from the savings account to the checking account.

1. Open the *Expenses:Electricity* account register and enter a simple 2 account transaction to pay the \$78 electrical bill at the end of the current month (eg: March 28, 2006). Enter a description (eg: Light Company) and the check number (eg: 102). The Transfer account should be *Assets:Checking*.
2. Open the *Assets:Checking* account register and enter a simple 2 account transaction to pay the \$45 phone bill at the end of the current month (eg: March 28, 2006). Enter a description (eg: Phone Company Name) and the check number (eg: 103). The Transfer account should be *Expenses:Phone*. Notice that you can enter expense transactions from either the credit side (the expense accounts) or the debit side (the asset account).
3. Open the *Expenses:Rent* account register and enter a simple 2 account transaction to pay the \$350 in rent at the end of the current month (eg: March 28, 2006). Enter a description (eg: April Rent) and the check number (eg: 104). The Transfer account should be *Assets:Checking*.
4. Duplicate this transaction using the Duplicate button in the *Toolbar*. Start by clicking on the current rent transaction, and click on the Duplicate icon. Enter the transaction date a month out in the future (eg: April 28, 2006), and notice the blue line separator that GnuCash uses to separate future transactions from current ones. In this way, you can enter transactions before they occur.

You could also set up a scheduled transaction to pay your rent, since the value of the rent is likely to be constant for the foreseeable future.

- Start by clicking on the current (April 28) rent transaction, and click on the Schedule icon
  - Change to Monthly, change description if needed and press OK
5. To transfer money from your savings account to your checking account, open the *Assets:Savings* account register, add a new transaction setting the Transfer to *Assets:Checking* in the amount of \$100 (date 6 March, 2006).
  6. As another example of a simple 2 account transaction, add another transaction to describe the purchase of \$45.21 worth of groceries on 5 of March. From within the *Assets:Checking* account, you would set Transfer to *Expenses:Groceries*. The account register should now appear:
  7. To add a paycheck transaction from the *Assets:Checking* account register window, click on a new transaction line, and click on Split. First enter the description of this transaction on the first line (eg: “Employers R Us”), as well as the date (14 March). In the “split” line below this, enter the deposit into *Assets:Checking* (eg:\$670). Follow this with the various tax deposits (*Assets:Checking* (eg: \$670), *Expenses:Taxes:Federal* account (eg: \$180), *Expenses:Taxes:Medicare* account (eg: \$90), and *Expenses:Taxes:Social Security* account (eg: \$60)) and lastly the gross total of your paycheck (eg: \$1000) as a withdrawal transfer from *Income:Salary*.
  8. You also need to pay for your Internet subscription of 20 USD on the 28th.



The screenshot shows the 'gcashdata\_4 - Accounts' window. It features a menu bar (File, Edit, View, Actions, Business, Reports, Tools, Windows, Help) and a toolbar with icons for Save, Close, Open, Edit, New, and Delete. Below the toolbar is a tab labeled 'Accounts'. The main area displays a table of accounts with columns for Account Name, Type, Commodity, and Total. The accounts are organized into a tree structure with expandable/collapsible icons. The 'Assets' account is expanded, showing sub-accounts like Checking, Savings, Equity, Expenses, Taxes, Income, and Liabilities. The 'USD, grand total' row at the bottom shows a total of USD 1,281.79 for Assets and a profit of -USD 218.21.

Account Name	Type	Commodity	Total
Assets	Asset	US Dollar	USD 1,781.79
Checking	Bank	US Dollar	USD 881.79
Savings	Bank	US Dollar	USD 900.00
Equity	Equity	US Dollar	USD 1,500.00
Expenses	Expense	US Dollar	USD 1,218.21
Electricity	Expense	US Dollar	USD 78.00
Groceries	Expense	US Dollar	USD 45.21
Internet	Expense	US Dollar	USD 20.00
Phone	Expense	US Dollar	USD 45.00
Rent	Expense	US Dollar	USD 700.00
Taxes	Expense	US Dollar	USD 330.00
Federal	Expense	US Dollar	USD 180.00
Medicare	Expense	US Dollar	USD 90.00
Social Se	Expense	US Dollar	USD 60.00
Income	Income	US Dollar	USD 1,000.00
Liabilities	Liability	US Dollar	USD 500.00
USD, grand total:			Assets: USD 1,281.79   Profits: -USD 218.21

### Save file

Before we go to the report section, let's save the GnuCash data file (gcashdata\_4).

### Reports

But only having data available on the computer screen will not make your accountant happy, therefore GnuCash comes with a large number of highly customizable reports.

Let's have a look at a Cash Flow, and a Transaction Report.

1. First let's have a look at the Cash Flow report for the month of March.

Select the cash flow report from Reports > Income & Expense > Cash Flow.

### Cash Flow - 01/03/06 to 31/03/06 for

Selected Accounts

- [Assets](#)
- [Assets:Checking](#)
- [Assets:Savings](#)

Money into selected accounts comes from	
<a href="#">Income:Salary</a>	USD 1,000.00
Money In	<b>USD 1,000.00</b>
Money out of selected accounts goes to	
<a href="#">Expenses:Electricity</a>	USD 78.00
<a href="#">Expenses:Groceries</a>	USD 45.21
<a href="#">Expenses:Internet</a>	USD 20.00
<a href="#">Expenses:Phone</a>	USD 45.00
<a href="#">Expenses:Rent</a>	USD 350.00
<a href="#">Expenses:Taxes:Federal</a>	USD 180.00
<a href="#">Expenses:Taxes:Medicare</a>	USD 90.00
<a href="#">Expenses:Taxes:Social Security</a>	USD 60.00
Money Out	<b>USD 868.21</b>
Difference	<b>USD 131.79</b>

To get this customized report, right click in the report and choose Report Options. Then set the time period, and specify which accounts you want in the report.

2. Now let's have a look at corresponding transaction report for the Checking account.

Select the transaction report from Reports > Transaction Report.

**Transaction Report****From 01/03/06 To 31/03/06**

Date	Num	Description	Transfer from/to	Amount
<u>Checking</u>				
05/03/06		Groceries	Groceries	<u>-USD 45.21</u>
06/03/06		Transfer	Savings	<u>USD 100.00</u>
14/03/06		Employers R Us	Split	<u>USD 670.00</u>
28/03/06		Internet subscription	Internet	<u>-USD 20.00</u>
28/03/06	102	Light Company	Electricity	<u>-USD 78.00</u>
28/03/06	103	Phone Company	Phone	<u>-USD 45.00</u>
28/03/06	104	April Rent	Rent	<u>-USD 350.00</u>
Total For Checking				<b>USD 231.79</b>
Grand Total				<b>USD 231.79</b>

3. Now let's change the transaction report to only show the various Expenses accounts.

**Transaction Report****From 01/03/06 To 31/03/06**

Date	Num	Description	Transfer from/to	Amount
March 2006				
05/03/06		Groceries	Checking	<u>USD 45.21</u>
14/03/06		Employers R Us	Split	<u>USD 180.00</u>
14/03/06		Employers R Us	Split	<u>USD 90.00</u>
14/03/06		Employers R Us	Split	<u>USD 60.00</u>
28/03/06		Internet subscription	Checking	<u>USD 20.00</u>
28/03/06	102	Light Company	Checking	<u>USD 78.00</u>
28/03/06	103	Phone Company	Checking	<u>USD 45.00</u>
28/03/06	104	April Rent	Checking	<u>USD 350.00</u>
Total For March 2006				<b>USD 868.21</b>
Grand Total				<b>USD 868.21</b>

## 2.4 Importing Data into GnuCash

This chapter will detail procedures for importing data into GnuCash.

### 2.4.1 Importing Transactions from Files

Imported transactions will generally be to a specific account in your account tree. In the following this will be referred to as the **import** or **base** account. It may or may not be specified in the data being imported, depending on the import

format. It is usually the first split of a transaction being imported.

All transactions will also must have a **destination** account for at least matching splits. This may or may not be supplied in the imported data. If it is not, an account can be assigned on the basis of the previous import history by matching to information in the imported data. The user may always over-ride this assignment.

Multi-split data previously exported from GnuCash may have both the import and destination accounts for transaction splits specified in the data file.

### File Import Formats

Gnucash allows transactions to be imported in the following formats:

- *QIF* (.qif) Quicken Interchange format - import data from Quicken financial software;
- *OFX/QFX* (.ofx,.qfx) Open Financial eXchange format (QXF is an Intuit/Quicken proprietary version of OFX);
- *CSV* (.csv) Comma Separated Values;
- *MT940* MT940
- *MT942* MT942
- *DTAUS* DTAUS

These import methods can be accessed from File > Import . . .

### Import Matcher

Several of the Import Assistants use an Import Matcher to implement a Bayesian approach to assign destination accounts, if such accounts are not specified in the imported data, to each imported transaction based on the previous import history of the import account. It also attempts to match the transactions being imported to any existing transactions based on the date and the description fields.

Transaction rows which match existing transactions already in the import account are flagged not to be imported. They will have a light green background and the A and U+R checkboxes will be unchecked and the R checkbox will be checked. To override and import the transaction, check the A checkbox. The U and R boxes will be unchecked automatically. The reliability of the match is indicated by a bar display in the Info column. If a destination account for the second split is assigned by the matcher it will be appended to the info column.

Transaction rows which do not match existing transactions in the import account, for which an assignment of a destination account cannot be made on the basis of the previous import history to the account, will be displayed with an orange-yellow background and the A box will be checked and U+R and R unchecked. A destination account must be specified for these transactions.

### Assign a Destination Account to a Single Transaction

The currently selected row is selected by Left-clicking it. It is displayed with a mid dark green background.

**Double click** on a row. This will select it and open an Account Selection dialog. Select the desired destination account in the dialog and click OK. The row background will change to a light green and the assigned destination account will be displayed in the Info column.

or alternatively, **Left-click** on a row to select it followed by a **Right-click** to bring up a popup menu then select "Assign a transfer account" to display the Account Selection dialog, select the destination account and click the OK button.



## Assign a Destination Account to Multiple Transactions

Sometimes you will have several transactions which will have the same destination account. GnuCash allows you to select multiple transactions and apply the same destination account to all transactions in the selection.

Rows in a selection are displayed with a mid dark green background.

Multiple rows may be selected to have the same destination account assigned to them.

To select rows either:

- **Left click** on first row and then **Ctrl-Left click** on other rows to add to the selection or
- **Left-click** on a first row and then **Shift-Left-click** on another row to select all rows between them.

then **Right-click** to display a popup menu and then select “**Assign a transfer account**” to open the Account Selection dialog. Select the desired destination account and click the OK button in the Account Selection dialog.

## Completing the Import

Once you have assigned destination accounts for all the imported transactions using the above methods (all row backgrounds will be a light green colour), check that the assigned destination accounts are correct and then press the OK button at the bottom of the Generic Import Matcher window. The transactions selected for import will have their splits added to the selected source and destination accounts.

The choices made for the destination accounts and description/memo fields are remembered and stored and used for future imports to the same account to automatically assign a destination account for transaction records not containing destination account information.

## Import QIF

To import data from Quicken®, MS Money, or other programs that use QIF(Quicken® Interchange Format), you must first export your data to a QIF file. One way to do this is to export each account as a separate QIF file. An easier way, available in Quicken® 98 and beyond, is to export all accounts at once into a single QIF file. Check your program’s manual to determine if this option is available.

### To import QIF files:

#### Load all of the QIF files containing data you wish to import

To do this, select **File -> Import -> Import QIF...** from the menu. When the QIF Import dialog box appears, click **Next** and follow the instructions to guide you through the process of loading your files.

This image shows the start of the QIF Import assistant.

You will be prompted for a filename to load. Use the Select button to select your QIF file and click Next to load it. Once the file is loaded, select Load another file if you have more files to load. When you have loaded all your QIF files, click Next to continue with the import process.

### Review the GnuCash accounts to be created.

The importer then matches up your QIF accounts and categories with GnuCash accounts and gives you a brief description of the matching process. Clicking Next will bring you to a view comparing your QIF accounts with the corresponding GnuCash accounts created. To change an account name, select the row containing that account name

and edit the name in the dialog box provided. Click **Next** when you have finished making changes, and proceed through a similar category matching process. QIF income and expense categories import as GnuCash income and expense accounts. Make changes to these account names if necessary, and click **Next** to continue.

**Note:** If you are not sure what changes are needed, it is safe to accept the GnuCash account names. It is easy to edit the accounts later if you find you need to make a change.

From the drop-down list, select a standard currency to be used for the imported accounts and click **Next** to continue. If you have stocks, mutual funds, or other securities, you will be prompted for additional information. The importer dialog will ask for the exchange or listing (i.e. Nasdaq), the security's full name, and the ticker symbol. If you do not have this information handy, you can edit the account information later, once the import is complete. Click **Next** to continue.

### Tell GnuCash to import the data.

The last step is the import. Once you have verified your account names and investment information, click **Finish** in the Update your GnuCash accounts page to complete the import process. Depending upon the size of your file, the import might take a few minutes to complete, so a progress bar displays the percentage finished. When the import process is complete, GnuCash will return you to the main window, which should now display the names of the accounts you imported.

### Import OFX/QFX

This opens a file selection dialog. Navigate to the file you wish to import, select a file with the appropriate extension (.ofx or .qfx), then press the **Import** button.

Gnucash opens an Account Selection dialog to select an account in your CoA corresponding to data source. Select the appropriate account from the account tree and press the **OK** button. On subsequent import of files from the same source (identified by tags in the file), the source is remembered and the account selection dialog is not displayed.

The generic import transaction matcher dialog is opened next. See the [Import Matcher](#) section (common to both OFX/QFX and CSV import formats) following the Import CSV section to continue the import process.

### Import CSV

Clicking on **Import CSV** in the Import menu will bring up the Import Assistant dialog. The first step brings up a file selection dialog. Navigate to the location where the file you wish to import is located and select the file to import then click the **OK** button.

The next window will allow you to set parameters for the importing of the file. All widgets have tooltips which explain what the setting affects and the options for the setting.

### Load and Save Settings

If this import is a regular occurrence, once you have set the other import parameters, you can save these settings by typing in a setting name in the **Load and Save Settings Entry** combo box and pressing the **Save** button just to the right of the box. Previously defined settings can be retrieved by selecting the appropriate setting name from the dropdown list activated by the down arrow at the right end of the text box. The trash can button to the right of the Save button can be used to remove the settings selected from the drop down list for the box. The settings group **"GnuCash Export Settings"** define a setting group for the export and reimport of GnuCash transaction data - use this if importing data previously exported from GnuCash.

## Account

This combo box allows you to select the base or import account into which the transactions will be imported. It may be left unset if the imported data contains a column listing the accounts associated with each split or the import data specifies the account for first split of the transaction.

## File Format

This section allows you to define whether the file has:

- **Fixed width columns** Selecting this radio button will allow you to define column boundaries by double clicking at the appropriate positions in the sample records displayed in the panel below. Single clicking in a column will narrow, widen or merge the column.
- **Separators** Selecting this radio button will allow you to define characters which will be used to distinguish columns in the input file. The default is comma separated however spaces, tabs,colons or semicolons or any combination of them may be used to separate columns in the input file by selecting the appropriate check boxes. You may also define custom separators by typing the required characters into the text box and selecting the **Custom** checkbox. This may be used in combination with any of the predefined separators.
- **Multi-split** Selecting this check box allows the splits for a single transaction to be defined on consecutive lines within the file with each line defining a single split. If not selected each line is assumed to contain the information for a single transaction including one or two splits.

## Miscellaneous

The miscellaneous settings allow you to set:

- **Encoding**This is usually the UTF-8 variant for your locale;
- **Date Format** This does not default to the Locale setting so check it matches the data you are importing;
- **Currency Format**;
- **Leading Lines to Skip**;
- **Trailing Lines to Skip**;
- **Skip alternate lines**;

to match the settings for the file you are importing. Tooltips may also contain information on the setting and options.

## Import Panel

The import panel shows the data being imported as it is interpreted using the settings chosen to define columns and formats. The dropdown lists in the headers for each column of the import allow you to associate a specific column in the imported data with a specific field in the display of a transaction in an account register. At a minimum to import data, columns in the imported data containing the following information **must** be specified:

- **Date**of transaction;
- **Account** into which transaction is to be imported (or alternatively set the base account as above);
- **Description** of the transaction;
- **Deposit or Withdrawal** column.

The **Skip Errors** check box will skip trying to import any rows with errors in matching the columns.

When you are happy with all the import settings, *save* them if you will use the same settings again, then press the **Next** button. This will bring up a window which allows you to map the accounts identified in the account column (Account Id) with accounts in the GnuCash account tree (Account name). Double click on a row to bring up a dialog to select the matching GnuCash account. When you have selected a match for all accounts, click on the **Next** button.

The Transaction Information panel allows review of data entry settings so far.

Clicking on Match Transactions will then bring up the main *Import Matcher* window described in the next section.

### Import MT940

Use this option to import MT940 data.

### Import MT942

Use this option to import MT942 data.

### Import DTAUS

Use this option to import DTAUS data.

## 2.5 Configuring GnuCash

This chapter will detail preference settings for GnuCash.

### 2.5.1 Overview

Configuring is fun.

### 2.5.2 Main Configuration Screens

GnuCash stores settings in two main locations.

#### Preferences: Accounting Period Tab

Stuff about the Accounting Period tab

#### Preferences: Accounts Tab

Stuff about the Accounts tab. Need this stub for the link in The Basics. More later.

#### Preferences: Online Banking

Stuff about the Accounts tab. Need this stub for the link in The Basics. More later.

- *Managing Personal Finances*

## 2.6 Checkbook

This chapter will give you all the specific information you need to manage your checkbook with GnuCash. Managing your checkbook register is the first step of tracking your finances, and GnuCash makes it much easier to manage than the traditional paper register does.

For one thing, as discussed in *Transactions*, data entry of common transactions is practically done for you in GnuCash, because of its auto-completion feature. GnuCash keeps a running balance of each account, and it makes reconciling these accounts easy. And the double-entry method helps you account for your spending by requiring a transfer account for withdrawals, so you can easily find out how much money you spend in different areas.

Once you are comfortable with using GnuCash for your checking and other bank accounts, you may wish to continue on with tracking other financial accounts. Chapters 6 through 9 examine methods to manage your other accounts.

### 2.6.1 Setting up Accounts

The first step in managing your checkbook is to set up the necessary accounts. You can either use the default GnuCash accounts or set up your own. For more detail on how to set up a new account, refer to *The Basic Top Level Accounts*. For instructions on importing your accounts from another program, refer to the GnuCash manual.

Let's start with the bank accounts you'll need. You need one GnuCash bank type account for each physical bank account you wish to track. If you are setting up your own accounts or using the default GnuCash accounts, make sure that you have an opening balance transaction for each bank account you own. The easiest way to get this number is to use the balance from your last bank statement as your opening balance. You can enter this in the account information window automatically as part of the New Account Hierarchy Setup assistant, or you can enter a manual transaction directly in the account. To enter the transaction manually, enter a transfer from an Opening Balances account (type equity) to the bank account.

The typical bank accounts you might track include:

- Checking - any institutional account that provides check-writing privileges.
- Savings - an interest-bearing institutional account usually used to hold money for a longer term than checking accounts.

Common transactions that affect these bank accounts are payments and deposits. *Payments* are transfers of money out of the bank account, usually to an expense account. *Deposits* are transfers of money into the bank account, usually from an income account. You will need to set up income and expense accounts to track where that money comes from and where it goes. Remember that a balanced transaction requires a transfer of an equal sum of money from at least one account to at least one other account. So if you deposit money in your checking account, you must also enter the account that money comes from. If you pay a bill from your checking account, you must also enter the account where that money goes.

### 2.6.2 Entering Deposits

Most deposit transactions are entered as a transfer from an income account to a bank account. Income may come from many sources, and it's a good idea to set up a separate income type account for each different source. For example, your income may come mainly from your paychecks, but you may also receive interest on your savings. In this case, you should have one income account for salary and another income account for interest income.

---

**Tip:** Be sure to check the Tax-Related box and assign an appropriate tax category in the Income Tax Information Dialog (Edit > Tax Report Options) when you set up taxable income accounts. Some types of income, such as gift income, may not be considered taxable, so check the appropriate tax rules to determine what is taxable. For ways to track capital gains income, refer to *Capital Gains*.

---

Before you start entering paycheck deposits, decide how much detail you want to track. The basic level of detail is to enter your net pay, just like you would in your paper register. This is easiest, but you can get even more information out of GnuCash if you enter your gross pay with deductions. It takes a bit more effort to enter the deductions, but entering your tax withholding information throughout the year allows you to run useful tax status reports in GnuCash at any time. These reports can help you determine whether you are withholding enough tax, and they can help you estimate your tax bill ahead of time. If you are unsure about the level of detail, start by entering net pay. You can always go back and edit your transactions later if you decide you want more detail.

### 2.6.3 Entering Withdrawals

When you withdraw money from your bank account, for whatever reason, you are transferring money from your bank account to some other location. In GnuCash, this other location is tracked as an account. The more detailed accounts you create and use for your spending, the more information you will get about where your money goes.

Withdrawals take many forms. ATM withdrawals are one of the most common transactions. Writing checks is one way to withdraw money to pay bills, to buy purchases, or to give to charity. Depending on your bank, you might also have service charges, where the bank withdraws the money from your account. Transfers of money out to another account are also withdrawals. We will take a look at each of these types of withdrawals and how to record them in GnuCash.

#### ATM/Cash Withdrawals

Cash withdrawals are handled as a transfer from a bank account to a cash account. GnuCash provides special *Cash* type accounts for tracking your cash purchases, so you should set up a cash account to record your ATM and other cash withdrawals.

Cash accounts can be used for different levels of detail. On a basic level of detail, you simply transfer money to it from your checking account. That tells you how much money you took out of checking on a given day, but it doesn't tell you where that cash was spent. With a little more effort, you can use the cash account to record your cash purchases as well, so that you can see where that cash went. You record these purchases as a transfer from the cash account to expense accounts.

Some people record every cash purchase, but this takes a lot of work. An easier way is to record the purchases for which you have receipts, but then adjust the balance of the account to match what is in your wallet.

It's a good idea to at least set up a cash account for your withdrawals. Then if you decide to track where your cash goes, you can enter transactions for the money you spend. You determine what level of detail you want to use.

### 2.6.4 Reconciling Your Accounts

GnuCash makes reconciliation of your bank account with your monthly bank statement much easier. [Reconciliation](#) gives instructions on how to reconcile your transactions with the monthly bank statement. This is the main reconciliation task that should be done every month.

But what about all those other accounts you created? Should those be reconciled too? If you receive a statement for the account, then you should consider reconciling that account. Examples include the checking account statement, the savings account statement, and the credit card statement. Credit card statements and credit card transactions are covered in the [Credit Cards](#), so if you are interested in tracking your credit cards in GnuCash, take a look at the instructions provided there.

Income and expense accounts are usually not reconciled, because there is no statement to check them against. You also don't need to reconcile cash accounts, for the same reason. With a cash account, though, you might want to adjust the balance every once in a while, so that your actual cash on hand matches the balance in your cash account. Adjusting balances is covered in the next section.

## 2.6.5 Putting It All Together

In *Transactions*, you entered some transactions in your checking account. In this chapter, we will add more transactions and then reconcile them.

### Opening Balances

So, let's get started by opening the gcashdata file you saved in the last chapter (gcashdata\_4). Your chart of accounts should look like this:

Account Name	Type	Commodity	Total
Assets	Asset	US Dollar	USD 1,601.79
Cash	Cash	US Dollar	USD 100.00
Checking	Bank	US Dollar	USD 1,101.79
Savings	Bank	US Dollar	USD 400.00
Equity	Equity	US Dollar	USD 1,500.00
Expenses	Expense	US Dollar	USD 1,398.21
Clothes	Expense	US Dollar	USD 0.00
Electricity	Expense	US Dollar	USD 78.00
Groceries	Expense	US Dollar	USD 120.21
Household	Expense	US Dollar	USD 50.00
Internet	Expense	US Dollar	USD 20.00
Phone	Expense	US Dollar	USD 45.00
Rent	Expense	US Dollar	USD 700.00
Service Charge	Expense	US Dollar	USD 5.00
Taxes	Expense	US Dollar	USD 330.00
Tools	Expense	US Dollar	USD 50.00
Imbalance-USD	Bank	US Dollar	USD 0.00
Income	Income	US Dollar	USD 1,000.00
Liabilities	Liability	US Dollar	USD 500.00
Orphan-USD	Bank	US Dollar	USD 0.00

USD: Assets: USD 1,101.79 Profits: -USD 398.21

### Add some transactions

1. Now open the *Checking* account from the main window. In the last chapter, you entered some paycheck transactions as deposits into Checking. Now we will enter another kind of deposit - a transfer of money from Savings into Checking. On a blank line, enter a transaction to transfer \$500 from *Savings* to *Checking* (In this case, the last transaction date was March 28, but this transfer transaction took place on the March 24). Your Transfer account will be *Savings*, since you are in the *Checking* account. Your *Checking* account should now look like this:



Date	Num	Description	Transfer	R	Deposit	Withdrawal	Balance
01/03/06		Opening Balance	Equity:Opening Balance	y	1,000.00		1,000.00
05/03/06	101	Grocery Store	Expenses:Groceries	y		45.21	954.79
05/03/06	105	HomeTown Grocery	Expenses:Groceries	y		75.00	879.79
06/03/06		Transfere Money	Assets:Savings	y	100.00		979.79
06/03/06	106	ABC Hardware	-- Split Transaction --	y		100.00	879.79
14/03/06		Employers R Us	-- Split Transaction --	y	670.00		1,549.79
24/03/06		Transfer Money	Assets:Savings	y	500.00		2,049.79
25/03/06		ATM Withdrawal	Assets:Cash	y		100.00	1,949.79
28/03/06		Internet Subscription	Expenses:Internet	y		20.00	1,929.79
28/03/06	102	Light Company	Expenses:Electricity	y		78.00	1,851.79
28/03/06	103	Phone Company	Expenses:Phone	y		45.00	1,806.79
28/03/06	104	April Rent	Expenses:Rent	y		350.00	1,456.79
31/03/06		Service Charge	Expenses:Service Charge	y	Deposit	5.00	1,451.79
28/04/06		May Rent	Expenses:Rent	n		350.00	1,101.79

Present:USD 1,451.79 Future:USD 1,101.79 Cleared:USD 1,451.79 Reconciled:USD 1,451.79 Projected Minimum:USD 1,101.79  
Monday 17 April 2006

2. Now let's write some checks on this account. First, write a check to HomeTown Grocery for \$75 (5th of March). Your transfer account is *Groceries*, since all of this money is going to buy groceries. Next, write a check to ABC Hardware for \$100 (6 of March), and split this amount between two expenses: *Household* \$50 and *Tools* \$50. You will need to create an *Expense* type account for each of these, then enter splits for them. Your checking account should now look like this;

Date	Num	Description	Transfer	R	Deposit	Withdrawal	Balance
01/03/06		Opening Balance	Equity:Opening Balance	y	1,000.00		1,000.00
05/03/06	101	Grocery Store	Expenses:Groceries	y		45.21	954.79
05/03/06	105	HomeTown Grocery	Expenses:Groceries	y		75.00	879.79
06/03/06		Transfere Money	Assets:Savings	y	100.00		979.79
06/03/06	106	ABC Hardware	-- Split Transaction --	y		100.00	879.79
14/03/06		Employers R Us	-- Split Transaction --	y	670.00		1,549.79
24/03/06		Transfer Money	Assets:Savings	y	500.00		2,049.79
25/03/06		ATM Withdrawal	Assets:Cash	y		100.00	1,949.79
28/03/06		Internet Subscription	Expenses:Internet	y		20.00	1,929.79
28/03/06	102	Light Company	Expenses:Electricity	y		78.00	1,851.79
28/03/06	103	Phone Company	Expenses:Phone	y		45.00	1,806.79
28/03/06	104	April Rent	Expenses:Rent	y		350.00	1,456.79
31/03/06		Service Charge	Expenses:Service Charge	y	Deposit	5.00	1,451.79
28/04/06		May Rent	Expenses:Rent	n		350.00	1,101.79

Present:USD 1,451.79 Future:USD 1,101.79 Cleared:USD 1,451.79 Reconciled:USD 1,451.79 Projected Minimum:USD 1,101.79  
Monday 17 April 2006

3. Suppose you now need to withdraw some money. You don't have a cash account set up in your chart of accounts, so you will need to create one. Create the account as Cash as a top-level account of type Asset. From your *Checking* account register, enter an ATM type withdrawal to transfer \$100 from *Checking* to *Cash* on the 25 of March.

Date	Num	Description	Transfer	R	Deposit	Withdrawal	Balance
01/03/06		Opening Balance	Equity:Opening Balance	y	1,000.00		1,000.00
05/03/06	101	Grocery Store	Expenses:Groceries	y		45.21	954.79
05/03/06	105	HomeTown Grocery	Expenses:Groceries	y		75.00	879.79
06/03/06		Transfere Money	Assets:Savings	y	100.00		979.79
06/03/06	106	ABC Hardware	-- Split Transaction --	y		100.00	879.79
14/03/06		Employers R Us	-- Split Transaction --	y	670.00		1,549.79
24/03/06		Transfer Money	Assets:Savings	y	500.00		2,049.79
25/03/06		ATM Withdrawal	Assets:Cash	y		100.00	1,949.79
28/03/06		Internet Subscription	Expenses:Internet	y		20.00	1,929.79
28/03/06	102	Light Company	Expenses:Electricity	y		78.00	1,851.79
28/03/06	103	Phone Company	Expenses:Phone	y		45.00	1,806.79
28/03/06	104	April Rent	Expenses:Rent	y		350.00	1,456.79
31/03/06		Service Charge	Expenses:Service Charge	y	Deposit	5.00	1,451.79
28/04/06		May Rent	Expenses:Rent	n		350.00	1,101.79

Present:USD 1,451.79 Future:USD 1,101.79 Cleared:USD 1,451.79 Reconciled:USD 1,451.79 Projected Minimum:USD 1,101.79  
Monday 17 April 2006

## Opening Balances

Now we are ready to reconcile this *Checking* account, using this sample bank statement:



Date	Description	Decreasing	Increasing	Balance
1/03/2006	Starting Balance			1,000.00
5/03/2006	Check: 101: Grocery Store	45.21		954.79
6/03/2006	Transfer		100.00	1,054.79
8/03/2006	Check: 106: ABC Hardware	100.00		954.79
14/03/2006	Salary		670.00	1,624.79
15/03/2006	Check: 105: HomeTown Grocery	75.00		1,549.79
24/03/2006	Transfer		500.00	2,049.79
25/03/2006	ATM	100.00		1,949.79
28/03/2006	Check: 102: Light Company	78.00		1,871.79
28/03/2006	Check: 103: Phone Company	45.00		1,826.79
28/03/2006	Check: 104: Apartment 4 Rent	350.00		1,476.79
28/03/2006	Best Internet	20.00		1,456.79
31/03/2006	Service Charge	5.00		1,451.79
31/03/2006	Ending Balance			1,451.79

1. Select Actions > Reconcile from the menu, and fill in the Closing balance as \$1451.79. Click OK to begin reconciling the account. Check off the entries as they appear on the sample statement. When you have checked off all your entries, the reconcile window should look like this:

Notice that your reconciled balance differs from your ending balance by \$5.00. If you look at the sample bank statement, you will see there is a \$5.00 service charge that has not been added to your *Checking* account.

2. So click on the *Checking* register and add the \$5.00 service charge to your *Checking* account. On a blank line of the *Checking* register, enter a transaction to transfer \$5.00 from *Checking* to a *Service Charges* account. (You will need to create the *Service Charges* account as type Expense.) Use the transaction date printed on the sample statement as the date you enter for this transaction. Your *Checking* account should now look like this:

3. Click back on the Reconcile window, and you should see the service charge now under Funds Out. Click on it to mark it as reconciled, and note that the difference amount below now becomes 0.00. Click the Finish button on the *Toolbar* to complete the reconciliation. The Reconcile R column in your *Checking* register should now show y for each transaction you just reconciled. Also observe the bottom status row that now indicates Reconciled:

USD 1451.79

Date	Num	Description	Transfer	R	Deposit	Withdrawal	Balance
01/03/06		Opening Balance	Equity:Opening Balance	y	1,000.00		1,000.00
05/03/06	101	Grocery Store	Expenses:Groceries	y		45.21	954.79
05/03/06	105	HomeTown Grocery	Expenses:Groceries	y		75.00	879.79
06/03/06		Transfer Money	Assets:Savings	y	100.00		979.79
06/03/06	106	ABC Hardware	-- Split Transaction --	y		100.00	879.79
14/03/06		Employers R Us	-- Split Transaction --	y	670.00		1,549.79
24/03/06		Transfer Money	Assets:Savings	y	500.00		2,049.79
25/03/06		ATM Withdrawal	Assets:Cash	y		100.00	1,949.79
28/03/06		Internet Subscription	Expenses:Internet	y		20.00	1,929.79
28/03/06	102	Light Company	Expenses:Electricity	y		78.00	1,851.79
28/03/06	103	Phone Company	Expenses:Phone	y		45.00	1,806.79
28/03/06	104	April Rent	Expenses:Rent	y		350.00	1,456.79
31/03/06		Service Charge	Expenses:Service Charge	y		5.00	1,451.79
28/04/06		May Rent	Expenses:Rent	n		350.00	1,101.79

## Save file

Go back to the main window and save your file with the new `gncashdata_5` name. Your chart of accounts is steadily growing, and it should now look like this:

Account Name	Type	Commodity	Total
Assets	Asset	US Dollar	USD 1,601.79
Cash	Cash	US Dollar	USD 100.00
Checking	Bank	US Dollar	USD 1,101.79
Savings	Bank	US Dollar	USD 400.00
Equity	Equity	US Dollar	USD 1,500.00
Expenses	Expense	US Dollar	USD 1,398.21
Clothes	Expense	US Dollar	USD 0.00
Electricity	Expense	US Dollar	USD 78.00
Groceries	Expense	US Dollar	USD 120.21
Household	Expense	US Dollar	USD 50.00
Internet	Expense	US Dollar	USD 20.00
Phone	Expense	US Dollar	USD 45.00
Rent	Expense	US Dollar	USD 700.00
Service Charge	Expense	US Dollar	USD 5.00
Taxes	Expense	US Dollar	USD 330.00
Tools	Expense	US Dollar	USD 50.00
Imbalance-USD	Bank	US Dollar	USD 0.00
Income	Income	US Dollar	USD 1,000.00
Liabilities	Liability	US Dollar	USD 500.00
Orphan-USD	Bank	US Dollar	USD 0.00

## Reports

As we did in the previous chapter, let's have a look at a Cash Flow, and a Transaction Report.

1. First let's have a look at the Cash Flow report for the month of March.

Select the cash flow report from Reports > Income & Expense > Cash Flow.

### Cash Flow - 01/03/06 to 31/03/06 for

Selected Accounts

- [Assets:Cash](#)
- [Assets:Checking](#)
- [Assets:Savings](#)

Money into selected accounts comes from	
<a href="#">Equity:Opening Balance</a>	USD 2,000.00
<a href="#">Income:Salary</a>	USD 1,000.00
<b>Money In</b>	<b>USD 3,000.00</b>
Money out of selected accounts goes to	
<a href="#">Expenses:Electricity</a>	USD 78.00
<a href="#">Expenses:Groceries</a>	USD 120.21
<a href="#">Expenses:Household</a>	USD 50.00
<a href="#">Expenses:Internet</a>	USD 20.00
<a href="#">Expenses:Phone</a>	USD 45.00
<a href="#">Expenses:Rent</a>	USD 350.00
<a href="#">Expenses:Service Charge</a>	USD 5.00
<a href="#">Expenses:Taxes:Federal</a>	USD 180.00
<a href="#">Expenses:Taxes:Medicare</a>	USD 90.00
<a href="#">Expenses:Taxes:Social Security</a>	USD 60.00
<a href="#">Expenses:Tools</a>	USD 50.00
<b>Money Out</b>	<b>USD 1,048.21</b>
<b>Difference</b>	<b>USD 1,951.79</b>

2. Now let's have a look at corresponding transaction report for the various Asset accounts.

Select the transaction report from Reports > Transaction Report.

## Transaction Report

### From 01/03/06 To 31/03/06

Date	Num	Description	Transfer from/to	Amount
<u>Cash</u>				
25/03/06		ATM Withdrawal	Checking	<u>USD 100.00</u>
Total For Cash				<b>USD 100.00</b>
<u>Checking</u>				
01/03/06		Opening Balance	Opening Balance	<u>USD 1,000.00</u>
05/03/06	101	Grocery Store	Groceries	<u>-USD 45.21</u>
05/03/06	105	HomeTown Grocery	Groceries	<u>-USD 75.00</u>
06/03/06		Transfere Money	Savings	<u>USD 100.00</u>
06/03/06	106	ABC Hardware	Split	<u>-USD 100.00</u>
14/03/06		Employers R Us	Split	<u>USD 670.00</u>
24/03/06		Transfer Money	Savings	<u>USD 500.00</u>
25/03/06		ATM Withdrawal	Cash	<u>-USD 100.00</u>
28/03/06	102	Light Company	Electricity	<u>-USD 78.00</u>
28/03/06	104	April Rent	Rent	<u>-USD 350.00</u>
28/03/06		Internet Subscription	Internet	<u>-USD 20.00</u>
28/03/06	103	Phone Company	Phone	<u>-USD 45.00</u>
31/03/06		Service Charge	Service Charge	<u>-USD 5.00</u>
Total For Checking				<b>USD 1,451.79</b>
<u>Savings</u>				
01/03/06		Opening Balance	Opening Balance	<u>USD 1,000.00</u>
06/03/06		Transfere Money	Checking	<u>-USD 100.00</u>
24/03/06		Transfer Money	Checking	<u>-USD 500.00</u>
Total For Savings				<b>USD 400.00</b>
Grand Total				<b>USD 1,951.79</b>

3. Now let's change the transaction report to only show the various *Expenses* account.

## Transaction Report

### From 01/03/06 To 31/03/06

Date	Num	Description	Account	Amount
05/03/06	101	Grocery Store	Groceries	<u>USD 45.21</u>
05/03/06	105	HomeTown Grocery	Groceries	<u>USD 75.00</u>
06/03/06	106	ABC Hardware	Household	<u>USD 50.00</u>
06/03/06	106	ABC Hardware	Tools	<u>USD 50.00</u>
14/03/06		Employers R Us	Federal	<u>USD 180.00</u>
14/03/06		Employers R Us	Social Security	<u>USD 60.00</u>
14/03/06		Employers R Us	Medicare	<u>USD 90.00</u>
28/03/06		Internet Subscription	Internet	<u>USD 20.00</u>
28/03/06	104	April Rent	Rent	<u>USD 350.00</u>
28/03/06	102	Light Company	Electricity	<u>USD 78.00</u>
28/03/06	103	Phone Company	Phone	<u>USD 45.00</u>
31/03/06		Service Charge	Service Charge	<u>USD 5.00</u>
<b>Grand Total</b>				<b>USD 1,048.21</b>

Notice that you have not yet used one of the accounts listed in your chart, the Credit Card account. Now that you know how to keep track of your bank and cash accounts in GnuCash, you may want to start tracking your credit cards as well. GnuCash provides a special type of account for credit cards, and this is discussed in the next chapter.

## 2.7 Expense Accounts

If managing your checkbook is the first step in tracking your finances, then using expense accounts to see where you are expending money is a close second step. This chapter will give you an understanding of how GnuCash uses expense accounts to help you keep track of many different categories of transactions.

### 2.7.1 Concepts

An expense type account is used to allow you to track how much you spend on specific expenses. Many people's first experience with tracking expenses comes from Quicken(tm), where transactions can be assigned to one or more categories. In GnuCash, these categories are set up as separate accounts, which are designated as Expense type accounts. This allows GnuCash to apply the rules of double-entry accounting consistently. Expense accounts can be as detailed or as general as you need. Some users need only a few accounts for personal expense tracking. Others use GnuCash expense accounts to manage their expenses in great detail. The level of detail you choose is up to you. Keep in mind that with GnuCash, you can change accounts for transactions, so if your needs change later on, it is possible to move transactions around.

### 2.7.2 Setting Up Accounts

## Simple Expense Account Setup

For many users, the easiest way to set up expense accounts is to check the “Common Accounts” when you create a new Account Hierarchy. This will establish many of the most common expense accounts that users need. See “New Account Hierarchy Setup” in Chapter 3 of the Help guide for more information.

## Complex Expense Account Setup

If you have different expense accounting needs, you can refer to *Transactions*, or Chapter 5.4 in the Help manual for instructions on how to create accounts.

Typical reasons for adding new or different expense accounts include: to track expenses for particular business purposes (e.g., specific types of supply expenses), to track expenses for particular tax purposes (e.g., tax expenses that must be reported to others), or simply to track expenses that are meaningful to you (e.g., payments made to a particular charity).

### 2.7.3 Entering Expense Transactions

While it is possible to enter transactions directly into expense accounts, it is not normally how these are entered. For most people, transactions for an expense account are added when the user is entering data into the other account in the transaction. In other words, if you have an expense account for Charitable Donations (e.g., Expenses:Charity), you will typically add a transaction to the expense account by assigning a check in your checking account register to the Charity account.

If you open an expense account, you will see a register similar to most others you find in GnuCash. The informal column headings for the transaction amounts are slightly different, however. The left (debit) column will read *Tot Expense*, while the right (credit) column will read *Tot Rebate*.

### 2.7.4 Other Considerations for Expense Accounts

Because expense accounts are generated entirely by you, there are no statements against which you would reconcile your data. Therefore, there is technically nothing to reconcile. You can, of course use the reconcile process for expense accounts, which will lock the transactions for future editing.

One point to consider is that as your use of GnuCash continues, the balances in these accounts will grow, since there are usually very few credit transactions that reduce the balances. There is nothing wrong with this situation, but some users may wish to clear the balances in their expense accounts periodically. Zeroing transactions can be entered that transfer the balance of the account to an Equity account. GnuCash includes a Closing Books procedure that includes zeroing out expense accounts. Keep in mind that this is not necessary, and that if you need to gather information on a given expense account, you can use various reports to extract that data without zeroing the account out.

## 2.8 Credit Cards

This chapter will show you how to manage your credit cards using GnuCash.

### 2.8.1 Concepts

Since you probably write a check or make an electronic payment to the credit card company each month, you may think of your credit card bill as an expense - but it really is not an expense. Why? A credit card account is a short-term

loan - you buy things on that loan account, and then you eventually have to pay back the money, often with interest (your finance charge). The purchases you make with that credit card are your expenses.

You have a couple of options when entering credit card transactions, so choose the one that fits your desired level of detail. The simplest method is to simply track monthly payments to the credit card company. From your bank account, you enter a transfer of money each month to the credit card expense account. This will show you the amount of money you are paying each month to the credit card company, but it won't show you any information about your credit card balance or credit card purchases.

A more complete way to track your credit card in GnuCash is to enter each purchase and payment as a separate transaction. Using the credit card account register, you enter your receipts throughout the month. When your credit card statement arrives, you reconcile the credit card account to the statement, and you enter your payment as a transfer of money from your checking account to your credit card account. This method gives you more information about your balance during the month and points out any discrepancies during reconciliation, but you will have to do more data entry.

## 2.8.2 Setting Up Accounts

To begin managing your credit cards in GnuCash, you should set up a Liability top level account and under this parent account create credit card type accounts for each credit card you use. If you are tracking only the payments you make to the credit card company, then all you need is a bank account and a credit card account to enter your transactions.

The charges you make on your credit card are expenses, so you will have to set up these accounts under the top-level account called Expenses. If you decide to keep a more detailed records of your purchases, you will need to create expense accounts for each kind of purchase you make. Since you will also be reconciling the balance to your credit card statements, you should also enter an opening balance in each credit card account. The easiest way to do this is to use your last statement balance as the opening balance.

### Simple Setup

If you do not want to track each expense made on the credit card, you can set up a simple account hierarchy like this:

```
-Assets
  -Bank
-Liabilities
  -Credit Card
-Expenses
  -Credit Card
```

In this example, if you enter your total amount charged per month as a transaction between *Liabilities:Credit Card* and *Expenses:Credit Card*. When you make a payment, you would enter a transaction between *Assets:Bank* and *Liabilities:Credit Card*.

The obvious limitation of this simple credit card setup is that you cannot see where your money is going. All your credit card expenses are being entered in the Credit Card expense account. This is, however, very simple to set up and maintain.

### Complete Setup

If you want to track your expenses more completely, you should set up multiple expense accounts named for the various kinds of expenses you have. Each charge on your credit card is then entered as a separate transaction between your Credit Card liability account and a specific expense account. Below is an example of an account hierarchy for this:

```
-Assets
  -Bank
-Liabilities
  -Credit Card
-Expenses
  -Food
  -Car
  -Clothes
  -Entertainment
  -Interest
  -Service
```

Clearly, you should enter specific expense accounts which fit your spending habits. The only difference with this setup as compared to the simple setup is that the expenses have been subdivided by groups. Also notice that there is an “Interest” expense, this is used for when your credit card charges interest on your monthly unpaid balance. The “Service” expense account is used to track service expenses associated with the credit card, such as the yearly usage fee if it exists. With this setup, you will be able to see where your money goes every month, grouped according to the expense accounts.

The rest of this chapter will assume you are using the complete setup.

### 2.8.3 Entering Charges

Entering your charges provides you with a more complete picture of your spending habits. Charges on a credit card are tracked as a transaction between the credit card liability account and the appropriate expense account.

When you pay for goods or services with your credit card, you are telling the credit card company to pay the merchant for you. This transaction will increase the amount of money you owe the credit card company, and the credit card balance will increase. The other side of these transactions will in most cases be an expense account. For example, if you buy clothing from a store with your credit card for \$50, you would be transferring that money from the credit account into Expenses:Clothing.

Entering these transactions into GnuCash allows you to track how much you owe the credit card company, as well as provides you a better picture of your overall accounts. It also allows you to monitor your account and ensure that fraud is avoided.

Adding transactions to a credit card account is similar to entering transactions in other accounts. You can enter them manually, or import them from your credit card company using a compatible import format.

For assistance with entering transactions, see Chapter 6 of the Help manual and [Transactions](#).

### 2.8.4 Entering Payments

Most payments to your credit card bill are entered as transfers from a bank account (asset) to the credit card account (liability). When you pay the monthly bill, you are withdrawing money from a bank account to pay down the credit card balance. This transaction decreases both your bank account balance and the amount of credit card debt you owe.

When you return a purchase, you receive a refund on your credit card. This is another type of payment in that it decreases the amount of credit card debt you owe. If you recorded the original purchase transaction as a transfer from the credit card account to the expense, you now simply reverse that transaction: transfer the money back from the expense to the credit card account. This transaction decreases both the expense account balance and the credit card account balance. For example, if you originally recorded a credit card purchase of clothing, the transaction is a transfer from the credit card account to the clothing expense account. If you then return that clothing for a refund, you simply transfer the money back from the clothing expense account to the credit card account.



**Note:** A common mistake is to enter a refund as income. It is not income, but rather a “negative expense”. That is why you must transfer money from the expense account to your credit card when you receive a refund.

To clarify this, let’s run through an example. You bought some jeans for \$74.99 on your VISA card, but realized one day later that they are too big and want to return them. The shop gracefully agrees, and refunds your credit card.

1. Start with opening the previous datafile we stored (gcashdata\_5), and store it as gcashdata\_6.
2. Open the *Liabilities:Visa* account register and enter a simple 2 account transaction to pay the \$74.99 jeans purchase. The Transfer account should be *Expenses:Clothes* and you Charge your *Visa* account with the \$74.99.

**Note:** Since we had not created the *Expenses:Clothes* account previously, GnuCash will prompt us to create it. Just remember to create it as an Expense account

3. Enter the refund in one of the following way.
  - Enter the same transaction as the purchase transaction, but instead of a “Charge” amount, use a “Payment” amount in the Credit Card account register.
  - Select the purchase transaction you want to refund (that is the Jeans transaction in our case), and selecting Transaction > Add Reversing Transaction. Modify the date as needed.

After reversing the transaction, your credit card account should look something like this:

The screenshot shows the 'Clothes' account register in GnuCash. The table has columns: Date, Num, Description, Transfer, R, Expense, Rebate, and Balance. The transactions are as follows:

Date	Num	Description	Transfer	R	Expense	Rebate	Balance
03/04/06		jeans R Us	Liabilities/Visa	n	74.99		74.99
05/04/06		jeans R Us	Liabilities/Visa	n		74.99	0.00
17/04/06			Transfer	n	Expense	Rebate	Balance

Below the table, it shows: Present:USD 0.00 Future:USD 0.00 Cleared:USD 0.00 Reconciled:USD 0.00 Projected Minimum:USD 0.00. At the bottom, there is a text entry field: 'Enter a description of the transaction'.

And the *Expenses:Clothes* register should look something like this:

This screenshot is identical to the one above, showing the 'Clothes' account register with the same transactions and summary information.

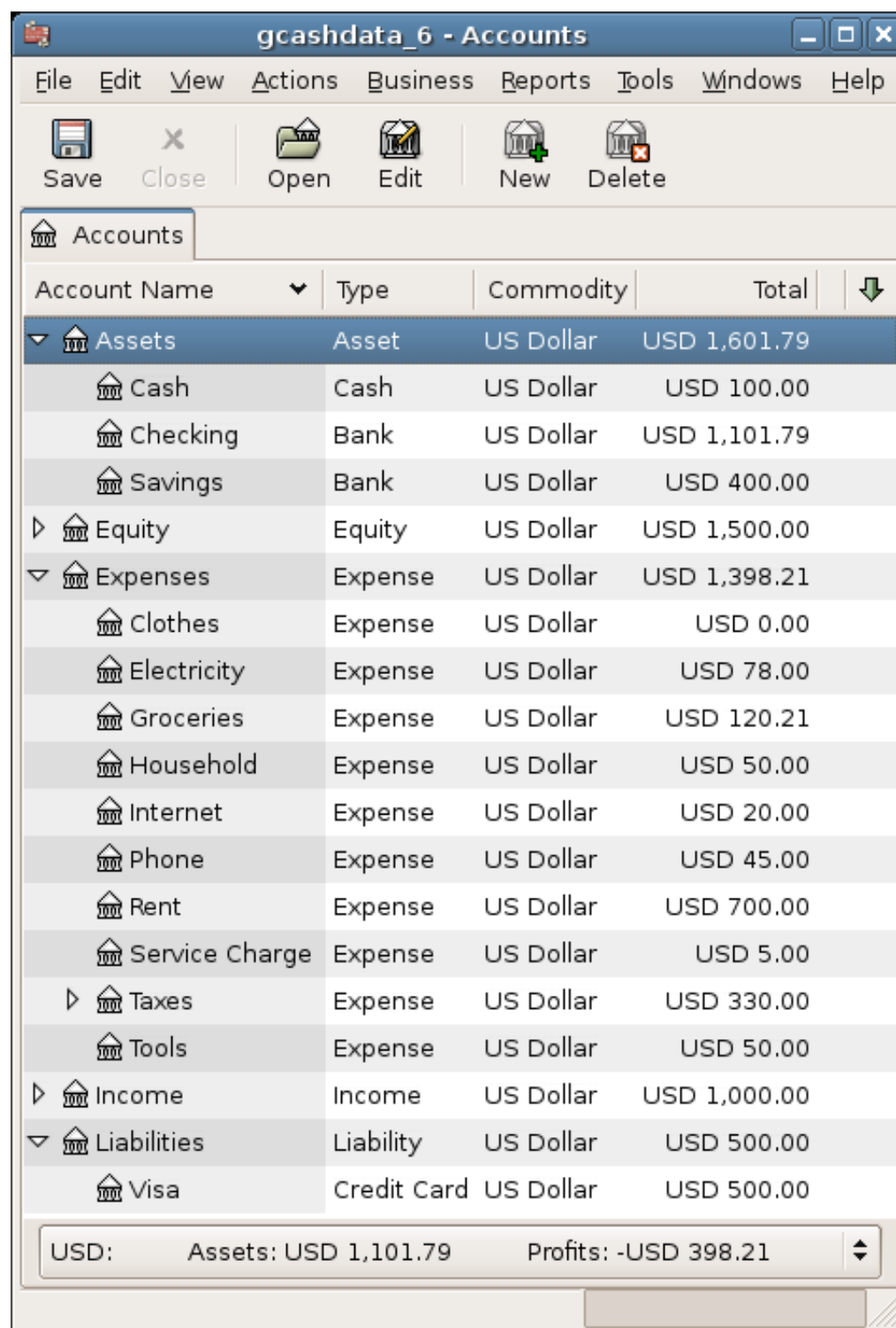
4. Save the GnuCash data file.

## 2.8.5 Putting It All Together

Now that we have covered the basic ideas behind the various transactions you must make to successfully track your credit card in GnuCash, let’s go through an example. In this example, we will make credit card purchases, refund two of the purchases, get charged interest on the unpaid balance, reconcile the credit card account, and finally make a partial payoff of the credit card.

## Open GnuCash file

Start with opening the previous datafile we stored, `gcashdata_5`, and store it as `gcashdata_6` directly. The main window should look something like this:



## Purchases

Let's make some purchases on our visa card. Start by buying \$25 worth of food from the Greasy Spoon Cafe, \$100 worth of clothing from Faux Pas Fashions, \$25 worth of gasoline from Premium Gasoline, \$125 worth of groceries and household items from Groceries R Us (split between \$85 in groceries and \$40 in household items) and finally, \$60 worth of household items from CheapMart.

We also redo the exercise in previous chapter, with purchasing a pair of Jeans for \$74.99 on April 3, and refund them two days later.

The register window for the credit card liability should look like this:

Date	Numb	Description	Transfer	R	Payment	Charge	Balance
01/03/06		Opening Balance	Equity:Opening Balance	n		500.00	500.00
03/04/06		Jeans R Us	Expenses:Clothes	n	74.99		574.99
05/04/06		Jeans R Us	Expenses:Clothes	n			500.00
10/04/06		Greasy Spoon Cafe	Expenses:Food	n		25.00	525.00
11/04/06		Faux Pas Fashion	Expenses:Clothes	n		100.00	625.00
12/04/06		Premium Gasolin	Expenses:Car	n		25.00	650.00
13/04/06		Groceries R Us	-- Split Transaction --	n		125.00	775.00
13/04/06		CheapMart	Expenses:Household	n		60.00	835.00

Present:USD 835.00 Future:USD 835.00 Cleared:USD 0.00 Reconciled:USD 0.00 Projected Minimum:USD 835.00  
Thursday 13 April 2006

## Refund

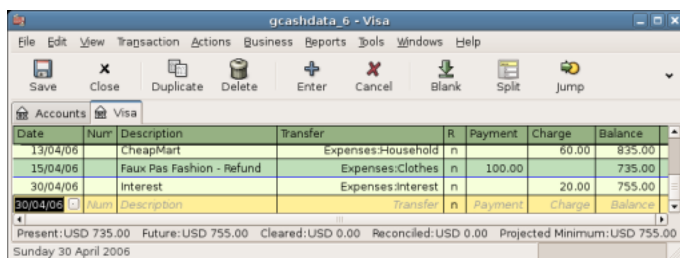
Now suppose that on April 15th you return the clothes you bought on April 11th from Faux Pas Fashions and they give you credit back on your credit card. Enter a transaction for the credit card refund for the full \$100 amount. Remember to use the same transfer account you used for the original purchase, and enter the amount under the Payment column. GnuCash will automatically complete the name and transfer account for you, but it will also automatically enter the \$100 in the Charge column. You will need to reenter the amount in the Payment column. The transaction looks like this:

Date	Numb	Description	Transfer	R	Payment	Charge	Balance
11/04/06		Faux Pas Fashion	Expenses:Clothes	n		100.00	625.00
12/04/06		Premium Gasolin	Expenses:Car	n		25.00	650.00
13/04/06		Groceries R Us	-- Split Transaction --	n		125.00	775.00
13/04/06		CheapMart	Expenses:Household	n		60.00	835.00
15/04/06		Faux Pas Fashion - Refund	Expenses:Clothes	n	100.00		735.00

Present:USD 735.00 Future:USD 735.00 Cleared:USD 0.00 Reconciled:USD 0.00 Projected Minimum:USD 735.00  
Enter a description of the transaction

## Interest Charge

After the month of spending, unfortunately, the credit card bill arrives in the mail or you access it on-line through the internet. You have been charged \$20 in interest on the last day of April because of the balance you carried from the previous month. This gets entered into the credit card account as an expense.



The screenshot shows the GnuCash interface for a 'Visa' account. The table lists transactions with columns for Date, Num, Description, Transfer, R, Payment, Charge, and Balance.

Date	Num	Description	Transfer	R	Payment	Charge	Balance
13/04/06		CheapMart	Expenses:Household	n		60.00	835.00
15/04/06		Faux Pas Fashion - Refund	Expenses:Clothes	n	100.00		735.00
30/04/06		Interest	Expenses:Interest	n		20.00	755.00
30/04/06			Transfer	n			

Summary: Present: USD 735.00, Future: USD 755.00, Cleared: USD 0.00, Reconciled: USD 0.00, Projected Minimum: USD 755.00. Sunday 30 April 2006.

## Reconciliation

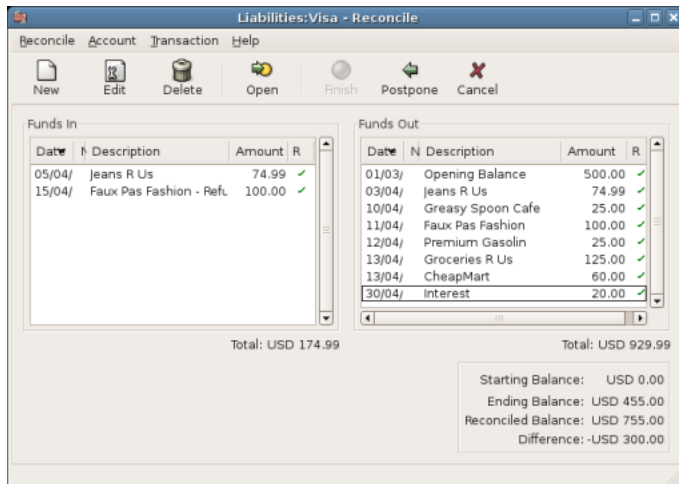
When your credit card bill arrives you should reconcile your credit card account to this document. This is done using GnuCash's built-in reconciliation application. Highlight the credit card account and click on Actions > Reconcile. . . This reconciliation procedure is described in detail in the [Reconciliation](#), but we will step through the process here as well. For this example, let's assume that the credit card statement is dated May 1st, with a final balance of \$455. Enter these values in to the initial Reconcile window as shown here.



The 'Liabilities: Visa - Reconcile' dialog box contains the following fields and controls:

- Reconcile Information**
  - Statement Date: 01/04/06 (dropdown menu)
  - Starting Balance: USD 0.00
  - Ending Balance: 455.00
  - ☐ Include subaccounts
  - Enter Interest Charge... (button)
- Buttons: Cancel, OK

During the reconciliation process, you check off each transaction in the account as you confirm that the transaction appears in both your GnuCash account and the credit card statement. For this example, as shown in the figure below, there is a \$300 difference between your GnuCash accounts and the credit card statement.



Some investigation uncovers that you forgot to record a payment you made on March 5th to the credit card company for \$300, you must enter this payment transaction from your bank account to the credit card. Now the credit card statement and your GnuCash account can be reconciled, with a balance of \$455.

## Payment

Assuming you have completed reconciliation of your credit card account, you need to make a payment to the credit card company. In this example, we owe \$455 but will make a partial payment of \$300 again this month. To do so, enter a transaction from your bank account to the credit card account for \$300, which should reduce your credit card balance to \$155. Your credit card account register should now appear like this:

Date	Num	Description	Transfer	R	Payment	Charge	Balance
01/03/06		Opening Balance	Equity:Opening Balance	y		500.00	500.00
05/03/06		Partial Payment	Assets:Checking	y	300.00		200.00
03/04/06		Jeans R Us	Expenses:Clothes	y		74.99	274.99
05/04/06		Jeans R Us	Expenses:Clothes	y		74.99	200.00
10/04/06		Greasy Spoon Cafe	Expenses:Food	y		25.00	225.00
11/04/06		Faux Pas Fashion	Expenses:Clothes	y		100.00	325.00
12/04/06		Premium Gasolin	Expenses:Car	y		25.00	350.00
13/04/06		Groceries R Us	-- Split Transaction --	y		125.00	475.00
13/04/06		CheapMart	Expenses:Household	y		60.00	535.00
15/04/06		Faux Pas Fashion - Refund	Expenses:Clothes	y		100.00	435.00
30/04/06		Interest	Expenses:Interest	y		20.00	455.00
05/05/06		Partial Payment of Visa Bill	Assets:Checking	n	300.00		155.00

Present: USD 435.00 Future: USD 155.00 Cleared: USD 455.00 Reconciled: USD 455.00 Projected Minimum: USD 455.00  
Friday 05 May 2006

Go back to the main window and save your file (`gcashdata_6`). Your chart of accounts is steadily growing, and it should now look like this:

gcashdata_6 - Accounts				
File Edit View Actions Business Reports Tools Windows Help				
Save          Close          Open          Edit          New          Delete				
Accounts				
Account Name	Type	Commodity	Total	
Assets	Asset	US Dollar	USD 1,001.79	
Cash	Cash	US Dollar	USD 100.00	
Checking	Bank	US Dollar	USD 501.79	
Savings	Bank	US Dollar	USD 400.00	
Equity	Equity	US Dollar	USD 1,500.00	
Expenses	Expense	US Dollar	USD 1,653.21	
Car	Expense	US Dollar	USD 25.00	
Clothes	Expense	US Dollar	USD 0.00	
Electricity	Expense	US Dollar	USD 78.00	
Food	Expense	US Dollar	USD 25.00	
Groceries	Expense	US Dollar	USD 205.21	
Household	Expense	US Dollar	USD 150.00	
Interest	Expense	US Dollar	USD 20.00	
Internet	Expense	US Dollar	USD 20.00	
Phone	Expense	US Dollar	USD 45.00	
Rent	Expense	US Dollar	USD 700.00	
Service Charge	Expense	US Dollar	USD 5.00	
Taxes	Expense	US Dollar	USD 330.00	
Tools	Expense	US Dollar	USD 50.00	
Income	Income	US Dollar	USD 1,000.00	
Liabilities	Liability	US Dollar	USD 155.00	
Visa	Credit Card	US Dollar	USD 155.00	
USD:      Assets: USD 846.79      Profits: -USD 653.21				

## Save file

Last, save the GnuCash data file (gcashdata\_6).

## Reports

As we did in the previous chapters, let's have a look at a Cash Flow, and a Transaction Report.

1. First let's have a look at the Cash Flow report for the liability account Visa during the month of March.

Select the cash flow report from Reports > Income & Expense > Cash Flow.

### Cash Flow - 01/03/06 to 30/04/06 for

Selected Accounts

● Liabilities:Visa

Money into selected accounts comes from	
<u>Assets:Checking</u>	USD 300.00
<u>Expenses:Clothes</u>	USD 174.99
<b>Money In</b>	<b>USD 474.99</b>
Money out of selected accounts goes to	
<u>Equity:Opening Balance</u>	USD 500.00
<u>Expenses:Car</u>	USD 25.00
<u>Expenses:Clothes</u>	USD 174.99
<u>Expenses:Food</u>	USD 25.00
<u>Expenses:Groceries</u>	USD 85.00
<u>Expenses:Household</u>	USD 100.00
<u>Expenses:Interest</u>	USD 20.00
<b>Money Out</b>	<b>USD 929.99</b>
<b>Difference</b>	<b>-USD 455.00</b>

2. Now let's have a look at corresponding transaction report for the Visa account.

Select the transaction report from Reports > Transaction Report.

**Transaction Report****From 01/03/06 To 30/04/06**

Date	Num	Description	Transfer from/to	Amount
<u>Visa</u>				
March 2006				
01/03/06		Opening Balance	Opening Balance	<u>USD 500.00</u>
05/03/06		Partial Payment	Checking	<u>-USD 300.00</u>
Total For March 2006				<b>USD 200.00</b>
April 2006				
03/04/06		Jeans R Us	Clothes	<u>USD 74.99</u>
05/04/06		Jeans R Us	Clothes	<u>-USD 74.99</u>
10/04/06		Greasy Spoon Cafe	Food	<u>USD 25.00</u>
11/04/06		Faux Pas Fashion	Clothes	<u>USD 100.00</u>
12/04/06		Premium Gasolin	Car	<u>USD 25.00</u>
13/04/06		Groceries R Us	Split	<u>USD 125.00</u>
13/04/06		CheapMart	Household	<u>USD 60.00</u>
15/04/06		Faux Pas Fashion - Refund	Clothes	<u>-USD 100.00</u>
30/04/06		Interest	Interest	<u>USD 20.00</u>
Total For April 2006				<b>USD 255.00</b>
Total For Visa				<b>USD 455.00</b>
<b>Grand Total</b>				<b>USD 455.00</b>

3. Now let's change the transaction report to only show the various Expenses accounts.



**Transaction Report****From 01/04/06 To 30/04/06**

Date	Num	Description	Account	Amount
<u>Car</u>				
12/04/06		Premium Gasolin	Car	<u>USD 25.00</u>
Total For Car				<b>USD 25.00</b>
<u>Clothes</u>				
03/04/06		Jeans R Us	Clothes	<u>USD 74.99</u>
05/04/06		Jeans R Us	Clothes	<u>-USD 74.99</u>
11/04/06		Faux Pas Fashion	Clothes	<u>USD 100.00</u>
15/04/06		Faux Pas Fashion - Refund	Clothes	<u>-USD 100.00</u>
Total For Clothes				<b>USD 0.00</b>
<u>Food</u>				
10/04/06		Greasy Spoon Cafe	Food	<u>USD 25.00</u>
Total For Food				<b>USD 25.00</b>
<u>Groceries</u>				
13/04/06		Groceries R Us	Groceries	<u>USD 85.00</u>
Total For Groceries				<b>USD 85.00</b>
<u>Household</u>				
13/04/06		Groceries R Us	Household	<u>USD 40.00</u>
13/04/06		CheapMart	Household	<u>USD 60.00</u>
Total For Household				<b>USD 100.00</b>
<u>Interest</u>				
30/04/06		Interest	Interest	<u>USD 20.00</u>
Total For Interest				<b>USD 20.00</b>
<u>Rent</u>				
28/04/06		May Rent	Rent	<u>USD 350.00</u>
Total For Rent				<b>USD 350.00</b>
Grand Total				<b>USD 605.00</b>

## 2.9 Loans

This chapter explains how to manage your loans with GnuCash.

### 2.9.1 Basic Concepts

A loan is defined as a financial transaction in which someone pays for the use of someone else's money. There are many familiar examples of loans: credits cards, auto loans, house mortgages, or a business loan.

## Terminology

Before discussing tracking loan in GnuCash specifically, it will be helpful to present a glossary of terminology. The terms presented below represent some of the basic concepts found concerning loans. It is a good idea to become familiar with these terms, or at least, refer back to this list if you encounter an unfamiliar word in the later sections.

- *Amortization* - the repayment plan which will insure that a loan is eventually paid off, typically utilizing equal valued monthly payments. These payments are usually split into principal and interest, where the amount of principal per payment increases (and interest decreases) as the amortization period elapses.
- *Borrower* - the person or company that receives the money from a loan.
- *Default* - when a borrower fails to repay a loan according to the terms agreed upon with the lender.
- *Deferment* - a temporary delay in the repayment of a loan.
- *Delinquency* - is the term that refers to late payments.
- *Disbursement* - amount of the loan paid to the borrower. Some loans have multiple disbursements, meaning the borrower does not receive the full amount of the loan at one time.
- *Interest* - the expense charged by the lender to the borrower for the use of the money loaned. This is typically expressed in terms of a yearly percentage charged on the principal borrowed, known as the *Annual Percentage Rate* or APR.
- *Lender* - the company or person who lends money to a borrower.
- *Loan Fee* - a processing fee removed from the principal at the time the borrower receives a loan.
- *Principal* - the original amount of the loan, or the amount of the original loan that is still owed. When you make a monthly payment on a loan, part of the money pays the interest, and part pays the principal.
- *Promissory Note* - the legal agreement between the borrower and lender concerning the loan.

### 2.9.2 Setting Up Accounts

When a borrower obtains a loan, it is usually with the intention to make a purchase of something of value. In fact, most loans require the borrower to buy some predetermined asset, such as a house. This asset is insurance against the borrower defaulting on the loan. There are, of course, examples of loans which do not necessarily have an associated high value asset, such as educational loans.

For the account structure presented here, we will assume the loan was used to purchase a compensating asset.

A loan is a liability, the interest you accrue on the loan is an on-going expense, and any administrative fees you may have to pay would be another expense. The thing purchased with the money from a loan is an asset. With these parameters, we can now present a basic loan account structure:

#### Basic Loan Account Structure

```
-Asset
  -Current Assets
    -Savings Account
  -Fixed Assets
    -Asset Purchased
-Liability
  -Loans
    -Mortgage Loan
-Expenses
  -Interest
```

(continues on next page)

(continued from previous page)

-Mortgage Interest  
-Mortgage Adm Fees

GnuCash has a number of predefined loan account hierarchies available, including Car Loans and Home Mortgage Loans. To access these predefined account structures, click on Actions > New Account Hierarchy... and select the loan types in which you are interested.

### 2.9.3 Calculations

Determining loan amortization schedules, periodic payment amounts, total payment value, or interest rates can be somewhat complex. To help facilitate these kinds of calculations, GnuCash has a built-in Loan Repayment Calculator. To access the calculator, go to Tools > Loan Repayment Calculator.

The Loan Repayment Calculator can be used to calculate any one of the parameters: Payment Periods, Interest Rate, Present Value, Periodic Payment, or Future Value given that the other 4 have been defined. You will also need to specify the compounding and payment methods.

- *Payment Periods* - the number of payment periods.
- *Interest Rate* - the nominal interest rate of the loan, ie: the yearly interest rate.
- *Present Value* - the present value of the loan, ie: current amount owed on the loan.
- *Periodic Payment* - the amount to pay per period.
- *Future Value* - the future value of the loan, ie: the amount owed after all payment periods are over.
- *Compounding* - two interest compounding methods exist, discrete or continuous. For discrete compounding select the compounding frequency from the popup menu with a range from yearly to daily.
- *Payments* - the popup menu allows you to select the payment frequency with a range from yearly to daily. You can also select whether your payments occur at the beginning or end of the period. Payments made at the beginning of the payment period have interest applied to the payment as well as any previous money paid or money still owed.

### Example: Monthly Payments

What is your monthly payment on a \$100000 30 year loan at a fixed rate of 4% compounded monthly?

This scenario is shown in the example image above. To perform this calculation, set Payment Periods to 360 (12 months x 30 years), Interest Rate to 4, Present Value to 100000, leave Periodic Payment empty and set Future Value to 0 (you do not want to owe anything at the end of the loan). Compounding is Monthly, Payments are Monthly, assume End of Period Payments, and Discrete Compounding. Now, click on the Calculate button next to the Periodic Payment area. You should see \$-477.42.

Answer: You must make monthly payments of 477.42.

### Example: Length of Loan

How long will you be paying back a \$20000 loan at 10% fixed rate interest compounded monthly if you pay \$500 per month?

To perform this calculation, leave Payment Periods empty, set Interest Rate to 10, Present Value to 20000, Periodic Payment is -500, and Future Value is 0 (you do not want to owe anything at the end of the loan). Compounding is Monthly, Payments are Monthly, assume End of Period payments, and Discrete Compounding. Now, click on the Calculate. You should see 49 in the Payment Periods field.

Answer: You will pay off the loan in 4 years and 1 month (49 months).

### Advanced: Calculation Details

In order to discuss the mathematical formulas used by the Loan Repayment Calculator, we first must define some variables.  $n$  == number of payment periods  $i$  == nominal interest rate, NAR, charged  $PV$  == Present Value  $PMT$  == Periodic Payment  $FV$  == Future Value  $CF$  == Compounding Frequency per year  $PF$  == Payment Frequency per year Normal values for  $CF$  and  $PF$  are: 1 == annual 2 == semi-annual 3 == tri-annual 4 == quarterly 6 == bi-monthly 12 == monthly 24 == semi-monthly 26 == bi-weekly 52 == weekly 360 == daily 365 == daily

### Converting between nominal and effective interest rate

When a solution for  $n$ ,  $PV$ ,  $PMT$  or  $FV$  is required, the nominal interest rate ( $i$ ) must first be converted to the effective interest rate per payment period ( $ieff$ ). This rate,  $ieff$ , is then used to compute the selected variable. When a solution for  $i$  is required, the computation produces the effective interest rate ( $ieff$ ). Thus, we need functions which convert from  $i$  to  $ieff$ , and from  $ieff$  to  $i$ .

To convert **from**  $i$  to  $ieff$ , the following expressions are used:

Discrete Interest:  $ieff = (1 + i/CF)^{(CF/PF)} - 1$

Continuous Interest:  $ieff = e^{(i/PF)} - 1 = \exp(i/PF) - 1$

To convert **from**  $ieff$  to  $i$ , the following expressions are used:

Discrete Interest:  $i = CF * [(1 + ie\text{ff})^{(PF/CF)} - 1]$

Continuous Interest:  $i = \ln[(1 + ie\text{ff})^{PF}]$

---

**Note:** In the equations below for the financial transaction, all interest rates are the effective interest rate, “ $ieff$ ”. For the sake of brevity, the symbol will be shortened to just “ $i$ ”.

---

## The basic financial equation

One equation fundamentally links all the 5 variables. This is known as the fundamental financial equation:

$$PV * (1 + i)^n + PMT * (1 + iX) * [(1+i)^n - 1] / i + FV = 0$$

Where: X = 0 **for** end of period payments, **and**  
X = 1 **for** beginning of period payments

From this equation, functions which solve for the individual variables can be derived. For a detailed explanation of the derivation of this equation, see the comments in the file `src/calculation/fin.c` from the GnuCash source code. The A, B, and C variables are defined first, to make the later equations simpler to read.

```
A = (1 + i)^n - 1
B = (1 + iX) / i
C = PMT * B
```

```
n = ln[(C - FV) / (C + PV)] / ln((1 + i))
PV = -[FV + A * C] / (A + 1)
PMT = -[FV + PV * (A + 1)] / [A * B]
FV = -[PV + A * (PV + C)]
```

The solution **for** interest **is** broken into two cases.  
The simple case **for** when PMT == 0 gives the solution:  
 $i = [FV/PV]^{(1/n)} - 1$

The case where PMT != 0 is fairly complex and will not be presented here. Rather than involving an exactly solvable function, determining the interest rate when PMT != 0 involves an iterative process. Please see the `src/calculation/fin.c` source file for a detailed explanation.

## Example: Monthly Payments

Let's recalculate *Example: Monthly Payments*, this time using the mathematical formulas rather than the Loan Repayment Calculator. What is your monthly payment on a \$100000 30 year loan at a fixed rate of 4% compounded monthly?

First, let's define the variables:  $n = (30 * 12) = 360$ ,  $PV = 100000$ ,  $PMT = \text{unknown}$ ,  $FV = 0$ ,  $i = 4\% = 4/100 = 0.04$ ,  $CF = PF = 12$ ,  $X = 0$  (end of payment periods).

The second step is to convert the nominal interest rate ( $i$ ) to the effective interest rate ( $ieff$ ). Since the interest rate is compounded monthly, it is discrete, and we use:  $ieff = (1 + i/CF)^{(CF/PF)} - 1$ , which gives  $ieff = (1 + 0.04/12)^{(12/12)} - 1$ , thus  $ieff = 1/300 = 0.0033333$ .

Now we can calculate A and B.  $A = (1 + i)^n - 1 = (1 + 1/300)^{360} - 1 = 2.313498$ .  $B = (1 + iX)/i = (1 + (1/300)*0)/(1/300) = 300$ .

With A and B, we can calculate PMT.  $PMT = -[FV + PV * (A + 1)] / [A * B] = -[0 + 100000 * (2.313498 + 1)] / [2.313498 * 300] = -331349.8 / 694.0494 = -477.415296 = -477.42$ .

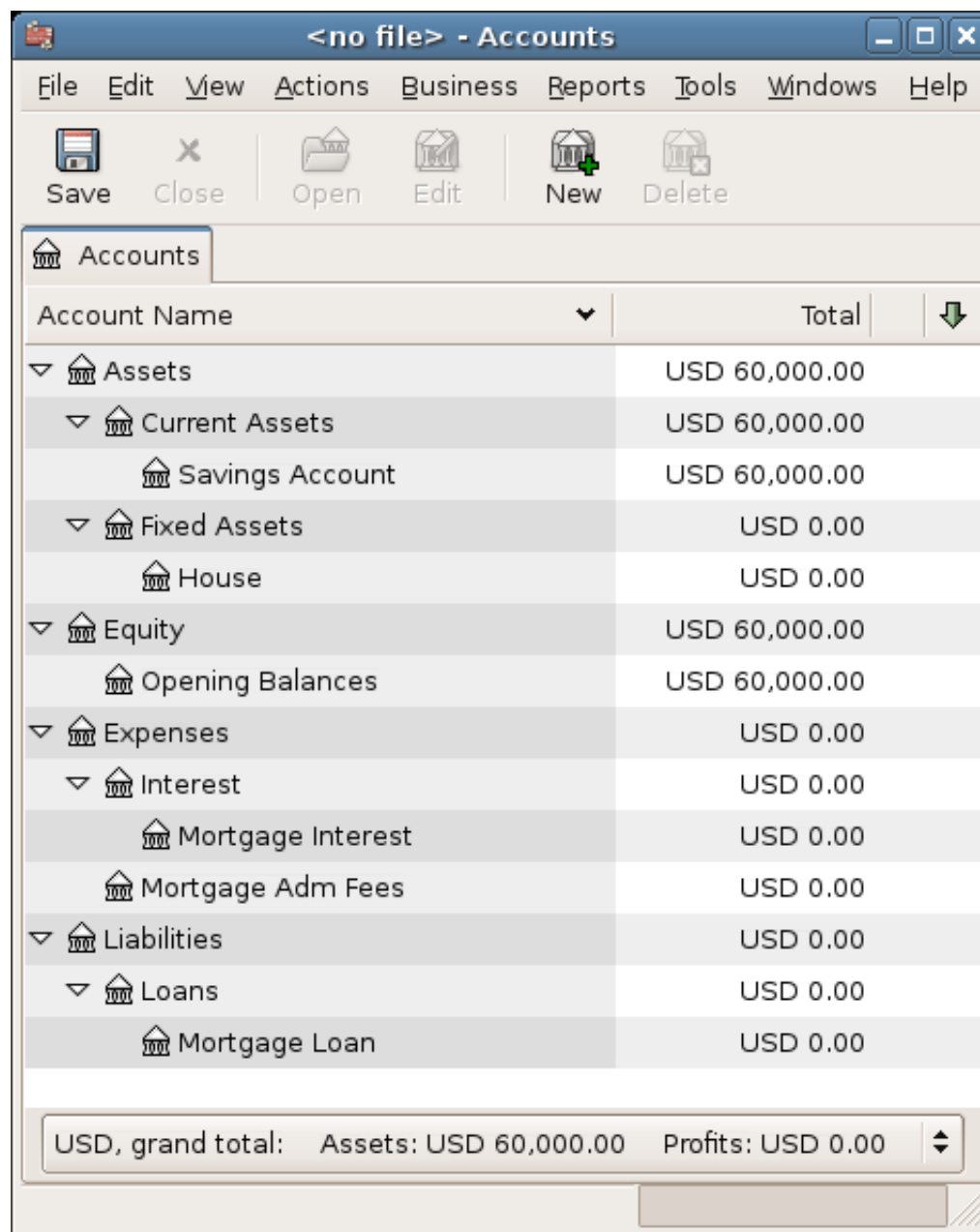
Answer: You must make monthly payments of 477.42.

## 2.9.4 House Mortgage (How-To)

A house mortgage can be setup using the account structure present in *Setting Up Accounts*.

As an example, assume you have \$60k in your bank account, and you buy a \$150k house. The mortgage is charging 6% APR, and has administrative fees (closing costs, etc) of 3%. You decide to put \$50k down, and thus will need to borrow \$103k, which will give you \$100 after the closing costs are paid (3% of \$100k).

Your accounts before borrowing the money:



The screenshot shows the 'Accounts' window in GnuCash. The title bar is '<no file> - Accounts'. The menu bar includes File, Edit, View, Actions, Business, Reports, Tools, Windows, and Help. Below the menu is a toolbar with icons for Save, Close, Open, Edit, New, and Delete. The main area displays a tree of accounts with a table showing the balance for each. The accounts are categorized into Assets, Equity, Expenses, and Liabilities. The 'Assets' category is expanded, showing 'Current Assets' (USD 60,000.00) and 'Fixed Assets' (USD 0.00). 'Current Assets' includes 'Savings Account' (USD 60,000.00). 'Fixed Assets' includes 'House' (USD 0.00). 'Equity' is USD 60,000.00, 'Expenses' is USD 0.00, and 'Liabilities' is USD 0.00. The 'Expenses' category is expanded, showing 'Interest' (USD 0.00), 'Mortgage Interest' (USD 0.00), and 'Mortgage Adm Fees' (USD 0.00). The 'Liabilities' category is expanded, showing 'Loans' (USD 0.00) and 'Mortgage Loan' (USD 0.00). At the bottom, a summary bar shows 'USD, grand total: Assets: USD 60,000.00 Profits: USD 0.00'.

Account Name	Total
Assets	USD 60,000.00
Current Assets	USD 60,000.00
Savings Account	USD 60,000.00
Fixed Assets	USD 0.00
House	USD 0.00
Equity	USD 60,000.00
Opening Balances	USD 60,000.00
Expenses	USD 0.00
Interest	USD 0.00
Mortgage Interest	USD 0.00
Mortgage Adm Fees	USD 0.00
Liabilities	USD 0.00
Loans	USD 0.00
Mortgage Loan	USD 0.00

USD, grand total: Assets: USD 60,000.00 Profits: USD 0.00

The purchase of the house is recorded with a split transaction in the *Assets:House* account, with \$50k coming from the bank (IE: your down payment), and \$100k coming from the Mortgage. You can place the \$3k closing costs in the same split, and we increase the house loan to \$103k to include the closing costs as well.

Table 5: Buying a House Split Transaction

Account	Increase	Decrease
Assets:Fixed Assets:House	\$150,000	
Assets:Current Assets:Bank		\$50,000
Liabilities:Loans:Mortgage Loan		\$103,000
Expenses:Mortgage Adm Fees	\$3000	

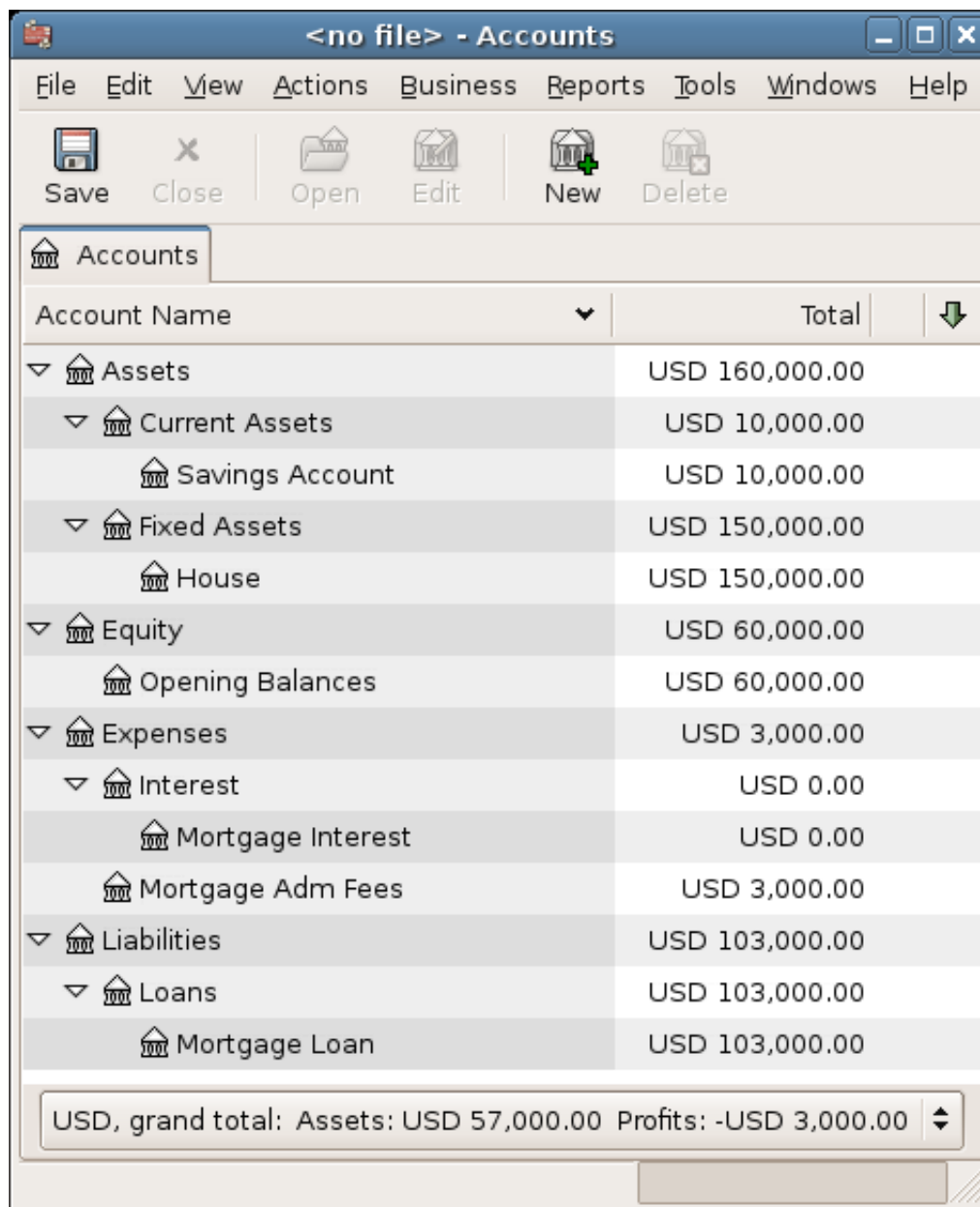
The split will look like this in the *Assets:Fixed Assets:House* Account:

Date	Description		Tot Increase	Tot Decrease	Balance
01/01/06	Buy house		150,000.00	150,000.00	
	Total value of house	Assets:Fixed Assets:House	n 150,000.00		
	Closing costs	Expenses:Mortgage Adm Fees	n 3,000.00		
	Down payment	Assets:Current Assets:Savings Account	n 50,000.00		
	Loan & Closing costs	Liabilities:Loans:Mortgage Loan	n 103,000.00		

Present: USD 150,000.00 Future: USD 150,000.00 Cleared: USD 0.00 Reconciled: USD 0.00 Projected Minimum: USD 150,000.00

Buy house

Which will give a Chart of Accounts like this:



The screenshot shows the 'Accounts' window in GnuCash. The window title is '<no file> - Accounts'. The menu bar includes File, Edit, View, Actions, Business, Reports, Tools, Windows, and Help. Below the menu bar are icons for Save, Close, Open, Edit, New, and Delete. The main area displays a tree of accounts with a 'Total' column and a sort icon. The accounts are organized as follows:

Account Name	Total
Assets	USD 160,000.00
Current Assets	USD 10,000.00
Savings Account	USD 10,000.00
Fixed Assets	USD 150,000.00
House	USD 150,000.00
Equity	USD 60,000.00
Opening Balances	USD 60,000.00
Expenses	USD 3,000.00
Interest	USD 0.00
Mortgage Interest	USD 0.00
Mortgage Adm Fees	USD 3,000.00
Liabilities	USD 103,000.00
Loans	USD 103,000.00
Mortgage Loan	USD 103,000.00

At the bottom, a summary bar shows: USD, grand total: Assets: USD 57,000.00 Profits: -USD 3,000.00

### 2.9.5 A Personal Loan to a friend (How-To)

It is not always you are borrowing money from the bank, sometimes you borrow money from your family, or perhaps even lend money to a friend. This How-To chapter will describe one way to handle lending money to a friend.

We are basing this How-To on the following generic account structure.

```
-Asset
  -Bank
    -Bank Account
  -Money owed to you
    -Person
-Income
```

(continues on next page)



(continued from previous page)

-Interest Income  
-Person

This example will show how to track a personal loan of 2,000 USD (default currency) to your friend Peter

## Loan Specifications

Peter wants to borrow \$2,000 dollars from you and plans to pay you back monthly for the next 18 months. Since he is your friend, (but not that close) you both agree on a yearly interest rate of 5%.

In summary we have the below details. Peter's loan details:

- Pinciple Amount - \$2,000
- Term - 18 months with 12 payments per year
- Annual Percentage Rate: 5%
- Monthly Payment : ??

So how do you calculate the Monthly Payment?

You have a number of different options, like paper and pen, Linux Calculator, Open Office's Calc module, but the easiest is to use GnuCash *Loan Repayment Calculator*. This tells you that the Monthly Payment should be \$115.56.

But you need to know how much of this is Interest and how much is Principal to be able to do a proper bookkeeping. For this you need a more powerful tool, something like the Calc module in OpenOffice.org, and in particular the PMT function.

		Start Amount	Interest	Principal	End Amount
Loan Amount	\$2,000.00	Jan 2,000.00	8.33	107.23	1,892.77
Number of months	18	Feb 1,892.77	7.89	107.67	1,785.10
Payments per year	12	Mar 1,785.10	7.44	108.12	1,676.97
Interest per year	5%	Apr 1,676.97	6.99	108.57	1,568.40
Monthly payment	\$115.56	May 1,568.40	6.54	109.03	1,459.37
		Jun 1,459.37	6.08	109.48	1,349.89
		Jul 1,349.89	5.62	109.94	1,239.96
		Aug 1,239.96	5.17	110.39	1,129.56
Monthly Payment=PMT(C8/C7,C6,-C5,0)		Sep 1,129.56	4.71	110.85	1,018.71
Interest=(\$C\$8/\$C\$7)*F4		Oct 1,018.71	4.24	111.32	907.39
Principal=\$C\$9-G4		Nov 907.39	3.78	111.78	795.61
End Amount=F4-H4		Dec 795.61	3.32	112.25	683.37
		Jan 683.37	2.85	112.71	570.65
		Feb 570.65	2.38	113.18	457.47
		Mar 457.47	1.91	113.65	343.81
		Apr 343.81	1.43	114.13	229.69
		May 229.69	0.96	114.60	115.08
		Jun 115.08	0.48	115.08	-0.00
		<b>Total</b>	<b>\$80.10</b>	<b>\$2,000.00</b>	

## Accounts for the loan

Let's start with the following accounts (all accounts have the same currency, in this case USD)

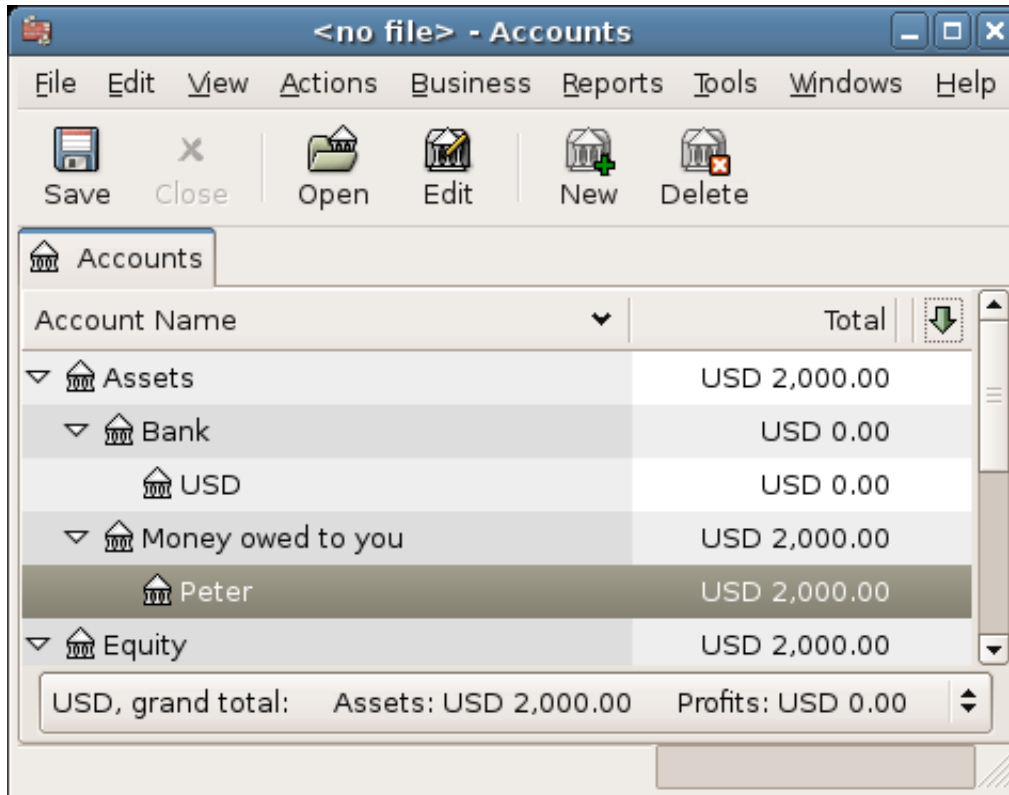
Assets:Bank:USD  
Assets:Money owed to you:Peter  
Income:Interest Income:Peter  
Equity:Opening Balances:USD

## Lending the money

When you have lent money to your friend, you have in fact moved money from an Asset account (like Bank, Checking or similar) to your Asset account *Money owed to you*. To record this you enter the following transaction into the *Assets:Money owed to you:Friend* account.

Table 6: Personal loan to a Friend

Account	Increase	Decrease
Assets:Money owed to you:Friend	\$2,000	
Assets:Bank:USD		\$2,000



## Receiving first payment

When the first payment (\$115.56) is received, you will need to determine how much is for the principal loan, and how much is for the loan interest.

- Outstanding loan amount this period = \$2,000
- Payment per month = \$115.56
- Payment breakdown
- $5\%/12 * \$2,000 = \$8.33$  Interest
- $\$115.56 - \$8.33 = \$107.23$  Principal

This can be translated to the following GnuCash entry

Date	Memo	Account	R	Deposit	Withdraw
01/01/06	Loan to Peter	Assets:Money owed to you:Peter	n	2,000.00	2,000.00
		Assets:Bank:USD	n		2,000.00
01/02/06	First payment	Assets:Bank:USD	n	115.56	107.23
	Principal	Assets:Money owed to you:Peter	n		107.23
	Interest	Income:Interest Income:Peter	n		8.33

Present:USD 1,892.77 Future:USD 1,892.77 Cleared:USD 0.00 Reconciled:USD 0.00 Projected Minimum:USD 1,892.77

The balance on Peter's loan is now  $\$2,000 - \$107.23 = \$1,892.77$

## Receiving second payment

When the second payment (\$115.56) is received, you will again need to determine how much is for the principal loan, and how much is for the loan interest.

- Outstanding loan amount this period = \$1,892.77
- Payment per month = \$115.56
- Payment breakdown
- $5\%/12 * \$1,892.77 = \$7.89$  Interest
- $\$115.56 - \$7.89 = \$107.67$  Principal

This can be translated to the following GnuCash entry

Date	Description	Account	Tot Depos	Tot Withd	Balance
01/01/06	Loan to Peter	Assets:Bank:USD	n	2,000.00	2,000.00
01/02/06	First payment	-- Split Transaction --	n	107.23	1,892.77
01/03/06	Second payment	Assets:Bank:USD	n	115.56	1,785.10
	Principal	Assets:Money owed to you:Peter	n		107.67
	Interest	Income:Interest Income:Peter	n		7.89

Present:USD 1,785.10 Future:USD 1,785.10 Cleared:USD 0.00 Reconciled:USD 0.00 Projected Minimum:USD 1,785.10

The balance on Peter's loan is now  $\$1,892.77 - \$107.67 = \$1,785.10$

The Chart of accounts looks now like this

Account Name	Total
Assets	USD 2,016.22
Bank	USD 231.12
USD	USD 231.12
Money owed to you	USD 1,785.10
Peter	USD 1,785.10
Equity	USD 2,000.00
Income	USD 16.22
Interest Income	USD 16.22
Peter	USD 16.22
USD, grand total: Assets: USD 2,016.22 Profits: USD 16.22	

As you can see, the interest varies for every month, as well as the principal amount. So for every payment you receive you need to calculate the proper amounts for your various split entries.

The interest amount will be less and less for every payment (since it is calculated on a smaller loan amount all the time), until the last payment where it is more or less 0. Please review the Figure of Detailed view over private loan to Peter for more details

## 2.9.6 Automobile Loan (How-To)

The Automobile Loan, or in common terms, Car Loan, is treated more or less exactly as the House loan. The only difference is different accounts, and different interest rates.

Basic Car Loan Account Structure -Asset -Current Assets -Savings Account -Fixed Assets -Car -Liability -Loans -Car Loan -Expenses -Interest -Car Loan Interest -Car Loan Adm Fees

For more information, please check *House Mortgage (How-To)*

## 2.9.7 Reconciling with the Loan Statements (How-To)

Reconciling a loan statement is no different from reconciling a bank or credit card statement.

During the period you should have recorded all the various loan related transactions, and every one of them are touching the Liability:Loans:Loan account. For instance, paying off a bit of the loan decreases your *Bank Account*,

and increases the *Loan account*, *Loan Interest* as well as perhaps *Loan administration fee*.

With the loan statement in your hands, open the Loan account, start the reconcile assistant, and tick off all the various transaction you have recorded. When you have finished, the reconciling difference should be 0, and if it is not, then you will have to go through the account and compare it with the loan statement to find the difference. When you have reached a 0 in difference, then your loan account is reconciled and you can finish the Reconcile assistant.

For more information on how to Reconcile, please check [Reconciliation](#)

## 2.9.8 Selling a house or a car (How-To)

When you will record the selling of your house in GnuCash you have some different options. Here we will go through two of them, one in which you only recorded the purchase amount, and now the selling amount. The other where you have followed the ups and downs on the property market and registered various Unrealized gains over the time.

### Simple Transaction

In this way you only record the proper sale amount.

Let's work through two samples of selling a house, one with a profit, and one with a loss. If you want to sell a car instead, just substitute the house account with a car account.

-Assets -Fixed Assets -House -Current Assets -Saving -Income -Capital Gains Long -House

- You bought a house for \$300,000 once upon a time, and now managed to sell it for \$600,000. How do you record this?

To record this you need to increase our bank account with the \$600k, and decrease some other accounts with \$600k. The house account only contains \$300k which is what you bought it for, so you move this amount to your bank account. That means you are lacking \$300k. This amount you fetch from the *Income:Capital Gains Long:House* account. The split transaction you enter into your *bank* account (*Assets:Current Assets:Saving*) should look like this.

Table 7: Selling an asset (house) with a profit

Account	Increase	Decrease
Assets:Current Assets:Saving	\$600,000	
Assets:Fixed Assets:House		\$300,000
Income:Capital Gains Long:House		\$300,000

- You bought a house for \$300,000 once upon a time, but due to a newly created airport, could only sell it for \$230,000. How do you record this?

To record this you need to increase your bank account with the \$230k, and decrease some other accounts with \$230k. The house account contains \$300k which is more than what you sold it for. So let's move \$230k of it to your bank account. After this you have \$70k remaining in your house account which needs to be removed. You move it to our *Income:Capital Gains Long:House* account, which will indicate a loss. The split transaction you enter into your *house* account (*Assets:Fixed Assets:House*) should look like this.

Table 8: Selling an asset (house) with a loss

Account	Increase	Decrease
Assets:Fixed Assets:House		\$300,000
Assets:Current Assets:Saving	\$230,000	
Income:Capital Gains Long:House	\$70,000	

## A More Complex Transaction

In this example, we will touch a little on some more complicated accounting principles. For more details on this subject, please check [Capital Gains](#)

Here we will only touch on the case when you have accurately estimated the current value of your house. For the other cases (over-, and under-estimated), please check [Capital Gains](#).

-Assets -Fixed Assets -House -Cost -Unrealized Gain -Current Assets -Saving -Income -Realized Gain -House -Unrealized Gain -House

You bought a house for \$300,000 once upon a time, and over the years kept a close look on the market and updated your records with the estimated current value of your house. At the time you want to sell it, you have determined that the current market value is \$600,000.

The difference between \$600,000 (estimated market value) and \$300,000 (purchase value) is the current Unrealized Gain value. Therefore you have a total of \$300,000 in your *Assets:Fixed Assets:House:Unrealized Gain* account

How do you record this sell transaction?

To record this you need to increase your bank account with the \$600k, and decrease some other accounts with \$600k. You must first change from unrealized gain to realized gain for your *Income* accounts. Lastly you need to transfer the full amounts from the *Assets:Fixed Assets:House* sub-accounts.

The transaction you enter into your *Income:Realized Gain:House* account should look like this.

Table 9: Selling an asset (house) with a profit

Account	Increase	Decrease
Income:Realized Gain:House	\$300,000	
Income:Unrealized Gain:House		\$300,000

The transaction you enter into your *Assets:Current Assets:Saving* account should look like this.

Table 10: Selling an asset (house) with a profit 2

Account	Increase	Decrease
Assets:Current Assets:Saving	\$600,000	
Assets:Fixed Assets:House:Cost		\$300,000
Assets:Fixed Assets:House:Unrealized Gain		\$300,000

After having recorded these transactions you see that your House Asset have a value of 0, your Savings account have increased with \$600,000, and lastly, the *Income:Realized Gain* have increased to \$300,000.

## 2.10 Investments

This chapter explains how to manage your investments with GnuCash. Most people have an investment plan, whether its just putting money into a CD account, investing through a company sponsored plan at your workplace or buying and selling stocks and bonds through a brokerage. GnuCash gives you tools to help you manage these investments such as the *Price Editor* which allows you to record changes in the prices of stocks you own.

### 2.10.1 Basic Concepts

An investment is something that you purchase in the hopes of generating income, or that you hope to sell in the future for more than you paid. Using this simple definition, many things could be considered investments: the house you

live in, a valuable painting, stocks in publicly traded companies, your savings account at the bank, or a certificate of deposit. These many types of investments will be discussed in this chapter in terms of how to track them using GnuCash.

## Terminology

Before discussing investments specifically, it will be helpful to present a glossary of investment terminology. The terms presented below represent some of the basic concepts of investing. It is a good idea to become familiar with these terms, or at least, refer back to this list if you encounter an unfamiliar word in the later sections.

**Capital gains** The difference between the purchase and selling prices of an investment. If the selling price is lower than the purchase price, this is called a *capital loss*. Also known as *realized gain/loss*.

**Commission** The fee you pay to a broker to buy or sell securities.

**Common stock** A security that represents a certain fractional ownership of a company. This is what you buy when you “buy stock” in a company on the open market. This is also sometimes known as *capital stock*.

**Compounding** The concept that the reinvested interest can later earn interest of its own (interest on interest). This is often referred to as *compound interest*.

**Dividends** Dividends are cash payments a company makes to shareholders. The amount of this payment is usually determined as some amount of the profits of the company. Note that not all common stocks give dividends.

**Equities** Equities are investments in which the investor becomes part (or whole) owner in something. This includes common stock in a company, or real estate.

**Interest** What a borrower pays a lender for the use of their money. Normally, this is expressed in terms of a percentage of the principal per year. For example, a savings account with 1% interest (you are the lender, the bank is the borrower) will pay you \$1 for every \$100 you keep there per year.

**Liquidity** A measure of how easily convertible an investment is to cash. Money in a savings account is very liquid, while money invested in a house has low liquidity because it takes time to sell a house.

**Principal** The original amount of money invested or borrowed.

**Realized vs Unrealized Gain/Loss** Unrealized gain or loss occurs when you’ve got a change in price of an asset. You realize the gain/loss when you actually sell the asset. See also *capital gain/loss*.

**Return** The total income plus capital gains or losses of an investment. See also *Yield*.

**Risk** The probability that the return on investment is different from what was expected. Investments are often grouped on a scale from low risk (savings account, government bonds) to high risk (common stock, junk bonds). As a general rule of thumb, the higher the risk the higher the possible return.

**Shareholder** A shareholder is a person who holds common stock in a company.

**Stock split** Occurs when a company offers to issue some additional multiple of shares for each existing stock. For example, a “2 for 1” stock split means that if you own 100 shares of a stock, you will receive an additional 100 at no cost to you. The unit price of the shares will be adjusted so there is no net change in the value, so in this example the price per share will be halved.

**Valuation** The process of determining the market value or the price the investment would sell at in a “reasonable time frame”.

**Yield** A measure of the amount of money you earn from an investment (IE: how much income you receive from the investment). Typically this is reported as a percentage of the principal amount. Yield does not include capital gains or losses (see Return). Eg: A stock sells for \$100 and gives \$2 in dividends per year has a yield of 2%.

### Types of Investments

Below is presented some of the broad types of investments available, and examples of each type.

- *Interest-bearing account or instrument*

This type of investment usually allows you immediate access to your money, and will typically pay you interest every month based on the amount of money you have deposited. Examples are bank savings accounts (and some interest bearing checking accounts) and cash accounts at your brokerage. This is a very low risk investment, in the US these accounts are often insured against loss, to a specified limit.

Sometimes an interest bearing investment is *time-locked*. This type of investment requires you to commit your money to be invested for a given period of time for which you receive a set rate of return. Usually, the longer you commit the higher the interest rates. If you withdraw your money before the maturity date, you will usually have to pay an early withdrawal penalty. This is a relatively lower risk investment. Examples are certificates of deposit or some government bonds. Other types of Bonds may have higher yields based on the higher risks from the quality of the issuer's "credit rating".

- *Stocks and Mutual Funds*

This is an investment you make in a company, in which you effectively become a part owner. There is usually no time lock on publicly traded stock, however there may be changes in the tax rates you pay on capital gains depending on how long you hold the stock. Thus, stocks are typically quite liquid, you can access your money relatively quickly. This investment is a higher risk, as you have no guarantee on the future price of a stock.

A mutual fund is a group investment mechanism in which you can buy into many stocks simultaneously. For example, a "S&P 500 index fund" is a fund which purchases all 500 stocks listed in the Standard and Poor's index. When you buy a share of this fund, you are really buying a small amount of each of the 500 stocks contained within the fund. Mutual funds are treated exactly like a single stock, both for tax purposes and in accounting.

- *Fixed Assets*

Assets that increase in value over time are another form of investment. Examples include a house, a plot of land, or a valuable painting. This type of investment is very difficult to determine the value of until you sell it. The tax implications of selling these items is varied, depending on the item. For example, you may have tax relief from selling a house if it is your primary residence, but may not receive this tax break on an expensive painting.

Fixed asset investments are discussed in [Capital Gains](#) and [Depreciation](#). Typically, there is not much to do in terms of accounting for fixed asset investments except recording the buying and selling transactions.

### 2.10.2 Setting Up Accounts

To setup investment accounts in GnuCash you can either use the predefined investment account hierarchy or create your own. The minimum you need to do to track investments is to setup an asset account for each type of investment you own. However, as we have seen in previous chapters, it is usually more logical to create a structured account hierarchy, grouping related investments together. For example, you may want to group all your publicly traded stocks under a parent account named after the brokerage firm you used to buy the stocks.

---

**Note:** Regardless of how you setup your account hierarchy, remember that you can always move accounts around later (without losing the work you've put into them), so your initial account hierarchy does not have to be perfect.

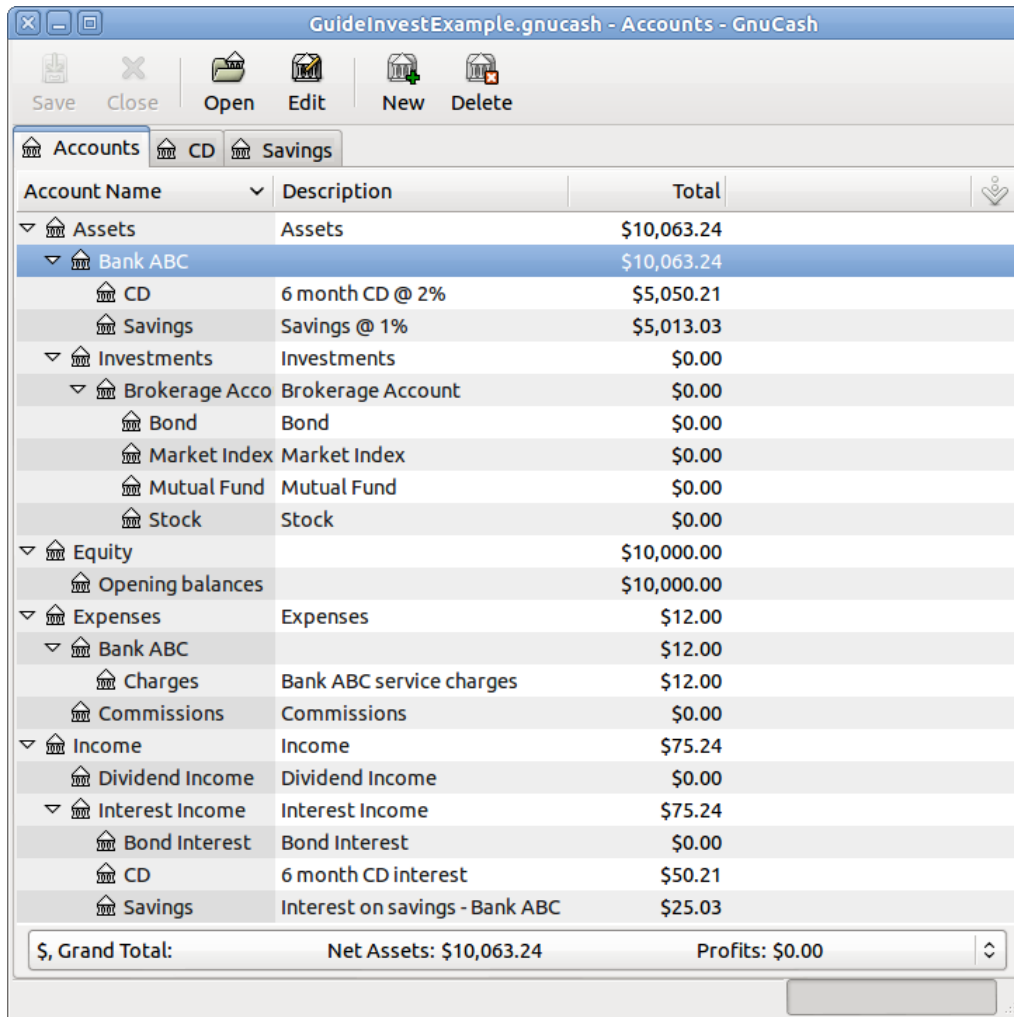
---



## Using Predefined Investment Accounts

The Investment Accounts option of the New Account Hierarchy Setup assistant will automatically create a basic investment account hierarchy for you. To access the predefined investment accounts hierarchy, you must make sure your GnuCash file is open, switch to the Accounts tab, and choose Actions > New Account Hierarchy. This will run the New Account Hierarchy Setup assistant and allow you to select additional accounts to add to your account hierarchy. Choose the Investment Accounts option (along with any others you are interested in). Assuming only investment accounts were selected, this will create an account hierarchy as shown below.

**Tip:** You can also run the New Account Hierarchy Setup assistant by creating a new GnuCash file.



The screenshot shows the GnuCash interface with the 'Accounts' window open. The window title is 'GuideInvestExample.gnucash - Accounts - GnuCash'. The 'Accounts' tab is selected, showing a hierarchical tree of accounts. The 'Assets' account is expanded, showing 'Bank ABC' as a sub-account. 'Bank ABC' is further expanded, showing 'CD' and 'Savings' as sub-accounts. 'CD' has a balance of \$5,050.21, and 'Savings' has a balance of \$5,013.03. The 'Investments' account is also expanded, showing 'Brokerage Account', 'Bond', 'Market Index', 'Mutual Fund', and 'Stock' as sub-accounts, all with a balance of \$0.00. The 'Equity' account is expanded, showing 'Opening balances' with a balance of \$10,000.00. The 'Expenses' account is expanded, showing 'Bank ABC' as a sub-account with a balance of \$12.00. 'Bank ABC' is further expanded, showing 'Charges' and 'Commissions' as sub-accounts, both with a balance of \$0.00. The 'Income' account is expanded, showing 'Dividend Income' and 'Interest Income' as sub-accounts, both with a balance of \$0.00. 'Interest Income' is further expanded, showing 'Bond Interest', 'CD', and 'Savings' as sub-accounts, with balances of \$0.00, \$50.21, and \$25.03 respectively. The 'Grand Total' row shows 'Net Assets: \$10,063.24' and 'Profits: \$0.00'.

Account Name	Description	Total
Assets	Assets	\$10,063.24
Bank ABC		\$10,063.24
CD	6 month CD @ 2%	\$5,050.21
Savings	Savings @ 1%	\$5,013.03
Investments	Investments	\$0.00
Brokerage Account	Brokerage Account	\$0.00
Bond	Bond	\$0.00
Market Index	Market Index	\$0.00
Mutual Fund	Mutual Fund	\$0.00
Stock	Stock	\$0.00
Equity		\$10,000.00
Opening balances		\$10,000.00
Expenses	Expenses	\$12.00
Bank ABC		\$12.00
Charges	Bank ABC service charges	\$12.00
Commissions	Commissions	\$0.00
Income	Income	\$75.24
Dividend Income	Dividend Income	\$0.00
Interest Income	Interest Income	\$75.24
Bond Interest	Bond Interest	\$0.00
CD	6 month CD interest	\$50.21
Savings	Interest on savings - Bank ABC	\$25.03
\$, Grand Total:		Net Assets: \$10,063.24 Profits: \$0.00

You will probably at least want to add a *Bank* account to the Assets and probably an *Equity:Opening Balances* account, as we have done in previous chapters. Don't forget to save your new account file with a relevant name!

## Creating Investment Accounts Manually

If you want to set up your own investment accounts hierarchy, you may of course do so. Investments usually have a number of associated accounts that need to be created: an asset account to track the investment itself; an income account to track dividend transactions; and expense accounts to track investment fees and commissions.

In a typical account structure, security accounts are sub accounts of an asset account representing an account at a brokerage firm. The brokerage account would be denominated in your local currency and it would include sub accounts for each security that you trade there.

Related purchases, sales, income and expense accounts should also be in the same currency as the brokerage account.

The security sub accounts would each be configured to contain units of a single security selected from the master (user defined) security list and they are expected to use the same currency as the brokerage account.

Security prices are kept in a separate area of GnuCash (the Price Database - Tools > Price Editor). This contains prices for individual securities (not security accounts). All prices for an individual security are in a single currency. If a security is traded in multiple currencies, then a separate security and separate accounts should be set up for each currency.

### Custom Accounts Example

The following is a somewhat more complicated example of setting up GnuCash to track your investments, which has the advantage that it groups each different investment under the brokerage that deals with the investments. This way it is easier to compare the statements you get from your brokerage with the accounts you have in GnuCash and spot where GnuCash differs from the statement. Assets Investments Brokerage Accounts I\*Trade Stocks ACME Corp Money Market Funds I\*Trade Municipal Fund Cash My Stockbroker Money Market Funds Active Assets Fund Government Securities Treas Bond xxx Treas Note yyy Mutual Funds Fund A Fund B Cash Income Investments Brokerage Accounts Capital Gains I\*Trade My Stockbroker Dividends I\*Trade Taxable Non-taxable My Stockbroker Taxable Non-taxable Interest Income I\*Trade Taxable Non-taxable My Stockbroker Taxable Non-taxable Expenses Investment Expenses Commissions I\*Trade My Stockbroker Management Fees I\*Trade My Stockbroker

---

**Tip:** There really is no standard way to set up your investment account hierarchy. Play around, try different layouts until you find something which divides your investment accounts into logical groups which make sense to you.

---

### 2.10.3 Interest Bearing Accounts

Investments which have a fixed or variable rate of interest are one of the simplest and most common form of investments available. Interest bearing investments include your bank account, a certificate of deposit, or any other kind of investment in which you receive interest from the principal. This section will describe how to handle these kinds of investments in GnuCash.

#### Account Setup

When you purchase the interest bearing investment, you must create an asset account to record the purchase of the investment, an income account to record earnings from interest, and an expense account to record bank charges. Below is an account layout example, in which you have an interest bearing savings account and a certificate of deposit at your bank.

Assets
Bank ABC
CD
Savings
Expenses
Bank ABC
Charges
Income
Interest Income

(continues on next page)

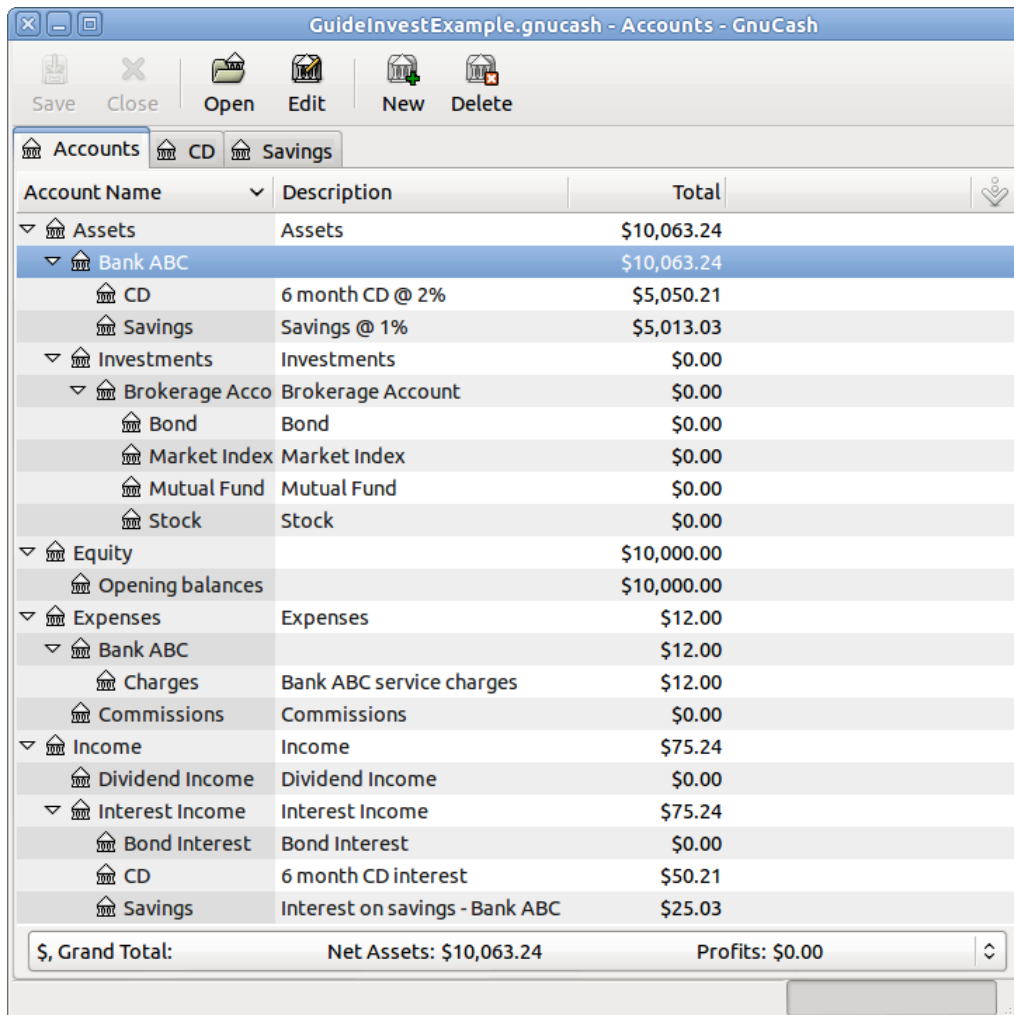
(continued from previous page)

CD  
Savings

As usual, this account hierarchy is simply presented as an example, you should create your accounts in a form which best matches your actual situation.

### Example

Now let's populate these accounts with real numbers. Let's assume that you start with \$10000 in your bank account, which pays 1% interest and you buy a \$5000 certificate of deposit with a 6 month maturity date and a 2% yield. Clearly, it is much better to keep your money in the CD than in the savings account. After the initial purchase, your accounts should look something like this:

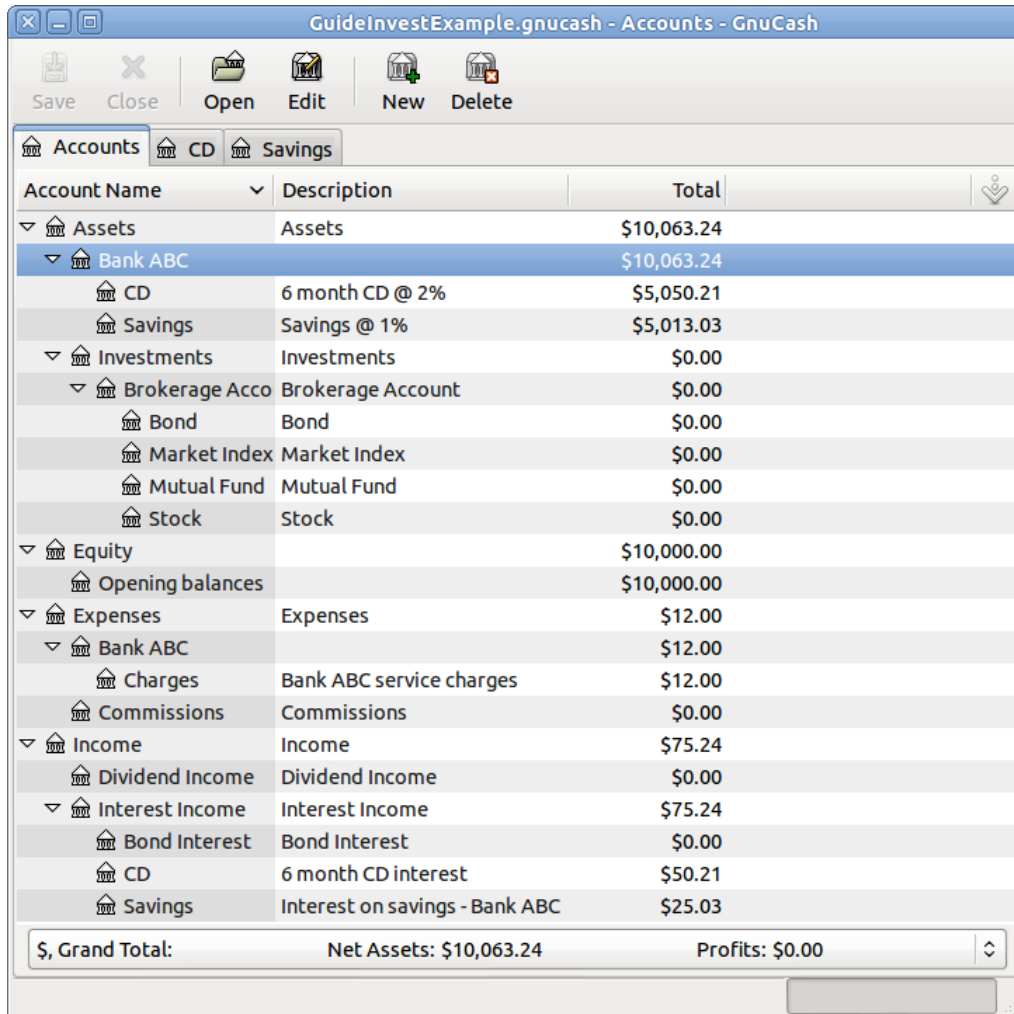


The screenshot shows the 'Accounts' window in GnuCash. The title bar is 'GuideInvestExample.gnucash - Accounts - GnuCash'. The window has a menu bar with 'Save', 'Close', 'Open', 'Edit', 'New', and 'Delete'. Below the menu bar are tabs for 'Accounts', 'CD', and 'Savings'. The main area displays a hierarchical tree of accounts with columns for 'Account Name', 'Description', and 'Total'. The 'Assets' section is expanded, showing 'Bank ABC' with a total of \$10,063.24. Under 'Bank ABC', there are 'CD' (6 month CD @ 2%, \$5,050.21) and 'Savings' (Savings @ 1%, \$5,013.03). Other sections include 'Investments' (total \$0.00), 'Equity' (total \$10,000.00), 'Expenses' (total \$12.00), and 'Income' (total \$75.24). The 'Income' section is further expanded, showing 'Dividend Income' (\$0.00), 'Interest Income' (\$75.24), 'Bond Interest' (\$0.00), 'CD' (6 month CD interest, \$50.21), and 'Savings' (Interest on savings - Bank ABC, \$25.03). At the bottom, a summary bar shows '\$, Grand Total: Net Assets: \$10,063.24 Profits: \$0.00'.

Account Name	Description	Total
Assets	Assets	\$10,063.24
Bank ABC		\$10,063.24
CD	6 month CD @ 2%	\$5,050.21
Savings	Savings @ 1%	\$5,013.03
Investments	Investments	\$0.00
Brokerage Acco	Brokerage Account	\$0.00
Bond	Bond	\$0.00
Market Index	Market Index	\$0.00
Mutual Fund	Mutual Fund	\$0.00
Stock	Stock	\$0.00
Equity		\$10,000.00
Opening balances		\$10,000.00
Expenses	Expenses	\$12.00
Bank ABC		\$12.00
Charges	Bank ABC service charges	\$12.00
Commissions	Commissions	\$0.00
Income	Income	\$75.24
Dividend Income	Dividend Income	\$0.00
Interest Income	Interest Income	\$75.24
Bond Interest	Bond Interest	\$0.00
CD	6 month CD interest	\$50.21
Savings	Interest on savings - Bank ABC	\$25.03
\$, Grand Total:		Net Assets: \$10,063.24 Profits: \$0.00

Now, during the course of the next 6 months, you receive monthly bank statements which describe the activity of your account. In our fictional example, we do nothing with the money at this bank, so the only activity is income from interest and bank charges. The monthly bank charges are \$2. After 6 months, the register window for the CD and for the savings account should look like these:

GuideInvestExample.gnucash - Accounts - GnuCash			
<div>  Save            Close            Open            Edit            New            Delete         </div>			
<div>  Accounts            CD            Savings         </div>			
Account Name	Description	Total	
Assets	Assets	\$10,063.24	
Bank ABC		\$10,063.24	
CD	6 month CD @ 2%	\$5,050.21	
Savings	Savings @ 1%	\$5,013.03	
Investments	Investments	\$0.00	
Brokerage Account	Brokerage Account	\$0.00	
Bond	Bond	\$0.00	
Market Index	Market Index	\$0.00	
Mutual Fund	Mutual Fund	\$0.00	
Stock	Stock	\$0.00	
Equity		\$10,000.00	
Opening balances		\$10,000.00	
Expenses	Expenses	\$12.00	
Bank ABC		\$12.00	
Charges	Bank ABC service charges	\$12.00	
Commissions	Commissions	\$0.00	
Income	Income	\$75.24	
Dividend Income	Dividend Income	\$0.00	
Interest Income	Interest Income	\$75.24	
Bond Interest	Bond Interest	\$0.00	
CD	6 month CD interest	\$50.21	
Savings	Interest on savings - Bank ABC	\$25.03	
\$, Grand Total:		Net Assets: \$10,063.24	Profits: \$0.00



The screenshot shows the 'Accounts' window in GnuCash. The title bar reads 'GuideInvestExample.gnucash - Accounts - GnuCash'. The window has a menu bar with 'Save', 'Close', 'Open', 'Edit', 'New', and 'Delete'. Below the menu bar are tabs for 'Accounts', 'CD', and 'Savings'. The main area displays a tree of accounts with columns for 'Account Name', 'Description', and 'Total'. The tree is expanded to show the 'Assets' category, which includes 'Bank ABC' (total \$10,063.24), 'CD' (\$5,050.21), 'Savings' (\$5,013.03), 'Investments' (\$0.00), 'Brokerage Account' (\$0.00), 'Bond' (\$0.00), 'Market Index' (\$0.00), 'Mutual Fund' (\$0.00), and 'Stock' (\$0.00). The 'Equity' category shows 'Opening balances' (\$10,000.00). The 'Expenses' category shows 'Bank ABC' (\$12.00), 'Charges' (\$12.00), and 'Commissions' (\$0.00). The 'Income' category shows 'Dividend Income' (\$0.00), 'Interest Income' (\$75.24), 'Bond Interest' (\$0.00), 'CD' (\$50.21), and 'Savings' (\$25.03). At the bottom, a summary bar shows '\$, Grand Total: Net Assets: \$10,063.24 Profits: \$0.00'.

Account Name	Description	Total
Assets	Assets	\$10,063.24
Bank ABC		\$10,063.24
CD	6 month CD @ 2%	\$5,050.21
Savings	Savings @ 1%	\$5,013.03
Investments	Investments	\$0.00
Brokerage Account	Brokerage Account	\$0.00
Bond	Bond	\$0.00
Market Index	Market Index	\$0.00
Mutual Fund	Mutual Fund	\$0.00
Stock	Stock	\$0.00
Equity		\$10,000.00
Opening balances		\$10,000.00
Expenses	Expenses	\$12.00
Bank ABC		\$12.00
Charges	Bank ABC service charges	\$12.00
Commissions	Commissions	\$0.00
Income	Income	\$75.24
Dividend Income	Dividend Income	\$0.00
Interest Income	Interest Income	\$75.24
Bond Interest	Bond Interest	\$0.00
CD	6 month CD interest	\$50.21
Savings	Interest on savings - Bank ABC	\$25.03
\$, Grand Total:		Net Assets: \$10,063.24 Profits: \$0.00

And this is the main GnuCash account window:

The screenshot shows the 'Accounts' window in GnuCash. The window title is 'GuideInvestExample.gnucash - Accounts - GnuCash'. The interface includes a menu bar with 'Save', 'Close', 'Open', 'Edit', 'New', and 'Delete'. Below the menu bar are tabs for 'Accounts', 'CD', and 'Savings'. The main area displays a tree view of accounts with columns for 'Account Name', 'Description', and 'Total'. The tree is expanded to show the following structure:

Account Name	Description	Total
Assets	Assets	\$10,063.24
Bank ABC		\$10,063.24
CD	6 month CD @ 2%	\$5,050.21
Savings	Savings @ 1%	\$5,013.03
Investments	Investments	\$0.00
Brokerage Acco	Brokerage Account	\$0.00
Bond	Bond	\$0.00
Market Index	Market Index	\$0.00
Mutual Fund	Mutual Fund	\$0.00
Stock	Stock	\$0.00
Equity		\$10,000.00
Opening balances		\$10,000.00
Expenses	Expenses	\$12.00
Bank ABC		\$12.00
Charges	Bank ABC service charges	\$12.00
Commissions	Commissions	\$0.00
Income	Income	\$75.24
Dividend Income	Dividend Income	\$0.00
Interest Income	Interest Income	\$75.24
Bond Interest	Bond Interest	\$0.00
CD	6 month CD interest	\$50.21
Savings	Interest on savings - Bank ABC	\$25.03
<b>\$, Grand Total:</b>		<b>Net Assets: \$10,063.24      Profits: \$0.00</b>

From the above image of the main GnuCash account window you see a nice summary of what happened to these investments over the 6 months. While the yield on the CD is double that of the savings account, the return on the CD was \$50.21 versus \$13.03 for the savings account, or almost 4 times more. Why? Because of the pesky \$2 bank charges that hit the savings account (which counted for \$12 over 6 months).

After this 6 month period, the CD has reached maturity which means you may sell it with no early withdrawal penalty. To do so, simply transfer the \$5050.21 from the CD account into the savings account.

## 2.10.4 Setup Investment Portfolio

Now that you have built an account hierarchy in the previous section, this section will show you how to create and populate the accounts with your investment portfolio. After this initial setup of your portfolio, you may have shares of stock purchased from before you started using GnuCash. For these stocks, follow the instructions in the [Entering Preexisting Shares](#) section below. If you have just purchased your stocks, then use the [Buying New Shares](#) section.

### Setup Accounts for Stocks and Mutual Funds

This section will show you how to add stocks and mutual fund accounts to GnuCash. In this section, we will assume you are using the basic account setup introduced in the previous section, but the principles can be applied to any account hierarchy.

You should have within the top level Asset account, a few levels down, an account entitled Stock. Open the account tree to this level by clicking on the “right facing triangle marker” signs next to the account names until the tree is opened to the depth of the new account. You will need to create a sub-account (of type *stock*) under the Stock account for every stock you own. Every stock is a separate account. The naming of these stock accounts is usually done using the stock ticker abbreviation, though account names may be anything that is clear to you and other users. So, for example, you could name your accounts *AMZN*, *IBM* and *NST* for your Amazon, IBM and NSTAR stocks respectively. Below is a schematic model of the layout (only showing the Assets sub-accounts).

```
Assets
  Investments
    Brokerage Accounts
      Bond
      Mutual Funds
      Market Index
      Stock
        AMZN
        IBM
        NST
```

**Note:** If you want to track income (dividends/interest/capital gains) on a per-stock or fund basis, you will need to create an *Income:Dividends:STOCKSYMBOL*, *Income:Cap Gain (Long):STOCKSYMBOL*, *Income:Cap Gain (Short):STOCKSYMBOL* and *Income:Interest:STOCKSYMBOL* account for each stock you own that pays dividends or interest.

### Example Stock Account

As an example, let's assume that you currently own 100 shares of Amazon stock. First, create the stock account *AMZN* by selecting the Stock account and click on the menu Actions > New Account. ... The New Account dialog will appear, follow the steps, in the sequence below to setup your new stock account.

**General** Opening Balance

**Identification**

Account name: AMZN

Account code:

Description: Amazon.com

Security/currency: Select...

Smallest fraction: Use Commodity Value

Account Color: Default

Notes:

☐ Tax related ☐ Hidden

☐ Placeholder

**Account Type** **Parent Account**

Stock

Mutual Fund

A/Receivable

Stock

Equity

Expenses

Help Cancel OK

1. *Account Name* - Usually, use the stock ticker abbreviation, IE: "AMZN"
2. *Account Code* - Optional field, use CUSIP, the newspaper listing symbol, mutual fund family ID or code of your own choosing.
3. *Description* - Optional field for detailed description of the commodity/stock. Note this field by default is displayed in the Account tab tree.
4. *Account Type* - Select the type of account you are creating from the lower left-hand list.
5. *Parent Account* - Select the parent account for the new account from the right hand listing. Expand list of accounts if necessary.
6. *Create the New Security* - To use a new stock, you must create the stock as a new commodity

---

**Note:** Be sure to first select Account Type *Stock* or *Mutual Fund* so that the Select... button brings up the list of securities rather than currencies.

---

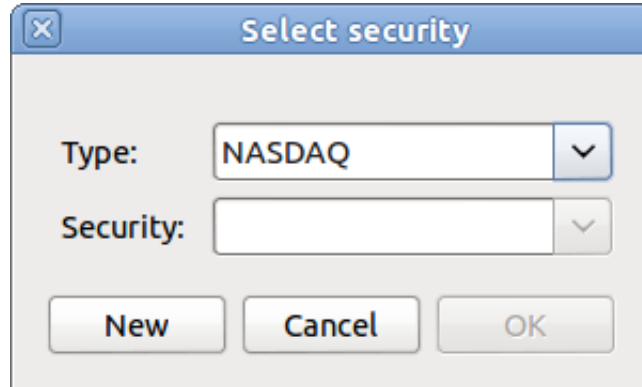
- *Select Security/Currency* - Click on the Select ... button next to the security/currency line. We must



change the security from the default (your default currency) to this specific stock. This will bring up the Select Security dialog.

- *Type* - Select the exchange where the security/commodity is traded (in this example NASDAQ).

Select the New button to open the New Security window.



- *Create the Security* - Click on the New... button and enter the appropriate information for this stock on the new form New Security.
  - The Full name: is “Amazon.com”.
  - The Symbol/abbreviation: is “AMZN”. The symbol is the stock ticker used in your quote source several lines down on the form. Note that different symbols will be utilized on different price sources for the same stock, an example is Ericsson on the Stockholm Exchange is ERIC-B while on Yahoo it is ERRICB.ST
  - The Type: should already be NASDAQ, because this is what was selected in the security selector, but you can change it here, including adding more categories. More information about this can be found in the Help Manual in section 7.7, “Security Editor”.
  - The ISIN, CUSIP or other code is where you can enter some other coding number or text (leave it blank in this example).
  - The Fraction traded should be adjusted to the smallest fraction of this security which can be traded, usually 1/100 or 1/10000.
  - The checkbox “Get Online Quotes”, the quote source and the timezone should be selected to define the sources for updating prices on-line. See Also “*Setting Stock Price Automatically*”.

---

**Note:** If the Get Online Quotes button is not highlighted, and it is not tickable, then the Finance::Quote package is not installed. See the section on “*Installing Finance::Quote*”.

---

Below is what this window should look like when finished: New Security Window New Security Window

- *Save Security* - Click on the OK button to save this new security, this will close the New Security window and return to the New Account window.
7. *Select the Security* - you should now see the newly created security available in the pull down menu for Security/Currency. Select it (it is probably already selected) and click on OK.
  8. *Smallest Fraction* - Specify the smallest fraction of the security/commodity that is traded.
  9. *Notes* - Enter any notes or messages related to this security/commodity.

10. *Tax Related* - Go to Edit > Tax Report Options to check this box if this account's transactions will relate to Income Taxes.
11. *Placeholder* - Check box if this account is a "Placeholder", that is it will contain no transactions.
12. *Finished* - You should now have been automatically returned to the New Account dialog, with the symbol/abbreviation: line set to "AMZN (Amazon.com)". Click on OK to save this new stock account.

You have now created the Amazon stock account, your main account should look something like this (notice that there are a few extra accounts here, a bank account, and an equity account):

Date	Action	Memo	Account	R	Shares	Price	Buy	Sell	Balance
01/03/2005		Initial IBM Purchase			50		50		50
	Buy	50 shares @\$100 = \$5000 + Commission \$100 = \$5100	Assets:Brokerage Account:Stock:IBM	n	50	102	5,100		
			Assets:Bank ABC	n				5,100.00	
							1		
16/03/2017					0.00				

Shares: 50 IBM Current Value: \$5,100.00  
 50 shares @\$100 = \$5000 + Commission \$100 = \$5100

Open the account register window for this AMZN stock account (double click on it). Here you see the **Commodity** view. This gives you an overview of the transactions in this commodity including the number of units (shares for a stock or mutual fund) bought or sold, the net price per unit, and the total amount. Obviously, we have not bought or sold any shares of AMZN yet, so the register should not contain any transactions.

## 2.10.5 Buying Shares

### Entering Preexisting Shares

The examples in this section use Transaction Journal view.

To register the initial 100 shares of this stock that you purchased previously, on the first (transaction) line, enter the date of the purchase (eg: Jan. 1 2005) and Description (eg: Opening Balance). On the first split line, enter 100 in Shares, delete the (unit) Price (it will be calculated when you Tab out of the split) and enter 2000 in the Buy column.

---

**Note:** It is also possible to use GnuCash to calculate Shares or Buy from the other 2 columns but to avoid rounding errors, it is better to automatically calculate Price.

---

Tab to the second split line, enter transfer from account *Equity:Opening Balances*. For simplicity, this example assumed there were no commissions on this transaction. Your AMZN Commodity view should now appear like this:

	Action	Memo	Account	R	Shares	Price	Buy	Sell	Balance
01/03/2005		Initial IBM Purchase			50		50		50
	Buy	50 shares @\$100 = \$5000 + Commission \$100 = \$5100	Assets:Brokerage Account:Stock:IBM	n	50	102	5,100		
			Assets:Bank ABC	n				5,100.00	
							1		
16/03/2017					0.00				

Shares: 50 IBM Current Value: \$5,100.00  
 50 shares @\$100 = \$5000 + Commission \$100 = \$5100

Notice that the Balance (last column) is in the units of the commodity (AMZN shares) not in currency units. Thus, the balance is 100 (AMZN units) rather than \$2000. This is how it should be.

## Buying New Shares

The main difference between setting up a new stock purchase versus the setup for preexisting stocks as described in the previous section is that instead of transferring the money used to purchase the stock from the *Equity:Opening Balances* account, you transfer from your *Assets:Bank ABC* or *Assets:Brokerage Account* account.

## Handling Commissions and Fees

For conciseness, this document will refer to the money you pay to a broker for buying and selling securities as *Commissions*. Government fees may also be payable. Unless otherwise stated, fees are handled in a similar way.

In GnuCash there are 2 alternate ways to handle commissions (for simplicity this document section will refer to these ways as net pricing and gross pricing):

- *Net Pricing*: You enter a net price (adjusted by commissions) when buying and selling securities. You do *not* also record commissions in a specific commissions account in order to later claim it as an expense, as this would be claiming commissions twice. This way *is* compatible with using *Selling Shares with Automatic Calculation of Capital Gain or Loss Using Lots*. This results in a slightly misleading price being added to the price database (the effective price you paid) but is not usually of any concern.

OR

- *Gross Pricing*: You enter the price not adjusted by commissions and enter the commissions expense on a separate split in the transaction. This enables the tracking of commissions but is *not* compatible with using *Selling Shares with Automatic Calculation of Capital Gain or Loss Using Lots*. Scrubbing doesn't know to deduct commissions and fees from the gains, so capital gains or losses must be manually calculated (see *Selling Shares with Manual Calculation of Capital Gain or Loss*).

Please get professional advice if you are unsure which of these ways are applicable to your jurisdiction.

## Example: Buying Shares with Gross Pricing

In this example you will purchase \$5000 of IBM stock, with a commission of \$100. First step will be to create the stock account for IBM. The existing *Expenses:Commission* account will be used. If you wish to track commissions to the individual stock level an additional sub-account would be necessary. E.g. *Expenses:Commission:IBM*.

Now for the transaction, on the first (transaction) line, enter the Date of the purchase (eg: Jan. 3 2005) and Description (eg: Initial IBM Purchase). On the first split line, enter 50 in Shares, delete Price (leave it empty so it will be calculated), and enter 5000 in Buy. You do not need to fill in the Price column, as it will be calculated for you when you Tab to the next split. The next line in the split transaction will be *Expenses:Commissions* and fill in Buy (\$100). The third line will be to transfer \$5100 from account *Assets:Bank ABC* to balance the transaction. Your IBM Commodity view should now appear like this:

Actio	Memo	Account	R	Shares	Price	Buy	Sell
01/03/2005	Initial IBM Purchase			50			50
	Buy 50 shares @ \$100 = \$5000 + Commission \$100 = \$5100	Assets:Brokerage Account:Stock:IBM	n	50	102	5,100	
		Assets:Bank ABC	n				5,100.00
16/03/2017				0.00			

Shares: 50 IBM Current Value: \$5,100.00  
50 shares @ \$100 = \$5000 + Commission \$100 = \$5100

### Example: Buying Shares with Net Pricing

Repeating the previous example using Net Pricing instead of Gross Pricing, in Transaction Journal view:

Purchase \$5000 of IBM stock being 50 Shares for \$100.00 each, plus a commission of \$100.

Now for the transaction, on the first (transaction) line, enter the Date of the purchase (eg: Jan. 3 2005) and a Description (eg: Initial IBM Purchase). On the first split line, optionally enter more details in Memo, then 50 in Shares, delete anything in Price (so it will be calculated by dividing Buy by Shares when you Tab out of the split), 5100 in Buy ( $50 * \$100.00 + \$100$ ). Alternatively use GnuCash to calculate Buy by entering the formula  $5000+100$  or  $(50 * 100) + 100$  in Buy (Buy will be calculated when you Tab

out of the column. Use the Tab key as many times as needed to proceed to the next split.

Do *not* enter a separate split for Commission as it has already been included in the Buy value. The second split line will be to transfer from *Assets:Bank ABC* \$5100 to balance the transaction. After the splits are all correct, use the Enter

key to save the transaction. Your IBM Commodity view should now appear like this:

Date	Action	Memo	Account	R	Shares	Price	Buy	Sell	Balance
01/03/2005	Initial IBM Purchase				50		50		50
	Buy	50 shares @\$100 = \$5000 + Commission \$100 = \$5100	Assets:Brokerage Account:Stock:IBM	n	50	102	5,100		
			Assets:Bank ABC	n				5,100.00	
							1		
16/03/2017					0.00				

Shares: 50 IBM Current Value: \$5,100.00  
 50 shares @\$100 = \$5000 + Commission \$100 = \$5100

## 2.10.6 Setting Share Price

The value of a commodity, such as a stock, must be explicitly set. The stock accounts track the quantity of stocks you own, but the value of the stock is stored in the *Price Editor*. The values set in the Price Editor can be updated manually or automatically.

### Initial Price Editor Setup

To use the Price Editor to track a stock value, you must initially insert the stock. To do so, open the Price Editor (Tools > Price Editor) and click on Add button. The first time a Commodity/Stock is entered this window will be blank except for the control buttons on the bottom. Select the appropriate Commodity you want to insert into the Price Editor. At this point, you can input the price of the commodity manually. There are 6 fields in the New Commodity window:

**Namespace** The exchange market where the security/commodity is traded (in this example NASDAQ)

**Security** The name of the commodity, must be chosen from the Select... list

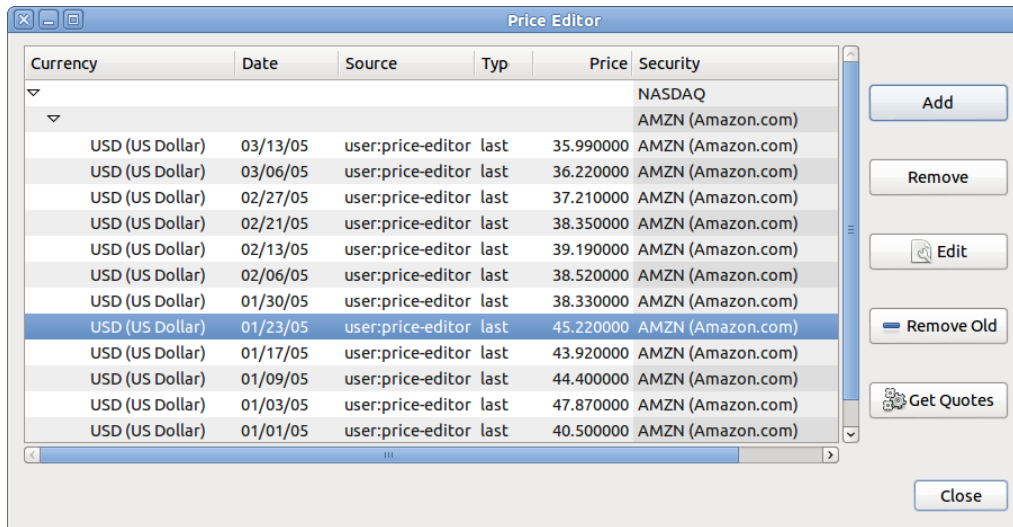
**Currency** The currency in which the Price is expressed.

**Date** Date that the price is valid

**Type** One of: Bid (the market buying price), Ask (the market selling price), Last (the last transaction price), Net Asset Value (mutual fund price per share), or Unknown. Stocks and currencies will usually give their quotes as one of bid, ask or last. Mutual funds are often given as net asset value. For other commodities, simply choose Unknown. This option is for informational purposes only, it is not used by GnuCash.

**Price** The price of one unit of this commodity.

As an example of adding the AMZN commodity to the price editor, with an initial value of \$40.50 per share.

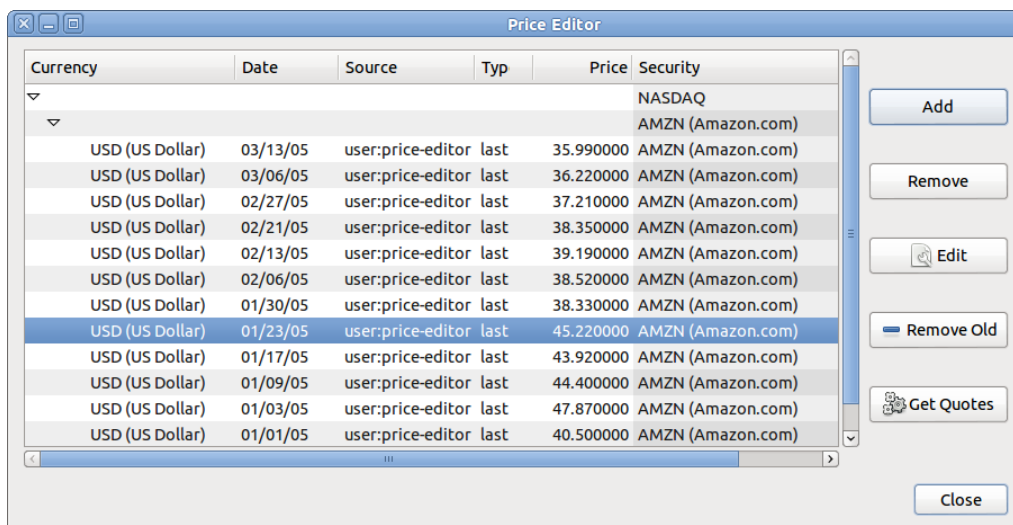


Click OK when finished. Once you have performed this initial placement of the commodity into the Price Editor, you will not have to do it again, even if you use the same commodity in another account.

**Note:** If you have online retrieval of quotes activated (see [Configuring for Automatic Retrieval of Quotes](#)), you can initialize a commodity without manually making an entry. When you initially add the security in the Security Editor, check Get Online Quotes and save the security. Then, in the Price Editor, click Get Quotes, and the new security will be inserted into the price list with the retrieved price.

## Setting Stock Price Manually

If the value of the commodity (stock) changes, you can adjust the value by entering the Price Editor, selecting the commodity, clicking on Edit and entering the new price.



## Configuring for Automatic Retrieval of Quotes

If you have more than a couple of commodities, you will tire of having to update their prices constantly. GnuCash has the ability to automatically download the most recent price for your commodities using the Internet. This is

accomplished through the Perl module `Finance::Quote`, which must be installed in order to activate this feature.

To determine if the Perl module `Finance::Quote` is already installed on your system, type `perldoc Finance::Quote` in a terminal window and check to see if there is any documentation available. If you see the documentation, then the module is installed, if you do not see the documentation, then it has not been installed.

## Installing Finance::Quote

### Microsoft Windows:

- Close GnuCash.
- Run **Install Online Price Retrieval** which can be found in the GnuCash “Start” menu entry.

**MacOS:** You need to have XCode installed. XCode is an optional item from your MacOS distribution DVD. Run the **Update Finance Quote** app in the GnuCash dmg. You can run it from the dmg or copy it to the same folder to which you copied GnuCash. It will open a Terminal window and run a script for you which will ask lots of questions. Accept the default for each unless you know what you’re doing.

### Linux:

- Close any running GnuCash instances.
- Locate the folder where GnuCash is installed by searching for `gnc-fq-update`
- Change to that directory, open a root shell
- Run the command `gnc-fq-update`

This will launch a Perl CPAN update session that will go out onto the internet and install the `Finance::Quote` module on your system. The `gnc-fq-update` program is interactive, however, with most systems you should be able to answer “no” to the first question: “Are you ready for manual configuration? [yes]” and the update will continue automatically from that point.

After installation is complete, you should run the “`gnc-fq-dump`” test program, in the same directory, distributed with GnuCash to test if `Finance::Quote` is installed and working properly.

---

**Note:** If you feel uncomfortable about performing any of these steps, please either email the GnuCash-user mailing list ([gncash-user@gncash.org](mailto:gncash-user@gncash.org)) for help or come to the GnuCash IRC channel on [irc.gnome.org](http://irc.gnome.org). You can also leave out this step and manually update your stock prices.

---

## Configuring Securities for Online Quotes

With `Finance::Quote` installed and functioning correctly, you must configure your GnuCash securities to use this feature to obtain updated price information automatically. Whether creating new securities or modifying securities that have already been setup, use the Tools > Security Editor, to edit the security and check the Get Online Quotes box. You will now be able to modify the radio buttons for Type of quote source, the pull-down menus to specify the specific source(s) and The timezone for these quotes. When finished editing, Close the Security Editor to return to the Price Editor and click on the Get Quotes button to update your stock prices on the Internet.

The command `gnucash --add-price-quotes $HOME/gnucash-filename` can be used to fetch the current prices of your stocks. The file specified `$HOME/gnucash-filename` will depend on the name and location of your data file. This can be determined by the name displayed in the top frame of the GnuCash window, before the “-”. The file name can also be found under File in the recently opened file list; the first item, numbered 1, is the name of the currently open file.

This can be automated by creating a crontab entry. For example, to update your file every Friday evening (16:00) after the relevant exchange markets close (modify the time accordingly for your time zone), you could add the following to your personal crontab:

```
0 16 * * 5 gncash --add-price-quotes $HOME/gnucash-filename > /dev/null 2>&1
```

Remember that Mutual Fund “prices” are really “Net Asset Value” and require several hours after the exchange closes before being available. If NAVs are downloaded before the current days NAVs are determined, yesterday’s NAVs are retrieved.

## Displaying Share Value

The main account window, by default, only shows the quantity of each commodity that you own, under the column heading Total. In the case of stocks, this commodity is the number of shares. Often, however, you will want to see the value of your stocks expressed in terms of some monetary unit. This is easily accomplished by entering the main window, selecting the Accounts tab, by clicking on the *Titlebar* Options button (the small down pointing arrow on the right side of the main account window titles bar), and selecting the option to display the account total field “Total (USD)”. You will see a new column in the main window entitled Total (USD) that will express the value of all commodities in the report currency.

Account Name	Description	Total	Total (USD)
Bond	Bond	\$0.00	\$0.00
Market Index	Market Index	\$0.00	\$0.00
Mutual Fund	Mutual Fund	\$0.00	\$0.00
Stock	Stock	\$12,543.09	\$12,543.09
AMZN	Amazon.com	100 AMZN	\$3,599.00
IBM	IBM stocks	51.7598 IBM	\$8,944.09
Equity		\$12,000.00	\$12,000.00
Opening balances		\$12,000.00	\$12,000.00
Expenses	Expenses	\$112.00	\$112.00
Bank ABC		\$12.00	\$12.00
Charges	Bank ABC service charges	\$12.00	\$12.00
Commissions	Commissions	\$100.00	\$100.00
Income	Income	\$75.24	\$75.24
Dividend Income	Dividend Income	\$0.00	\$0.00
Interest Income	Interest Income	\$75.24	\$75.24
Bond Interest	Bond Interest	\$0.00	\$0.00
\$, Grand Total:		Net Assets: \$17,506.33	Profits: \$0.00

## The “Price Source” in Reports

Most GnuCash reports have options to set/modify a number of parameters for the report. The Options dialog box is displayed by selecting the report tab then clicking on either the Options icon in the *Menubar* or selecting Edit > Report Options. Price Source determines how accounts denominated in commodities different from the report



currency are converted to the report currency. Depending on the report the selector may appear in either the General, the Commodities, or the Display tab of the Report Options dialog box.

---

**Note:** In the example below, the report is a customization of the **Average Balance** report in the Assets & Liabilities reports submenu.

---

Determining Stock Price/Currency Exchange Rate Source in Reports Determining the value of a stock commodity or a currency other than the report currency in a report by setting the Price Source option.

- **Weighted Average:** Calculates the price by summing the absolute value of the amount and the absolute value of every split in every account denominated in the commodity, excluding those splits with a zero amount, and dividing the sum of values by the sum of amounts to obtain a price. For example, if you had a buy transaction for 200 shares of XYZ for a total of 2000 and a sell of 100 for 1300 the weighted average would be  $3300/300$  or 11/share.

---

**Note:** Gain/Loss splits have an amount of 0 and are *not* included in this calculation.

---

- **Average Cost:** Calculates the price by summing the amounts and values of every split in every account denominated in the commodity, including the zero amount splits. In the example above, with an additional split (either part of the sale transaction or in a separate transaction) booking the gain at 0 shares and a 300 gain, the average cost is  $1000/100$  ( $2000$  original cost -  $1300$  proceeds from the sale +  $300$  gain) / ( $200 - 100$ ) shares or 10/share.

---

**Note:** Gain/Loss splits *are* included in this calculation.

---

---

**Note:** This is the *only* Price Source that will balance the Trial Balance Report and in order for it to balance you *must* correctly book your gains and losses.

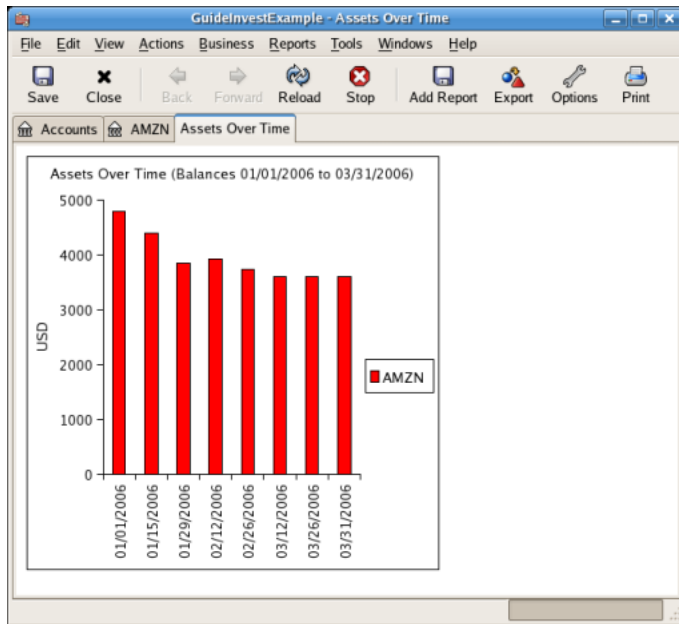
---

- **Most Recent:** Uses the latest price from the price database.
- **Nearest in time:** Uses the price nearest in time to the report date—the datum date for time series reports like Assets Over Time—from the price database.

---

**Note:** The nearest date isn't necessarily before the date in question.

---



### 2.10.7 Selling Shares

Entering an investment sale transaction is done in a similar way to entering a buy transaction (see [Buying New Shares](#)) except the amount entered in the *Shares* column is negative and the proceeds of the transaction is entered in the *Sell* column. The net proceeds from the sale should be transferred from the shares account to your bank or brokerage account.

For information on handling *commissions* and the use of *Net Pricing* or *Gross Pricing*, please see [Handling Commissions and Fees](#).

If you will be recording a capital gain or loss on the sale, please see [Capital Gains](#) and [Depreciation](#) for more information on this topic.

To use the GnuCashAutomatic *Calculation of Capital Gain or Loss Using Lots* feature, please see [Selling Shares with Automatic Calculation of Capital Gain or Loss Using Lots](#) otherwise continue to the next section.

### Selling Shares with Manual Calculation of Capital Gain or Loss

**Note:** In order for GnuCash to commit the zero-share, zero-price split for account *Assets:Stock:SYMBOL* to the transaction in the following schemes, you *\*must\** Tab out of that split. If you use the Enter key, GnuCash will convert the split into shares of the commodity.

In the schemes of transaction splits presented below, the following symbols are used:

- NUM\_SHARES: number of shares you are selling
- COMMISSION: brokerage commissions or fees
- GROSS\_SELL\_PRICE: unit price for which you sold the shares, not reduced by COMMISSION
- NET\_SELL\_PRICE: unit price for which you sold the shares, reduced by COMMISSION
- GROSS\_BUY\_PRICE: unit price for which you bought the shares, not increased by COMMISSION
- NET\_BUY\_PRICE: unit price for which you bought the shares, increased by COMMISSION

- GROSS\_BUY: total price for which you bought shares, excluding COMMISSION, equal to NUM\_SHARES \* GROSS\_BUY\_PRICE
- NET\_BUY: amount of money for which you bought shares including COMMISSION, equal to GROSS\_BUY + COMMISSION.
- GROSS\_SALE: total price for which you sold shares, equal to NUM\_SHARES \* GROSS\_SELL\_PRICE
- NET\_SALE: amount of money received from the sale, equal to GROSS\_SALE - COMMISSION
- GROSS\_PROFIT: amount of money you made on the sale, not reduced by COMMISSION
- NET\_PROFIT: amount of money you made on the sale, reduced by COMMISSION

There are 2 ways of manually recording the capital gain or loss. The capital gain/loss can be combined with the sale in one transaction or it can be entered in a separate transaction.

### Combine the Sale and Capital Gain/Loss in One Transaction

This has the advantage that all parts of the sale event are kept together. This is *not* compatible with using *scrubbing* (see *Selling Shares with Automatic Calculation of Capital Gain or Loss Using Lots*). If you may in future use scrubbing on a specific security, save some work later by entering the capital gain/loss splits in a separate transaction now.

When the capital gain/loss splits are combined with the sale splits in one transaction, there are 2 splits for the security account in the same transaction, so the transaction must be entered with the security register in Auto-Split Ledger or Transaction Journal view. One of the splits for the security account is for the sale and the other is for the capital gain or loss. The security account split for the capital gain or loss must be entered with 0 number of shares and 0 price per share to stop the automatic recalculation of these fields.

Account for the profit or loss as coming from an *Income:Capital Gains* or *Expenses:Capital Loss* account.

### Combined, Gross Pricing

Combined, Gross Pricing

<i>Account</i>	<i>Number of Shares</i>	<i>Share Price</i>	<i>Total Buy</i>	<i>Total Sell</i>
Assets:Bank ABC			NET_SALE	
Assets:Stock:SYMBOL	0	0	GROSS_PROFIT	(Loss)
Expenses :Commis-sion			COMMISS-ION	
Assets:Stock:SYMBOL	NUM_SHARES	GROSS_SELL_PRICE		GROSS_SALE
Income:Capital Gains			(Loss)	GROSS_PROFIT

### Combined, Net Pricing

Combined, Net Pricing

<i>Account</i>	<i>Number of Shares</i>	<i>Share Price</i>	<i>Total Buy</i>	<i>Total Sell</i>
Assets:Bank ABC			NET_SALE	
Assets:Stock:SYMBOL	0	0	NET_PROFIT	(Loss)
Assets:Stock:SYMBOL	NUM_SHARES	NET_SELL_PRICE		NET_SALE
Income:Capital Gain			(Loss)	NET_PROFIT

### Separate the Capital Gain/Loss Transaction from the Sale Transaction.

This is *required* if using *scrubbing* to automatically calculate and create capital gain/loss transactions (otherwise scrubbing will not detect them and will create an incorrectly valued gain/loss transaction).

### Separated, Gross Pricing

Separate Transactions, Sale Transaction

<i>Account</i>	<i>Number of Shares</i>	<i>Share Price</i>	<i>Total Buy</i>	<i>Total Sell</i>
Assets:Bank ABC			NET_SALE	
Expenses :Commission			COMMISSION	
Assets:Stock:SYMBOL	NUM_SHARES	GROSS_SELL_PRICE		GROSS_SALE

Separate Transactions, Capital Gain/Loss Transaction

<i>Account</i>	<i>Number of Shares</i>	<i>Share Price</i>	<i>Total Buy</i>	<i>Total Sell</i>
Assets:Stock:SYMBOL	0	0	GROSS_PROFIT	(Loss)
Income:Capital Gain			(Loss)	GROSS_PROFIT

### Separated, Net Pricing

Separate Transactions, Sale Transaction

<i>Account</i>	<i>Number of Shares</i>	<i>Share Price</i>	<i>Total Buy</i>	<i>Total Sell</i>
Assets:Bank ABC			NET_SALE	
Assets:Stock:SYMBOL	NUM_SHARES	NET_SELL_PRICE		NET_SALE

Separate Transactions, Capital Gain/Loss Transaction

Account	Number of Shares	Share Price	Total Buy	Total Sell
Assets:Stock:SYMBOL	0	0	NET_PROFIT	(Loss)
Income:Capital Gain			(Loss)	NET_PROFIT

## Examples of Selling Shares with Manually Entry of Capital Gain or Loss

In these examples we will use the AMZN account created in the previous section.

### Example: Sale of Shares with Profit, Manual Profit/Loss Calculation, Sale & Profit Combined, Gross Pricing

In this example you bought 100 shares of AMZN for \$20 per share, then later sell them all for \$36 per share with a commission of \$75. In the split transaction scheme above, GROSS\_BUY\_PRICE is \$20 (the original buying price), NUM\_SHARES is 100, GROSS\_BUY is \$2000 (the original buying cost), GROSS\_SALE is \$3600, and finally GROSS\_PROFIT is \$1600 (GROSS\_SALE - GROSS\_BUY).

Pricing

Account	Shares	Price	•	Sell
			Buy*	
Assets:Bank ABC			352 5.00	
Assets:Brokerage Account:Stock:AMZN	0	0	160 0.00	
Expenses:Commission			7 5.00	
Assets:Brokerage Account:Stock:AMZN	100	36.00		360 0.00
Income:Capital Gain (Long Term):AMZN				160 0.00

Date	Action	Memo	Account	R	Shares	Price	Buy	Sell	Balance
01/01/2007	Buy	100 @ 19 ea = \$1900 + \$100 commission = \$2000	Assets:Brokerage Account:Stock:AMZN	n	100	20	2,000.00		100
	Buy		Assets:Bank ABC	n				2,000.00	
03/21/2008	Sell	stock for profit, Net Pricing, Separated, Sale Transaction	Assets:Bank ABC	n	-100			100	0
		Net Sale = Gross Sale 3600 - Commission 75 = 3525	Assets:Bank ABC	n			3,525.00		
03/21/2008	Sell	Net Sale = Sell 100 @ 36 Ea = 3600 - Commission 75 = 3525	Assets:Brokerage Account:Stock:AMZN	n	-100	35.25		3,525.00	
		Capital Gain Transaction	Assets:Brokerage Account:Stock:AMZN	n	0				0
		Net Profit = Net Sale 3525 - Net Buy 2000 = 1525	Assets:Brokerage Account:Stock:AMZN	n			1,525.00		
		Capital Gain (Long) = Net Profit	Income:Capital Gain (Long Term):AMZN	n				1,525.00	
01/25/2017	Memo			n	0.00				

Shares: 0 AMZN Current Value: \$0.00

Enter an action type, or choose one from the list

**Note:** In the above screenshot, it appears there are 2 transactions for Mar 21st 2006. This is because the register is in Auto-Split Ledger view and there are 2 splits for the register account in the 1 transaction. Transaction Journal

view may be clearer. Refer to *Simple vs. Split Transactions*. As there are 2 splits for the register account in the sale transaction, this transaction must be entered in Auto-Split Ledger or Transaction Journal view. It cannot be entered in Basic Ledger view.

Action	Memo	Account	R	Shares	Price	Buy	Sell
01/01/2007	Buy 100 @ 19 ea = \$1900 + \$100 commission = \$2000			100		100	100
	Buy	Assets:Brokerage Account:Stock:AMZN	n	100	20	2,000.00	
		Assets:Bank ABC	n				2,000.00
03/21/2008	Sell stock for profit, Net Pricing, Separated, Sale Transaction			-100			100
	Net Sale = Gross Sale 3600 - Commission 75 = 3525	Assets:Bank ABC	n			3,525.00	
	Sell Net Sale = Sell 100 @ 36 Ea = 3600 - Commission 75 = 3525	Assets:Brokerage Account:Stock:AMZN	n	-100	35.25		3,525.00
03/21/2008	Capital Gain Transaction			0			0
	Net Profit = Net Sale 3525 - Net Buy 2000 = 1525	Assets:Brokerage Account:Stock:AMZN	n			1,525.00	
	Capital Gain (Long) = Net Profit	Income:Capital Gain (Long Term):AMZN	n				1,525.00
	Memo	Account	R	Shares	1	Buy	Sell
01/25/2017				0.00			
					1		

Shares: 0 AMZN Current Value: \$0.00  
Enter an action type, or choose one from the list

### Example: Sale of Shares with Profit, Manual Profit/Loss Calculation, Sale & Profit Combined, Net Pricing

In this example you bought 100 shares of AMZN for \$20 per share (including commissions), then later sell them all for \$36 per share with a commission of \$75. In the split transaction scheme above, NUM\_SHARES is 100, NET\_BUY\_PRICE is \$20 (the original buying price), NET\_BUY is \$2000 (the original buying cost), NET\_SELL\_PRICE is \$35.25  $((\$3600 - \$75) / 100)$ , GROSS\_SALE is \$3600, NET\_SALE is \$3525, and finally NET\_PROFIT is \$1525 (NET\_SALE NET\_BUY).

#### Pricing

Account	Shares	Price	• Buy*	Sell
Assets:Bank ABC			352 5.00	
Assets:Brokerage Account:Stock:AMZN	0	0	152 5.00	
Assets:Brokerage Account:Stock:AMZN	100	35.25		352 5.00
Income:Capital Gain (Long Term):AMZN				152 5.00

Action	Memo	Account	R	Shares	Price	Buy	Sell	
01/01/2007	Buy 100 @ 19 ea = \$1900 + \$100 commission = \$2000			100		100		100
	Buy	Assets:Brokerage Account:Stock:AMZN	n	100	20	2,000.00		
		Assets:Bank ABC	n				2,000.00	
03/21/2008	Sell stock for profit, Net Pricing, Separated, Sale Transaction			-100			100	0
	Net Sale = Gross Sale 3600 - Commission 75 = 3525	Assets:Bank ABC	n			3,525.00		
	Sell Net Sale = Sell 100 @ 36 Ea = 3600 - Commission 75 = 3525	Assets:Brokerage Account:Stock:AMZN	n	-100	35.25		3,525.00	
03/21/2008	Capital Gain Transaction			0				0
	Net Profit = Net Sale 3525 - Net Buy 2000 = 1525	Assets:Brokerage Account:Stock:AMZN	n			1,525.00		
	Capital Gain (Long) = Net Profit	Income:Capital Gain (Long Term):AMZN	n				1,525.00	
	Memo							
01/25/2017				0.00				
						1		

Shares: 0 AMZN Current Value: \$0.00

Enter an action type, or choose one from the list

### Example: Sale of Shares with Profit, Manual Profit/Loss Calculation, Sale & Profit Separated, Gross Pricing

You bought 100 shares of AMZN for \$20 per share and commissions \$20, then later sell them all for \$36 per share with a commission of \$75. In the split transaction scheme above, GROSS\_BUY\_PRICE is \$20 (the original buying price), NUM\_SHARES is 100, GROSS\_BUY is \$2000 (the original buying cost), GROSS\_SALE is \$3600, and finally GROSS\_PROFIT is \$1600 (GROSS\_SALE GROSS\_BUY).

Table 11: Selling Shares Split Scheme, Sale Transaction, Gross Pricing

Account	Shares	Price	•	Sell
			Buy*	
Assets:Bank ABC			352 5.00	
Expenses:Commission			7 5.00	
Assets:Brokerage Account:Stock:AMZN	100	36.00		360 0.00

Table 12: Selling Shares Split Scheme, Gain Transaction, Gross Pricing

Account	Shares	Price	•	Sell
			Buy*	
Assets:Brokerage Account:Stock:AMZN	0	0	160 0.00	
Income:Capital Gain (Long Term):AMZN				160 0.00

Action	Memo	Account	R	Shares	Price	Buy	Sell	
01/01/2007	Buy 100 @ 19 ea = \$1900 + \$100 commission = \$2000			100		100		100
	Buy	Assets:Brokerage Account:Stock:AMZN	n	100	20	2,000.00		
		Assets:Bank ABC	n				2,000.00	
03/21/2008	Sell stock for profit, Net Pricing, Separated, Sale Transaction			-100			100	0
	Net Sale = Gross Sale 3600 - Commission 75 = 3525	Assets:Bank ABC	n			3,525.00		
	Sell Net Sale = Sell 100 @ 36 Ea = 3600 - Commission 75 = 3525	Assets:Brokerage Account:Stock:AMZN	n	-100	35.25		3,525.00	
03/21/2008	Capital Gain Transaction			0				0
	Net Profit = Net Sale 3525 - Net Buy 2000 = 1525	Assets:Brokerage Account:Stock:AMZN	n			1,525.00		
	Capital Gain (Long) = Net Profit	Income:Capital Gain (Long Term):AMZN	n				1,525.00	
	Memo							
01/25/2017				0.00				
			n		1			

Shares: 0 AMZN Current Value: \$0.00

Enter an action type, or choose one from the list

### Example: Sale of Shares with Profit, Manual Profit/Loss Calculation, Sale & Profit Separated, Net Pricing

You bought 100 shares of AMZN for \$20 per share (including commissions), then later sell them all for \$36 per share with a commission of \$75. In the split transaction scheme above, NUM\_SHARES is 100, NET\_BUY\_PRICE is \$20 (the original buying price), NET\_BUY is \$2000 (the original buying cost), NET\_SELL\_PRICE is \$35.25 ((\$3600 \$75) / 100)), GROSS\_SALE is \$3600, NET\_SALE is \$3525, and finally NET\_PROFIT is \$1525 (NET\_SALE NET\_BUY).

Table 13: Selling Shares Split Scheme, Sale Transaction, Net Pricing

Account	Shares	Price	• Buy*	Sell
Assets:Bank ABC			352 5.00	
Assets:Brokerage Account:Stock:AMZN	100	35.25		352 5.00

Table 14: Selling Shares Split Scheme, Gain Transaction, Net Pricing

Account	Shares	Price	• Buy*	Sell
Assets:Brokerage Account:Stock:AMZN	0	0	152 5.00	
Income:Capital Gain (Long Term):AMZN				152 5.00



Action	Memo	Account	R	Shares	Price	Buy	Sell	
01/01/2007	Buy 100 @ 19 ea = \$1900 + \$100 commission = \$2000			100		100		100
	Buy	Assets:Brokerage Account:Stock:AMZN	n	100	20	2,000.00		
		Assets:Bank ABC	n				2,000.00	
03/21/2008	Sell stock for profit, Net Pricing, Separated, Sale Transaction			-100			100	0
	Net Sale = Gross Sale 3600 - Commission 75 = 3525	Assets:Bank ABC	n			3,525.00		
	Sell Net Sale = Sell 100 @ 36 Ea = 3600 - Commission 75 = 3525	Assets:Brokerage Account:Stock:AMZN	n	-100	35.25		3,525.00	
03/21/2008	Capital Gain Transaction			0				0
	Net Profit = Net Sale 3525 - Net Buy 2000 = 1525	Assets:Brokerage Account:Stock:AMZN	n			1,525.00		
	Capital Gain (Long) = Net Profit	Income:Capital Gain (Long Term):AMZN	n				1,525.00	
	Memo	Account	R	Shares	1	Buy	Sell	
01/25/2017				0.00				
			n		1			

Shares: 0 AMZN Current Value: \$0.00

Enter an action type, or choose one from the list

## Selling Shares with Automatic Calculation of Capital Gain or Loss Using Lots

### Introduction

Wikipedia includes the following definition of a lot

“a set of goods for sale together in an auction; or a quantity of a financial instrument”.

GnuCash has a built-in lot management facility that can be used to keep track of capital gains or losses resulting from security sales. Buy and sell transactions are put into lots for the purpose of calculating the cost of the sale. More specifically, a lot is used to link particular buy and sell transaction splits. Lots can be automatically or manually created and linked. Capital gain or loss can be automatically calculated and transaction(s) created for the difference between the sale value and the cost of the securities sold. GnuCash refers to this process as *scrubbing*.

The term *scrub* is used because security accounts need to be cleaned after sales to ensure the difference between the cost paid for securities, and value received from selling them, is accounted for as capital gain or loss. If the capital gain/loss is not correct, the Trial Balance (Reports > Income & Expense > Trial Balance) bottom line total debits will not balance to total credits.

**Note:** If you make an error, you can delete the lot(s) and capital gain/loss transaction(s) and retry. Ensure you delete the lot, or at least unlink sale transactions from the lot, before you delete a capital gain/loss transaction. Otherwise, the Lots in Account screen will recreate the capital gain/loss transaction when you select the lot.

If you are not familiar with FIFO, LIFO or Average costing, please see Wikipedia [FIFO and LIFO accounting](#) and [Average cost method](#).

If you are not familiar with the difference between GnuCash transactions and splits, please see [Simple vs. Split Transactions](#).

The GnuCash lot management facility can be a useful feature, reducing manual calculation, especially if dividends have been reinvested over years and there are many different costs involved. It can automatically link buy transactions to sell transactions using FIFO cost method and one can manually link specific buy transactions to sell transactions in order to use LIFO. Advanced Portfolio Report basis costs and gains/losses will agree with the costs and gain/loss transactions created by scrubbing if either the FIFO or LIFO cost methods are used.

## Lots in Account Window

The Lots in Account SSSS window, where SSSS is a security account, is used to manually or automatically link security transaction splits to lots and create capital gain/loss transactions to account for the difference between the costs of buying a security and the value received by selling it.

To open the Lots in Account window, open the security account register, then select Actions > View Lots.

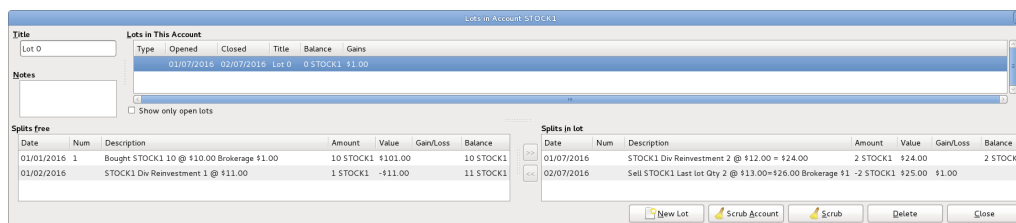


Fig. 12: Selling Shares - Capital Gains - Lots in Account window

Refer to the Help Manual, Chapter 8 Tools & Assistants, [Lots in Account](#) for details of the Lots in Account screen elements.

## Procedure Summary

Using the lot management facility for the automatic calculation of capital gain or loss typically follows these steps:

1. Record the sale transaction using Net Pricing (but stop short of entering the Capital Gain transaction as it will be created by scrubbing). See *Example: Sale of Shares with Profit, Manual Profit/Loss Calculation, Sale & Profit Separated, Net Pricing*.
2. *Manual Lot Creation and Linking* (Optional depending on cost method)
3. *Automatic Creation of Capital Gain Or Loss Transactions*
4. *Change Orphaned Gains-CCC to Gain/Loss Account*
5. *Run a Trial Balance* report to ensure total debits balance to total credits

## Manual Lot Creation and Linking

Before using this feature, ensure you have read *Considerations*.

This functionality allows the manual linking of specific buy and sell transactions. It may be used in the case where a user wishes to use a different cost method than the automatic linking method (FIFO). Effectively, if one wishes the cost basis and capital gains in the Advanced Portfolio Report to be consistent with the capital gains transactions created by scrubbing, manual lot creation only needs to be used when using LIFO or “sale of designated lots” (the same thing for securities as far as US personal tax law is concerned). This is because the scrub function can automatically do FIFO linking so there is no need to do it manually and scrubbing cannot be used for average costing.

See *Example 1: Manual Lot Creation and Linking*.

## Automatic Creation of Capital Gain Or Loss Transactions

---

**Note:** Do *NOT* do this unless you are using FIFO or LIFO to cost sales. See *Considerations*

---

GnuCash can automatically calculate and create security sale capital gain/loss transactions. Lots are used to link buy transaction splits with sell transaction splits so the correct cost of the securities sold can be determined. GnuCash will use any existing lots, and create new lots for any buy transaction splits not already linked to a lot. Buy and sell transaction splits are linked to lots using FIFO method.

See:

*Example 2: Automatic Creation of Capital Gain Or Loss Transactions*

*Example 3: Automatic Creation of Capital Gain Or Loss Transactions, 2 Sales at Once*

*Example 4: Automatic Creation of Capital Gain Or Loss Transactions - After a Simple Stock Split*

## Change Orphaned Gains-CCC to Gain/Loss Account

The capital gain/loss transaction(s) created by scrubbing uses an automatically created generic *Orphaned Gains-CCC* account (where CCC is the security currency) because GnuCash doesn't know which capital gain or loss account should be used. After scrubbing, the user should edit the *Orphaned Gains-CCC* transaction split to re-assign the income account to a more meaningful income (or expense) gain or loss account (e.g. *Income:Long Term Capital Gain:IBM*).

See *Example 5: Changing the Orphaned Gains-CCC to Gain/Loss Account*.

## Run a Trial Balance

Running a Trial Balance report (Reports > Income & Expense > Trial Balance) after creating capital gain/loss transactions, is a basic check that capital gains/losses are correctly accounting for the difference between the cost paid for securities, and value received from selling them. At the end of the report, total debits should equal total credits.

---

**Tip:** A Trial Balance may not balance due to some other problem. To determine if the cause of an imbalance is from incorrectly accounting for capital gain/loss:

If necessary, temporarily change the date of the sell transaction and the capital gain/loss transaction, so they are the only transactions for a particular date, then run the Trial Balance as at the day before. If the Trial Balance is still out by the same amount, it is not the capital gain/loss that is causing the problem.

If you find a prior *out of balance* Trial Balance, keep running the Trial Balance report with different dates until you find the date it starts being out of balance. Temporarily change the transaction dates for each transaction on the problem date to the following day, then change the dates back to the correct date 1 at a time, running the Trial Balance each time, until you identify the problem transaction. When you change the date of a security sell transaction, you also need to change the date of the corresponding capital gain transaction as it is only the sum of these that will balance in the Trial Balance.

---

## Considerations

There are some points that should be considered before using the lot management facility.

1. GnuCash implements only the First In/ First Out (FIFO) cost method when automatically linking buy transactions to sell transactions. I.e. The oldest securities are always sold first. The Last In First Out (LIFO) cost method may be used by manually linking the most recent buy security splits to the sell split before scrubbing.
2. The Advanced Portfolio Report does not use lot information when calculating costs, just the security transaction splits. It calculates the cost basis and gains or losses using the selected *Basis calculation method* report option

(Average, FIFO or LIFO). If one wishes the Advanced Portfolio Report costs and gains/losses to be consistent with the capital gain/loss transactions created by scrubbing, the same cost model must be used in both places.

3. Scrubbing does not recognize commissions or fees so makes no allowance for them in the calculation of gain or loss. Therefore you must use *Net Pricing* rather than *Gross Pricing* if you wish to use scrubbing. See [Handling Commissions and Fees](#).
4. Scrubbing does not recognize capital gain/loss transaction splits if they have been manually entered as part of the sale transaction. Therefore ensure previous sales are recorded as 2 transactions:

reduction of the number of shares

<i>Account</i>	<i>Tot Shares</i>	<i>(Unit) Price</i>	<i>Buy (Debit)</i>	<i>Sell (Credit)</i>
Brokerage or Bank			Debit	
Security	NumSold	SaleUnitPrice		SaleValue

Table 15: Transaction 2 capital gain/loss (loss in this example)

<i>Account</i>	<i>Tot Shares</i>	<i>(Unit) Price</i>	<i>Buy (Debit)</i>	<i>Sell (Credit)</i>
Capital Loss			Debit	
Security	0	0		Credit

5. The automatic capital gains calculations can handle straightforward buy, sell, and return of capital transactions but any transaction that affect the number of shares, even simple splits, will cause it to produce wrong answers so those cases must be handled manually.

### Example 1: Manual Lot Creation and Linking

Here is an example of selling part of a security holding using the LIFO method. In this example, the most recent buy transaction (dated 01/07/2016, a reinvested dividend), is linked to a lot, along with the sell transaction, and the GnuCash scrub function is used to calculate capital gain or loss and create the capital gain/loss transaction.

1. Open the security account's register.
2. Ensure all previous capital gain/loss transactions are separate transactions to the sell transactions which record the reduction in the number of shares and the value received.
3. Select Actions > View Lots to open the Lots in Account SSSS window where SSSS is the security account.
4. Create a new lot using the New Lot button. Initially this lot is not linked to any buy or sell split.
5. Highlight the new lot in the Lots in This Account panel.
6. Highlight the buy split (dated 01/07/2016) of the security to be sold in the Splits free panel.
7. Click the >> button to link the buy split with the highlighted lot. The split moves from the Splits free panel to the Splits in Lot panel.
8. Repeat the previous 2 steps for any other buy splits that should be included in the lot (in this example, there is only 1 buy split in the sale).
9. Highlight the sell split in the Splits free panel.
10. Click the >> button to link the sell split with the highlighted lot.
11. Check the lot Balance is as expected. I.e. in this example the lot balance should be zero as the number of securities sold in the lot, is matched with the same number of security buys.

12. Click the Scrub button (*NOT* the Scrub Account button).

The Lots in Account window has not changed after using the Scrub button so no example screen image is supplied.

13. Close the Lots in Account SSSS window and return to the security account register.
14. Continue to *Change Orphaned Gains-CCC to Gain/Loss Account*

### Example 2: Automatic Creation of Capital Gain Or Loss Transactions

Create the capital gains transaction by following these steps:

1. Open the security account's register.
2. Ensure any previous manually entered capital gain/loss transaction splits have been entered in separate transactions to the the sell transactions.
3. Select Actions > View Lots to open the Lots in Account SSSS window where SSSS is the security account.
4. If using LIFO, use the above procedure *Manual Lot Creation and Linking* to create a lot for each sell transaction, link the lot with the sell transaction and each of the buy transactions that make up the sale.
5. Click the Scrub Account button which:
  - Creates lots for any buy transactions that are not already linked to a lot and links them to sell transactions splits using the FIFO method. As a transaction split can only be linked to 1 lot, if a sell transaction needs to be linked to multiple lots, the sell transaction split is itself split into multiple subsplits. In the case of multiple subsplits, it is possible to have different splits from the same transaction in both the Splits free and Splits in lot panels.
  - Creates a separate transaction per lot for capital gain/loss.
6. Continue to *Change Orphaned Gains-CCC to Gain/Loss Account*

### Example 3: Automatic Creation of Capital Gain Or Loss Transactions, 2 Sales at Once

Here is an example of FIFO scrubbing without manual lot creation. In this example, the transactions for 2 sales are scrubbed at once but usually scrubbing would be performed after each sale. One reason for scrubbing 2 sales at once, could be because there were multiple sales on the same day.

1. Select Actions > View Lots to open the Lots in Account SSSS window where SSSS is the security account.
2. Click the Scrub Account button.

---

**Note:** After using the Scrub Account button only the last lot is shown, so the above image is after the Lots in Account window has been closed and reopened so all the lots show.

---

3. Close the Lots in Account SSSS window and return to the security account register.

---

**Note:** The security splits in the sell transactions have been split into subsplits, one subsplit per lot, and a capital gain transaction has been created for each security subsplit of each sell transaction.

---

4. Continue to *Change Orphaned Gains-CCC to Gain/Loss Account*

\*GuideInvestExample.gnucash - STK2 - GnuCash

File Edit View Transaction Actions Business Reports Tools Windows Help

Save Close Split

Accounts STK2

Date	Num	Description		Tot Sh		Tot Buy	Tot Sell	Balance
01/01/2016		Bought 10 @ \$10.00 Brokerage \$1.00		10		10		10
	Buy		Assets:Investments:Broker1:STK2	n	10	10.1		
			Assets:Investments:Broker1	n			101.00	
01/02/2016		Div Reinvestment 1 @ \$11.00		1		1		11
	Buy	Div qty 10 @ CPS \$2.00 = \$20.00	Assets:Investments:Broker1:STK2	n	1	11	11.00	
		Increase in Residual \$20.00 - \$11.00 = \$9.00	Assets:Investments:DRP Residual:STK2	n			9.00	
			Income:Dividend:Franked:STK2	n			20.00	
01/07/2016		Div Reinvestment 2 @ \$12.00 = \$24.00		2		2		13
	Buy	Div qty 11 @ CPS \$2.00 = \$22.00	Assets:Investments:Broker1:STK2	n	2	12	24.00	
		Decrease in Residual = (\$9.00 - \$7.00) = \$2.00	Assets:Investments:DRP Residual:STK2	n			2.00	
			Income:Dividend:Franked:STK2	n			22.00	
02/07/2016		Sell FIFO Qty 11 @ \$13.00=\$143.00		-11			11	2
		Net Sale: \$143.00 - Commission \$1.00 = \$142.00	Assets:Investments:Broker1	n			142.00	
	Sell	Cost (10@10.10)+(1@11.00) = \$112.00	Assets:Investments:Broker1:STK2	n	-11	12.9091		142.00
01/08/2016		Div Reinvestment 1 @ \$14.00 = \$14.00		1		1		3
	Buy	Div qty 2 @ CPS \$4.00 = \$8.00	Assets:Investments:Broker1:STK2	n	1	14	14.00	
		Decrease in Residual = (\$7.00 - \$1.00) = \$6.00	Assets:Investments:DRP Residual:STK2	n			6.00	
			Income:Dividend:Franked:STK2	n			8.00	
02/08/2016		Sell FIFO Qty 3 @ \$15.00=\$45.00		-3			3	0
		Net Sale: \$45.00 - Commission \$1.00 = \$44.00	Assets:Investments:Broker1	n			44.00	
	Sell	Cost (2@12.00)+(1@14.00) = \$38.00	Assets:Investments:Broker1:STK2	n	-3	14.6667		44.00
28/02/2017	Num			0.00				
						1		

Shares: 0 STK2 Current Value: \$0.00

Enter a description of the transaction

Fig. 13: Selling Shares - Capital Gains - Register before Scrub Account

Lots in Account STK2

Title

Notes

Show only open lots

Splits free

Date	Num	Description	Amount	Value	Gain/Loss	Balance
01/01/2016	Bought 10 @ \$10.00 Brokerage \$1.00	10 STK2	\$101.00	10 STK2		
01/02/2016	Div Reinvestment 1 @ \$11.00	1 STK2	-\$11.00	11 STK2		
01/07/2016	Div Reinvestment 2 @ \$12.00 = \$24.00	2 STK2	-\$24.00	13 STK2		
02/07/2016	Sell FIFO Qty 11 @ \$13.00=\$143.00	-11 STK2	\$142.00	2 STK2		
01/08/2016	Div Reinvestment 1 @ \$14.00 = \$14.00	1 STK2	-\$14.00	3 STK2		
02/08/2016	Sell FIFO Qty 3 @ \$15.00=\$45.00	-3 STK2	\$44.00			

Splits in lot

Date	Num	Description	Amount	Value	Gain/Loss	Balance
------	-----	-------------	--------	-------	-----------	---------

New Lot Scrub Account Scrub Delete Close

Fig. 14: Selling Shares - Capital Gains - Lots before Scrub Account

Lots in Account STK2

Title Lot 0

Notes

Show only open lots

Splits free

Date	Num	Description	Amount	Value	Gain/Loss	Balance
01/01/2016	02/07/2016	Lot 0	0 STK2	\$28.09		
01/02/2016	02/07/2016	Lot 1	0 STK2	\$1.91		
01/07/2016	02/08/2016	Lot 2	0 STK2	\$5.33		
01/08/2016	02/08/2016	Lot 3	0 STK2	\$0.67		

Splits in lot

Date	Num	Description	Amount	Value	Gain/Loss	Balance
01/01/2016	Bought 10 @ \$10.00 Brokerage \$1.00	10 STK2	\$101.00	10 STK2		
02/07/2016	Sell FIFO Qty 11 @ \$13.00=\$143.00	-10 STK2	\$129.09	\$28.09		

New Lot Scrub Account Scrub Delete Close

Fig. 15: Selling Shares - Capital Gains - Lots after Scrub Account

### Example 4: Automatic Creation of Capital Gain Or Loss Transactions - After a Simple Stock Split

Here is an example of FIFO scrubbing without manual lot creation/linking, where the Stock Split Assistant has been used for a simple stock split. In this example, 100 shares of security XYZ were bought for \$10.00 each, there was a simple 2 for 1 stock split for zero cost (so the holding was then 200 shares @ \$5.00 each), then all 200 shares were sold for \$6.00 each.

	Action	Memo	Account	R	Shares	Price	Buy	Sell	
01/01/2008	Buy qty 100 @ \$10.00				100		100		100
	Buy		Assets:Investments:Broker1:XYZ	n	100	10	1,000		
			Assets:Current Assets:Bank1	n				1,000.00	
							1		
01/01/2009	Stock Split				100		100		200
	Split		Assets:Investments:Broker1:XYZ	n	100				
01/07/2009	Sell 200 for \$6.00 ea = \$1200.00				-200			200	0
			Assets:Current Assets:Bank1	n			1,200.00		
	Sell		Assets:Investments:Broker1:XYZ	n	-100	6		600.00	
	Sell		Assets:Investments:Broker1:XYZ	n	-100	6		600.00	
01/07/2009	Realized Gain/Loss				0				0
	Realized Gain/Loss		Assets:Investments:Broker1:XYZ	n			600.00		
	Realized Gain/Loss		Orphaned Gains-AUD	n				600.00	
01/07/2009	Realized Gain/Loss				0				0
	Realized Gain/Loss		Orphaned Gains-AUD	n			400.00		
	Realized Gain/Loss		Assets:Investments:Broker1:XYZ	n				400.00	

Shares: 0 XYZ Current Value: \$0.00

Fig. 16: Selling Shares - Capital Gains - Register after Scrub Account

The above screen shots show that scrubbing created:

2 lots. A separate lot for each buy (it essentially treats the stock split as a buy of 100 for no cost)

2 capital gain transactions (one for each lot) on the date of the sale:

- Lot 0: 1/7/2009 loss \$400 (sale \$600 cost \$1000)
- Lot 1: 1/7/2009 gain \$600 (sale \$600 cost \$0)

Total gain \$200 is correct. Whether the gain is a single long-term one or one each of long-term and short-term or whether there's even a distinction depends on the user's tax jurisdiction and the way the split is structured. If the user needs help figuring it out they should consult a professional.

### Example 5: Changing the Orphaned Gains-CCC to Gain/Loss Account

1. Close the Lots in Account SSSS window if open and return to the security account register.
2. Find each new *Realized Gain/Loss* transaction in the security account register (they will have the same date as the sell transactions). Edit the *Orphaned Gains-CCC* transaction split to re-assign the income account to a more meaningful income (or expense) gain or loss account (e.g. *Income:Long Term Capital Gain:IBM*).

**Tip:** You may like to split the capital gain/loss into taxable and non taxable parts if that is in accord with your tax laws.

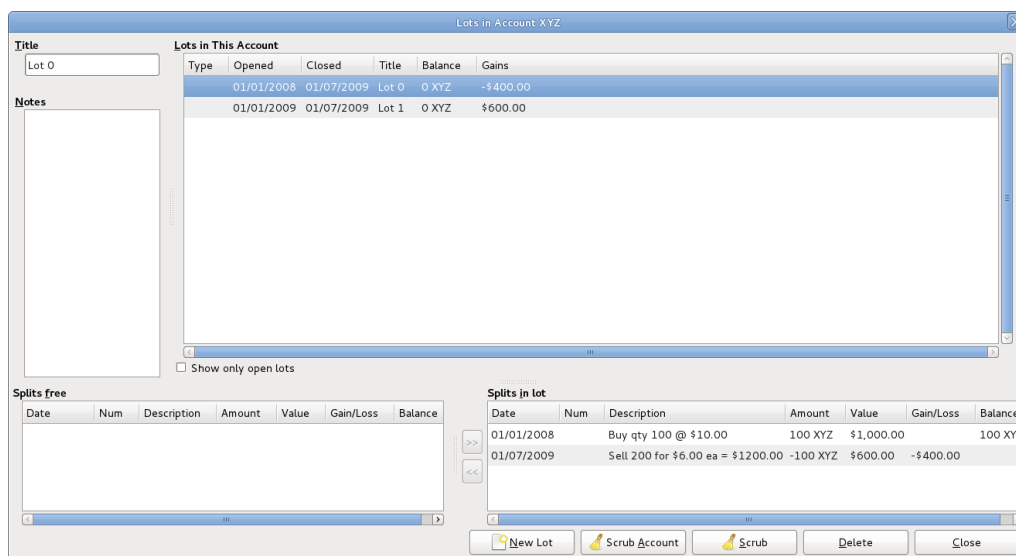


Fig. 17: Selling Shares - Capital Gains - Lot 0 after Scrub Account

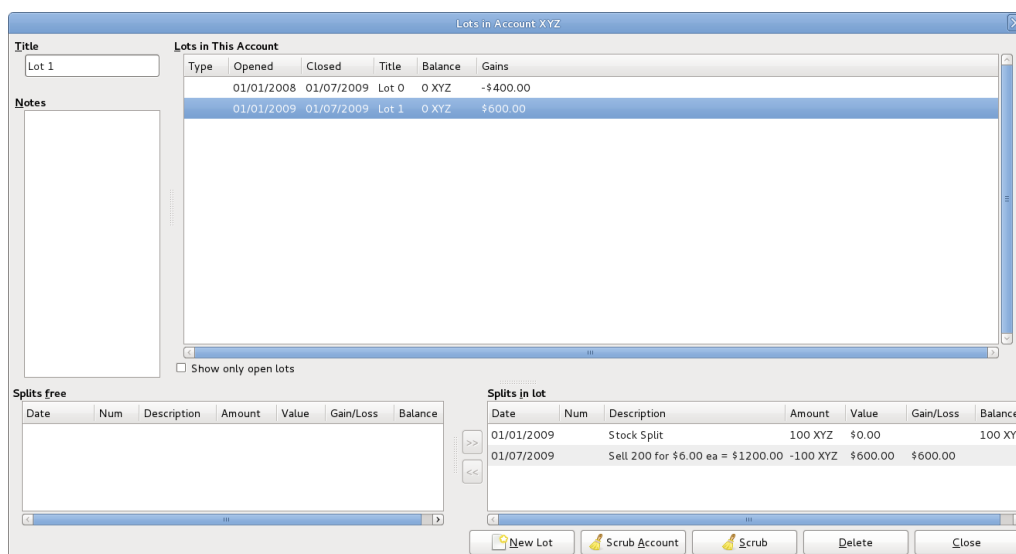


Fig. 18: Selling Shares - Capital Gains - Lot 1 after Scrub Account



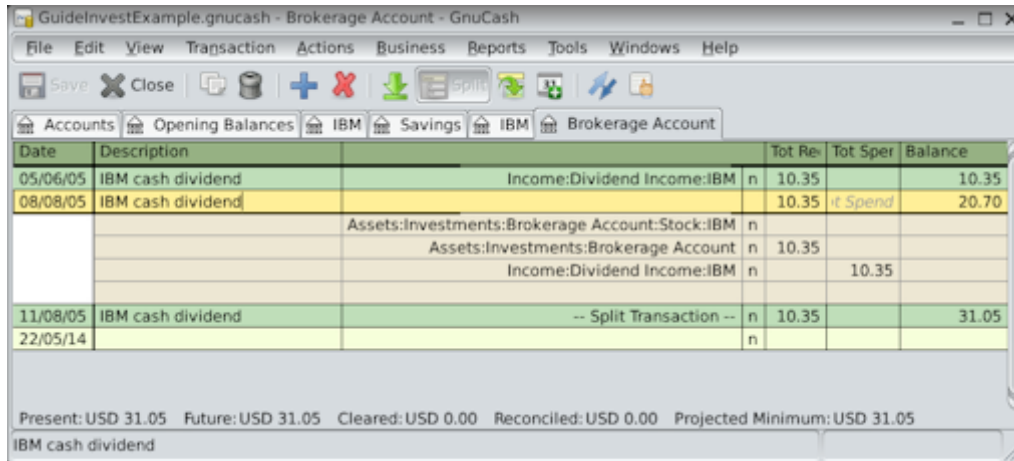
## 2.10.8 Dividends

Some companies or mutual funds pay periodic dividends to shareholders. Dividends are typically given in one of two ways, either they are automatically reinvested into the commodity or they are given as cash. Mutual funds are often setup to automatically reinvest the dividend, while common stock dividends usually pay cash.

### Dividends in Cash

If the dividend is presented as cash, you should record the transaction in the asset account that received the money, as income from *Income:Dividends*. Additionally if you want to tie the cash dividend to a particular stock holding then add a dummy transaction split to the stock account with quantity 0 price 1 value 0.

As an example consider the following; the dividends deposited as cash into the *Broker* Account with a tie to the stock account.



The screenshot shows the GnuCash interface for a 'Brokerage Account'. The transaction history table is as follows:

Date	Description		Tot Re	Tot Sper	Balance
05/06/05	IBM cash dividend	Income:Dividend Income:IBM	n 10.35		10.35
08/08/05	IBM cash dividend		10.35	it Spend	20.70
		Assets:Investments:Brokerage Account:Stock:IBM	n		
		Assets:Investments:Brokerage Account	n 10.35		
		Income:Dividend Income:IBM	n	10.35	
11/08/05	IBM cash dividend	-- Split Transaction --	n 10.35		31.05
22/05/14			n		

Summary: Present: USD 31.05 Future: USD 31.05 Cleared: USD 0.00 Reconciled: USD 0.00 Projected Minimum: USD 31.05

**Note:** If you want to track dividends on a per-stock basis, you would need to create an *Income:Dividends:STOCKSYMBOL* account for each stock you own that pays dividends.

### Dividends Re-Invested

If you receive the dividend in the form of an automatic reinvestment, the transaction for this should be handled within the stock or mutual fund account as income from "Income:Dividend" for the appropriate number of reinvested shares. This type of reinvest account is often referred to as a DRIP (Dividend Re-Investment Program).

As an example consider the following purchase of NSTAR (NST) stock with the dividends reinvested into a DRIP Account. Mutual fund re-investments would be the same.

Starting with the purchase of 100 shares on Jan. 3, 2005, all dividends will be reinvested and an account is created to track the dividend to the specific stock. GnuCash simplifies the entry by allowing calculations within the cells of the transaction. If the first dividend is \$.29/share, enter \$53.28 (purchase price + dividend) in the share Price cell and 100\*.29 in the Buy cell. GnuCash will calculate for you the corresponding number of Shares

The screenshot shows the GnuCash NST window. The table has columns: Date, Num, Description, Transfer, R, Shares, Price, Buy, Sell, and Balance. The data rows are:

Date	Num	Description	Transfer	R	Shares	Price	Buy	Sell	Balance
01/03/05		Initial Purchase	Equity:Opening Balances	n	100	52.99	5,299.00		100
01/06/05	Num	Reinvested Dividend	Income:DRIPs:NST	n	0.5443	53.2794	29.00	Sell	100.5443
02/24/14				n			1		

Shares: 100.5443 NST    Current Value: \$5,417.33  
Reinvested Dividend

### 2.10.9 Return of Capital

This refers to a transaction where an investment returns capital to the investor and doesn't have any accounting implications other than reducing the cost basis. The number of shares held is not changed.

A Return of Capital transaction can be entered in the stock register by entering the stock split with

Shares	0
Price	0
Sell	Return of Capital value

The other side of the double entry would usually be a debit to the brokerage bank account.

The screenshot shows the GnuCash STOCK window. The table has columns: Action, Memo, Account, R, Shares, Price, Buy, Sell, and Balance. The data rows are:

Action	Memo	Account	R	Shares	Price	Buy	Sell	Balance
01/04/2016	Buy 100 shares @ \$5.00 each			100		100		100
	Buy	Assets:Investments:Brokerage:STOCK	n	100	5	500.00		
		Assets:Current Assets:Bank	n				500.00	
02/04/2016	Return of Capital			0				100
		Assets:Current Assets:Bank	n			100.00		
	Action Memo	Assets:Investments:Brokerage:STOCK	n		Price	Buy	100	
						1		
16/04/2016				0.00				
			n			1		

Shares: 100 STOCK    Current Value: \$500.00  
Enter the number of shares bought or sold

**Note:** It is not possible to use the Stock Split Assistant to do this type of transaction.

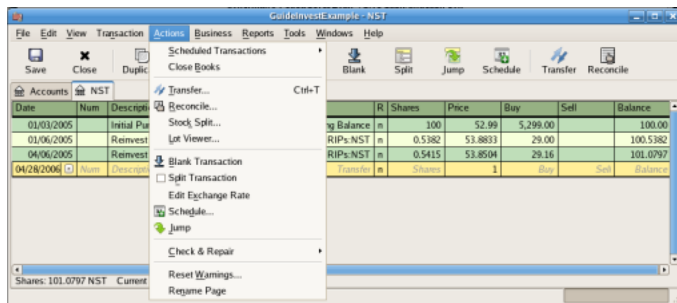
**Tip:** If you accidentally entered a non-zero price in the stock split, GnuCash may have created an unwanted price database entry which could cause reports to be wrong. Check for and remove such an unwanted entry from the price database using Tools > Price Editor.

## 2.10.10 Splits and Mergers

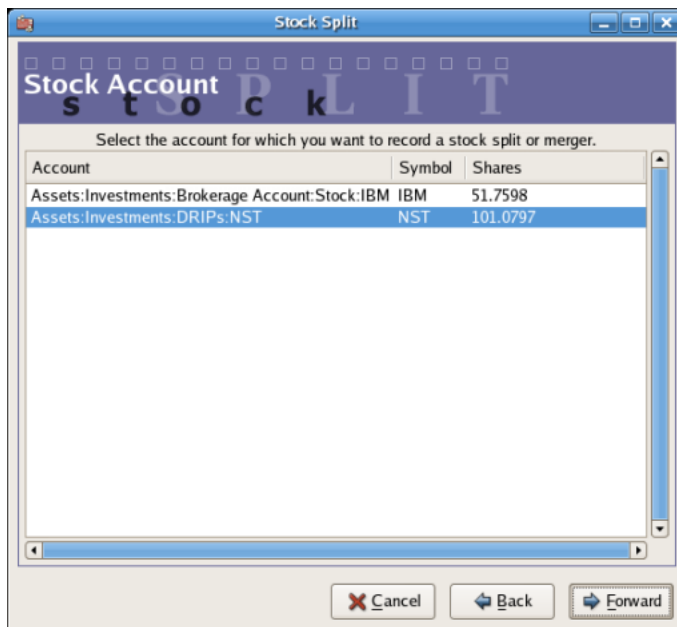
Companies may split their stock for many reasons but the most common is that the price has risen higher than management thinks is a reasonable price for many investors. Some of these splits are simple exchanges (eg 2 for 1 or 3 for 2) and some are complex exchanges with cash distributions. Splits may also result in fewer shares if the exchange rate is a reverse split (1 for 3 or .75 for 1).

### Simple Stock Split

As an example, our holding of NST stock declared a 2 for 1 stock split effective June 6, 2005. The process for entering this transaction is; select Actions > Stock Split to start the assistant.



The first screen is an Introduction, select Forward to display the selection of the account and stock for the split. You will need to create an entry for each *Account:Stock* combination you hold.



Select the *Assets:Investments:DRIPs:NST* and click on Forward.

The next screen presents 5 fields in the Stock Splits Details window:

- Date - Enter the date of the split.
- Shares - The number of shares increased (or decreased) in the transaction.

In our example it is a 2 for 1 split so the number of additional shares is the number of shares currently in the register.

- Description - The Description should give a brief explanation of the transaction.
- New Price - If desired the new price of the stock, after the split, may be entered.
- Currency - The currency of the transaction is required. This should be the same as the stock purchase currency.

Click on the Forward button.

The screenshot shows the 'Stock Split' dialog box with the 'Stock Split Details' tab selected. The dialog has a title bar with standard window controls. Below the title bar is a header area with the text 'Stock Split Details' and a decorative background. The main area contains instructions: 'Enter the date and the number of shares you gained or lost from the stock split or merger. For stock mergers (negative splits) use a negative value for the share distribution. You can also enter a description of the transaction, or accept the default one.' Below this are three input fields: 'Date:' with the value '06/06/2005', 'Shares:' with the value '101.0797', and 'Description:' with the value 'NST 2:1 Stock Split'. Below these is another instruction: 'If you want to record a stock price for the split, enter it below. You may safely leave it blank.' This is followed by a 'New Price:' input field and a 'Currency:' dropdown menu set to 'USD (US Dollar)'. At the bottom are three buttons: 'Cancel', 'Back', and 'Forward'.

The next screen will be skipped in this example as there was no "Cash in Lieu".

The screenshot shows the 'Stock Split' dialog box with the 'Cash In Lieu' tab selected. The dialog has a title bar with standard window controls. Below the title bar is a header area with the text 'Cash In Lieu' and a decorative background. The main area contains instructions: 'If you received a cash disbursement as a result of the stock split, enter the details of that payment here. Otherwise, just click "Forward".' Below this are two input fields: 'Amount:' and 'Memo:' with the value 'Cash In Lieu'. Below these are two columns of account lists. The left column is titled 'Income Account' and contains a list of accounts: Income, Bank ABC, Interest, CD, Savings, Brokerage Account, Cap Gain(Long), Dividends, Interest, DRIPs, and NST. The right column is titled 'Asset Account' and contains a list of accounts: Assets, Bank ABC, CD, Savings, Investments, Brokerage Account, DRIPs, and Orphan-USD. At the bottom are three buttons: 'Cancel', 'Back', and 'Forward'.

A final Finish screen will give a last option to; Cancel, Back to modify any data entered or Apply to complete the stock split with the data entered.

Date	Num	Description		Total Shares	Total Buy	Total Sell	Balance
01/03/2005		Initial Purchase	Equity Opening Balance	100	52.99	5,299.00	100.00
01/06/2005		Reinvest Dividend	Income DRIPs NST	0.5382	\$3.8833	29.00	100.5382
04/06/2005		Reinvest Dividend	Income DRIPs NST	0.5415	\$3.8504	29.36	101.0797
06/06/2005	100	2:1 Stock Split		101.0797			202.1594
		Sale	Assets:Investments:DRIPs:NST	101.0797			
05/09/2006						1	

Shares: 202.1594 NST Current Value: \$5,611.97  
2:1 Stock Split

## Moderately Complex Stock Merger

As an example, assume you held AT&T stock during the Nov. 18, 2005 merger of SBC with AT&T. For this example you will have purchased AT&T on April 1, 2005, any dividends will have been paid in cash, therefore not entered into the AT&T stock register.

The conditions of the merger were .77942 share of SBC stock were exchanged for each share of AT&T stock. The merged company continued to use the symbol "T" from AT&T.

AT&T paid a "dividend" of \$1.20/share on the transaction date, however this will not appear in the stock account as it was a cash distribution.

The process for entering this transaction is identical to the simple split until the "Details" screen. You will need to create an split entry in each *Investment Account:Stock* account combination that has shares splitting.

Stock Split

Select the account for which you want to record a stock split or merger.

Account	Symbol	Shares
Assets:Investments:Brokerage Account:Stock:IBM	IBM	51.7598
Assets:Investments:Brokerage Account:Stock:T	T	100
Assets:Investments:DRIPs:NST	NST	202.1594

Cancel Back Forward

Select the *Assets:Investments:Brokerage Account:Stock:T* and click on Forward.

The next screen presents 5 fields in the Stock Splits Details window:

- Date - Enter the date of the split. Here we'll enter November 18, 2005.
- Shares - The number of shares increased (or decreased) in the transaction.

In our example it is a .77942 for 1 split so the number of shares will decrease from the number of shares currently in the register. You may use GnuCash's ability to perform calculations on an entry form by entering data directly

(E.g. “(.77942\*100)100”) to calculate the decrease in shares from the split.

- Description - The Description should give a brief explanation of the transaction.
- New Price - If desired the new price of the stock, after the split, may be entered.
- Currency - The currency of the transaction is required. This should be the same as the stock purchase currency.

Click on the Forward button.

**Stock Split**

**Stock Split Details**

Enter the date and the number of shares you gained or lost from the stock split or merger. For stock mergers (negative splits) use a negative value for the share distribution. You can also enter a description of the transaction, or accept the default one.

Date: 11/18/2005

Shares: (.77942\*100)-100

Description: SBC / AT&T merger

If you want to record a stock price for the split, enter it below. You may safely leave it blank.

New Price: 24.38

Currency: USD (US Dollar)

Cancel Back Forward

The next screen will be skipped in this example as there was no “Cash in Lieu”.

A final “Finish” screen will give a last option to Back to modify any data entered or Apply to complete the stock split with the data entered.

Account	Advanced Portfolio	T
04/01/2005	Initial Purchase	Equity Opening Balance
11/18/2005	SBC / AT&T merger	Assets Investments Brokerage
04/01/2005		

Shares: 77.942 T Current Value: \$1,900.23  
Assets Investments Brokerage Account Stock T

## 2.11 Reports

GnuCash is a powerful double entry accounting software package that allows users to enter and track their money in a reliable manner. However, putting this information into GnuCash is only a part of the process. To be truly helpful, you need to be able to extract this information in meaningful ways. GnuCash’s reporting features allow you to just that.

GnuCash’s reporting features allow you to display nearly any group of transactions in a wide variety of formats. This makes it easy to answer questions about your finances, such as “How much did I spend on groceries last month?” or

“How much did I earn in the previous six months?”

GnuCash includes a number of common report types, which can be modified to meet your specific needs. If these common reports are insufficient, it is possible to modify or even write your own custom reports (although this is not recommended for beginners).

### 2.11.1 Overview

There are many standard reports pre-built in GnuCash, all available from the Reports pulldown menu in the main account window.

When you select a report from the list of reports, that report is first run with its default settings. Once you have opened the report, you can modify its parameters by clicking on the Options button on the toolbar. Under Options, you will see the different settings that you can change for each report. Note that for different reports, the options will be different.

### 2.11.2 Saved Reports Configurations Concepts

Once you have modified a report to meet your needs, you may wish to save that report for reuse at a later point. GnuCash allows custom reports to be saved, using the Save Report Configuration command.

To save a report configuration:

- Go to the Reports Menu and choose the desired report.
- Change the settings on the report until it includes what is needed.
- Go to the General tab of the report’s options and change the Report Name to something meaningful (Do not confuse this with the Report Title).
- Apply the changes and close the dialog.
- Click the Save Report Configuration or Save Report Configuration As... button

This will store the report options in a file in your home directory.

The first time you save a report with a name that has not already been saved, you can use either the Save Report Configuration or the Save Report Configuration As... button. You can modify the report name before saving it.

Once a report has been saved with the current name, the Save Report Configuration button will immediately update the saved report configuration. Use the Save Report Configuration As button to save the current report configuration with a new name.

Saved report configurations are available for use under the Reports->Saved Report Configurations entry. They will also be available for use on multicolumn reports.

Saved report configurations can be deleted in the Saved Report Configurations dialog by clicking the trashcan icon.

To edit saved report configurations, open the report via Reports->Saved Report Configurations, edit and apply the new options, and click Save Report Configuration.

### 2.11.3 Standard Reports Overview

The standard reports that are included in GnuCash are presented in this chapter. For each report, a short description is given, which explains what the report is intended to show and its primary purpose. We begin with a description of the Transaction Report, which is an all-purpose report with many features and settings that allow sophisticated examination of the data in a file.

### Transaction Report

The Transaction Report is a fundamental report that allows users to retrieve a wide variety of useful information from the financial records. The report includes many features that allow powerful and flexible reporting from the single basic report.

- Debit and Credit subtotals can be displayed, and within a grouping section, if debit totals > credit totals, the group subtotal will be displayed in the debit column.
- Multiple data columns enable display of transaction amounts in report currency and original currency in separate columns.
- Additional sortkeys e.g. sort by reconcile status (unreconciled -> cleared -> reconciled), sort by weeks, allow additional views into the financial records.
- Full text filter enabled for description, notes and memo, and for account full names. This allows easier selection of source account names and tailored reporting. Filters can optionally use full POSIX regular expressions.
- Indenting of columns will indent the report columns for easier understanding of grouping and sorting. (This is now enabled by default for GnuCash 3.0 onwards)
- Subtotal table is useful to understand the relationship in the data. This feature compiles the primary and secondary subtotals into a table view, allowing additional insight into financial records, e.g. account subtotals in different time series.

For further ideas on how to use the Transaction Report to produce common reports, see *Using the Transaction Report*.

### Assets & Liabilities Group

Reports in this group provide general information about assets and liabilities.

### Advanced Portfolio

The Advanced Portfolio produces a report on commodity accounts (stock and mutual fund type accounts) using price information stored in the GnuCash price database and within the GnuCash transaction data. If you do not have stock price information in your file, the report will indicate an error. This report includes extended information about commodity holdings, including information about the basis, gain, and return of individual commodities.

### Advanced Portfolio Capital Gains

The Advanced Portfolio report doesn't use the capital gain splits to calculate capital gains. It calculates the gains from the various buy and sell transactions in the account without regard to whether the gains and losses are recorded or not. Any realized gain splits are ignored. Realized gain splits are recognized as two splits, one in the stock's account with a zero number of shares and a non-zero value, the other in an income or expense account with a value that is the negative of the split in the stock account. These two splits can be in a separate transaction (as created by scrubbing) or in the same transaction as the sale (this will cause incorrect future scrubbing). The income or expense split can be split into multiple splits, say for taxable/non-taxable or short/long term gains, without affecting this report.

### Asset Barchart

The Asset Barchart presents the value of assets on a monthly basis in barchart form. By default, the report displays the eight largest accounts that have specific asset types assigned to them, and it displays bars for the current financial period.



Several settings on this report can greatly affect the information included.

- On the Accounts tab, the “Show Accounts until level” option changes how the report aggregates account totals. Change this value to see information at deeper levels of the account structure.
- On the Display tab, the “Maximum Bars” option will display more bars in the chart, allowing information for more accounts to display. Additionally, the “Show table” option enables the display of chart information in tabular form below the chart.
- On the General tab, the “Price Source” option can significantly affect the reported value of various commodities included in the report.

### **Asset Piechart**

The Asset Piechart presents the value of assets on a monthly basis in piechart form. By default, the report shows the seven largest accounts, that have specific asset types assigned to them, arranged in descending order by value as of the end of the current accounting period.

Several settings on this report can greatly affect the information included.

- On the Accounts tab, the “Show Accounts until level” option changes how the report aggregates account totals. Change this value to see information at deeper levels of the account structure.
- On the Display tab, the “Maximum Slices” option will display more slices in the chart, allowing information for more accounts to display.
- On the General tab, the “Price Source” option can significantly affect the reported value of various commodities included in the report.

### **Average Balance**

The Average Balance report displays monthly averages for the current accounting period.

### **Balance Sheet**

The Balance Sheet lists Asset, Liability, and Equity account balances for all such accounts, and provides totals as of a given date. Balance sheets are commonly run for the last day of each fiscal year to give an overall sense of the financial condition of the entity.

### **General Journal**

The General Journal produces a register of all transactions (beginning to end) in order by date, showing the accounts and the amounts involved, and totals the Net Change by all currencies and assets.

This report is not customizable by date or account, though you can include more or fewer details about the individual transactions, and whether to include running balances and totals for the credits and debits. If you need a report restricted to particular accounts, consider the Transaction Report or open a particular account and choose the Account Transaction Report.

### **General Ledger**

The General Ledger produces information about all transactions for a selected set of accounts. When first run, this report loads no data, and the report options must be changed to retrieve information from the file.

## **Investment Portfolio**

The Investment Portfolio produces a report of commodity accounts (that is, accounts with type “Stock” or “Mutual Fund”), giving holdings, price and value information about commodities in the file.

## **Liability Barchart**

The Liability Barchart presents the value of liabilities on a monthly basis in barchart form. By default, the report displays the eight largest accounts that have specific asset types assigned to them, and it displays bars for the current financial period.

Several settings on this report can greatly affect the information included.

- On the Accounts tab, the “Show Accounts until level” option changes how the report aggregates account totals. Change this value to see information at deeper levels of the account structure.
- On the Display tab, the “Maximum Bars” option will display more bars in the chart, allowing information for more accounts to display. Additionally, the “Show table” option enables the display of chart information in tabular form below the chart.
- On the General tab, the “Price Source” option can significantly affect the reported value of various commodities included in the report.

## **Liability Piechart**

The Liability Piechart presents the value of liabilities on a monthly basis in piechart form. By default, the report shows the seven largest accounts, that have specific asset types assigned to them, arranged in descending order by value as of the end of the current accounting period.

Several settings on this report can greatly affect the information included.

- On the Accounts tab, the “Show Accounts until level” option changes how the report aggregates account totals. Change this value to see information at deeper levels of the account structure.
- On the Display tab, the “Maximum Slices” option will display more slices in the chart, allowing information for more accounts to display.
- On the General tab, the “Price Source” option can significantly affect the reported value of various commodities included in the report.

## **Net Worth Barchart**

The Net Worth Barchart summarizes Asset accounts, Liability accounts, and overall Net Worth as bars on a graph on a monthly basis for the current financial period. This report provides a graphic overview of the file over time.

## **Net Worth Linechart**

The Net Worth Linechart summarizes Asset accounts, Liability accounts, and overall Net Worth as a line graph on a monthly basis for the current financial period. This report provides a graphic overview of the file over time.

## **Price Scatterplot**

The Price Scatterplot displays the value of one commodity relative to another commodity, for example the value of a stock relative to a currency. When first run, this report loads no data, and the report options must be changed to retrieve information from the file. Specifically, the “Price of Commodity” setting on the Price options tab must be changed to a specific commodity.

## **Budget Group**

Budget reports in GnuCash allow you to gather summary information related to budgets you may have created. In order for these reports to work, you must first create a budget. The reports in this group are specifically based on budget information. To use these reports, you need to have a budget saved in your file.

### **Budget Balance Sheet**

### **Budget Barchart**

### **Budget Flow**

### **Budget Income Statement**

### **Budget Profit & Loss**

### **Budget Report**

## **Business Group**

Reports in this group provide general information about activities related to a business.

## **Customer Report**

### **Customer Summary**

Customer Summary is a customer profit report that can help with job analysis by comparing the income and expenses for a specific customer.

All invoices have an Owner in GnuCash, so invoices that are made will show a customer and show in the report. When creating a Bill, the Default Chargeback Customer is blank. To use the profit report, this field needs an entry, since this is the tag that decides the line to which to attach the expense. Left blank, the bill will be assigned to “No Customer.” Similarly, when income is entered directly in a register rather than creating an invoice, it will also be assigned to “No Customer.”

Thus, if this report includes an entry for “No Customer”, this means that the report may be inaccurate, as the results are not all properly labeled.

Possible use scenarios include:

- Tracking retail sales from different cities
- Tracking rental properties
- Tracking types of business

- Tracking commission sales

Each of these scenarios assumes that the account structure includes breakdowns for individual tracked categories. Changing settings on the Income and Expense tabs under Options can hone the information displayed. By default all income and expense accounts are included; however, since GnuCash can't really predict the names and classification of income and expense accounts, it must group them all into the "No Customer" entry.

Note that inventory-based businesses won't benefit from this report because of its nature.

Useful options:

- The Expense Accounts tab allows the selection of one or more expense accounts.
- The Income Accounts tab allows the selection of one or more income accounts.
- The Display tab allows sorting by name, profit percentage, or amount of profit.

### Easy Invoice

### Employee Report

### Fancy Invoice

### Payable Aging

### Printable Invoice

### Receivable Aging

This report provides a listing of all customers, their current balance, and how much they have outstanding from invoices over different time periods—how much they owe from 0-30 days, from 31-60 days, from 61-90 days, and over 90 days. The report also contains links to each customer and to their current customer report.

### Vendor Report

### Income and Expense Group

Reports in this group provide information about Income and Expense

### Cash Flow

This report shows the change in value for a set of accounts (the flow of cash) over a given period of time. By default, this report is based on accounts in Assets and Special Accounts, and covers the current financial period. The report enumerates all money coming in to and going out of the base accounts, broken down by the other account.

### Equity Statement

This report can be seen as extension of the Balance Sheet report. The Balance Sheet states the balance of Assets, Liabilities and Equity at a specific point of time. The Equity Statement focuses on the Equity Accounts by showing the cash flow to and from them for a given period of time.

By balancing this cash flow with income, the report shows the available capital at the beginning and end of the selected time period.

## Expense Barchart

The Expense Barchart presents the value of expenses on a monthly basis in barchart form. By default, the report displays the eight largest accounts that have specific expense types assigned to them, and it displays bars for the current financial period.

Several settings on this report can greatly affect the information included.

- On the Accounts tab, the “Show Accounts until level” option changes how the report aggregates account totals. Change this value to see information at deeper levels of the account structure.
- On the Display tab, the “Maximum Bars” option will display more bars in the chart, allowing information for more accounts to display. Additionally, the “Show table” option enables the display of chart information in tabular form below the chart.
- On the General tab, the “Price Source” option can significantly affect the reported value of various commodities included in the report.

## Expense Piechart

The Expense Piechart presents the value of expenses on a monthly basis in piechart form. By default, the report shows the seven largest accounts, that have specific expense types assigned to them, arranged in descending order by value as of the end of the current accounting period.

Several settings on this report can greatly affect the information included.

- On the Accounts tab, the “Show Accounts until level” option changes how the report aggregates account totals. Change this value to see information at deeper levels of the account structure.
- On the Display tab, the “Maximum Slices” option will display more slices in the chart, allowing information for more accounts to display.
- On the General tab, the “Price Source” option can significantly affect the reported value of various commodities included in the report.

## Expenses vs. Day of Week

Expenses vs. Day of Week presents a pie chart showing the totals for selected expense accounts totaled by the day of the week of the transaction. The report options enable you to make some adjustments (such as accounts, display options, and the date range) but the account selector only allows expense accounts to be chosen. The report aggregates expense transactions by day of week, not by any other period or category. Due to these limitations, the report may be considered a demonstration or an example to someone wanting to examine the source code for composing a useful custom report.

## Income Barchart

The Income Barchart presents the value of income on a monthly basis in barchart form. By default, the report displays the eight largest accounts that have specific income types assigned to them, and it displays bars for the current financial period.

Several settings on this report can greatly affect the information included.

- On the Accounts tab, the “Show Accounts until level” option changes how the report aggregates account totals. Change this value to see information at deeper levels of the account structure.
- On the Display tab, the “Maximum Bars” option will display more bars in the chart, allowing information for more accounts to display. Additionally, the “Show table” option enables the display of chart information in tabular form below the chart.
- On the General tab, the “Price Source” option can significantly affect the reported value of various commodities included in the report.

## Income & Expense Chart

### Income Piechart

The Income Piechart presents the value of income on a monthly basis in piechart form. By default, the report shows the seven largest accounts, that have specific income types assigned to them, arranged in descending order by value as of the end of the current accounting period.

Several settings on this report can greatly affect the information included.

- On the Accounts tab, the “Show Accounts until level” option changes how the report aggregates account totals. Change this value to see information at deeper levels of the account structure.
- On the Display tab, the “Maximum Slices” option will display more slices in the chart, allowing information for more accounts to display.
- On the General tab, the “Price Source” option can significantly affect the reported value of various commodities included in the report.

### Income Statement

This report lists Income and Expense account totals for a set period. By default, it shows all Expense and Income accounts down to 3 levels of sub-accounts for the current financial period.

An Income Statement is also called a “Profit and Loss” report or “Revenue Statement.”

In earlier versions of GnuCash, this report was called “Profit & Loss,” but with version 2, the report was renamed “Income Statement” to use more common accounting terminology.

The Income Statement helps show where money is coming from and where it is going for a given time period.

### Income vs. Day of Week

Income vs. Day of Week presents a piechart showing the totals for selected income accounts totaled by the day of the week of the transaction. The report options enable you to make some adjustments (such as accounts, display options, and the date range) but the account selector only allows income accounts to be chosen. The report aggregates income transactions by day of week, not by any other period or category. Due to these limitations, the report may be considered a demonstration or an example to someone wanting to examine the source code for composing a useful custom report.

### Trial Balance

Trial Balance lists the ending balances in all accounts as of a particular date. It is typically run at the end of an accounting period and is primarily used to ensure that the total of all debits equals the total of all credits.

## Income and GST Statement

The Income and GST Statement is a specialised Transaction Report designed to print Business-related Sales and Receipts, as well as their GST or VAT components. This report is designed for use in Australia, but is also usable in any jurisdictions implementing value-added taxes (or goods and services taxes) during regular business sales and receipts; and the business owner is expected to submit periodic reports of total sales, purchases, and taxes collected on sales from clients or paid to a supplier.

This report makes some assumptions upon the accounts used for sales, purchases, and taxes collected on sales, and paid on purchases.

- Net Sales must be an Income-type account
- Net Purchases must be an Expense-type account
- GST/VAT on Sales must be a Liability-type account
- GST/VAT on Purchases must be an Asset-type account
- There may be multiple GST accounts, e.g. reduced GST, standard GST etc. The GST accounts can be printed individually, or summarised for sales and purchases.
- There may be multiple sales and expenses accounts, e.g. Income:Sales, Income:Grants, Expenses:Suppliers, Expenses:Marketing, Expenses:Insurance. These amounts may be reported individually, or summarised for sales and purchases.
- The transactions may be entered manually as a multi-split transaction, or they may be entered via the Business Invoices and Bills tools.

As an example, consider a business has the following transactions: It incurs sales of \$1,000 plus 5% GST, receiving \$1,050. It also incurs GST-exempt sales of \$600. It purchases goods/services worth \$400 plus 5% = \$420. The following transactions will record the business activities:

Table 16: Sample Chart of Accounts for Income and GST Statement

Date	Description	Account	Debit	Credit
01/01/2018	Sales \$1,000 + 5% GST	Income:Sales		\$1,000
	GST on Sales [Liability]	\$50		
	Asset:Bank	\$1,050		
02/01/2018	GST-Free Sales \$600	Income:Sales		\$600
	Asset:Bank	\$600		
03/01/2018	Purchase \$400 + 5% GST	Expenses:Purchases	\$400	
	Asset:Bank	\$420		
	GST on Purchases [Asset]	\$20		

For the Income and GST Statement, the `Accounts` selected will be the Income and Expense Accounts, and the `Tax accounts` are used to select the GST accounts. The GST period is selected via the General tab. The resulting report is shown as follows:

Table 17: Income and GST Statement

Date	Description	Total Sales	Net Sales	Tax on Sales	Total Purchases	Net Purchases	Tax on Purchases
01/01/2018	Sales \$ 1,000 + 5%	\$ 1,050	\$ 1,000	\$50			
02/01/2018	GST -Free Sales \$600	\$600	\$600				
03/01/2018	Purchase \$400 + 5% GST				\$420	\$400	\$20
Totals		\$ 1,650	\$ 1,600	\$50	\$420	\$400	\$20

This report would indicate that there was a total of \$50 GST collected, and \$20 GST on purchases, therefore a \$30 GST liability is payable to the authorities.

### Sample & Custom Group

The reports in this group offer examples on how reports can be customized or modified to suit personal need.

### Custom Multicolumn Report

This report provides a base that allows several standard and custom reports to be combined into one view. Note that this report opens with an empty window; you must open the options and designate which reports to include for display. Once the reports have been selected, the settings for individual reports in the multicolumn display can be edited.

### Sample Report with Examples

This is a sample report that users can examine to learn how to write their own reports.

### Welcome Sample Report

This report demonstrates how the Multicolumn Report can be used to create custom dashboard-type reports.

### Other Reports

Several reports are included on the main Reports menu. These are described below.

### Account Summary

This lists the balances of all accounts and subaccounts as of a particular date. By default, this report shows accounts and totals down to third-level subaccounts.

This report gives effectively the same information as the Chart of Accounts. You can use this report to export and print the Chart of Accounts.

---

**Note:** To generate a report of account totals over a particular time period (especially if you do not close your books at regular intervals), you might consider using the Income Statement, or Cash Flow reports.

---



## Balance Forecast

This creates a balance chart which aims to track asset (or any other account type) balances, including amounts from future scheduled transactions. The balances will include all regular booked amounts and future amounts from numeric scheduled transaction templates.

This report can also add additional data charts to forecast future minimum balances. Reserve line can be displayed to confirm if future minimum will dip below the reserve amount. A target line may also be displayed above the reserve line for financial planning.

## Future Scheduled Transactions Summary

### Tax Report & TXF Export

Generates a report and a downloadable .txf file of taxable income and deductible expenses for a particular accounting period. To download the report data, choose the Export button on the toolbar and choose between html and .txf downloadable versions.

To use this report, you must use Edit -> Tax Options to identify which form the taxing authority uses for each income or expense account. Note that you can see but not modify the “Tax related” checkbox in Edit -> Edit Account.

### Saved Report Configurations

Selecting this will open a dialog with a list of available Saved Report Configurations. “Saved Report Configurations” means sets of customized settings for standard reports.

These sets must be saved by the user before they appear here. See Report Concepts above for instructions on how to save report configurations.

### Account Report

The Account Report menu entry only appears when an account register is the active tab. This report produces a list of all transactions in the open register.

Note that if you conduct a search that retrieves several transactions, the results are displayed in a new search register, which can then be used to create a report for just those transactions.

### Account Transaction Report

This report also only appears when an account register is the active tab. However, this report only lists transactions that have been selected (e.g. by mouse click) in the current register. If no transactions are selected, an empty report will be generated.

### Using the Transaction Report

The Transaction Report can be heavily customised to produce most reports appropriate for personal finance and business bookkeeping purposes. It aims to retrieve and filter information from the database, and present transactions and totals, useful for the user and the accountant.

The following guide to the Transaction Report will assume the user has set up the chart of accounts according to conventional bookkeeping practice. For example, the following describes a typical household book with skeleton accounts. Further accounts will undoubtedly be necessary.

Table 18: Sample Chart of Accounts for the Transaction Report

Account Name	Account Type
Asset	ASSET (placeholder)
Asset:Bank	BANK
Asset:Property	ASSET
Liability	LIABILITY (placeholder)
Liability:Credit Card	CREDIT CARD
Liability:Home Loan	LIABILITY
Income	INCOME (placeholder)
Income:Salary	INCOME
Income:Interest	INCOME
Expense	EXPENSE (placeholder)
Expense:Groceries	EXPENSE
Expense:Auto	EXPENSE
Expense:Medical	EXPENSE
Equity	EQUITY (placeholder)
Equity:Opening Balances	EQUITY

Conventionally, the oldest transaction would be equity transfers from Equity:Opening Balances to Asset and Liability Accounts. Most subsequent transactions would be transfers from Income/Expense accounts to Assets/Liability accounts (representing day-to-day activity e.g. receiving salary, paying utility bill), or transfers between assets and liability accounts (representing movements between loans and assets, e.g. paying credit card bill, receiving a loan).

The following use cases will be illustrated, and the options to be selected explained:

### Using the Transaction Report to show previous year expenses

The Transaction Report can show how much was spent on expense accounts last year. This will usually be the most useful basic transaction report format. To create this report:

1. Open the report options, from the Accounts tab, click the Expense placeholder account and Select Children.
2. From the General tab, Choose relative dates “Start of Previous Year” and “End of Previous Year”.
3. From the Sorting tab, set Primary Key to Accounts, and enable Primary Key Subtotal.
4. Set the Secondary Key to Date, and set the Secondary Key Subtotal to None.

### Secondary level grouping

The following will modify the above Transaction Report to use a secondary grouping strategy. The first grouping (i.e. Primary Key = Accounts, Subtotal = enabled) will group transactions from the same accounts, whereas the secondary grouping (i.e. Secondary Key = Date, Subtotal = Monthly) will also calculate monthly sums within each account.

1. Select options as above
2. From the Sorting tab, set Secondary Key Subtotal to Monthly

## Using secondary grouping for periodic comparison

The Display / Subtotal Table adds a subtotal summary table, which will ease comparison of accounts across date periods. This displays the same subtotal calculations as the main table, presented in a grid structure.

In addition, if there are multiple primary-key groups (e.g. Date grouping with monthly subtotals, across multiple months) the subtotal table will also compute and display the primary-key grouping Average. The average is computed as the total amount *per row*, divided by the *number of columns*. This may cause confusion when computing monthly spend averages, for instance, if a month has no transactions and is not displayed in the table, it will *not* create a corresponding column in the table, therefore the average amount will not reflect a true monthly average.

1. Select options as above
2. From the Display tab, enable the Subtotal Table.

## Use filtering to limit accounts and transactions

Accounts and transactions can be filtered for reporting according to account full name (e.g. “:Business:” will select account structure Income:Business:Sales, Income:Business:Grants, Expenses:Business:Rent, Expenses:Business:Utilities) or transaction description/notes/memo (select transactions tagged #gift). The filtering text may be either a standard string, or a POSIX regular expression.

1. From the Filter tab, complete the Account Name Filter, or Transaction Filter. Optionally toggle the regular expression check boxes to toggle full POSIX regex matching. Regular expressions will allow more complex queries e.g. the transaction filter can be set to (#gift#travel) which will filter transactions with either #gift or #travel. They can be complicated and are best learned from external guides.

## Using the Transaction Report to generate a reconciliation statement.

This sortkey ‘reconcile status’ will separate reconciled, cleared and unreconciled transactions. This report may be useful in producing a printable reconciliation report.

Alternatively, the Reconciliation Report will preset these defaults and requires only the Accounts selection.

1. From the Accounts tab, select the relevant bank or credit card account
2. From the General tab, choose an appropriate date range e.g. past 3 months
3. From the Sorting tab, set the Primary Key to Reconcile Status, Primary Key Subtotal to enabled, and set the Secondary Key to Date. The Secondary Date Subtotal is left to None

## 2.12 Capital Gains

This chapter will present some of the techniques used to keep track of the unrealized and realized gains and losses, better known as capital gains and losses.

### 2.12.1 Basic Concepts

This chapter will present some of the techniques used to keep track of the unrealized and realized gains and losses, better known as capital gains and losses.

Certain resellable assets can change value over time, such as stocks, bonds, houses, or cars. Some assets (eg: a stock) could increase in value, some (eg: a car) could decrease in value. It is important to be able to track some of these time-dependent asset valuations, this chapter will show you how.

Probably everything you own will increase or decrease in value over time. So, the question is for which of these assets should you track this changing value? The simple answer is that you only need to track this for items which could be sold for cash in the future or which relate to taxation.

Consumable and disposable items (eg: food, gas for your car, or printer paper) are obviously not involved. Thus, even though the new clothes you recently bought will certainly depreciate, you would not want to track this depreciation since you have no intention of reselling the clothes and there is no tax implications to the depreciation on clothing. So, for this example, the purchase of new clothes should be recorded as a pure expense... you spent the money, and it is gone.

Asset appreciation occurs when something you own increases in value over time. When you own an asset which has increased in value, the difference between the original purchase price and the new value is known as *unrealized gains*. When you sell the asset, the profit you earn is known as *realized gains* or *capital gains*. An example of an asset from which you could have unrealized gains, and eventually capital gains, is stock in a publicly traded company.

### 2.12.2 Estimating Valuation

As mentioned in the introduction to this chapter, capital gains are the profits received from the sale of an asset. This section will describe how to record capital gains in GnuCash.

The accounting methods for handling asset appreciation differs somewhat from depreciation because typically you are only concerned with the moment you sell the asset and realize the capital gains. This is opposed to the continuous nature of tracking depreciation. Capital gains are an important subject in the world of taxation, because governments tend to be quite interested in taxing capital gains in one manner or another.

---

**Note:** As always, there are exceptions. If you hold a bond that pays all of its interest at maturity, tax authorities often require that you recognize interest each year, and refuse this to be treated as a capital gain. Consult the appropriate tax codes to determine the preferred treatment for each type of asset you have which may be affected by capital gains taxes.

---

Estimating the increasing value of assets is generally not simple, because often it is difficult to know its exact value until the moment it is sold.

Securities traded daily on open markets such as stock exchanges are possibly the easiest type of asset to predict the value of, and selling the asset at market prices may be as simple as calling a broker and issuing a Market Order. On the other hand, estimating the value of a house is more difficult. Homes are sold less often than stocks, and the sales tend to involve expending considerable effort and negotiations, which means that estimates are likely to be less precise. Similarly, selling a used automobile involves a negotiation process that makes pricing a bit less predictable.

Values of collectible objects such as jewelry, works of art, baseball cards, and “Beanie Babies” are even harder to estimate. The markets for such objects are much less open than the securities markets and less open than the housing market. Worse still are one-of-a-kind assets. Factories often contain presses and dies customized to build a very specific product that cost tens or hundreds of thousands of dollars; this equipment may be worthless outside of that very specific context. In such cases, several conflicting values might be attached to the asset, none of them unambiguously correct.

The general rule of thumb in accounting for estimating unrealized gains (or losses) is that you should only revalue assets such as stocks which are readily sellable and for which there are very good estimates of the value. For all other assets, it is better to simply wait until you sell them, at which time you can exactly determine the capital gains. Of course, there is no hard rule on this, and in fact different accountants may prefer to do this differently.

### 2.12.3 Account Setup

As with most accounting practices, there are a number of different ways to setup capital gains accounts. We will present here a general method which should be flexible enough to handle most situations. The first account you will need is an *Asset Cost* account (GnuCash account type Asset), which is simply a place where you record the original purchase of the asset. Usually this purchase is accomplished by a transaction from your bank account.

In order to keep track of the appreciation of the asset, you will need three accounts. The first is an *Unrealized Gains* asset account in which to collect the sum of all of the appreciation amounts. The *Unrealized Gains* asset account is balanced by a *Unrealized Gains* income account, in which all periodic appreciation income is recorded. Finally, another income account is necessary, called a *Realized Gains* in which you record the actual capital gains upon selling the asset.

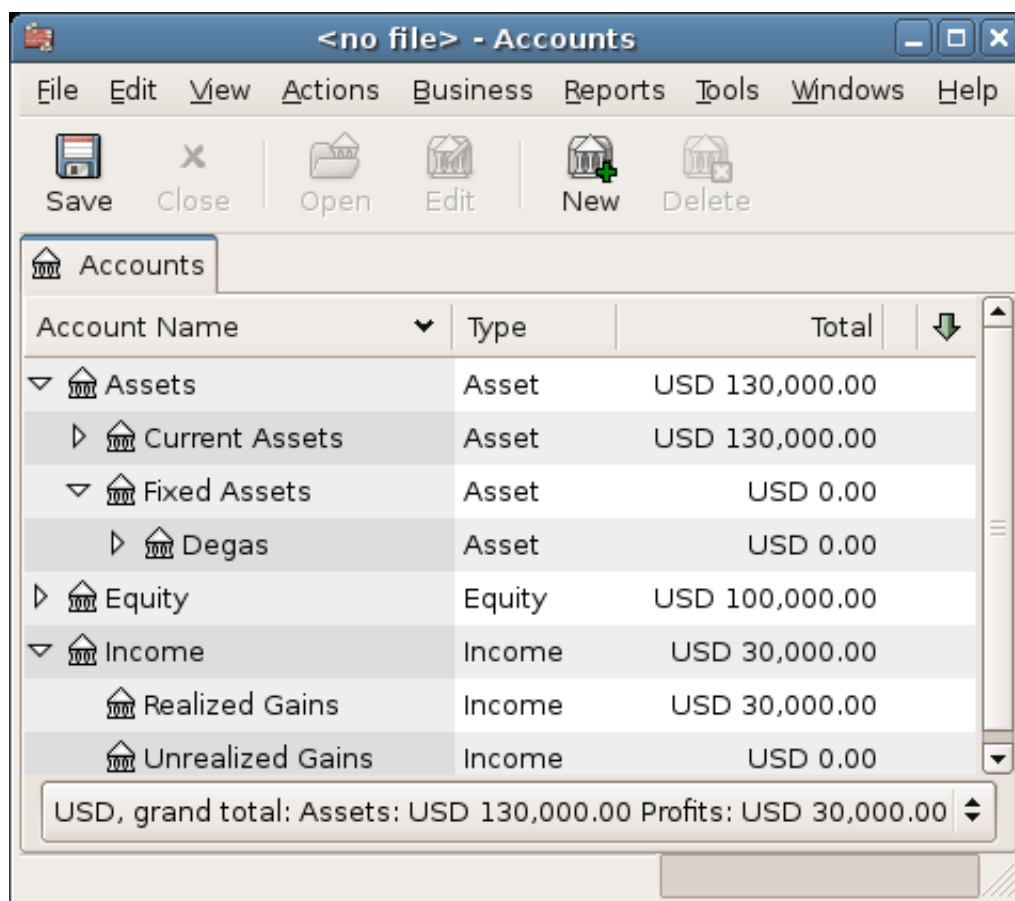
Below is a generic account hierarchy for tracking the appreciation of 2 assets, *ITEM1* and *ITEM2*. The *Assets:Fixed Assets:ITEM1:Cost* accounts are balanced by the *Assets:Current Assets:Savings Account* account, the *Assets:Fixed Assets:ITEM1:Unrealized Gains* accounts are balanced by the *Income:Unrealized Gains* account (similar for *ITEM2*).

```
-Assets
  -Current Assets
    -Savings Account
  -Fixed Assets
    -ITEM1
      -Cost
      -Unrealized Gain
    -ITEM2
      -Cost
      -Unrealized Gain
-Income
  -Realized Gains
  -Unrealized Gains
```

### 2.12.4 Example

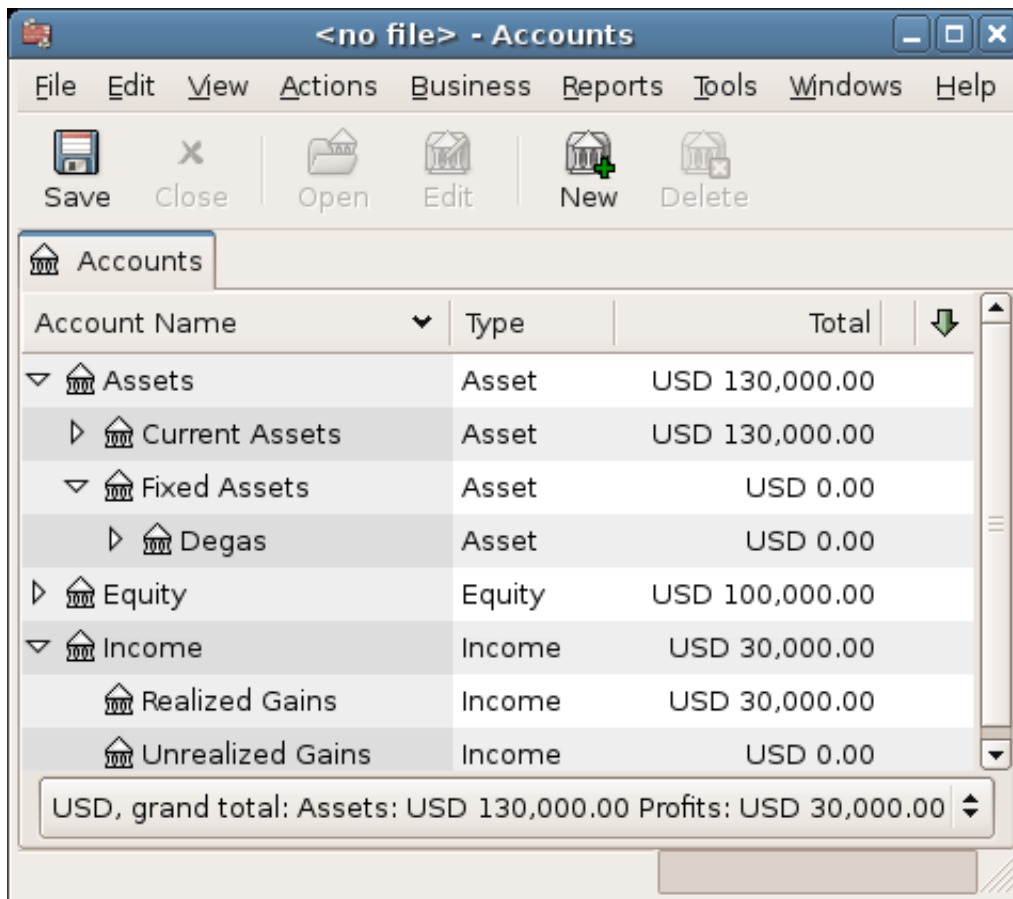
Let's suppose you buy an asset expected to increase in value, say a Degas painting, and want to track this. (The insurance company will care about this, even if nobody else does.)

Start with an account hierarchy similar to that shown in *Account Setup*, but replace "ITEM1" with "Degas" and you can remove the "ITEM2" accounts. We will assume that the Degas painting had an initial value of one hundred thousand dollars. Begin by giving your self \$100,000 in the bank and then transferring that from your bank account to your *Assets:Fixed Assets:Degas:Cost* account (the asset purchase transaction). You should now have a main account window which appears like this:



## Unrealized Gains

A month later, you have reason to suspect that the value of your painting has increased by \$10,000 (an unrealized gain). In order to record this you transfer \$10,000 from your *Accrued Gains* income account (*Income:Unrealized Gains*) to your asset Unrealized Gains account (*Assets:Fixed Assets:Degas:Unrealized Gain*). Your main window will resemble this:



The screenshot shows the 'Accounts' window in GnuCash. It features a menu bar (File, Edit, View, Actions, Business, Reports, Tools, Windows, Help) and a toolbar with icons for Save, Close, Open, Edit, New, and Delete. Below the toolbar is a tab labeled 'Accounts'. The main area displays a tree view of accounts with columns for 'Account Name', 'Type', and 'Total'. The accounts are organized hierarchically: Assets (USD 130,000.00), Current Assets (USD 130,000.00), Fixed Assets (USD 0.00), Degas (USD 0.00), Equity (USD 100,000.00), Income (USD 30,000.00), Realized Gains (USD 30,000.00), and Unrealized Gains (USD 0.00). A summary bar at the bottom shows 'USD, grand total: Assets: USD 130,000.00 Profits: USD 30,000.00'.

Account Name	Type	Total
Assets	Asset	USD 130,000.00
Current Assets	Asset	USD 130,000.00
Fixed Assets	Asset	USD 0.00
Degas	Asset	USD 0.00
Equity	Equity	USD 100,000.00
Income	Income	USD 30,000.00
Realized Gains	Income	USD 30,000.00
Unrealized Gains	Income	USD 0.00

USD, grand total: Assets: USD 130,000.00 Profits: USD 30,000.00

## Selling

Let's suppose another month later prices for Degas paintings have gone up some more, in this case about \$20,000, you estimate. You duly record the \$20,000 as an unrealized income like above, then decide to sell the painting.

Three possibilities arise. You may have *accurately estimated* the unrealized gain, *underestimated* the unrealized gain, or *overestimated* the unrealized gain.

### 1. *Accurate estimation* of unrealized gain.

Your optimistic estimate of the painting's value was correct. First you must record that the profits made are now realized gains, not unrealized gains. Do this by transferring the income from the *Income:Unrealized Gains* to the *Income:Realized Gains* account.

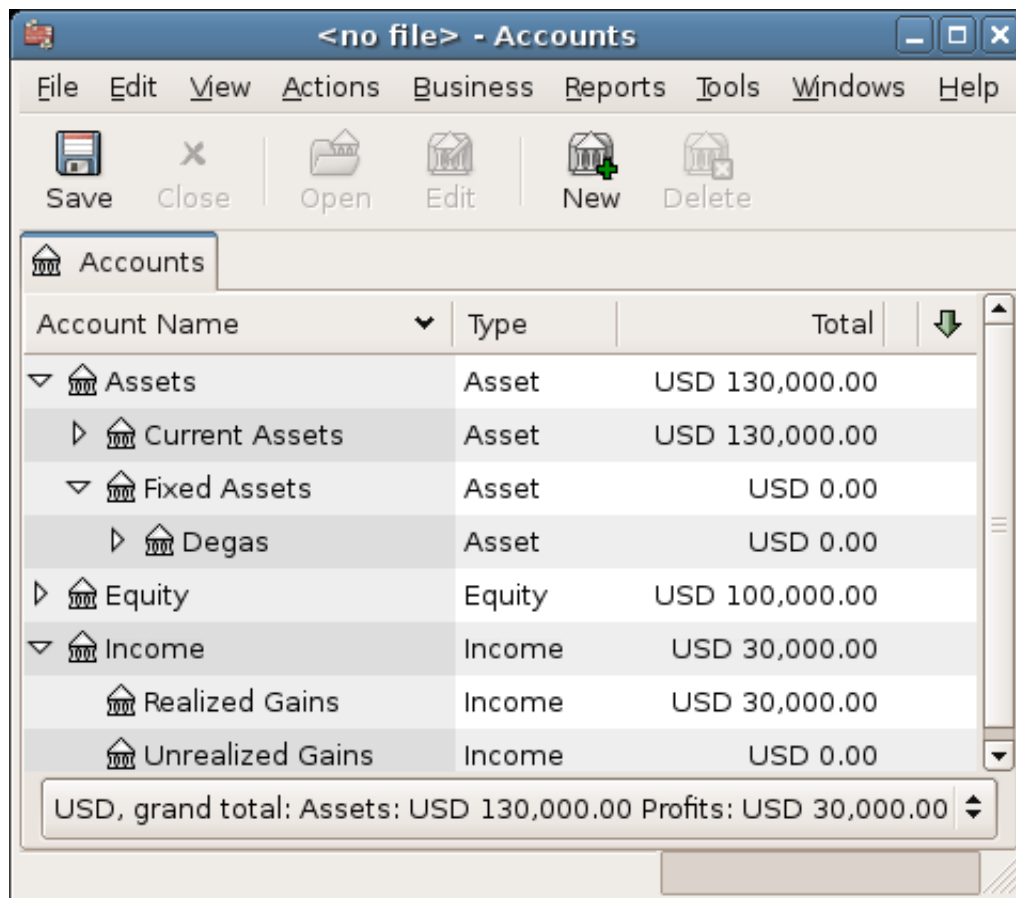
Secondly, you must credit your bank account with the selling price of the painting. This money comes directly from your *Assets:Fixed Assets:Degas* sub-accounts. Transfer the full *Assets:Fixed Assets:Degas:Cost* value into *Assets:Current Assets:Savings Account*, and the full *Assets:Fixed Assets:Degas:Unrealized Gain* into *Assets:Current Assets:Savings Account*.

These transactions should now appear as follows:

Estimation

<i>Account</i>	<i>Transfer to</i>	<i>Transaction Amount</i>	<i>Account Total</i>
Income:Unrealized Gains	Income:Realized Gains	\$30,000	\$0
Assets:Fixed Assets:Degas:Cost	Assets:Current Assets:Savings Account	\$100,000	\$0
Assets:Fixed Assets:Degas:Unrealized Gains	Assets:Current Assets:Savings Account	\$30,000	\$0

This leaves the *Assets:Current Assets:Savings Account* account with a total of \$130000 and *Income:Realized Gains* with a total of \$30000.



## 2. Under estimation of unrealized gain.

You were over-optimistic about the value of the painting. Instead of the \$130000 you thought the painting was worth you are only offered \$120000. But you still decide to sell, because you value \$120000 more than you value the painting. The numbers change a little bit, but not too dramatically.

The transactions should now appear as follows (observe the last transaction which balances the *Unrealized Gains* accounts):

Underestimation



<i>Account</i>	<i>Transfer to</i>	<i>Transaction Amount</i>	<i>Account Total</i>
Income:Unrealized Gains	Income:Realized Gains	\$20,000	\$10,000
Assets:Fixed Assets:De gas:Cost	Assets:Current Assets:Savings Account	\$100,000	\$0
Assets:Fixed Assets:De gas:Unrealized Gains	Assets:Current Assets:Savings Account	\$20,000	\$10,000
Assets:Fixed Assets:De gas:Unrealized Gains	Income:Unrealized Gains	\$10,000	\$0

This leaves the *Assets:Current Assets:Savings Account* account with a total of \$120,000 and *Income:Realized Gains* with a total of \$20,000.

### 3. Over estimation of unrealized gain.

You manage to sell your painting for more than you thought in your wildest dreams (\$150,000). The extra value is, again, recorded as a gain, i.e. an income.

The transactions should now appear as follows (observe the last transaction which balances the Unrealized Gains accounts):

Overestimation

<i>Account</i>	<i>Transfer to</i>	<i>Transaction Amount</i>	<i>Account Total</i>
Income:Unrealized Gains	Income:Realized Gains	\$50,000	\$-20,000
Assets:Fixed Assets:De gas:Cost	Assets:Current Assets:Savings Account	\$100,000	\$0
Assets:Fixed Assets:De gas:Unrealized Gains	Assets:Current Assets:Savings Account	\$50,000	\$-20,000
Income:Unrealized Gains	Assets:Fixed Assets:De gas:Unrealized Gains	\$20,000	\$0

This leaves the *Assets:Current Assets:Savings Account* account with a total of \$150,000 and *Income:Realized Gains* with a total of \$50,000.

## 2.12.5 Taxation

Taxation policies vary considerably between countries, so it is virtually impossible to say anything that will be universally useful. However, it is common for income generated by capital gains to not be subject to taxation until the date that the asset is actually sold, and sometimes not even then. North American home owners usually find that when they sell personal residences, capital gains that occur are exempt from taxation. It appears that other countries treat sale of homes differently, taxing people on such gains. German authorities, for example, tax those gains only if you owned the property for less than ten years.

Chris Browne has a story from his professional tax preparation days where a family sold a farm, and expected a considerable tax bill that turned out to be virtually nil due to having owned the property before 1971 (wherein lies a critical “Valuation Day” date in Canada) and due to it being a dairy farm, with some really peculiar resulting deductions. The point of this story is that while the presentation here is fairly simple, taxation often gets terribly complicated...

## 2.13 Multiple Currencies

This chapter will show how to set up GnuCash accounts to use multiple currencies.

### 2.13.1 Basic Concepts

GnuCash supports over a hundred currencies, from the Andorran Franc to the Zimbabwe Dollar. For example, you can have a bank account set up in Euros, and another using Hong Kong Dollars.

Some of the issues which arise when using multiple currencies are, how do you transfer funds between accounts with different currencies? How do you calculate the overall value when you have mixed currency accounts? How do reports deal with mixed currencies?

---

**Note:** An alternative way to manage multiple currency accounts from the one presented in the next sections, is to use the trading accounts capabilities of GnuCash. This feature, which was introduced with GnuCash version 2.3.14, can be enabled by going to the Accounts tab under File > Properties.

For a complete guide on trading accounts, see [this tutorial by Peter Selinger](#).

---

### 2.13.2 Account Setup

Your default account currency is set in the Account tab under Edit > Preferences (GnuCash > Preferences on MacOS).

Similarly, GnuCash offers an option to set your preferred currency for displaying reports (like the balance sheet and income statement). The option is called Default Report Currency, and is in the Reports tab of the GnuCash Preferences screen. You'll want to set both options when you start using GnuCash because if (for example) your accounts are all in Canadian Dollars but the generated reports are all in US Dollars, the reports will just say that there are "no data/transactions (or only zeroes) for the selected time period".

When you create a new account, you have the option to define the commodity in which that account is denominated. For accounts denominated in a currency, you can specify any of the currencies supported by GnuCash by simply selecting it from the currency commodity list. You will notice that the default currency is the currency that is defined for the parent account of the new account.

As an example, let's set up a bank account scenario where you mostly work in US Dollars, but do also have a European bank account using the Euro currency, as well as one bank account in Hong Kong using Hong Kong Dollars. So, set up 3 bank accounts, one denominated in US Dollars, one using Euros, and the third in Hong Kong Dollars. One possible account structure for this would be:

```
-Assets (USD)
  -Current Assets (USD)
    -US Bank (USD)
    -European Bank (EUR)
    -HK Bank (HKD)
  -Equity (USD)
    -Opening Balances (USD)
      -USD (USD)
      -EUR (EUR)
      -HKD (HKD)
```

---

**Note:** The currency of each account is shown in parenthesis.

---

Since in this example you mostly work in USD, all of the parent accounts are set to USD. Of course, if you mostly work in Euros, you could change the currency of these parent accounts to EUR. The totals shown in the account tree window will always be converted to the currency of each particular account. Notice, we also set up 3 Starting Balances equity accounts, used to initially populate the 3 banks.

**Note:** You could also set up just a single Starting Balance account and use a currency transfer to populate the “different currency” accounts. However, this is more advanced option, which is explained in a later section (*Purchase of an Asset with Foreign Currency*).

Below you see the result of this example, in which you start with USD 10,000, EUR 10,000 as well as HKD 10,000 in the three bank accounts. Notice that the total of the parent accounts only shows the value of the currency of sub-accounts with matching currencies. In other words, the Total Assets and Total Equity values only reflect USD amounts, because GnuCash has no way of evaluating the value of EUR or HKD yet. Once you set up exchange rates between the currencies, the parent accounts will calculate the converted value of all sub-accounts. See the later section (*Recording and Updating Currency Exchange Rates*) on ways to do this.

Account Name	Type	Total	Total (USD)
Assets	Asset	\$10,000.00	\$10,000.00
Current Assets	Asset	\$10,000.00	\$10,000.00
European Bank	Bank	EUR 10,000.00	\$0.00
Hong Kong Bank	Bank	HKD 10,000.00	\$0.00
US Bank	Bank	\$10,000.00	\$10,000.00
Equity	Equity	\$10,000.00	\$10,000.00
Opening Balances	Equity	\$10,000.00	\$10,000.00
EUR	Equity	EUR 10,000.00	\$0.00
HKD	Equity	HKD 10,000.00	\$0.00
USD	Equity	\$10,000.00	\$10,000.00
\$, Grand Total:		Assets: \$10,000.00	Profits: \$0.00

Notice that the “Total (Report)” column is being displayed. This is configured in the column header row, select Arrow down and select “Total(USD)”.

## User-Defined Currencies

Usually when we talk about currencies, we mean government-backed currencies (or more precisely, currencies defined in ISO 4217). GnuCash does not allow you to create your own currencies. If you want to track non-ISO currencies, you can use either of two workarounds, depending on which fits your needs better.

The **first method** is to treat these as if they were a security—that is, like a stock or mutual fund. The second method is to use one of the “dummy” currencies for them.

Let’s say for example that you want to track loyalty points you’ve earned by buying from a certain group of businesses. The account which tracks your loyalty points will be *Assets:Other:LoyaltyGroupRewardMiles*.

In the first method, you define a new security, of type FUND, called RewardMiles. This is pretty straightforward—when you create the new *LoyaltyGroupRewardMiles* account, just set the account type to Stock or Mutual

Fund, click the Select... button next to the Security/currency: box, and click New to define a new security of type FUND.

This is not really what the stock and mutual fund account types are meant for, but GnuCash will allow it. The downside is that you'll have to enter a "price" for every transaction involving this account, because GnuCash needs the prices to figure out the monetary value of the points and treat them as one of your assets.

In the **second method**, you use one of the dummy currencies to track the loyalty points. These currencies are "XTS (Code for testing purposes)" and "XXX (No currency)". If you use one of these for your LoyaltyGroupRewardMiles account, you can enter transactions into the account without having to enter share prices for every transaction. And, you can keep using the same two dummy currencies to track all sorts of amounts—vacation dollars earned and used so far this year, vacation hours earned and used, health insurance benefits allowance used and remaining, and so on.

The drawback with this second method is that you cannot define exchange rates for the dummy currencies to convert them to ISO currencies. If you want to do that, you really should use the first method.

### 2.13.3 Recording and Updating Currency Exchange Rates

GnuCash allows you to update the Currency Exchange Rates both manually and automatically. This process is essentially the same as setting share prices for investments (see *Setting Share Price*). In the following two sections we will work through both methods.

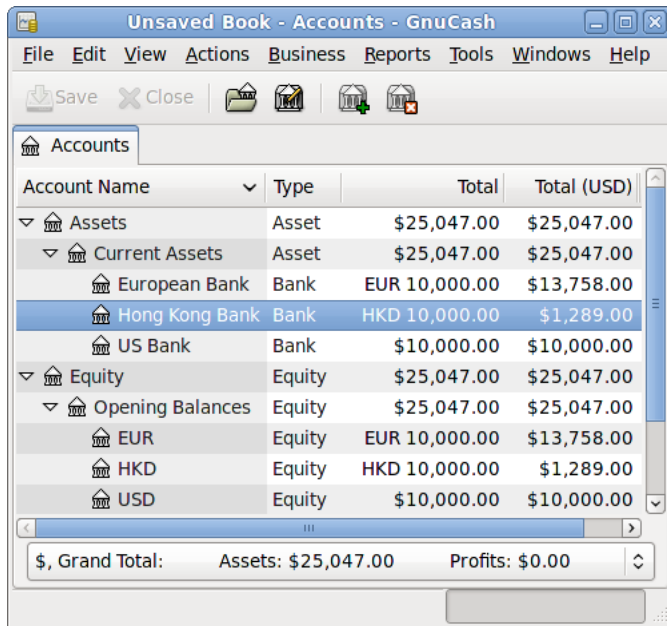
Before we start, let's have a quick look at the Chart of Accounts

Account Name	Type	Total	Total (USD)
Assets	Asset	\$10,000.00	\$10,000.00
Current Assets	Asset	\$10,000.00	\$10,000.00
European Bank	Bank	EUR 10,000.00	\$0.00
Hong Kong Bank	Bank	HKD 10,000.00	\$0.00
US Bank	Bank	\$10,000.00	\$10,000.00
Equity	Equity	\$10,000.00	\$10,000.00
Opening Balances	Equity	\$10,000.00	\$10,000.00
EUR	Equity	EUR 10,000.00	\$0.00
HKD	Equity	HKD 10,000.00	\$0.00
USD	Equity	\$10,000.00	\$10,000.00
\$, Grand Total:		Assets: \$10,000.00	Profits: \$0.00

As you see, the overall balances do not yet reflect any value for EUR or HKD holdings. Adding currency exchange rates will fix this.

#### Manually Updating Exchange Rates

Open the Price Editor by going to Tools > Price Editor.

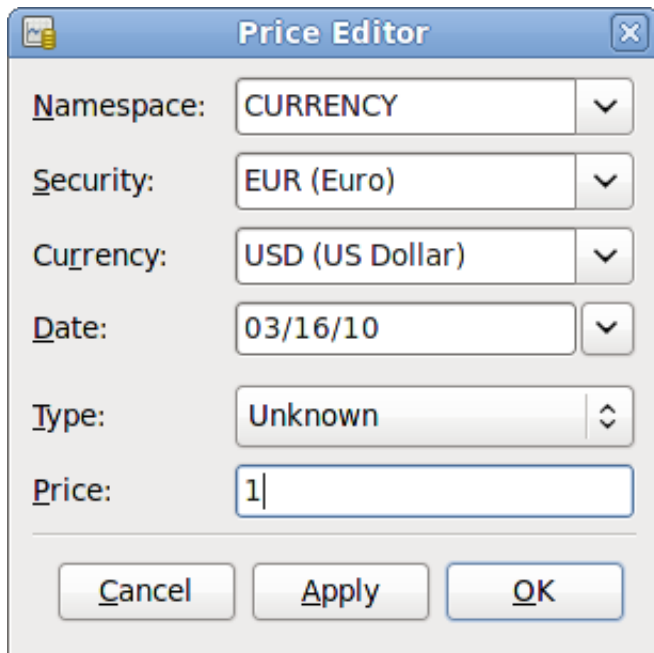


The screenshot shows the 'Accounts' window in GnuCash. It displays a tree view of accounts on the left and a summary table at the bottom. The summary table shows the following data:

Account Name	Type	Total	Total (USD)
Assets	Asset	\$25,047.00	\$25,047.00
Current Assets	Asset	\$25,047.00	\$25,047.00
European Bank	Bank	EUR 10,000.00	\$13,758.00
Hong Kong Bank	Bank	HKD 10,000.00	\$1,289.00
US Bank	Bank	\$10,000.00	\$10,000.00
Equity	Equity	\$25,047.00	\$25,047.00
Opening Balances	Equity	\$25,047.00	\$25,047.00
EUR	Equity	EUR 10,000.00	\$13,758.00
HKD	Equity	HKD 10,000.00	\$1,289.00
USD	Equity	\$10,000.00	\$10,000.00

At the bottom of the window, there is a summary bar showing: \$, Grand Total: Assets: \$25,047.00 Profits: \$0.00

Click on the Add button to add a new currency exchange rate. A window will appear in which you can set up a new exchange rate. This window should appear like this:

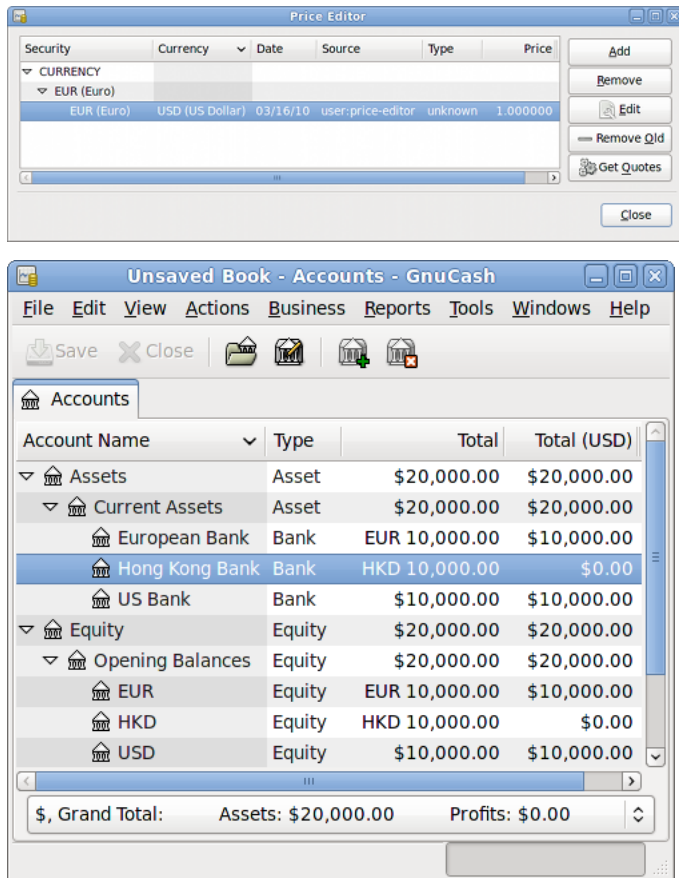


The screenshot shows the 'Price Editor' dialog box. It contains the following fields and values:

- Namespace: CURRENCY
- Security: EUR (Euro)
- Currency: USD (US Dollar)
- Date: 03/16/10
- Type: Unknown
- Price: 1

At the bottom of the dialog box, there are three buttons: Cancel, Apply, and OK.

Set the Namespace to Currency and the Security to EUR (Euro). Then set the exchange rate between the selected security and the selected currency. The price box defines how many units of currency are required to purchase one unit of the security. In this case, how many dollars it will take to purchase one Euro. In this example, you will set the exchange rate to 1 EUR for 1 USD.

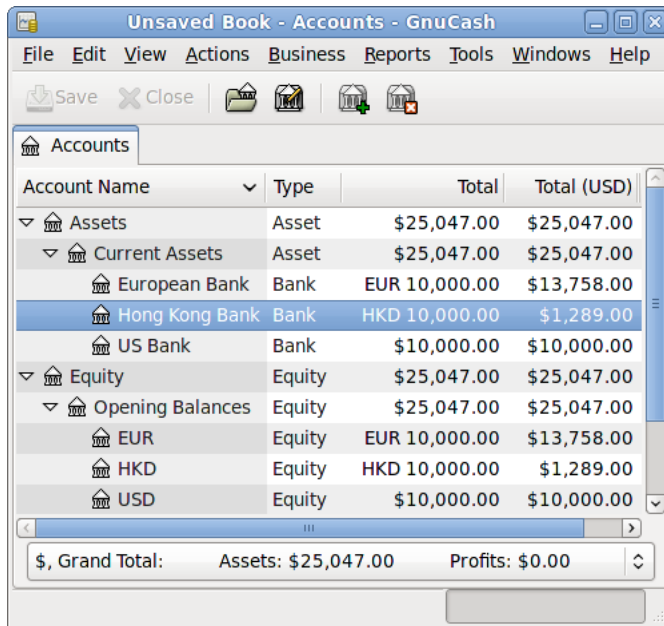


Observe that since you have no exchange rate for HKD, GnuCash doesn't convert the HKD accounts to USD. This will be added in the next section.

### Automatic Updating Exchange Rates (How-To)

In the previous section you saw how to manually define a new currency exchange rate, but GnuCash includes an automatic price update feature, which will now be described.

Open the Price Editor by going to Tools > Price Editor.



The screenshot shows the 'Accounts' window in GnuCash. It features a tree view on the left and a summary table at the bottom. The tree view shows the following structure:

- Assets
  - Current Assets
    - European Bank (Bank, EUR 10,000.00, \$13,758.00)
    - Hong Kong Bank (Bank, HKD 10,000.00, \$1,289.00)
    - US Bank (Bank, \$10,000.00, \$10,000.00)
- Equity
  - Opening Balances
    - EUR (Equity, EUR 10,000.00, \$13,758.00)
    - HKD (Equity, HKD 10,000.00, \$1,289.00)
    - USD (Equity, \$10,000.00, \$10,000.00)

The summary table at the bottom shows:

Account Name	Type	Total	Total (USD)
\$, Grand Total:		Assets: \$25,047.00	Profits: \$0.00

Click on the Get Quotes button to automatically load the various exchange rates you need.

**Note:** If the Get Quotes button is disabled, that means that the Perl module `Finance::Quote` is not installed. For information on how to install it, please see [Configuring for Automatic Retrieval of Quotes](#)

Price Editor window Price Editor Window after we obtained online quotes.

GnuCash downloads exchange rates for all currencies that are in use in your various accounts. This will happen every time you click on Get Quotes or request GnuCash to download quotes as per [Configuring for Automatic Retrieval of Quotes](#)

Now when you check the main Chart of Accounts you will see that GnuCash has automatically converted the HKD amount to USD amount on the parent accounts that are in USD, as well as on the Total (USD) column. Also the Euro accounts have been updated with the latest exchange rate.

The screenshot shows the 'Accounts' window in GnuCash. It features a tree view on the left and a summary table on the right. The tree view shows the following structure:

- Assets
  - Current Assets
    - European Bank (Bank, EUR 10,000.00, \$13,758.00)
    - Hong Kong Bank (Bank, HKD 10,000.00, \$1,289.00)
    - US Bank (Bank, \$10,000.00, \$10,000.00)
  - Equity
    - Opening Balances
      - EUR (Equity, EUR 10,000.00, \$13,758.00)
      - HKD (Equity, HKD 10,000.00, \$1,289.00)
      - USD (Equity, \$10,000.00, \$10,000.00)

The summary table at the bottom shows:

Account Name	Type	Total	Total (USD)
Assets	Asset	\$25,047.00	\$25,047.00
Current Assets	Asset	\$25,047.00	\$25,047.00
European Bank	Bank	EUR 10,000.00	\$13,758.00
Hong Kong Bank	Bank	HKD 10,000.00	\$1,289.00
US Bank	Bank	\$10,000.00	\$10,000.00
Equity	Equity	\$25,047.00	\$25,047.00
Opening Balances	Equity	\$25,047.00	\$25,047.00
EUR	Equity	EUR 10,000.00	\$13,758.00
HKD	Equity	HKD 10,000.00	\$1,289.00
USD	Equity	\$10,000.00	\$10,000.00

At the bottom of the window, a summary bar shows: \$, Grand Total: Assets: \$25,047.00 Profits: \$0.00

### Disabling Exchange Rate Retrieval

Whenever you create an account that uses a non-default currency, exchange rate retrieval will be automatically enabled for that currency. However, if you later delete that account, GnuCash will not automatically disable exchange rate retrieval for that currency.

If you have deleted the last account for a particular currency, and you do not wish to retrieve exchange rates for that currency anymore, do the following:

- Open the Securities window by selecting Tools > Security Editor.
- Make sure the Show National Currencies box is selected.
- Expand the CURRENCY row.
- Double click on the currency for which you want to disable exchange rate retrieval.
- Deselect the Get Online Quotes box and click OK.

### 2.13.4 Recording Purchases in a Foreign Currency

Purchases in a foreign currency can be managed in two different ways.

- 1) Use GnuCash's built-in currency exchange functions when you do your transactions. This is mainly used for one-time transactions, and nothing which happens regularly.
- 2) Use separate accounts to track transactions, where all involved accounts use the same currency. This is the recommended method, since it allows much better tracking and follow up. In this way, you do one currency exchange transaction, and after that you do normal transactions in the native currency.

The rest of this section will explain more based upon option 2).

#### Purchase of an Asset with Foreign Currency

You are using USD as your default currency. But, you decide to purchase a boat in Jamaica. To do this, you opened a bank account in Jamaica, moved some money from the US, and then purchased your boat.



To record this in GnuCash we use the following basic account structure:

```
-Assets (USD)
  -Current Assets (USD)
    -US Bank (USD)
    -Jamaican Bank (JMD)
  -Fixed Assets (USD)
    -Boat (JMD)
-Equity (USD)
  -Opening Balances (USD)
    -USD (USD)
```

**Note:** The currency of each account is shown in parenthesis.

First you need to transfer some money (\$10,000) to Jamaica, and you use your US bank account (with a balance of \$100,000) for that. The bank gives you an exchange rate of USD 1 = JMD 64, but charges you USD 150 to transfer the money.

#	Memo	Account	R	Deposit	Withdrawal
2006-01-01	Opening Balance	Equity:Opening Balances:USD	n	100,000.00	100,000.00
2006-02-01	Transfere money to jamaica			10,000.00	90,000.00
		Expenses:Service Charge	n	150.00	
		Assets:Current Assets:jamaican Bank	n	9,850.00	
		Assets:Current Assets:US Bank	n		10,000.00

Present:USD 90,000.00 Future:USD 90,000.00 Cleared:USD 0.00 Reconciled:USD 0.00 Projected Minimum:USD 90,000.00

Select the Jamaica transaction line (\$9,850.00), right click and select Edit Exchange Rate

**Transfer Funds**

**Basic Information**

Amount: 9,850.00

Date: 2006-02-01

Num:

Description: Transfere money to jamaica

Memo:

**Transfer From**      **Transfer To**

Currency: USD (US Dollar)      Currency: JMD (Jamaican Dollar)

**Currency Transfer**

☒ Exchange Rate: 64.00      1 USD = 64.000000 JMD

☐ To Amount: 630,400.00      1 JMD = 0.015625 USD

As Exchange Rate, you enter 1 USD = 64 JMD, since this is the rate your bank gave. Press ok in the Transfer Funds (Edit Exchange Rate) window, and then save this split transaction. Below is how it now looks in the main Chart of Accounts.

The screenshot shows the 'Currency Exchange Updates - Accounts' window. The 'Accounts' tab is selected, displaying a tree view of the chart of accounts. The table below represents the data shown in the window.

Account Name	Type	Total	Total (USD)
Assets	Asset	USD 99,850.00	USD 99,850.00
Current Assets	Asset	USD 99,850.00	USD 99,850.00
Jamaican Bank	Bank	JMD 630,400.00	USD 9,850.00
US Bank	Bank	USD 90,000.00	USD 90,000.00
Fixed Assets	Asset	USD 0.00	USD 0.00
Boat	Asset	JMD 0.00	USD 0.00
Equity	Equity	USD 100,000.00	USD 100,000.00
Opening Balances	Equity	USD 100,000.00	USD 100,000.00
USD	Equity	USD 100,000.00	USD 100,000.00
Expenses	Expens	USD 150.00	USD 150.00
Service Charge	Expens	USD 150.00	USD 150.00

At the bottom of the window, a summary bar shows: USD, grand total: Assets: USD 99,850.00 Profits: -USD 150.00

You choose to buy a boat for JMD 509,000. To record this transaction in GnuCash, you will need to enter a simple transaction in *Assets:Current Assets:Jamaican Bank* withdrawing JMD 509,000 and transferring it to *Assets:Fixed Assets:Boat*

The screenshot shows the same 'Currency Exchange Updates - Accounts' window after recording the transaction. The 'Jamaican Bank' account has been reduced by JMD 509,000, and the 'Boat' account has been increased by the same amount. The table below represents the updated data.

Account Name	Type	Total	Total (USD)
Assets	Asset	USD 99,850.00	USD 99,850.00
Current Assets	Asset	USD 91,896.88	USD 91,896.88
Jamaican Bank	Bank	JMD 121,400.00	USD 1,896.88
US Bank	Bank	USD 90,000.00	USD 90,000.00
Fixed Assets	Asset	USD 7,953.12	USD 7,953.12
Boat	Asset	JMD 509,000.00	USD 7,953.12
Equity	Equity	USD 100,000.00	USD 100,000.00
Opening Balances	Equity	USD 100,000.00	USD 100,000.00
USD	Equity	USD 100,000.00	USD 100,000.00
Expenses	Expens	USD 150.00	USD 150.00
Service Charge	Expens	USD 150.00	USD 150.00

The summary bar at the bottom remains the same: USD, grand total: Assets: USD 99,850.00 Profits: -USD 150.00

The Chart of Accounts now reflects that your bank account has been reduced by JMD 509,000, and that your Fixed Assets boat account has been increased by the same amount. If you also have turned on the CoA (Column Choice) “Total (USD)” you will see the corresponding value in USD. The USD value will always reflect the latest currency exchange rate you have either automatically or manually retrieved.

## Purchasing Foreign Stocks

This example will show how to purchase stocks that are priced in a currency other than your primary currency.

Assume that you live in New York and therefore you have set the default currency to USD. You decide to purchase a stock traded in Hong Kong that is priced in HKD. You would also like to be able to track the various income and expense amounts per stock and broker.

You decide to purchase stock in the Beijing Airport (Hong Kong). The ticker for this stock is 0694.HK on Yahoo! Since you wanted to track all various income and expense amounts, here is the necessary account structure:

```
Assets:Investments:Brokerage Accounts:Boom:0694.HK (0694.HK)
Assets:Investments:Brokerage Accounts:Boom:Bank (HKD)
Equity:Opening Balances:HKD (HKD)
Expenses:Commissions:Boom.0694.HK (HKD)
Income:Investments:Dividend:Boom:0694.HK (HKD)
```

The Chart of Accounts looks like this after creating all the needed accounts:

Account Name	Type	Commodity	Total	Total (USD)
Assets	Asset	US Dollar	USD 0.00	USD 0.00
Investments	Asset	US Dollar	USD 0.00	USD 0.00
Brokerage Account	Bank	US Dollar	USD 0.00	USD 0.00
Boom	Bank	Hong Kong Dollar	HKD 0.00	USD 0.00
0694.HK	Stock	Beijing Airport	0 0694.HK	USD 0.00
Bank	Bank	Hong Kong Dollar	HKD 0.00	USD 0.00
Equity	Equity	US Dollar	USD 0.00	USD 0.00
Opening Balances	Equity	US Dollar	USD 0.00	USD 0.00
HKD	Equity	Hong Kong Dollar	HKD 0.00	USD 0.00
Expenses	Expense	US Dollar	USD 0.00	USD 0.00
Commissions	Expense	US Dollar	USD 0.00	USD 0.00
Boom	Expense	Hong Kong Dollar	HKD 0.00	USD 0.00
0694.HK	Expense	Hong Kong Dollar	HKD 0.00	USD 0.00
Income	Income	US Dollar	USD 0.00	USD 0.00
Investments	Income	US Dollar	USD 0.00	USD 0.00
Dividend	Income	US Dollar	USD 0.00	USD 0.00
Boom	Income	Hong Kong Dollar	HKD 0.00	USD 0.00
USD, grand total:			Assets: USD 0.00	Profits: USD 0.00

The stock definition can be seen in the Security Editor. (Tools > Security Editor)

Namespace	Symbol	Name	Source	Timezone
CURRENCY				
HK				
0694.HK		Beijing Airport	yahoo_asia	Europe/London

☒ Show National Currencies

Buttons: Add, Remove, Edit, Close

If you have not moved money (HKD 50,000) into the brokerage cash account (*Assets:Investments:Brokerage Account:Boom:Bank*), do so now, either using the Equity (HKD) account, or an existing bank account (Currency Transfer).

There are two ways to enter the actual purchase transaction: you can enter it from the cash account (shown below), or you can enter it from the stock account. *If entered from the stock account, the stock is assumed to be priced in the currency of the parent account.*

Let's assume that the stock price is HKD 3 per share. To record the purchase, open the brokerage's HKD cash account (*Assets:Investments:Brokerage Account:Boom:Bank*), and enter the following:

### Buy Stocks

Assets:Investments:Brokerage Account:Boom:Bank	Withdrawal	50,000
Expenses:Investments:Commission:Boom_HKD	Deposit	500
Assets:Investments:Brokerage Account:Boom:0694	Deposit	49,500 (16,500 shares)

If the exchange rate dialog box does not appear automatically, right-click on the stock row, and select Edit Exchange Rate. Enter the number of shares (16,500) as the To Amount.

**Transfer Funds**

**Basic Information**

Amount: 49,500.00

Date: 2006-01-02

Num:

Description: Purchase Beijing Airport

Memo:

**Transfer From**      **Transfer To**

Currency: HKD (Hong Kong Dollar)      Currency: 0694.HK (Beijing Airport)

**Currency Transfer**

☐ Exchange Rate: 0.333333      1 HKD = 0.333333 0694.HK

☒ To Amount: 16,500.00      1 0694.HK = 3.000000 HKD

When you return to the Chart of Accounts, you will see the purchased shares reflected in the stock account's total.

Account Name	Type	Commodity	Total	Total (USD)
Assets	Asset	US Dollar	USD 0.00	USD 0.00
Investments	Asset	US Dollar	USD 0.00	USD 0.00
Brokerage Account	Bank	US Dollar	USD 0.00	USD 0.00
Boom	Bank	Hong Kong Dollar	HKD 0.00	USD 0.00
0694.HK	Stock	Beijing Airport	16,500 0694.HK	USD 0.00
Bank	Bank	Hong Kong Dollar	HKD 0.00	USD 0.00
Equity	Equity	US Dollar	USD 0.00	USD 0.00
Opening Balances	Equity	US Dollar	USD 0.00	USD 0.00
HKD	Equity	Hong Kong Dollar	HKD 50,000.00	USD 0.00
Expenses	Expense	US Dollar	USD 0.00	USD 0.00
Commissions	Expense	US Dollar	USD 0.00	USD 0.00
Boom	Expense	Hong Kong Dollar	HKD 500.00	USD 0.00
Income	Income	US Dollar	USD 0.00	USD 0.00
USD, grand total:			Assets: USD 0.00	Profits: USD 0.00

However, as you can see, the USD totals may be zero if GnuCash doesn't have an exchange rate between USD and HKD. To fix this, go to Tools > Price Editor and click the Get Quotes button to automatically retrieve the exchange rates you need.

**Note:** To reiterate, this example shows how stock can be purchased in any currency by entering the transaction in the register of the cash account used to make payment. It is also possible to enter the purchase in the stock account's register, but be aware that the stock is assumed to be priced in the currency of the stock account's parent.

In this example, the stock account's parent (*Assets:Investments:Brokerage Account:Boom*) is denominated in HKD. Since this is same currency as the stock price, the purchase can be safely entered in the stock account's register.

## 2.13.5 Tracking Currency Investments

Currency investment is when you decide to invest in a different country's currency, and hope that it will rise in value relative your own currency.

When you enter these transactions into GnuCash, you will have to decide on how much detail you would like to have.

If you are not interested in detail at all, a very simple account structure would suffice:

```
Assets:Investments:Currency:Bank (USD)
Assets:Investments:Currency:XXX (XXX)
```

You would simply enter transfers between the two accounts, noting exchange rates as you went.

But, if you do want to be able to track capital gains or losses, as well as any fees, you do need a more complex account structure, such as:

```
Assets:Investments:Currency:Bank (USD)
Assets:Investments:Currency:Currency Bank:XXX (XXX)
Expenses:Investments:Currency:Currency Bank:XXX (XXX)
Income:Investments:Currency Bank:Capital Gains:XXX (XXX)
```

## Purchasing Currency

When purchasing another currency, you will buy a certain number of units of foreign currency with your own currency, at a particular rate. For example, you might buy USD 10,000 worth of Andorran Francs, at 5 Francs to the dollar, with a transaction fee of \$150.

### Buy Currency

Assets:Investments:Currency:Bank	Withdrawal	10,000
Expenses:Investments:Currency:Currency Bank:ADF	Deposit	150
Assets:Investments:Currency:ADF	Deposit	49,250

The Exchange Rate window should pop up when you leave the last row in the split above (Currency Transaction). If this window does not pop up, right click on the row or select Actions, and select Edit Exchange Rate. In the Exchange Rate window you specify the exchange rate you got from the bank.

## Selling a currency investment

Entering a currency sale is done in the same way as a currency buy except that you are now transferring money from the Currency account to your Savings account (very similar to *Selling Shares*).

The proper recording of the currency sale *must* account for realized gains or losses. This can be done using a split transaction. In the split transaction, you must account for the profit (or loss) as coming from an *Income:Capital Gains* account (or *Expenses:Capital Loss*). To balance this income, you will need to enter the Currency asset twice in the split—once to record the actual sale (using the correct amount and correct exchange rate), and once to balance the income profit (setting the amount to 0).

In short, a selling Currency transaction should look something like below, seen again from the *Assets:Investments:Currency:Bank*.

Table 19: Selling a currency with a Split Transaction Scheme

Account	Deposit	Withdrawal
Assets:Investments:Currency:Bank	Sold Amount - Exchange Fee	
Expenses:Investments:Currency:Currency Bank:XXX	Exchange Fee	
Assets:Investments:Currency:XXX		Sold Amount
Income:Investments:Currency Bank:Capital Gains:XXX	[LOSS]	PROFIT
Assets:Investments:Currency:XXX	PROFIT (with To Amount = 0)	[LOSS (with To Amount = 0)]

## 2.13.6 Reconciling Statements in a Foreign Currency

Reconciling foreign statement are done in the same manner as when you reconcile your local bank statement. If you have created a Chart of Accounts structure which allows you to have the same currency per account as your statement, it is actually exactly the same as reconciling your local bank statement.

If you have different currencies you might have to manually convert the amounts from one currency to another while you reconcile the accounts.

- *Managing Business Finances*

## 2.14 Business Features

### 2.14.1 Introduction to Business Features

The accounting needs of a business are quite different from that of a person. Businesses have customers that owe money, vendors which are owed money, employee payroll, more complex tax laws, etc. GnuCash offers business oriented features to facilitate these needs.

*Accounts Receivable* (A/R) are used by businesses to record sales for which they are not immediately paid. This is represented on the balance sheet as an asset, because the expectation is that you will receive payment soon.

*Accounts Payable* (A/P) record bills that businesses have received, but may not pay until later. This is represented on the balance sheet as a liability because you will have to pay for them.

A/R and A/P accounts are used primarily when you have a lot of bills and receipts flowing in and out, and do not want to lose track of them just because you do not pay or get paid right away. For most home users, A/R and A/P are too complicated to be worthwhile.

### 2.14.2 Business Setup

To set up GnuCash to handle accounts receivable or accounts payable for a company, these preliminary steps must be done.

- Build an appropriate Account Hierarchy.
- Set up Sales Tax Tables.
- Enter the company information in GnuCash.
- Set Business Preferences.
- Set up Billing Terms.

#### Account Setup

There are many different ways to set up a business account hierarchy. You can start with the Business Accounts setup which is available from the New Account Hierarchy assistant, or you could build one manually. To access the prebuilt Business Accounts, start GnuCash and click on File > New File and proceed until you see the list of available accounts, select Business Accounts.

The prebuilt Business Account hierarchy will not meet your needs exactly. You will need make adjustments to the hierarchy so that it will meet your particular needs. It should be close enough, however, that it is recommended you begin with it.

To use GnuCash's integrated accounts receivable system, you must first set up a special account (usually a sub-account under Assets) to hold transactions for receivables. This account must be defined with account type *A/Receivable*. GnuCash will use this account to place transactions related to the integrated accounts receivable system.

To use GnuCash's integrated accounts payable system, you must first set up an account (usually a sub-account under Liabilities) to hold transactions for payables. This account must be defined with account type *A/Payable*. GnuCash will use this account to place transactions related to the integrated accounts payable system.

Basic A/R **and** A/P Account Hierarchy:

```
-Assets
  -Accounts Receivable
  -Checking
```

(continues on next page)

(continued from previous page)

```
-Expenses
  ... (as required)
-Income
  -Sales
-Liabilities
  -Accounts Payable
  -Tax
    -Tax on Purchases
    -Tax on Sales
```

You need to add additional accounts to this hierarchy for it to be useful.

---

**Note:** You do not need to create an individual A/R account for each customer. GnuCash keeps track of customers internally and provides per-customer reports based on the internal tracking. The same applies to A/P and vendors.

---

---

**Note:** If you deal with customers in more than one currency you will need a separate Accounts Receivable account *for each currency*.

If you deal with vendors in more than one currency you will need a separate Accounts Payable account *for each currency*.

---

---

**Note:** Transactions involving an Accounts Receivable or Accounts Payable account should not be added, changed or deleted in any way other than by using Post/Unpost Bill/Invoice/Voucher or Process Payment!

---

### Setting up Sales Tax Tables

Sales Tax Tables can be used to determine the tax for customer invoices (or vendor bills).

A tax table entry can be assigned to an invoice line or bill line.

Set up distinct tax tables for customers and vendors.

The default invoice tax table entry can be assigned to each customer and the default bill tax table entry can be assigned to each vendor.

The default tax table entry for new customers or new vendors can be specified in the *Book Options* window which can be accessed by File > Properties > Business tab.

Sales Tax Tables are maintained using the *Sales Tax Table* editor which is accessed via menu Business > Sales Tax Table.

- Name This is the tax table name.
- Type Either Percent % or Value \$.
- Value This is the percentage or value depending on Type.
- Account This is the account to which tax will be posted. For tax collected from customers, this should probably be a Liability account as it must be paid to the government. For tax paid to vendors, if tax laws allow tax paid to vendors to offset tax collected from customers, this should probably also be a Liability account (even though it will usually have a debit balance) so that the net tax owed to the government can be easily observed.

If you set up Tax on Purchases and Tax on Sales as subaccounts of Liabilities:Tax then the net tax will be rolled up and can be seen in the GnuCash Accounts tab.



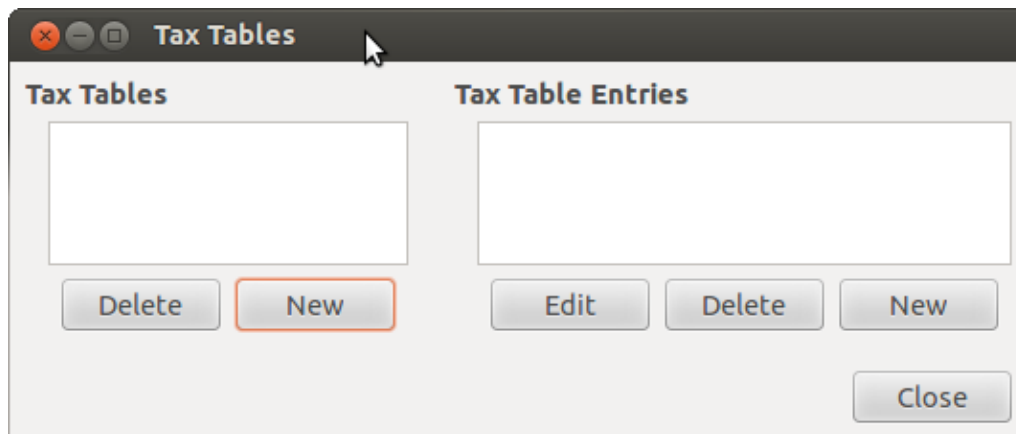


Fig. 19: Sales Tax Tables Editor

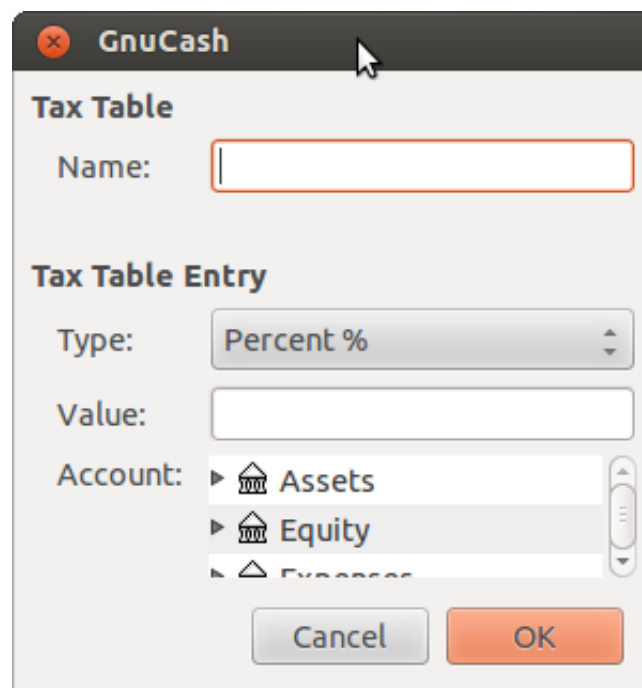


Fig. 20: New Sales Tax Table Entry

*If unsure about tax law requirements, get professional advice.*

The following charts illustrate sample tax tables and may be used as starting points to determine the setup appropriate for a particular jurisdiction.

Belgium, 20% / 5% / 0% UK etc.) (2017)

Tax Table	Tax Table Entries [Asset/Liability]	Percentage or Amount
Standard VAT Sales	VAT:Sales:Standard [L]	21%
Reduced VAT Sales	VAT:Sales:Reduced [L]	6%
Zero-Rated VAT Sales	VAT:Sales:Zero [L]	0%
EC Sales	VAT:Sales:EC [L]	21%
VAT:Sales:Reverse EC [L]	-21%	
Standard VAT Purchases	VAT:Purchases:Standard [A]	21%
Reduced VAT Purchases	VAT:Purchases:Reduced [A]	6%
Zero-Rated VAT Purchases	VAT:Purchases:Zero [A]	0%

Table 20: Sample Tax Table Entries for Australia (2017)

Tax Table	Tax Table Entries [Asset/Liability]	Percentage or Amount
Standard GST Sales	GST:Sales:Standard [L]	10%
GST-free Sales	GST:Sales:Zero [L]	0%
Standard GST Purchases	GST:Purchases:Standard [A]	10%
GST-free Purchases	GST:Purchases:Zero [A]	0%

Table 21: Sample Tax Table Entries for Cook County, Illinois (2017)

Tax Table	Tax Table Entries [Asset/Liability]	Percentage or Amount
Chicago Sales Taxes	Taxes:Sales:State [L]	6.25%
Taxes:Sales:City [L]	1.25%	
Taxes:Sales:County [L]	1.75%	
Taxes:Sales:Region [L]	1%	

## Enter Company Information

After you have built the account structure and defined your tax tables, designate your company in the GnuCash file. To do this, select the Business tab in the Book Options window, which can be accessed from File > Properties.

Here you can:

- Enter the name of your company along with contact information such as your phone number, fax number, e-mail address and website URL.
- Enter your company's tax payer id number in the Company ID field.
- Select default tax tables applicable to your most common customers and vendors.

## Setting Business Preferences

Set options on the Business tab of the GnuCash preferences, which is accessed via Edit > Preferences (GnuCash > Preferences on MacOS). See Help manual chapter 10.3.3 Business Book Options Tab.

The screenshot shows the 'Book Options' dialog box with the 'Business' tab selected. The dialog contains several input fields for company information, tax table selections, and date format settings. The 'Business' tab is highlighted with a red border and a mouse cursor. The 'Fancy Date Format' section includes radio buttons for 'Number', 'Abbreviation', and 'Name', with 'Number' selected. There is also a checkbox for 'Include Century' which is checked. The 'Format' field shows '%d/%m/%Y' and a 'Sample' date is displayed as '29/08/2015'. At the bottom, there are buttons for 'Help', 'Cancel', 'Apply', and 'OK', along with a 'Reset defaults' button.

**Book Options**

Accounts Budgeting **Business** Counters

Company Name:

Company Address:

Company Contact Person:

Company Phone Number:

Company Fax Number:

Company Email Address:

Company Website URL:

Company ID:

Default Customer TaxTable:

Default Vendor TaxTable:

Fancy Date Format:

Months: ☒ Number ☐ Abbreviation ☐ Name

Years: ☒ Include Century

Format:

Sample: 29/08/2015

Fig. 21: Entering Company Information

## Setting Billing Terms

Billing Terms can be used to determine the payment due date and be a guide for determining discount for early payment of invoices (or vendor bills).

**Note:** As of GnuCash 2.6.7, Billing Terms are only partially supported. Date due is calculated using the Billing Terms but discount amount is not.

Discount for early invoice payment is not implemented. There are 2 ways this may be done, although neither is recommended, and professional advice should be used to confirm that regulations are being complied with:

- After creating and posting a payment which pays the invoice in full, manually edit the payment transaction (usually strongly discouraged) and split the payment to reduce it by the amount of the discount and create a compensating split in an income (discount) account.
- Alternatively, after creating and posting a payment for the discounted amount, create a credit note for the discount using a specific negative sales income (discount) account for the transfer account.

You can specify the billing terms on each invoice/bill. Invoice billing terms will default from the customer billing terms. Bill billing terms will default from the vendor billing terms.

Billing Terms are maintained using the Billing Terms Editor which is accessed via menu Business > Billing Terms Editor.

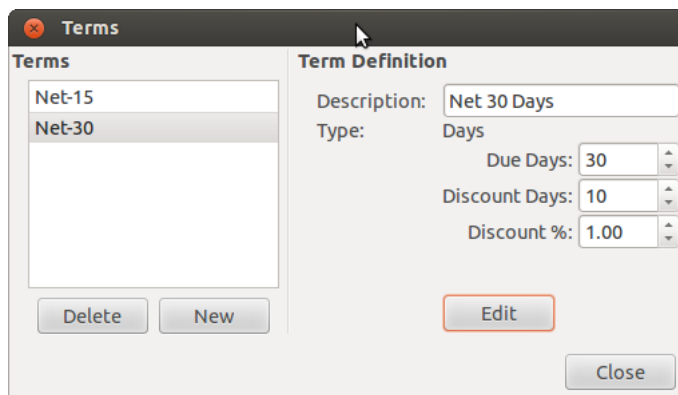


Fig. 22: Billing Terms Editor

- Name The internal name of the billing term. For some examples of billing term names and descriptions see ‘<[https://wiki.gnucash.org/wiki/Payment\\_Terms](https://wiki.gnucash.org/wiki/Payment_Terms)>’.
- Description The description of the billing term, printed on invoices
- There are 2 types of billing terms, with different information to be entered
  - Type Days
    - \* Due Days The invoice or bill is due to be paid within this number of days after the post date
    - \* Discount Days The number of days after the post date during which a discount will be applied for early payment
    - \* Discount % The percentage discount applied for early payment
  - Type Proximo
    - \* Due Day The day of the month bills are due

Fig. 23: New Billing Term

- \* Discount Day The last day of the month for the early payment discount
- \* Discount % The discount percentage applied if paid early
- \* Cutoff Day The cutoff day for applying bills to the next month. After the cutoff, bills are applied to the following month. Negative values count backwards from the end of the month.

### 2.14.3 Accounts Receivable

Accounts Receivable (or A/R) refers to products or services provided by your company for which payment has not yet been received.

#### System Components

Transactions generated by the A/R system are recorded within the Accounts Receivable account. *You should not work directly with this account.* Instead, you will work with the four integrated GnuCash A/R application components available through the Business > Customer sub-menu. These four components are:

- Customers are people or companies to whom you sell products or services on credit.
- Invoices represent the physical invoice you send to a customer to request payment. This invoice contains an itemized list of things you sold.

In addition, GnuCash also has support for Credit Notes which represent the inverse of Invoices. A credit note is usually handed to a customer to correct items that were incorrectly invoiced or returned.

Both document types will be set up using the same menu items. Credit notes were introduced starting with GnuCash stable release 2.6.0.

- Jobs (optional) is where you register Customer Jobs. Jobs are a mechanism by which you can group multiple invoices to a particular customer.

- Process Payments is used to register payments you received from a customer.

## Customers

Customers are people or companies to whom you sell goods or services. They must be registered within the A/R system.

### New

To register a new customer, enter the menu Business > Customer > New Customer. Fill in customer information, such as Company Name, Address, Phone, Fax, etc.

The screenshot shows a window titled "New Customer - ABC Inc ()". It has three tabs: "Customer", "Billing Information", and "Shipping Address". The "Customer" tab is selected. Under the "Identification" section, there is a "Customer Number" field, a "Company Name" field filled with "ABC Inc", and a checked "Active" checkbox. Below this is the "Billing Address" section, which includes fields for "Name" (filled with "Bob McBob"), "Address" (filled with "123 First Ave." and "Somecity, SS 12345"), "Phone" (filled with "515-234-5678"), "Fax" (filled with "515-235-5679"), and "Email" (filled with "abc@abc.com"). At the bottom is a "Notes" section with a text area containing "Bob McBob, Sales Dept.". The window has "Help", "Cancel", and "OK" buttons at the bottom.

Fig. 24: New Customer Registration Window

## Find and Edit

To search for an existing customer, use the Business > Customer > Find Customer window. You select a customer to View/Edit from the results of the search. This window is also used to look up customers when creating invoices and processing payments.

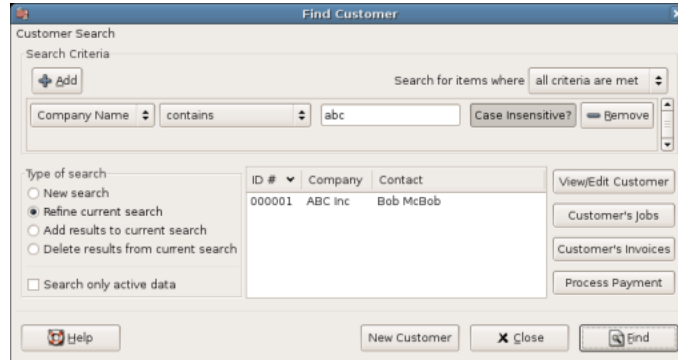


Fig. 25: Find Customer Window

If many customers match the search criteria you provide, the search can be refined by running an additional search within the current results. The current result set is searched when the Refine Current Search radio button is selected. In fact, GnuCash selects this option for you after you run the initial search.

If the customer you are searching for does not match the supplied search criteria, change the search criteria, click the New Search radio button and then the Find button. The relevant step is the New Search selection. If the customer is not in the result of the original search, and you only search within this set, the customer cannot be found, regardless of new search criteria.

---

**Note:** To return a list of all registered active customers, set the search criterion to matches regex, and place a single dot “.” in the text field area. Make sure Search only active data is checked, then click Find. The regular expression “.” means to match anything.

---

## Invoices

An invoice is the paperwork you send to a customer to request payment for products or services rendered. GnuCash can generate and track invoices.

A credit note is the paperwork you send to a customer to correct products or services rendered that were incorrectly invoiced. GnuCash can generate and track credit notes via the same menu entries as invoices.

---

**Note:** This section applies to both invoices and credit notes. In order to avoid repeating everything twice and to keep the text easier to read it will refer only to invoices. You can apply it equally to credit notes. Only where the behavior of credit notes differs from invoices this will be explicitly mentioned.

---

## New

To send an invoice to a customer you must first create the new document. To create an invoice use Business > Customer > New Invoice. The New Invoice window must be filled in appropriately.

When you click the OK button, the Edit Invoice window opens.

**New Invoice**

Invoice Information

Invoice ID

Date Opened

Billing Information

Customer  

Job  

Billing ID

Terms

Notes

Your personal notes go here.  
Notes do not appear on printed invoices.

Help Cancel OK

Fig. 26: Creating a New Invoice



## Edit

From the Edit Invoice window you can enter an itemized list of goods and services you sold on this invoice in a manner similar to how the account register works. For credit notes you enter an itemized list of goods and services you refunded instead.

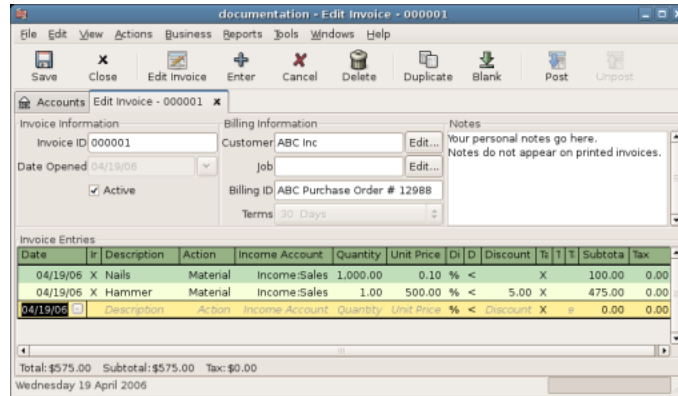


Fig. 27: Edit Invoice Window

When you have finished entering all the items, you can Post and print the invoice.

## Post

When you finish editing an invoice and are ready to print, you must Post the invoice. The invoice does not have to be posted immediately. In fact, you should only post an invoice when you are ready to print it. Posting an invoice places the transactions in an accounts receivable account.

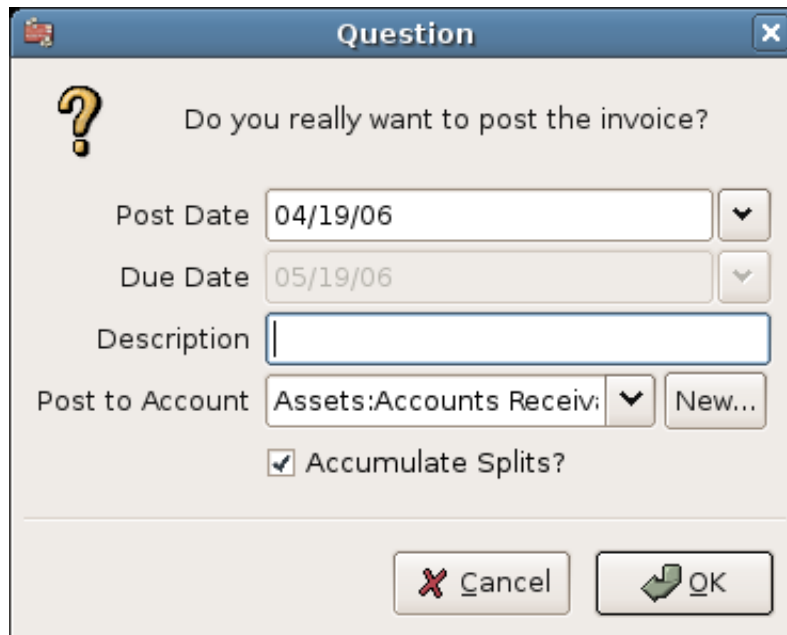


Fig. 28: Post Invoice Window

### Find

To find an existing invoice, use the Business > Customer > Find Invoice menu item. From the results of the search, you can select an invoice to edit or view.

---

**Note:** Before you can edit a posted invoice, you will need to Unpost it.

---

One of the design goals in GnuCash's Account Receivable system was to allow different processes to get to the same state, so you can reach an invoice from different directions based on the way you think about the problem:

- You can search for the customer first, then list their invoices.
- You can search for invoices by number or by company name.
- You can list invoices associated with a customer job.

### Print

After you post an invoice, you should print it and send it to your customer. To print an invoice use File > Print Invoice menu item.

<b>Invoice #000001</b>							
Invoice Date: 04/19/06				My Company			
Due Date: 05/19/06				9876 Second St.			
				Othercity, SS 12345			
				April 19, 2006			
ABC Inc.							
123 First Ave.							
Somecity, SS 12345							
Reference: ABC Purchase Order # 12988							
Terms: 30 Days - same as cash							
Date	Description	Charge Type	Quantity	Unit Price	Discount	Taxable	Total
04/19/06	Nails	Material	1,000.00	\$0.10	0.00%		\$100.00
04/19/06	Hammer	Material	1.00	\$500.00	5.00%		\$475.00
<b>Subtotal</b>							<b>\$575.00</b>
<b>Tax</b>							<b>\$0.00</b>
<b>Amount Due</b>							<b>\$575.00</b>
Thank you for your patronage							

Fig. 29: Invoice Print Output

---

**Note:** You can modify the appearance of the invoice, IE: add a company logo, etc. To do so, see the [Changing the Invoice Appearance](#).

---

Invoices can also be printed from the main window by selecting Reports > Business Reports > Printable Invoice from the main menu. The resulting report window states that no valid invoice is selected. To select the invoice to print:

1. Use the Options *Toolbar* button or select Edit > Report Options from the main menu.
2. Select the General tab of the report options dialog.
3. Click the Select button next to the Invoice Number field.
4. Search for the invoice as usual.

You can also print invoices from within the Process Payment dialog. See the [Process Payment](#) for instructions on how to do so.

### Assign Starting Invoice Number

By default, GnuCash starts with invoice number 1 and increments from there. You can manually type an invoice number into the text box each time you create an invoice, but this gets tiring and sooner or later leads to duplicate numbers.

You can change the starting invoice number if it is important you. Use File > Properties, access the Counters tab, change the Invoice number value to be one less than your desired starting invoice number and click the OK button or the Apply button.

### Customer Jobs

Customer Jobs are used to group multiple invoices and credit notes to the same customer. Use of the Customer Jobs feature is optional. The feature is useful when you have multiple jobs for the same customer, and would like to view all the invoices and credit notes related to a single job.

The screenshot shows a dialog box titled "New Job - ABC Deck Construction ()". It contains the following fields and controls:

- Job Information:**
  - Job Number: (empty text box)
  - Job Name: ABC Deck Construction
- Owner Information:**
  - Customer: ABC Inc (text box) and Select... (button)
  - Billing ID: ABC Purchase Order # 12988
- Active:**
  - ☒ Job Active
- Buttons:** Help, Cancel, and OK.

Fig. 30: New Customer Job

To use customer jobs, you must create them using the Business > Customer > New Job menu item. You will see the New Job window.

To edit an existing customer job, use the Business > Customer > Find Job menu item. Select the desired job in the search results, and click the View/Edit Job button.

To select from the invoices and credit notes associated with a given job, use Business > Customer > Find Job menu item. Select the desired job in the search results and click the View Invoices button. A window listing invoices and credit notes associated with this job appears. Select an invoice or credit note and click the View Invoice button to open an invoice editor in the main application window.

## Process Payment

Eventually, you will receive payment from your customers for outstanding invoices. To register these payments, use the Process Payment application found in Business > Customer > Process Payment.

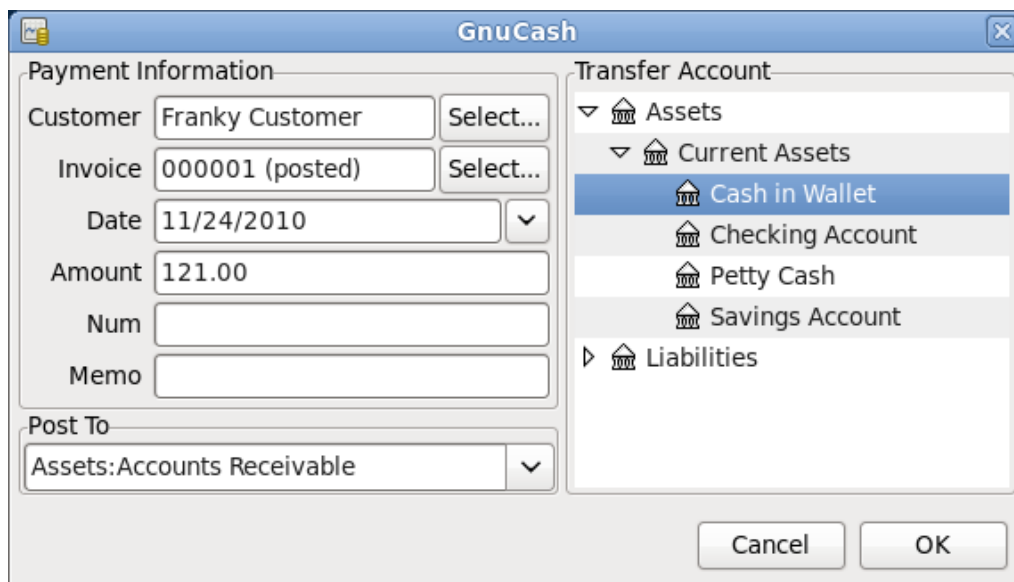


Fig. 31: Process Customer Payment

## Changing the Invoice Appearance

The default Invoice style, as shown in [Print](#), is fairly barren. The default invoice style leaves the top part of the form blank, so you can print on company letterhead paper. There are some things you can do to change invoice appearance.

Use File > Properties to enter your Company information in the Business tab of the Book Options window. Some of the entered information is printed on the right side of invoices.

To add a customized background, heading banner or logo to invoices, modify the invoice style sheets. To do this, go to Edit > Style Sheets and select the New button in the Select HTML Style Sheet window that will appear. You will then see a window like this:

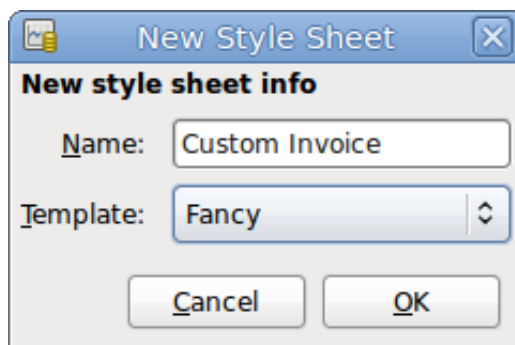


Fig. 32: The New Style Sheet Window

Give a Name to the new style sheet (e.g. Custom Invoice) and select the Fancy Template. When you click the OK button, the HTML Style Sheet Properties window is displayed. This window presents you five sections listed in the

left pane: Colors, Fonts, General, Images, and Tables. The Colors section allows you to change the colors of various items of the invoice. The Fonts section lets you set fonts type and dimensions. The General section allows you to set the Preparer and Prepared for information, and to Enable Links. The Images section allows you to import graphics into the style sheet. The Tables section allows you to adjust the spacing around the tables which make up the invoice.

To include a company logo, banner heading and background image, use your favorite graphics application such as The Gimp or OpenOffice Draw to save the images in either GIF or PNG format. Then import them into the style sheet using the Images section described above.

Below is an example that imports all three types of images.

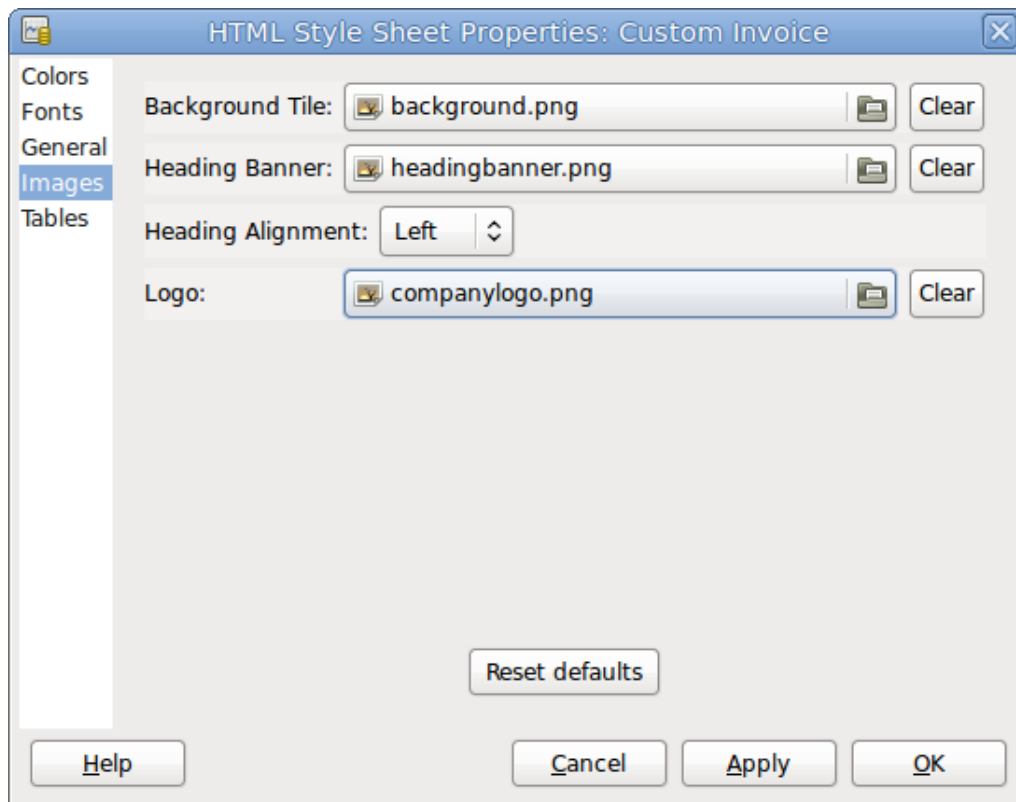


Fig. 33: HTML Style Sheets Example Window

**Note:** The images are placed in the invoice as follows. The Background Tile is tiled to become the background image, the Heading Banner goes to above the invoice text, and the Logo is placed in the upper left corner of the invoice to the left of the Heading Banner. You will probably have to try a few different sized images until you get the invoices to print nicely. Some sizing suggestions are that the Logo should be 1 square cm (~0.5 inch), and the Heading Banner should be 15 cm (~6 inches) wide and 1 cm (~0.5 inch) tall.

With the style sheet configured, when you print the invoice, you select the style sheet to use from the Options menu. Below is the resultant invoice after applying the style sheet demonstrated above.

#### 2.14.4 Accounts Payable

Accounts Payable (or A/P) refers to the accounting of products or services which a company has bought and needs to pay for.

<b>My Company Logo</b>		<b>Heading Banner</b>				
Invoice #000001		Invoice Date: 04/19/06			My Company	
Due Date: 05/19/06		ABC Inc			9876 Second St.	
123 First Ave.		Somecity, SS 12345			Othercity, SS 12345	
Reference: ABC Purchase Order # 12988		Terms: 30 Days - same as cash			April 23, 2006	
Date	Description	Charge Type	Quantity	Unit Price	Discount	Total
04/19/06	Nails	Material	1,000.00	\$0.10	0.00%	\$100.00
04/19/06	Hammer	Material	1.00	\$500.00	5.00%	\$475.00
<b>Subtotal</b>						<b>\$575.00</b>
<b>Tax</b>						<b>\$0.00</b>
<b>Amount Due</b>						<b>\$575.00</b>
Thank you for your patronage						

Fig. 34: HTML Style Sheets Example Output

## System Components

GnuCash has an integrated accounts payable system. The transactions generated by the A/P system are placed within the Accounts Payable account, as a record of what occurs. Generally you do not directly work with this account but use the four integrated GnuCash A/P application components. The A/P components are available from the Business > Vendor sub-menu. These A/P components are:

- Vendors are people or companies from which you buy products or services on credit.
- Bills represent the physical bills vendors send to request payment from you. A bill contains an itemized list of things you purchased.

In addition, GnuCash also has support for Credit Notes which represent the inverse of Bills. A credit note is usually received from a vendor to correct items that were erroneously billed or returned.

Both document types will be set up using the same menu items.

- Jobs (optional) is where you register Vendor Jobs. Jobs are mechanism by which you can group multiple bills from a particular vendor.
- Process Payments is where you register payments to a vendor to whom you owe money.

The following sections introduce the individual Accounts Payable application components.

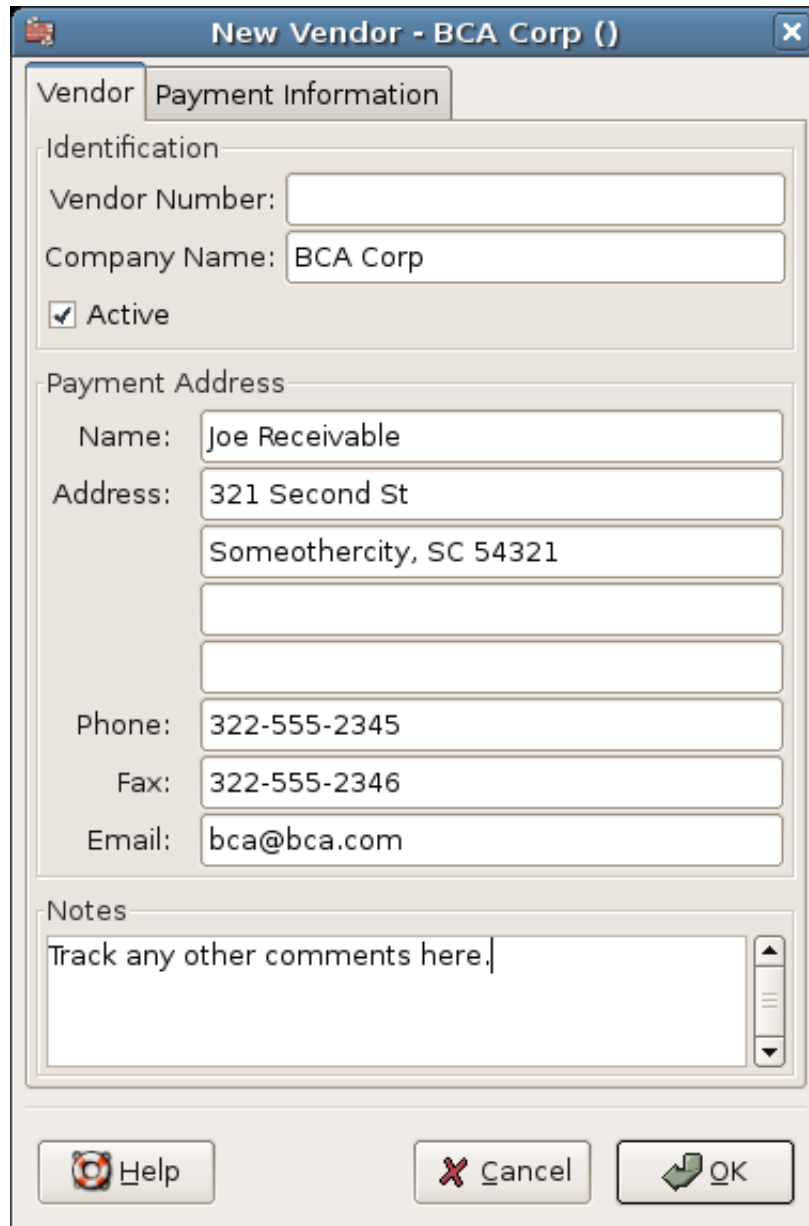
## Vendors

A vendor is a company or person from whom you purchase goods or services. Vendors must be registered within the A/P system.

## New

To register a new vendor, select the Business > Vendor > New Vendor menu item. Fill in general information about the vendor, such as Company Name, Address, Phone, Fax, etc. Below is a list of the other options:

This is what the New Vendor registration window looks like:



The image shows a 'New Vendor' registration window for 'BCA Corp'. It has two tabs: 'Vendor' (selected) and 'Payment Information'. The 'Vendor' tab contains an 'Identification' section with fields for 'Vendor Number' (empty), 'Company Name' (filled with 'BCA Corp'), and a checked 'Active' checkbox. Below this is a 'Payment Address' section with fields for 'Name' (filled with 'Joe Receivable'), 'Address' (filled with '321 Second St' and 'Someothercity, SC 54321'), 'Phone' (filled with '322-555-2345'), 'Fax' (filled with '322-555-2346'), and 'Email' (filled with 'bca@bca.com'). At the bottom is a 'Notes' section with a text area containing 'Track any other comments here.' and a scrollbar. The window has 'Help', 'Cancel', and 'OK' buttons at the bottom.

**New Vendor - BCA Corp ()**

**Vendor** | Payment Information

Identification

Vendor Number:

Company Name:

☒ Active

Payment Address

Name:

Address:

Phone:

Fax:

Email:

Notes

Track any other comments here.




 Help  Cancel  OK

Fig. 35: New Vendor Registration Window

## Find and Edit

To search for an existing vendor, use the Business > Vendor > Find Vendor window. You select a vendor to View/Edit from the results of the search. This window is also used to look up a vendor when entering bills and processing payments.

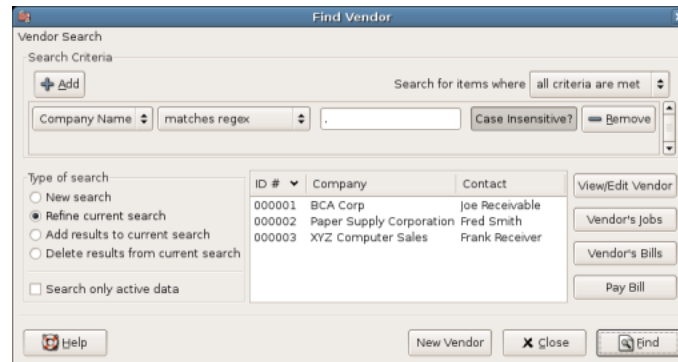


Fig. 36: Find Vendor Window

If many vendors match the search criteria you provide, the search can be refined by running an additional search within the current results. The current result set is searched when the Refine Current Search radio button is selected. In fact, GnuCash selects this option for you after you run the initial search.

If the vendor you are searching for does not match the supplied search criteria, change the search criteria, click the New Search radio button and then the Find button. The relevant step is the New Search selection. If the vendor is not in the result of the original search, and you only search within this set, the vendor cannot be found, regardless of new search criteria.

---

**Note:** To return a list of all registered active vendors, set the search criterion to matches regex, and place a single dot “.” in the text field area. Make sure Search only active data is checked, then click Find. The regular expression “.” means to match anything.

---

## Bills

A bill is a request for payment you receive from a vendor. GnuCash can track bills.

A credit note is the document you receive from a vendor to correct products or services rendered that you were incorrectly charged for on a bill. GnuCash can generate and track credit notes via the same menu entries as bills.

---

**Note:** This section applies to both bills and credit notes. In order to avoid repeating everything twice and to keep the text easier to read it will refer only to bills. You can apply it equally to credit notes. Only where the behaviour of credit notes differs from bills this will be explicitly mentioned.

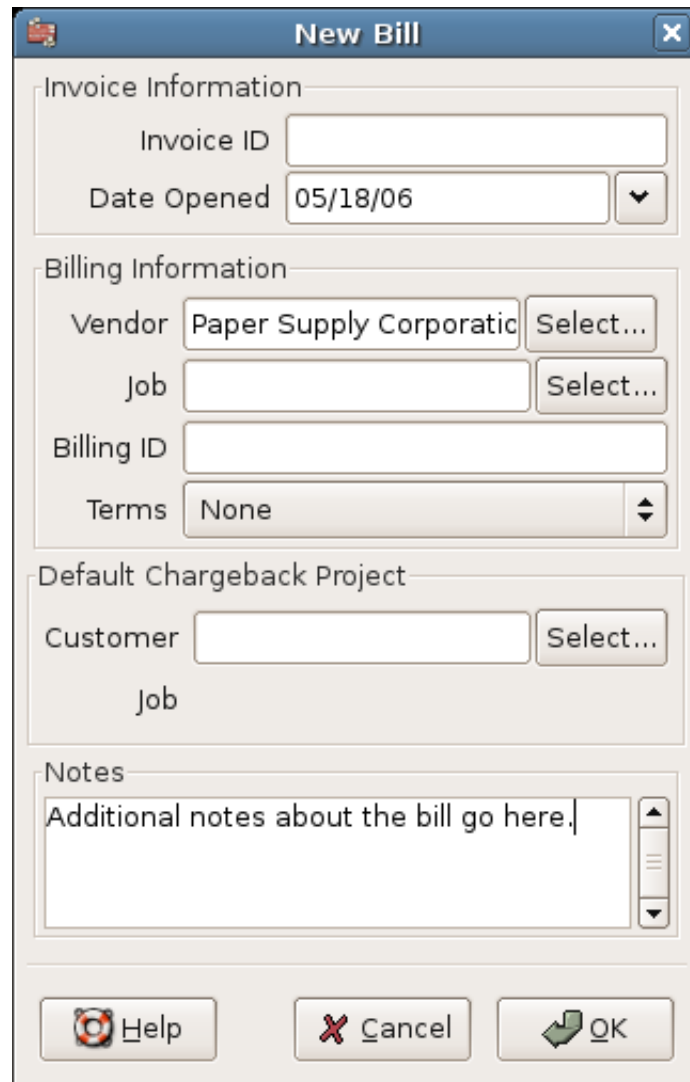
---

## New

When you receive a bill from a vendor and want to enter it into GnuCash, you must create a new bill. To create a new bill use the Business > Vendor > New Bill menu item, and fill in the resulting window appropriately.

When you click the OK button the Edit Bill window opens.





The "New Bill" window is a dialog box with a title bar containing a red "X" icon and the text "New Bill". It is divided into four main sections: "Invoice Information", "Billing Information", "Default Chargeback Project", and "Notes".

- Invoice Information:** Contains two text input fields. The first is labeled "Invoice ID" and is empty. The second is labeled "Date Opened" and contains the text "05/18/06". To the right of the date field is a small downward-pointing arrow icon.
- Billing Information:** Contains four fields. The first is labeled "Vendor" and contains the text "Paper Supply Corporatic", with a "Select..." button to its right. The second is labeled "Job" and is empty, with a "Select..." button to its right. The third is labeled "Billing ID" and is empty. The fourth is labeled "Terms" and contains the text "None", with a double-headed vertical arrow icon to its right.
- Default Chargeback Project:** Contains two fields. The first is labeled "Customer" and is empty, with a "Select..." button to its right. The second is labeled "Job" and is empty.
- Notes:** Contains a large text area with the placeholder text "Additional notes about the bill go here.". To the right of the text area is a vertical scrollbar with up and down arrow icons.

At the bottom of the window are three buttons: "Help" (with a red circular icon), "Cancel" (with a red "X" icon), and "OK" (with a green checkmark icon).

Fig. 37: New Bill Registration Window

## Edit

From the Edit Bill window you can enter an itemized list of goods and services you purchased, in a manner similar to how the account register works. For credit notes you enter an itemized list of goods and services the vendor refunded instead.

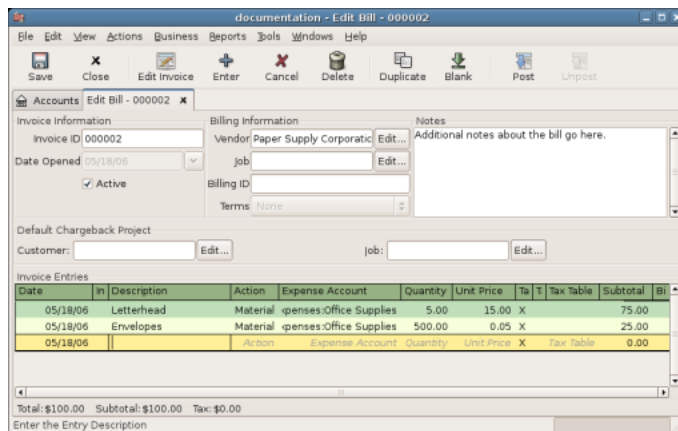


Fig. 38: Edit Bill Window

When you have finished entering all the items, Post the bill.

## Post

When you finish editing a bill, you should Post the bill. You do not have to post the bill, you can close it and return to it later. You have to post the bill eventually. Posting a bill places its transactions into an accounts payable account. The Post Bill window appears and asks you to enter information:



Fig. 39: Post Bill Window

## Find

To find an existing bill, use the Business > Vendor > Find Bill menu item. From the results of the search, you can select a bill to edit, or view.

---

**Note:** Before you can edit a posted bill, you will need to Unpost it.

---



---

**Note:** There are other ways to access an existing bill. These are similar to accessing invoices for your customers. See [Find](#) for more information.

---

## Vendor Jobs

Vendor Jobs are used to group multiple bills and credit notes from a single vendor. Use of the vendor jobs feature is optional. The feature is useful when you have multiple jobs for the same vendor, and would like to view all the bills and credit notes for a single job.

To use vendor jobs, you must create them using the Business > Vendor > New Job menu item. You will see the New Job window.

The image shows a window titled "New Job - 2006 Computer Upgrades". It contains the following fields and controls:

- Job Information:**
  - Job Number: [empty text box]
  - Job Name: 2006 Computer Upgrades
- Owner Information:**
  - Vendor: XYZ Computer Sales [Select... button]
  - Billing ID: [empty text box]
- Active:**
  - ☒ Job Active
- Buttons:** Help, Cancel, OK

Fig. 40: New Vendor Job

To edit an existing vendor job, use the Business > Vendor > Find Job menu item. Select the desired job in the search results, and click the View/Edit Job button.

To select from the bills associated with a given job, use Business > Vendor > Find Job menu item. Select the desired job in the search results and click the View Invoices button. A window listing bills and credit notes associated with this job appears. Select a bill or credit note and click the View Invoice button to open a bill editor in the main application window.

## Process Payment

Eventually, you need to pay your bills. To do so, use the Process Payment application found in Business > Vendor > Process Payment.

Below is the GnuCash Accounts Payable payment window.

Fig. 41: Process Vendor Payment

## 2.14.5 Payroll

Payroll is a financial record of wages, net pay, paid vacations, and deductions for an employee. This section demonstrates how to track payroll using GnuCash.

### Basic Concepts

Payroll is a financial record of wages, net pay, paid vacations, and deductions for an employee. Basically, anything that relates to giving money or benefits to an employee. Payroll is one of the more complex tasks in accounting, because there are many different accounts, people, and agencies involved in paying salaries.

Payroll is typically accounted for as an expense. Sometimes accountants “store” some payroll expenses in a short term liability account. This is useful for things such as payroll taxes, which may be paid at a different time than the employee. The employee might get paid biweekly, while taxes are paid quarterly. This chapter presents a methodology which expenses payroll immediately for salaries, but stores taxes in liability accounts.

---

**Note:** GnuCash does not have an integrated payroll system. While you can track payroll expenses in GnuCash, the calculation of taxes and deductions has to be done outside of GnuCash.

---

## Account Setup

Local tax law must be considered when setting up accounts. Because there are many different ways payroll taxes are handled throughout the world, this section presents a very simple structure. From this, you should be able to adapt your particular payroll deductions setup.

Assume that you must pay 2 taxes, Tax1 and Tax2, and that each has an employee contribution and an employer contribution.

The employee's salary and these two taxes are expense accounts. The tax components are liability accounts. The tax liability accounts are where you accumulate the taxes withheld for all of your employees. The taxes are later paid to the appropriate government agency.

Simple Payroll Account Layout: -Assets -Checking -Liabilities -Tax1 (short term "storage" account) -Tax2 (short term "storage" account) -Expenses -Salaries -Tax1 -Tax2

---

**Note:** Resist the temptation to create per-employee sub-accounts to track individual salaries. Creating a sub-account for each employee leads to unmanageably large lists of accounts. Imagine the account structure after a few years of employees coming and going. It is much simpler to keep all of your employees' payroll records within a single account (*Expenses:Salaries* for example) and use reports to view per-employee information.

The Transaction report can be used to sort and total by description or memo (but not by part of them).

To report salary transactions for a specific employee where the employee name or code is entered in the transaction description or memo, use the Find Transaction assistant (Edit > Find) to select the transactions, and then report on them using Reports > Account Report. Further formatting or analysis may be done by copying and pasting the report into a spreadsheet. See [Find Transaction](#) in the GnuCash Help Manual.

---

## Payroll Protocol Sample

GnuCash does not have an integrated payroll system. GnuCash can track your payroll expenses, but you need to develop a payroll protocol and perform the calculations outside of GnuCash, in a spreadsheet for example. In this section, one such protocol is presented. You can use the sample protocol as a model.

### Step 1: Deductions list

The first step to the payroll protocol is to create a list of all the possible taxes and deductions for each employee. Each entry should include definitions and formulas for calculating each value. Once the protocol is established it needs to be changed only when payroll laws or tax rates change.

In the proposed scenario, such a list would look like this:

- *E\_GROSS\_SALARY* - Employee gross salary
- *E\_TAX1* - Employee contribution to tax1 (X% of *E\_GROSS\_SALARY*)
- *E\_TAX2* - Employee contribution to tax2 (X% of *E\_GROSS\_SALARY*)
- *C\_TAX1* - Company contribution to tax1 (X% of *E\_GROSS\_SALARY*)
- *C\_TAX2* - Company contribution to tax2 (X% of *E\_GROSS\_SALARY*)

---

**Note:** The employee's net salary (*E\_NET\_SALARY*) is defined as *E\_GROSS\_SALARY* - *E\_TAX1* - *E\_TAX2* and need not be placed in this list since it is composed of items that already exist.

---

Place the actual formulas for calculating each deduction in this list. Sometimes these formulas are quite complex, and sometimes they simply say “look it up in table XYZ of the tax codes”.

Notice that you can calculate some interesting values using the above definitions. One such value is the total cost to the company:  $E\_GROSS\_SALARY + C\_TAX1 + C\_TAX2$ .

## Step 2: Create the Transaction Map

When you record payroll in GnuCash, do so with a single split transaction. This split transaction populates the appropriate expense and liability accounts. If you need to look the payroll details at a later time, open the split transaction.

With the deductions list from above, an employee split transaction map can be generated. Each of the items in the list is mapped to a GnuCash account.

Table 22: Payroll Transaction Map

Account	Increase	Decrease
Assets:Checking		E_NET_SALARY
Expenses:Salaries	E_GROSS_SALARY	
Liabilities:Tax1		E_TAX1
Liabilities:Tax2		E_TAX2
Expenses:Tax1	C_TAX1	
Liabilities:Tax1		C_TAX1
Expenses:Tax2	C_TAX2	
Liabilities:Tax2		C_TAX2

Note that the C\_TAX1 and C\_TAX2 components have entries in the both the liability and expense accounts. The company component of each tax is expensed at the time of payroll, but remains a liability until taxes are due.

## Step 3: Pay the Employee

Go to the account from which the employee will be paid, for example your *Assets:Checking* account. Open a split transaction and enter the real values using the Transaction Map above as a guide. Repeat this for all employees.

---

**Tip:** This manual process is tedious, especially if you have a large number of employees.

One GnuCash tool you certainly want use when entering employee payroll is duplicate transaction (use the Duplicate *Toolbar* button). This saves you from having to enter all the transaction splits for each employee. You still need to change the amounts of money to match each employee’s real payroll values, but you will not have to build the split for each employee.

If payroll transactions do not change significantly every pay period, you can also use the duplicate transaction feature to duplicate each employee’s most recent payroll transaction for the current pay period. If you find you are doing so all the time, read about the Schedule Transactions feature and save even more time!

---

## Step 4: Pay the Government

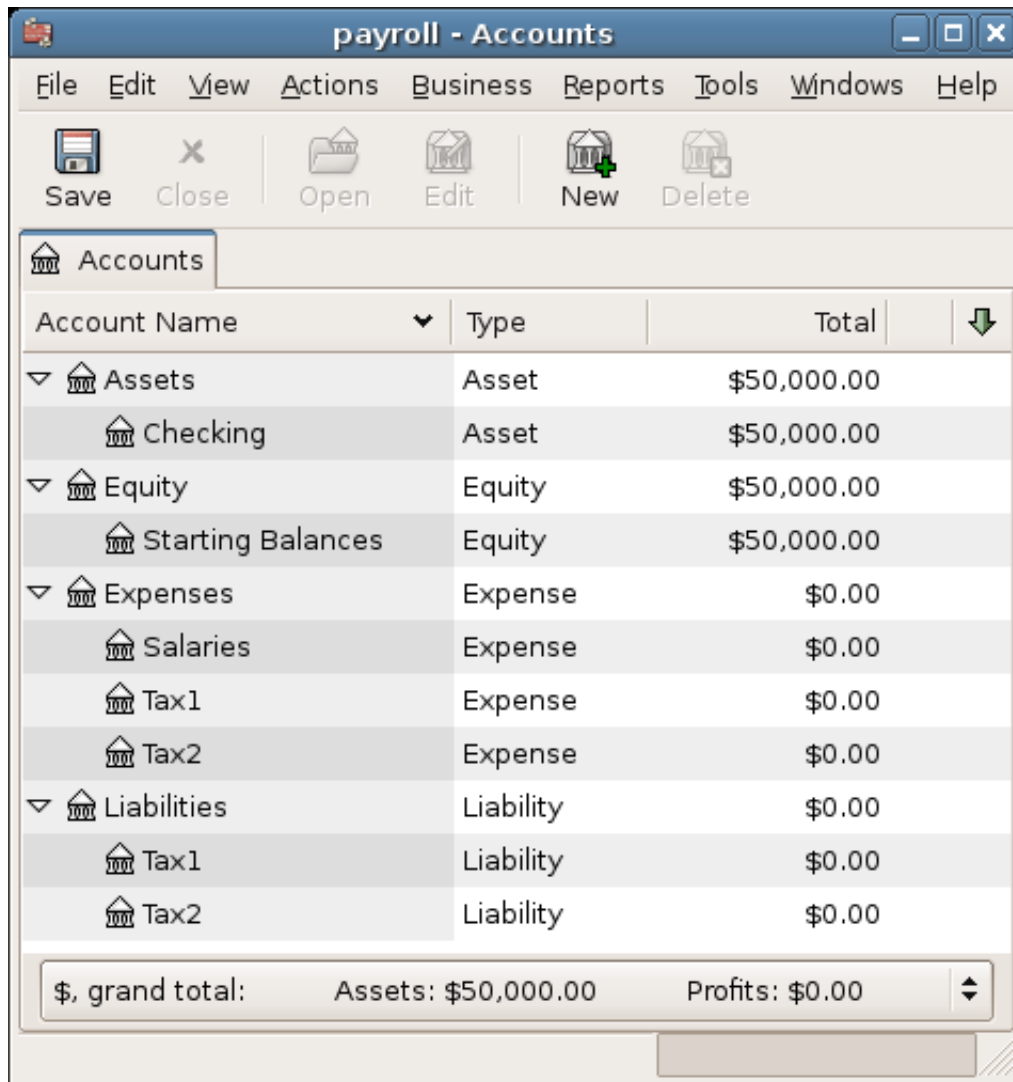
The final thing to do is to pay the taxes to the government. The liability accounts have been collecting the taxes for various government agencies, and periodically you need to send a check to the government to pay this charge. To do so, you simply enter a 2 account transaction in (for example) your checking account to pay off the tax liability. The

transaction is between the checking account and the liability account, no expense account is involved. The expense accounts are charged at the time the tax liability is recorded.

### Business Payroll Example

Using the account setup seen previously, let's go through an example. Assume that there are 2 employees (E1 and E2) who each earn \$1000 per month gross salary. The employee contribution to Tax1 and Tax2 are 10% and 5% respectively. The company contribution to Tax1 and Tax2 are 15% and 10% each on top of the employee's gross salary.

Starting with \$50k in the bank, and before doing any payroll, the account hierarchy looks like this:



Account Name	Type	Total
Assets	Asset	\$50,000.00
Checking	Asset	\$50,000.00
Equity	Equity	\$50,000.00
Starting Balances	Equity	\$50,000.00
Expenses	Expense	\$0.00
Salaries	Expense	\$0.00
Tax1	Expense	\$0.00
Tax2	Expense	\$0.00
Liabilities	Liability	\$0.00
Tax1	Liability	\$0.00
Tax2	Liability	\$0.00

\$, grand total:      Assets: \$50,000.00      Profits: \$0.00

Fig. 42: Payroll Example: Initial Setup

The deductions list for employee 1 are:

- *E\_GROSS\_SALARY* - Employee gross salary - \$1000
- *E\_TAX1* - Employee contribution to tax1 - \$100 (10% of *E\_GROSS\_SALARY*)
- *E\_TAX2* - Employee contribution to tax2 - \$50 (5% of *E\_GROSS\_SALARY*)

- *C\_TAX1* - Company contribution to tax1 - \$150 (15% of E\_GROSS\_SALARY)
- *C\_TAX2* - Company contribution to tax2 - \$100 (10% of E\_GROSS\_SALARY)

Table 23: Payroll Transaction Map for Employee 1

Account	Increase	Decrease
Assets:Checking		\$850 (E_NET_SALARY)
Expenses:Salaries	\$1000 (E_GROSS_SALARY)	
Liabilities:Tax1		\$100 (E_TAX1)
Liabilities:Tax2		\$50 (E_TAX2)
Expenses:Tax1	\$150 (C_TAX1)	
Liabilities:Tax1		\$150 (C_TAX1)
Expenses:Tax2	\$100 (C_TAX2)	
Liabilities:Tax2		\$100 (C_TAX2)

From the checking account, enter the split transaction for employee 1. It should look like this:

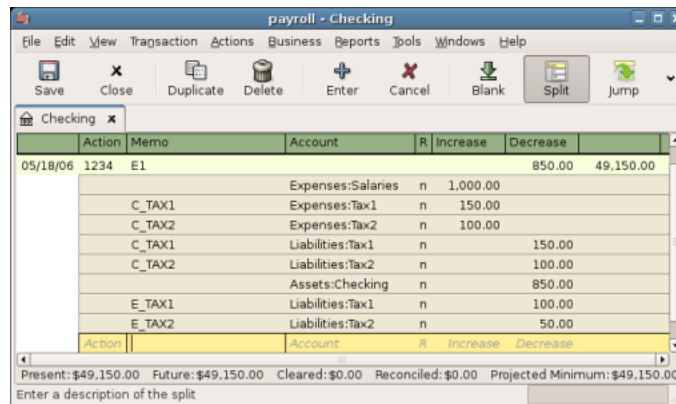


Fig. 43: Payroll Example: Employee Split Transaction

**Tip:** When paying employees, enter only the employee name in the Description area. If you decide to use GnuCash's check printing capabilities, the check is automatically made out to the correct employee. If you want to record other information in the transaction besides the employee name, use the Notes area, available when viewing the Register in double-line mode.

Repeat this for the second employee, which leaves the account hierarchy looking like this:

Now, you will enter the company contributions. The *Liabilities:Tax1* and *Liabilities:Tax2* accounts track how much you must pay to the government for each tax type. When it is time to pay these agencies, you will make a transaction from the checking account to these liability accounts. No expense accounts are involved. The main account will then appear like this:

## 2.15 Budgets

This chapter explains how to create and use budgets with GnuCash.



Account Name	Type	Total
Assets	Asset	\$48,300.00
Checking	Asset	\$48,300.00
Equity	Equity	\$50,000.00
Starting Balances	Equity	\$50,000.00
Expenses	Expense	\$2,500.00
Salaries	Expense	\$2,000.00
Tax1	Expense	\$300.00
Tax2	Expense	\$200.00
Liabilities	Liability	\$800.00
Tax1	Liability	\$500.00
Tax2	Liability	\$300.00
\$, grand total:		Assets: \$47,500.00    Profits: -\$2,500.00

Fig. 44: Payroll Example: Accounts After Salaries Paid

Account Name	Type	Total
Assets	Asset	\$47,500.00
Checking	Asset	\$47,500.00
Equity	Equity	\$50,000.00
Starting Balances	Equity	\$50,000.00
Expenses	Expense	\$2,500.00
Salaries	Expense	\$2,000.00
Tax1	Expense	\$300.00
Tax2	Expense	\$200.00
Liabilities	Liability	\$0.00
Tax1	Liability	\$0.00
Tax2	Liability	\$0.00
\$, grand total:		Assets: \$47,500.00    Profits: -\$2,500.00

Fig. 45: Payroll Example: Accounts After Paying Government

### 2.15.1 Basic Concepts

A budget is a tool for estimating expected income and expenses. You can use it to help you plan how you intend for your finances to change over a period of time, and to examine how your actual financial transactions for the period compare to your planned transactions.

The budgeting concept is quite general, so GnuCash offers a budgeting tool that is both simple and flexible. You, the user, have to decide how complex or simple you want to make your budget. This guide will help you make some of those decisions.

#### Terminology

There are a few helpful terms listed below that will be used to discuss budgeting.

- *Budget* - A financial plan describing the expected revenues and/or disbursements for a particular time period
- *Cash Budget* - A budget planning for expected cash receipts and cash disbursements. This type of budget tracks cash flow – where your money comes from, where it goes, and, of course, how much.
- *Expense Budget* - A budget chiefly for planning what you spend your money on. This type of budget tracks your expenses. It is typically not concerned with things like appreciation or repayment of liabilities. However, it would account for interest charges. For example, if you buy \$100 worth of groceries with your credit card, you incur an \$100 expense for groceries, and a \$100 liability to your credit card company. When you pay the credit card bill for \$110, you are incurring an additional interest expense of \$10. An expense budget plans for the transaction of buying the groceries and paying the interest, but not the transaction of repaying the credit card company.
- *Capital Budget* - A budget that describes a plan for paying for a large future expense, often through a combination of saving and borrowing money. Note: Capital budgets can sometimes get quite complex because they can try to answer the question “Can we afford to do such-and-such?” by exploring various hypothetical scenarios that can involve hypothetical accounts.
- *Budget Period* - The period of time during which the plan is expected to take place. The most common budget periods are annual and monthly. Sometimes, you may budget for several consecutive periods at once, for convenience or for finer-grained planning. For example, an annual budget may include 12 monthly budget periods.

### 2.15.2 Creating a Budget

Even before you begin to make a budget, it's important to have given some thought to your account hierarchy. For example, if you want to budget a certain amount for your electric bill and a certain amount for your water bill, you can't have only an *Expenses:Utilities* account. Your accounts must be at least as specific as your budget.

#### Choose Which Accounts To Budget For

The first step in creating a budget is to decide what it is you want to plan for. This decision will affect which accounts you include in your budget. For example, if you are only interested in tracking your expenses, you may create an expense budget by only entering amounts for expense accounts. On the other hand, if you want to track all of your cash flow, you may create a cash flow budget by entering amounts for asset, liability, income and expense accounts.

Before you begin to create your budget, you need to make two decisions: What accounts do I want to budget for? and When do I want my budget to be for? You can always change your mind later, after you've created a budget, but you need to start with something.

---

**Tip:** As a rule of thumb, if you mostly care about *what* you spend your money on, you may want to make an expense report. If you're also concerned about having enough money in the right places at the right times, you may want to use a cash-flow budget.

---

### Choosing a Budget Period

Before creating a budget you must also decide what period of time you want to plan for. The most common budget periods are monthly and annual. If you want your budget to plan for changes in financial patterns over time, then you should include multiple budget periods in your budget. For example, if you want to plan on having higher utility expenses in the winter than in the summer, then you might break your annual budget into 4 quarters or even 12 months, and budget a higher value for the winter periods than for the summer periods.

### Getting Started

To create your first budget click on Actions > Budget > New Budget. You will immediately see a new budget with the default settings and no entries. Then click on the Options button. The most important options are the budget period and the number of periods. For the budget period, choose the beginning date and the smallest period of time that you want to plan for. Then, for the number of periods, choose how many periods you want to plan for.

The budget page now shows a list of accounts with a column for each budget period. The date shown in the title of each column is the beginning of that budget period.

### Entering Budget Values

Now, you must enter the budget values - the amounts that you expect the account balances to change during the budget period. There are two ways to enter budget values. The first way is to simply click on the cell and enter an amount.

If you have past transactions recorded in GnuCash, the second way is to let GnuCash estimate the budget values by looking at those transactions. First, select the accounts you want GnuCash to estimate. Then click on the Estimate *Toolbar* button. In the Estimate Budget Values dialog, select the date past which GnuCash should look for past transactions. GnuCash will start at that date and look forward for the duration of your budget. For example, if you are making an annual budget, and you select Jan. 1, 2005, GnuCash will look at all the transactions in that account from Jan. 1, 2005 through Dec. 31, 2005.

## 2.15.3 Budget Reporting

You've already done the hardest part - creating your budget. But now you want to know how your actual financial transactions compare to your plan. You need to run the Budget Report.

Click on Reports > Budget > Budget Report. For each account, the Budget Report will show the budgeted and the actual amounts in two adjacent columns for each period in the budget. If you have created multiple budgets, you can use the Budget Report Options to select which budget to use in the report.

Two other types of budget reports are commonly used in the small business setting. They are the *Budgeted Income Statement* and the *Budgeted Balance Sheet*.

### Budgeted Income Statement

The budgeted income statement is similar to the income statement. Both show the revenues and expenses for a given period as well as the profit, which is the difference revenue - expenses. The income statement is based on historical data, but the *budgeted* income statement is based on the predictions made in the budget.

## Budgeted Balance Sheet

The budgeted balance sheet is similar to the balance sheet. Both show the assets, liabilities, and equity. The difference is that the balance sheet is based on historical data, and the *budgeted* balance sheet is based on the predictions made in the budget.

## 2.16 Other Assets

### 2.16.1 General Concepts

This chapter presents many additional accounting treatments for frequently encountered business and less-frequently found personal activities that need recording in accounting books. The explanations below cover both the description and purpose of the activity, and they include also the usual accounting treatments (bookings or recordings) for these transactions.

These concepts have evolved over centuries of experience by those keeping accounting records and will help you maximize your record keeping's utility and meaningfulness.

This section introduces categorization of assets in the balance sheet based on time or the asset's useful life (current and long-term). Sometimes assets are also considered from the standpoint of their *liquidity*, which is regarded as how close or distant the asset is from being turned into cash. Near-cash assets are relatively quickly converted to cash (e.g., accounts receivable), while assets requiring rather a long time to convert to cash are considered to be relatively *fixed* in their non-cash state (e.g., heavy equipment, buildings, land). (Fixed does not mean they were repaired!)

You should find that current assets parallel those with more liquidity, while long-term and fixed assets are those with much less liquidity. Finally, below you will find a few assets that could be either current or long-term based on the nature of the facts constituting them.

### 2.16.2 Other Assets Described

#### Current Assets

Current Assets are those activities whose normal expected life would be one year or less. Such activities could be tracking reimbursable expenses, travel advances, short-term loans to a friend or family member, prepaid expenses, annual insurance premium amortization, and so on. The individual entity could have many other kinds of short term activities that reflect what it is doing. (These asset types are explained individually below.)

#### Long-term (Fixed) Assets

Long-term (Fixed) Assets are those activities whose normal expected life exceeds one or more years. This grouping covers both tangible and intangible assets. Examples of tangible assets are land, buildings, and vehicles (cars, trucks, construction equipment, factory presses, etc.) Intangible assets include such things as patents, copy rights, goodwill, etc. Because the lives of some of these assets show wear and tear and deterioration in value over time, businesses and individuals can allow for that diminution in value by calculating depreciation on such assets. For example, land normally does not depreciate, but buildings do, as do equipment and vehicles. (These asset types are explained individually below.)

### 2.16.3 Current Assets

This section explains short-term receivables, reimbursable expenses, travel advances, prepaid premiums, prepaid rent, suspense or wash accounts.

## Short-term Receivables

This kind of account is useful to reflect an agreement made with someone you trust. Suppose you lent someone \$500 and he agreed to repay you \$50 a month. If he paid on time, the loan you made would be paid off within a year, which is why it is classified as a short-term receivable. So you could record that loan initially in this account tree: *OtherAssets:Current Assets:LoanToJoe*. At the time you give him the money: your entry is debit (increase) *LoanToJoe* \$500 and credit (decrease) *Bank* \$500. Each time you receive Joe's payment you record \$50 debit (increase) to *Bank* and credit (decrease) *LoanToJoe*.

---

**Tip:** Don't become confused by the use of the word "Loan". "Loan-To" is the tipoff that you really have a receivable, that is, you will receive from Joe, the money you previously loaned. Until he actually pays the money owed you, you reflect his debt in your books by an account describing your expectation—you will receive the money owed you, hence the word "receivable".

---

## Reimbursable Expenses

This kind of activity is one in which you spend your own money on behalf of someone else (your employer, perhaps) and later you receive repayment of what you spent. The case might be a business trip. The employer has a policy of covering (paying for) all authorized expenses. After the trip is over, the employee submits a report listing dates and amounts spent with receipts for all the expenditures. The employer reviews the report and pays for all items that it considers as having a valid business reason. (Normally, employees know in advance what the employer will reimburse, so only those items are recorded as a reimbursable expense on the employee's books.)

Because a business trip can involve different kinds of expenditures (air travel, lodging, transportation at the destination, etc.), different kinds of expenditures would be recorded in the one account as long as the expenditures all related to the same trip. In other words, if a second trip is made before the first is fully settled, a second account for a different event could be set up. It would make sense to do this, if it would help to keep separate all the details of one trip from those of another. It is up to the person making the trip to decide how much trouble it would be to put separate trips in separate accounts or to put them all in the same account. The trip taker should remember that the account must be reconciled in order to know with certainty that all expenses have been reimbursed.

Recording the expenditures on the trip would be much the same. That is, if you paid trip expenses by cash you would debit (increase) the reimbursable expense account for the money paid in cash, because it is a receivable to you until it has been reimbursed to you. The credit offsetting your expenditure would decrease the account that shows the cash in your pocket or the account from which you drew the cash for the payment made. If you paid by credit card, the debit side would be the same as just described, but the credit would be an increase to the credit card company account on your books.

When you received your reimbursement, then the journal entry (or transaction) to record receipt of the funds from the employer would be: debit (increase) *Bank* for the check amount and credit (decrease) the reimbursable expense account for the check amount.

If it turns out that the reimbursable expense account is not zero balance after processing the employer's payment, then it means that there is a difference between you and the employer in handling the expense, which needs to be investigated. If the balance is a debit (a positive balance), your account has some money that was not reimbursed. If the balance is a credit (a negative balance), you were paid for more than what you recorded as due you. In both of those situations you should reconcile the difference between what you recorded and what was paid. That effort should disclose exactly what is causing the discrepancy. You will need to contact the employer's bookkeeper to know what was paid, if the reimbursement check was not accompanied by a detailed list of the items being paid you.

In the event the employer refused to reimburse you for an expenditure, that effectively makes it your expense. In that case, you would make this entry: debit (increase) your own *Expense* (appropriately named) and credit (decrease) the *Reimbursable Expense* account. That entry should result in a zero balance in the *Reimbursable Expense* account. If not, reconcile until you identify the difference.

**Tip:** Sometimes there are small differences that don't match an individual entry. In those cases divide the amount by 2 or by 9. If the unresolved amount is divisible by two, it suggests that both you and the employer entered the item in the same manner: both as debits or both as credits. If it is divisible by 9, then likely one of you transposed adjoining numbers; e.g., one entered 69 and the other entered 96. If the difference is divisible neither by 2 or by 9, then it could be that more than one error is present.

---

## Travel Advances

These are very similar to Reimbursable Expenses. The difference is that someone gives you money first; you spend it, and then you give a report accounting for what you spent it on. The report is supported by invoices establishing who, what, where, when, and how much for each expenditure. In the Reimbursable Expense case, you spent your money first and later recovered it.

In the Travel Advance case when you receive the advance, you record on your books this entry: debit (increase) Bank for the travel advance amount received (say, \$500); credit (increase) the short-term liability Travel Advance (\$500). This is a liability, because you are not gifted with the money, but only loaned it for the purpose of having funds to spend when doing the employer's business.

Frequently, the way these monetary arrangements work is that at the beginning of for example a salesperson's employment, he or she receives the advance and monthly (or more frequently) turns in a report about who, what, where, when, and how much he spent. The money in the report is reimbursed if approved.

During the period after receiving the advance and before filing a request for reimbursement report, the salesperson can record his or her expenditures into the advance liability account. In that case, the balance in the account will show how much of the advance has not yet been spent (assuming the Travel Advance balance is a credit). If no mistakes have been made and all expenses are approved, then the sum of the unspent account balance and the reimbursing check amount will equal the original travel advance amount.

It makes sense for the salesperson to record the travel expenses to this advance account (and not to his or her own expense accounts), because the money is being spent on behalf of the employer, for the employer's authorized expenses. It is not the employee's own money, and therefore not his or her own expense.

When the salesperson receives the report reimbursement (say, \$350), he or she debits (increases) Bank, and credits (increases) again the Travel Advance liability account, assuming that previously he or she had been recording expenditures to the travel advance account. Tracking activity in this manner causes the account to always show the amount that is owed the employer.

See [Reimbursable Expenses](#) above for what to do if the employer does not accept an item the employee put on the travel advance reimbursement request report. The difference resolution effort is essentially the same for both types of accounts.

## Prepaid Premiums or Prepaid Rent

Some types of expenses are usually billed as semi-annual or annual amounts. For example, the insurance industry will bill home insurance annually, while car insurance premiums can be annual or semi-annual. For those that pay an amount that covers several months or a full year, the proper accounting treatment is to reflect in each accounting period the amount that expresses the benefit applying to that period.

In the case of someone who pays a full-year's insurance premium at the beginning of the insurance period, the entry to record this is debit (increase) Prepaid Insurance Premium for say, \$1,200, and credit (decrease) Bank for \$1,200.

Then a monthly recurring journal entry (scheduled transaction) is created that debits (increases) Insurance Expense \$100 and credits (decreases) Prepaid Insurance Premium \$100. This technique spreads the cost over the periods that receive the insurance coverage benefit. Businesses following generally accepted accounting practices would normally use this technique, especially if they had to present financial statements to banks or other lenders. Whether individuals



do depends on the person and how concerned they are to match cost with benefit across time periods. Another factor influencing use of this technique would be the number of such situations the person encounters. It is relatively easy to remember one or two, but more difficult if having to manage 10 to 20. You would set up as many or as few as proved useful and important to you.

### **Suspense or Wash Accounts**

The purpose of these accounts is to provide a device to track “change of mind” situations. The objective of these accounts is to provide a temporary location to record charges and credits that are not to be included permanently in your books of record. When the transactions reflected in these accounts have been fully completed, Wash/Suspense accounts will normally carry a zero balance.

For example, say in the grocery store you see canned vegetables on sale, so you buy 6 cans at \$1 per can. Say that the total purchases were \$50. When you come home and are putting things in the cupboard you discover you already had 12 cans. You decide to return the 6 you just bought. Some persons in this situation would charge (increase) the whole bill to Grocery Expense; and when they returned the cans, they would credit (decrease) Grocery Expense. That is one way of handling that. The effect of this method is to leave recorded on your books the cost of items that you really did not purchase from a permanent standpoint. It is only when the items have actually been returned and the vendor’s return receipt has also been recorded that the distortion this method generates will then be removed.

Actually, there are several treatments, depending on when and how the original transaction was booked/recorded and when you decided to return the items purchased. Basically, did you change your mind before you recorded the transaction or after doing so?

If you decided to return the items after recording the purchase transaction, you may originally have charged Grocery Expense for the full amount (\$50) of all items. In that scenario, what you kept and the amount of the items to be returned were grouped into one account. You could edit the original transaction and restate the amount charged to the Grocery Expense account to be the difference (\$44) between the total paid (\$50) for groceries and the value of the items to be returned. That leaves the returned-item value as the amount (\$6) you should record to the Suspense account.

Obviously, if you decided to return items before you recorded your purchase, then you would book the original entry as a charge to Grocery Expense for the amount kept (\$44) and as a charge to Suspense for the amount returned (\$6). The off-setting credit (\$50) to cash or credit card is not affected by these treatments.

When there are several persons shopping and at different vendors, there can be a case where there are several returns happening at once and in overlapping time frames. In that case the Wash Account is charged (increased) at time of changing the mind, and either Bank or Credit Card is credited. When the return occurs, the reverse happens: Bank or Credit Card is debited for the cash value of the returned items and the Wash/Suspense Account is credited in the same amount.

If the wash account has a non-zero balance, scanning the debit and credit entries in the account will show the non-matched items. That is, debits not matched by offsetting credits indicate items intended to be returned but not actually returned yet. The reverse (credits not matched by offsetting debits) indicates that returns were made but the original charge was not recorded in the Wash Account.

These differences can be cleared up by returning unreturned items or recording charges (debits) for items already returned. The mechanics of doing that likely will be finding the original expense account the item was charged to and making an entry like: debit Wash Account, credit original expense. It also could be as described above where the original recording is adjusted by adding a charge to Wash/Suspense account and decreasing the amount charged to the original account.

### **2.16.4 Short or Long-term Assets**

This section explains why some types of assets may be short or long-term and presents an example.



An example is deposits (e.g., utility, rental, security). If the deposit agreement contains a provision to recover the deposit at the end of a year, the treatment could be that of a short-term asset. However, when the agreement is that the deposit holder returns the funds only upon successful inspection at the end of the relationship, then at the start of the relationship or agreement, the person paying the deposit has to decide whether to write it off as a current expense or to track it for eventual recovery at the end of the agreement (not infrequently, moving to a new location).

Whichever decision is made, the accounting treatment is to debit (increase) expense (assuming the write-off decision) or debit (increase) Deposits Receivable (assuming the intent is to recover the deposit in the future) and credit (decrease) Bank for the amount of the deposit (if paid by cash) or credit (increase) credit card if paid using that payment method.

### 2.16.5 Long-term (Fixed) Assets

This section illustrates long-term assets (those whose useful lives exceed a year) and discusses these types: land, buildings, leasehold improvements, intangibles, vehicles and other equipment.

#### Land

Land is not a wasting asset. That is, it does not get used up over time and rarely suffers damage such that it loses value. For that reason, it usually is recorded at cost at the time of purchase. Appreciation in its value over decades is not recorded and is not recognized in any way on the books of the owner. It is only after land has been sold that sale price and purchase cost are compared to calculate gain or loss on sale.

Land is frequently sold/purchased in combination with structures upon it. That means that the cost has to become separated from the cost of structures on it. Land valuation is usually part of the transfer of ownership process and its value is shown on the purchase documents separately from that of any structures it supports.

Land values shown on purchase documents frequently arise from the process of value determination managed by assessors whose job it is to assign values to land for tax purposes. Local and regional areas of a state or province use the values determined by assessors in their tax formulas, which provide revenues for local and regional governing authorities to finance their required community services.

Should land be acquired in a situation not subject to a history of land valuation by a formal valuation system, then the purchaser can appeal to real estate agents and an examination of recent sale transactions for information that would allow calculating a reasonable amount to express the value of the land.

#### Buildings

Buildings are the man-made “caves” in which much of life’s human interaction occurs. These structures are wasting assets, because in their use they or their components gradually wear. Over time they begin to lose some of their function and they can suffer damage due to planetary elements or human action.

Accepted accounting practice is to record the cost of the building determined at time of ownership transfer (purchase) or at conclusion of all costs of construction. Because buildings are frequently used for decades, and due to the need to be able to calculate gain or loss on sale, accounting practice preserves the original cost by not recording declines in value in the account containing the original purchase or construction cost.

Instead, the depreciation technique is used to show (in the balance sheet) the structure’s net book value (original cost reduced by accumulated depreciation). Depreciation is a separate topic treated elsewhere in this Guide.

#### Leasehold Improvements

When a business does not own the building where it operates, and instead has a long-term lease, it is not uncommon for the business tenant to make improvements to the premises so that the structure obtains both function and appearance that enhances conducting its business activities.

In these cases, the expenditures that the business incurs are recorded in a Leasehold Improvements account: increase (debit) Leasehold Improvements, decrease (credit) Bank or increase (credit) a suitable liability account (which could be a liability to a contractor or a bank or a credit card, etc.).

## **Vehicles or Equipment**

Vehicles or Equipment of all kinds usually last for several years, but their useful lives are much shorter than that of assets that have little movement in their functioning. Because they do wear out over time, common accounting practice in business is to record depreciation using life spans and depreciation methods appropriate to the nature and use of the asset. Frequently, the life and depreciation methods chosen are influenced by what is permitted per national tax regulations for the kind of asset being depreciated.

Usually, businesses depreciate their assets. Individuals can do so as well to the degree that taxing authorities permit. Very wealthy persons employ accountants and attorneys to track and manage their investments and assets holdings to take advantage of all tax benefits permitted by law.

## **Intangibles**

The mechanics of accounting (debiting and crediting appropriate accounts) for these assets are relatively simple, much the same as for any of the above assets. Where the difficulty lies is in their valuation, which is an advanced topic and not something that individual persons and small businesses would likely encounter. For that reason further discussion of items such as patents, copyrights, goodwill, etc. are left out of this Guide.

## **2.17 Depreciation**

This chapter will introduce the concept of depreciation in accounting and give some real life examples for using it.

### **2.17.1 Basic Concepts**

Depreciation is the accounting method for expensing capital purchases over time. There are two reasons that you may want to record depreciation; you are doing bookkeeping for your own personal finances and would like to keep track of your net worth, or you are doing bookkeeping for a small business and need to produce a financial statement from which you will prepare your tax return.

The method of recording depreciation is the same in either case. but the end goal is different. This section will discuss the differences between the two. But first, some terminology.

- *Accumulated depreciation* - the accumulated total of book depreciation taken over the life of the asset. This is accumulated in the depreciation account in the asset section.
- *Book depreciation* - this is the amount of depreciation that you record in your financial statements per accounting period.
- *Fair market value* - the amount for which an asset could be sold at a given time.
- *Net book value* - this is the difference between the original cost and the depreciation taken to date.
- *Original cost* - this is the amount that the asset cost you to purchase. It includes any cost to get the asset into a condition in which you can use it. For example - shipping, installation costs, special training.
- *Salvage value* - this is the value that you estimate the asset can be sold for at the end of its useful life (to you).
- *Tax depreciation* - this is the amount of depreciation that you take for income tax purposes.

## Personal Finances

Depreciation is used in personal finances to periodically lower an asset's value to give you an accurate estimation of your current net worth. For example, if you owned a car you could keep track of its current value by recording depreciation every year. To accomplish this, you record the original purchase as an asset, and then record a depreciation expense each year (See *Example* for an example). This would result in the net book value being approximately equal to the fair market value of the asset at the end of the year.

Depreciation for personal finance has no tax implications, it is simply used to help you estimate your net worth. Because of this, there are no rules for how you estimate depreciation, use your best judgement.

For which assets should you estimate depreciation? Since the idea of depreciation for personal finances is to give you an estimate of your personal net worth, you need only track depreciation on assets of notable worth that you could potentially sell, such as a car or boat.

## Business

As opposed to personal finance where the goal is tracking personal worth, business is concerned with matching the expense of purchasing capital assets with the revenue generated by them. This is done through book depreciation. Businesses must also be concerned with local tax laws covering depreciation of assets. This is known as tax depreciation. The business is free to choose whatever scheme it wants to record book depreciation, but the scheme used for tax depreciation is fixed. More often than not this results in differences between book and tax depreciation, but steps can be taken to reduce these differences.

Now, what purchases should be capitalized? If you expect something that you purchase to help you earn income for more than just the current year, then it should be capitalized. This includes things like land, buildings, equipment, automobiles, and computers - as long as they are used for business purposes. It does not include items that would be considered inventory. So if you made a purchase with the intent to resell the item, it should not be capitalized.

In addition to the purchase of the asset itself, any costs associated with getting the asset into a condition so that you can use it should be capitalized. For example, if you buy a piece of equipment and it needs to be shipped from out of town, and then some electrical work needs to be done so you can plug the machine in, and some specialized training is needed so you know how to use the machine, all these costs would be included in the cost of the equipment.

You also need to know the estimated salvage value of the asset. Generally, this is assumed to be zero. The idea behind knowing the salvage value is that the asset will be depreciated until the net book value (cost less depreciation) equals the salvage value. Then, when the asset is written off, you will not have a gain or loss resulting from the disposal of the asset.

The last step is to determine the method of depreciation that you want to use. This will be discussed on the next few pages.

**Warning:** Be aware that different countries can have substantially different tax policies for depreciation; all that this document can really provide is some of the underlying ideas to help you apply your “favorite” tax/depreciation policies.

### 2.17.2 Estimating Valuation

A central issue with depreciation is to determine how you will estimate the future value of the asset. Compared to the often uncertain estimates one has to do where appreciation of assets is concerned, we are on somewhat firmer ground here. Using sources listed below should make it fairly straight forward to estimate the future value of your depreciating assets.

- *Tax Codes*: For businesses that want to use depreciation for tax purposes, governments tend to set up precise rules as to how you are required to calculate depreciation. Consult your local tax codes, which should explicitly state how to estimate depreciation.
- *Car Blue Book*: For automobiles, it is easy to look up in references such as “Blue Books” estimates of what an automobile should be worth after some period of time in the future. From this you will be able to develop a model of the depreciation.

## Depreciation Schemes

A *depreciation scheme* is a mathematical model of how an asset will be expensed over time. For every asset which undergoes depreciation, you will need to decide on a depreciation scheme. An important point to keep in mind is that, for tax purposes, you will need to depreciate your assets at a certain rate. This is called tax depreciation. For financial statement purposes you are free to choose whatever method you want. This is book depreciation. Most small businesses use the same rate for tax and book depreciation. This way there is less of a difference between your net income on the financial statements and your taxable income.

This section will present 3 of the more popular depreciation schemes: *linear*, *geometric*, and *sum of digits*. To simplify the examples, we will assume the salvage value of the asset being depreciated is zero. If you choose to use a salvage value, you would stop depreciating the asset once the net book value equals the salvage value.

1. *Linear depreciation* diminishes the value of an asset by a fixed amount each period until the net value is zero. This is the simplest calculation, as you estimate a useful lifetime, and simply divide the cost equally across that lifetime.

Example: You have bought a computer for \$1500 and wish to depreciate it over a period of 5 years. Each year the amount of depreciation is \$300, leading to the following calculations:

Table 24: Linear Depreciation Scheme Example

<i>Year</i>	<i>Depreciation</i>	<i>Remaining Value</i>
0	•	1500
1	300	1200
2	300	900
3	300	600
4	300	300
5	300	0

2. *Geometric depreciation* is depreciated by a fixed percentage of the asset value in the previous period. This is a front-weighted depreciation scheme, more depreciation being applied early in the period. In this scheme the value of an asset decreases exponentially leaving a value at the end that is larger than zero (i.e.: a resale value).

Example: We take the same example as above, with an annual depreciation of 30%.

Table 25: Geometric Depreciation Scheme Example

<i>Year</i>	<i>Depreciation</i>	<i>Remaining Value</i>
0	•	1500
1	450	1050
2	315	735
3	220.50	514.50
4	154.35	360.15
5	108.05	252.10

**Note:** Beware: Tax authorities may require (or allow) a larger percentage in the first period. On the other hand, in Canada, this is reversed, as they permit only a half share of “Capital Cost Allowance” in the first year. The result of this approach is that asset value decreases more rapidly at the beginning than at the end which is probably more realistic for most assets than a linear scheme. This is certainly true for automobiles.

3. *Sum of digits* is a front-weighted depreciation scheme similar to the geometric depreciation, except that the value of the asset reaches zero at the end of the period. This is a front-weighted depreciation scheme, more depreciation being applied early in the period. This method is most often employed in Anglo/Saxon countries. Here is an illustration:

Example: First you divide the asset value by the sum of the years of use, e.g. for our example from above with an asset worth \$1500 that is used over a period of five years you get  $1500/(1+2+3+4+5)=100$ . Depreciation and asset value are then calculated as follows:

Table 26: Sum of Digits Depreciation Scheme Example

<i>Year</i>	<i>Depreciation</i>	<i>Remaining Value</i>
0	•	1500
1	$100*5=500$	1000
2	$100*4=400$	600
3	$100*3=300$	300
4	$100*2=200$	100
5	$100*1=100$	0

### 2.17.3 Account Setup

As with most accounting practices, there are a number of different ways to setup depreciation accounts. We will present here a general method which should be flexible enough to handle most situations. The first account you will need is an *Asset Cost* account (GnuCash account type *Asset*), which is simply a place where you record the original purchase of the asset. Usually this purchase is accomplished by a transaction from your bank account.

In order to keep track of the depreciation of the asset, you will need two depreciation accounts. The first is an *Accumulated Depreciation* account in which to collect the sum of all of the depreciation amounts, and will contain negative values. In GnuCash, this is an account type *asset*. The *Accumulated Depreciation* account is balanced by a *Depreciation Expense* account, in which all periodic depreciation expenses are recorded. In GnuCash, this is an account type *expense*.

Below is a generic account hierarchy for tracking the depreciation of 2 assets, *ITEM1* and *ITEM2*. The *Asset Cost* accounts are balanced by the *Bank* account, the *Accumulated Depreciation* account is balanced by the *Expenses:Depreciation* account.

```
-Assets
  -Fixed Assets
    -ITEM1
      -Cost                (Asset Cost account)
      -Depreciation        (Accumulated Depreciation account)
    -ITEM2
      -Cost                (Asset Cost account)
      -Depreciation        (Accumulated Depreciation account)
  -Current Assets
    -Bank
```

(continues on next page)

(continued from previous page)

-Expense	
-Depreciation	(Depreciation Expense account)

One of the features of the account hierarchy shown above is that you can readily see some important summary values about your depreciating asset. The *Assets:Fixed Assets:ITEM1* account total shows you the current estimated value for item1, the *Assets:Fixed Assets:ITEM1:Cost* shows you what you originally paid for item1, *Assets:Fixed Assets:ITEM1:Depreciation* shows you your accrued depreciation for item1, and finally, *Expenses:Depreciation* demonstrates the total accrued depreciation of all your assets.

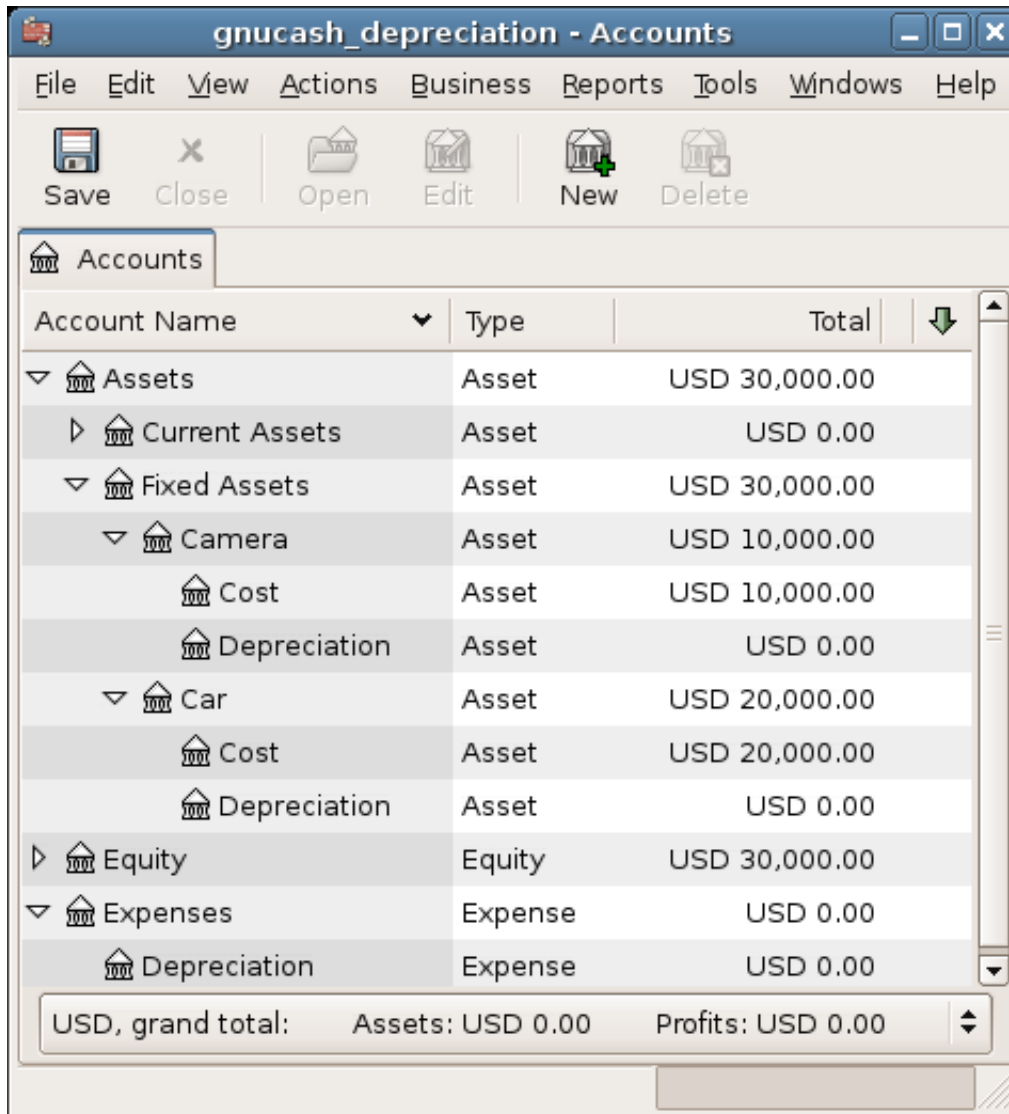
It is certainly possible to use a different account hierarchy. One popular account setup is to combine the *Asset Cost* and *Accrued Depreciation* asset accounts. This has the advantage of having fewer accounts cluttering your account hierarchy, but with the disadvantage that to determine some of the summary details mentioned in the paragraph above you will have to open the account register windows. As with most things, there are many ways to do it, find a way that works best for you.

The actual input of the depreciation amounts is done by hand every accounting period. There is no way in GnuCash (as of yet) to perform the depreciation scheme calculations automatically, or to input the values automatically into the appropriate accounts. However, since an accounting period is typically one year, this really is not much work to do by hand.

### 2.17.4 Example

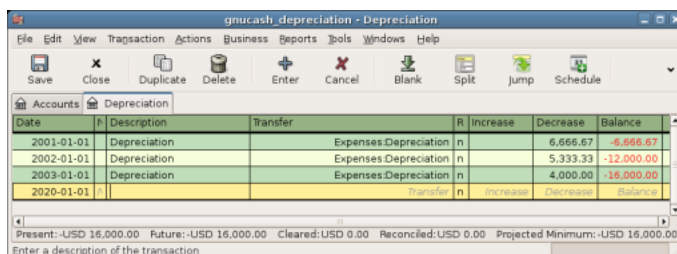
Let's go ahead and step through an example. Imagine you are a photographer and you use a car and an expensive camera for your personal business. You will want to track the depreciation on these items, because you can probably deduct the depreciation from your business taxes.

The first step is to build the account hierarchy (as shown in the previous section, replace *ITEM1* and *ITEM2* with "car" and "camera"). Now, record the purchase of your assets by transferring the money from your bank account to the appropriate *Asset Cost* accounts for each item (eg: the *Assets:Fixed Assets:Car:Cost* account for the car). In this example, you start with \$30k in the bank, the car cost \$20k and the camera cost \$10k and were both purchased on January 1, 2000.



Looking at the tax codes, we realize that we must report depreciation on these items using the “sum of digits” scheme, over a 5 year period. So, the yearly depreciation amounts for the car come to \$6667, \$5333, \$4000, \$2667, \$1333 for years 1 to 5 respectively, rounded to the nearest dollar. The yearly depreciation amounts for the camera are \$3333, \$2667, \$2000, \$1333, \$667. Consult the previous section on Depreciation Schemes for the formula for calculating these values.

For each accounting period (IE: fiscal year) you record the depreciation as an expense in the appropriate *Accrued Depreciation* account (eg: the *Assets:Fixed Assets:Car:Depreciation* account for the car). The two windows below show your car’s accrued depreciation account and the main window after the third year (IE: three periods) of depreciation using this sum of digits scheme.



Account Name	Type	Total
Assets	Asset	USD 6,000.00
Current Assets	Asset	USD 0.00
Fixed Assets	Asset	USD 6,000.00
Camera	Asset	USD 2,000.00
Cost	Asset	USD 10,000.00
Depreciation	Asset	-USD 8,000.00
Car	Asset	USD 4,000.00
Cost	Asset	USD 20,000.00
Depreciation	Asset	-USD 16,000.00
Equity	Equity	USD 30,000.00
Expenses	Expense	USD 24,000.00
Depreciation	Expense	USD 24,000.00

USD, grand total: Assets: -USD 24,000.00 Profits: USD 0.00

**Note:** A Word of Caution: Since depreciation and tax issues are closely related, you may not always be free in choosing your preferred method. Fixing wrong calculations will cost a whole lot more time and trouble than getting the calculations right the first time, so if you plan to depreciate assets, it is wise to make sure of the schemes you will be permitted or required to use.

## 2.18 Python Bindings

GnuCash historically has always been a traditional application in the sense that you open it, use it to manipulate your financial data via the windows it presents, save your data and close the windows again. This has the inherent limitation that you can only do whatever the windows, menus and toolbars allow you to do.

Sometimes you might need a little more flexibility. For example, you need a report with just a little different information than what the built-in reports provide, or you want to automate a frequently recurring action. Such custom manipulations are ideal candidates to write in one or the other scripting language.



Starting with GnuCash version 2.4 you can write Python scripts to manipulate your financial data.

---

**Note:** The Python extensions are an optional feature in the source code. To be able to use Python scripts, GnuCash must have been compiled with this option enabled, otherwise all what follows won't work. At present this option is not enabled by default, so if you need this, you may have to compile GnuCash from source yourself.

---

The Python extensions come with a couple of ready to use scripts. This chapter will show you how to use some of these.

---

**Note:** This chapter is not about how to write your own Python scripts. Refer to the developer documentation for that instead.

---

## 2.19 Importing Business Data

### 2.19.1 Importing Bills and Invoices

#### General

This functionality creates invoices or bills from a csv import file containing rows of invoice entry data. The import file may contain rows for new and/or existing invoices. If an invoice already exists, GnuCash adds the imported entries to the invoice (unless the invoice is already posted). If the import file contains posting data for an invoice, then GnuCash will also attempt to post the invoice. If any row of an invoice contains an error, GnuCash will ignore all rows of the same invoice.

The field separator in the csv file must be either a comma or a semicolon; field values may be enclosed in double quotes.

For the sake of readability, in this chapter the term “invoice” by itself is used to refer to both customer invoices and vendor bills.

#### The format of the import file

The import file should contain rows of invoice entry data. Any row contains both header and entry fields, but GnuCash takes the invoice header data from the first row of an invoice ID. For informational purposes, the header data may be repeated for each subsequent row of the same invoice.

There is no information in the file to indicate whether it concerns customer invoice or vendor bill data. Instead, a user option in the import dialog makes that distinction.

Each row should contain the fields listed below, in the same sequence, separated by a comma or a semicolon. The fields are listed here by their technical name, which GnuCash uses in the preview of the import data.

- `“id”` - The invoice ID. If the invoice ID is blank, GnuCash replaces it with the invoice ID from the previous row. If the invoice ID already exists, GnuCash will add the entries to the existing invoice (unless it is already posted).
- `“date_opened”` - Use the same date format as defined in Preferences. Defaulted to today's date if left blank, or if the date provided is not valid.
- `“owner_id”` - Customer or vendor number. Mandatory in the first data row of an invoice. If not provided, all rows of the same invoice will be ignored.
- `“billingid”` - Billing ID. Optional

- “*notes*” - Invoice notes. Optional.
- “*date*” - The date of the entry. Defaulted to date opened if left blank, or if the date provided is not valid.
- “*desc*” - Description. Optional
- “*action*” - Action. Optional
- “*account*” - Account for the entry. Mandatory in each row. If not provided or invalid, all rows of the same invoice will be ignored.
- “*quantity*” - Quantity. Defaulted to 1 if left blank.
- “*price*” - Price. Mandatory for each row. If not provided, all rows of the same invoice will be ignored.
- “*disc\_type*” - Type of discount. Optional. Only relevant for invoices, not for bills. Use “%” or blank for percentage value, anything else for monetary value.
- “*disc\_how*” - Discount how. Optional. Only relevant for invoices, not for bills. Use “>” for discount applied after tax, “=” for discount and tax applied before tax, and “<”, blank or anything else for discount applied before tax.
- “*discount*” - Amount or percentage of discount. Optional. Only relevant for invoices, not for bills
- “*taxable*” - Is this entry taxable? Optional. Use “Y” or “X” for yes, “N” or blank for no.
- “*taxincluded*” - Is tax included in the item price? Optional. Use “Y” or “X” for yes, “N” or blank for no.
- “*tax\_table*” - Tax table. Optional. If the tax table provided does not exist, it will be blank in the invoice.
- “*date\_posted*” - Date posted. Optional. Use the same date format as defined in Preferences. If you provide a date posted for the first row of an invoice, GnuCash will attempt to also post the invoice (as opposed to only saving or updating it).
- “*due\_date*” - Due date. Optional. Use the same date format as defined in Preferences. Defaulted to date posted, if left blank. Only relevant in the first row of an invoice, if the invoice is posted.
- “*account\_posted*” - Post to account, for vendor or customer posting. Only mandatory in the first row of an invoice, if the invoice is posted.
- “*memo\_posted*” - Memo. Optional. Only relevant in the first row of an invoice, if the invoice is posted.
- “*accu\_splits*” - Accumulate splits? Optional. Use “Y” or “X” for yes, “N” or blank for no. Only relevant in the first row of an invoice, if the invoice is posted. If you use a spreadsheet program to create the import file, it is advised not to use blank for no, because a final column with only blanks may not be recognized as relevant data when the spreadsheet program creates the csv file.

---

**Note:** If you use the field separator character within a field, the field value should be enclosed in double quotes. Only for the fields description (desc) and notes, can you also include a double quote within a quoted field value, by doubling the double quote. E.g. "This field value uses the separator, and a "quoted" word", would be imported as This field value uses the separator, and a "quoted" word.

---

Example content for two bills; one of 2 entries, and one of 3 entries. The first is saved and posted, the second only saved. Using comma field separator, decimal point and dd/mm/yyyy date format.

```
1204;15/12/2018;2001;PO 210220;Special delivery;16/12/2018;Pride and Prejudice;pc;
↪Expenses:Books;1;30.00;;;X;A1;17/12/2018;17/1/2019;Liabilities:Accounts Payable;;X
1204;15/12/2018;2001;PO 210220;Special delivery;16/12/2018;Electronic principles;pc;
↪Expenses:Books;1;50.00;;;X;A1;17/12/2018;17/1/2019;Liabilities:Accounts Payable;;X
1205;15/12/2018;2044;PO 21099;16/12/2018;Ultimate Guide;pc;Expenses:Books;1;10.01;;;
↪;;;;;;;;;
```

(continues on next page)

(continued from previous page)

```
1205;15/12/2018;2044;PO 21099;;16/12/2018;Dinner & drinks;pc;Expenses:Dining;1;10.01;;
↪;;;;;;;;
1205;15/12/2018;2044;PO 21099;;16/12/2018;UG course;pc;Expenses:Education;1;10.01;;;;
↪;;;;;;;;
```

Example content for one customer invoice, with one entry, including tax and discount. Using comma field separator, decimal point and dd/mm/yyyy date format. The the value of the description field contains the separator character.

```
20221;16/12/2018;1001;Order 3378;Discount as agreed;4/12/2018;"Accounting part 1, 2";
↪ea;Income:Other Income;1;769.95;%;=;10;X;N;A1;16/12/2018;16/01/2019;Assets:Accounts_
↪Receivable;Posted by import;X
```

## Import your data

To import your invoice data, navigate to File > Import > Import Bills & Invoices... to open a new import dialog, and provide the necessary information.

- 1. Choose the file to import - Select your import file, or manually type the path and file name.
- 2. Select import type - Select the import type, either Bill or Invoice.
- 3. Select import options - Select your csv format. Use the with quotes options if your file contains fields enclosed in double quotes. These options also match fields not enclosed in quotes; except for the fields for description and notes, fields should not contain the quote character itself. See *note\_title* above. Use one of the other options if your file does not have fields enclosed in quotes; any quote characters in the file will be imported as is.
- 4. Preview - Once you have selected your import file and csv format, GnuCash shows you a preview of the data. You can verify if your data is listed in the correct columns. If you do not see any rows in the preview, then GnuCash was not able to match your import data rows to the selected csv format. See *What could go wrong?* below.
- 5. Afterwards - You can choose if GnuCash should open tabs for the invoices after the import. Either for all invoices, or for the invoices that are saved but not posted, or for none of the invoices. Opening tabs slows down the import process considerably.
- *Start the import* - If you are satisfied with your selections, hit the OK button to start the import.

If your data file contains invoice IDs that already exist, then GnuCash will ask you (once per import session) to confirm that you want to update existing invoices. If not confirmed, all rows for existing invoices will be ignored.

**Note:** Internally, GnuCash uses so called regular expressions to match the import rows to the data fields. The import option Custom regular expression offers the option to use your own regular expression for this matching process. Obviously, this option requires that you are well versed in regular expressions. When you choose the option Custom regular expression, GnuCash opens a window in which you can edit the GnuCash regular expression, or replace it with your own. Your regular expression should contain a named subpattern for each of the fields of the csv file (using the technical names). A custom regular expression could be useful if the rows of your source data file contain all required fields, but in a different order or format. E.g. if the format of your source data file starts with customer number, followed by invoice ID, followed by the due date, and uses | as separator, your regular expression would start with something like this:

```
^(?<owner_id>[^\|]*)\|(?<id>[^\|]*)\|(?<due_date>[^\|]*)
```

With a custom regular expression, GnuCash could import your source data files, without the need to convert them to the GnuCash import format.

### Feedback and statistics

GnuCash executes the import process in three steps:

- “*Import*” - Imports the data file and attempts to match each row to the data fields.
- “*Validation and adjustment*” - Validates the data fields and replaces data with defaults if applicable.
- “*Processing*” - Handles the currency related validations, and creates, updates and posts the invoices.

After all steps have finished, GnuCash issues information about the result of the process. The initial dialog shows the informational or error messages from the validation and processing steps. The second dialog shows the statistics of the process:

- Import - rows ignored: the number of rows that could not be matched to the data fields.
- Import - rows imported: the number of rows that were successfully matched to the data fields.
- Processing and validation - rows fixed: the number of rows for which a default value was used for a field.
- Processing and validation - rows ignored: the number of rows for that were not processed because of a validation error.
- Processing and validation - invoices created: the number of invoices created.
- Processing and validation - invoices updated: the number of invoices that were updated.

If there were unmatched rows in the import step, a final dialog shows the actual rows that could not be matched.

### What could go wrong?

#### Errors in the import step

If the statistics show unmatched rows under “Import - rows ignored”, then there is some issue with the format of your import file. Verify that you use and select the correct separator. Verify that your data rows have exactly 21 separator characters (1 for each field, except for the last). Verify whether you use the separator character within a data field; if so, enclose the field in double quotes.

If you use one of the with quotes import options, verify if you use the double quote character in any of the data field values; if within the description or notes fields, make sure that the field value is quoted, and precede each double quote within the field with an extra double quote; if within any other field, remove the double quote character.

#### Errors in the validation step

The following errors can occur in the validation step. Any error in a data row will cause all rows of the same invoice to be ignored.

---

**Note:** In versions 3.4 and earlier, an error in a data row would cause just that row to be ignored, as opposed all rows of the same invoice.

---

- The field `ID` is blank. GnuCash replaces a blank invoice ID with the invoice ID from the previous row. But the first row of the import file should always have an invoice ID.
- The field `owner_id` is blank. Every first row of an invoice should have an `owner_id`.
- The customer or vendor number in field `owner_id` does not exist. The `owner_id` in the first row of an invoice should be an existing customer (for invoices) or vendor (for bills).

- The date in field `date_posted` is not a valid date. If you provide a value for `date_posted` in the first row of an invoice, it should be a valid date. Did you use the date format as set in Preferences?
- The account in the field `account_posted` does not exist. If you provide a value for the field `date_posted` in the first row of an invoice, the field `account_posted` should be an existing account.
- The account in the field `account_posted` is not of type Accounts Receivable (for invoices) or Accounts Payable (for bills). If you provide a value for the field `date_posted` in the first row of an invoice, the field `account_posted` should be an account of the correct type.
- The field `price` is blank. Every row should have a value for the field `price`.
- The account in the field `account` does not exist. Every row should have an existing account in the field `account`.

Any error in the validation step is listed after the overall import process completes. Correct your data file accordingly.

### Errors in the processing step

The following errors can occur in the processing step.

- The invoice cannot be updated because it is already posted. All rows of the same invoice will be ignored. If you want to update the existing invoice, unpost it first in GnuCash.
- The currency of the invoice differs from the currency of the account posted (“Invoice x NOT posted because currencies don’t match”). GnuCash determines the currency of the invoice either from the customer or vendor master data (for a new invoice) or from the invoice itself (for an existing invoice). The currency of the invoice must agree with the currency of the post to account in the field `account_posted`. GnuCash creates the invoice but cannot post it. Manually correct the invoice in GnuCash.
- The invoice requires currency conversion. (“Invoice x NOT posted because it requires currency conversion”). The invoice contains entries on accounts with different currencies, or the currency of the entries differs from the currency of the post to account. For such an invoice, GnuCash needs exchange rates to translate the currency amounts. GnuCash creates the invoice but cannot post it. Post the invoice manually in GnuCash, and provide the requested exchange rates.

### Not supported invoice functionality

Currently the invoice import function does not support (at least) the following:

- Import of billing terms and job.
- Import of customer and job in default chargeback project for bills.
- Application of billing terms from customer or vendor master data.
- Automatic numbering of invoices.
- Credit notes.

## 2.19.2 Importing Customers and Vendors

### General

This functionality creates and updates customers and vendors from a csv import file containing rows of vendor/customer master data. The import file may contain rows for new and/or existing customers/vendors. If a customer/vendor already exists, GnuCash updates the existing customer/vendor.

## The format of the import file

The import file should contain rows of customer/vendor data, one row for each customer/vendor. The customer/vendor is identified by the customer/vendor number in the field `id` of the data rows. If the field is blank, GnuCash will use the next number from the relevant counter (set in the Counters tab under File > Properties).

There is no data in the file to indicate whether it concerns customer or vendor master data. Instead, a user option in the import dialog makes that distinction.

Each row should contain the fields listed below, in the same sequence, separated by a comma or a semicolon. The fields are listed here by their technical name, which GnuCash uses in the preview of the import data.

- `“id”` - The customer/vendor number. If it is for an existing customer/vendor, GnuCash will update the customer/vendor. Note that in GnuCash e.g. ‘000010’ is a different customer number than ‘10’. If the id field is empty, GnuCash will use the next number from the relevant counter.
- `“company”` - The company name. If it is left blank, it is defaulted to the value of field `name`. If that is also blank, then the row is ignored.
- `“name”` - Billing address - Name. Optional.
- `“addr1”` - Billing address - Address line 1. At least one of the four address lines of the billing address must be filled. If not, then the row is ignored.
- `“addr2”` - Billing address - Address line 2.
- `“addr3”` - Billing address - Address line 3.
- `“addr4”` - Billing address - Address line 4.
- `“phone”` - Billing address - Phone. Optional
- `“fax”` - Billing address - Fax. Optional
- `“email”` - Billing address - Email. Optional
- `“notes”` - Notes. Optional
- `“shipname”` - Shipping information - Name. Optional. Not relevant for vendors.
- `“shipaddr1”` - Shipping information - Address line 1. Optional. Not relevant for vendors.
- `“shipaddr2”` - Shipping information - Address line 2. Optional. Not relevant for vendors.
- `“shipaddr3”` - Shipping information - Address line 3. Optional. Not relevant for vendors.
- `“shipaddr4”` - Shipping information - Address line 4. Optional. Not relevant for vendors.
- `“shipphone”` - Shipping information - Phone. Optional. Not relevant for vendors.
- `“shipfax”` - Shipping information - Fax. Optional. Not relevant for vendors.
- `“shipmail”` - Shipping information - Email. Optional. Not relevant for vendors.

Example content for a customer with a separate shipping address. Using a semicolon for separator.

```
2201;All Star Company;All Star Company;Union Avenue 776;San Juan;CA;;  
0482938838;;contact@allstar.com;Last contacted on 4/4/2018.;All Star  
Company; John Alderman, Office 456;Union Avenue 777;San Juan;CA;78998766;;  
alderman@allstar.com
```

Example content for a vendor; no ID given, so GnuCash will take the next number from the counter. Using a comma for separator.

```
,Johnson Supplies,Johnson Supplies,Electric Park 56,Plains,VA,,0482986538,,  
jack@johnson.com,Discount negotiated,,,,,,,,
```

All fields by technical name in the required sequence.

```
id, company, name, addr1, addr2, addr3, addr4, phone, fax, email, notes,
shipname, shipaddr1, shipaddr2, shipaddr3, shipaddr4, shipphone, shipfax,
shipmail
```

## Import your data

To import your customer or vendor data, navigate to File > Import > Import Customer & Vendors... to open a new import dialog, and provide the necessary information.

- 1. Choose the file to import - Select your import file, or manually type the path and file name.
- 2. Select import type - Select the import type, either Customer or Vendor.
- 3. Select import options - Select your csv format. Use the with quotes options if your file contains fields enclosed in double quotes. These options also match fields not enclosed in double quotes, but fields should not contain the double quote character itself. Use one of the other options if your file does not have fields enclosed in quotes; any double quote characters in the file will then be imported as is.
- 4. Preview - Once you have selected your import file and csv format, GnuCash shows you a preview of the data. You can verify if your data is listed in the correct columns. If you do not see any rows in the preview, then GnuCash was not able to match your import data rows to the selected csv format. See [What could go wrong?](#) below.
- *Start the import* - If you are satisfied with your selections, hit the OK button to start the import.

**Note:** Internally, GnuCash uses so called regular expressions to match the import rows to the data fields. The import option Custom regular expression offers the option to use your own regular expression for this matching process. Obviously, this option requires that you are well versed in regular expressions. When you choose the option Custom regular expression, GnuCash opens a window in which you can edit the GnuCash regular expression, or replace it with your own. Your regular expression should contain a named subpattern for each of the fields of the csv file (using the technical names). A custom regular expression could be useful if the rows of your source data file contain all necessary fields, but in a different order or format. E.g. if the format of your source data file starts with customer number, followed by company name, name, and one address field, and that is all you want to import, then your custom regular expression would be something like this (using comma as a separator):

```
^(?<company>[^\,]*) , (?<id>[^\,]*) , (?<name>[^\,]*) , (?<addr1>[^\,]*) , (?<addr2>[^\,]*) , (?
↪<addr3>[^\,]*) $
```

With a custom regular expression, GnuCash could import your source data files, without the need to convert them to the GnuCash import format.

## Feedback and statistics

GnuCash executes the import process in three steps:

- “*Import*” - Imports the data file and attempts to match each row to the data fields.
- “*Validation and adjustment*” - Validates the data fields and replaces data with defaults if applicable.
- “*Processing*” - Creates or updates the vendor or customer master data.

After all steps have finished, GnuCash issues information about the result of the process. The initial dialog shows the statistics of the process:

- Import results - lines ignored: the number of rows that could not be matched to the data fields.

- Import results - lines imported: the number of rows that were successfully matched to the data fields.
- Import results - customers/vendors fixed: the number of rows for which a default value was used for a field.
- Import results - customers/vendors ignored: the number of rows for that were not processed because of a validation error.
- Import results - customers/vendors created: the number of customers/vendors created.
- Import results - customers/vendors updated: the number of customers/vendors that were updated.

If there were unmatched rows in the import step, a final dialog shows the actual rows that could not be matched.

### What could go wrong?

#### Errors in the import step

If the statistics show unmatched rows under “Import results - lines ignored”, then there is some issue with the format of your import file. Verify that you use and select the correct separator. Verify that your data rows have exactly 18 separator characters (1 for each field, except for the last). Verify whether you use the separator character within a data field; if so, enclose the field in double quotes.

If you use one of the with quotes import options, verify if you use the double quote character in any of the data field values. If so, remove them; importing double quotes as is, is not supported when using the with quotes import options.

#### Errors in the validation step

If the statistics show rows under “Import results - customers/vendors ignored”, then data rows were ignored because of one of the errors below:

- The field `company` and the field `name` are both blank. The field `company` is mandatory; if it is blank, then it is defaulted to the value of the field `name`, but if both are blank, then the data row cannot be processed.
- The fields `addr1`, `addr2`, `addr3` and `addr4` are all blank. At least one of these fields must have a value, otherwise the data row cannot be processed.

#### Not supported customer/vendor functionality

Currently the customer/vendor import function does not support (at least) the following:

- Import of any of the fields in the customer tab for billing information: currency, terms, discount, credit limit, tax included and tax table.
- Import of any of the fields in the vendor tab for payment information: currency, terms, tax included and tax table.

## 2.20 GnuCash Glossary

This is a glossary of common terms used in GnuCash. Some entries here are taken from [Wikipedia](#).

**Account** An account keeps track of what you own, owe, spend or receive.

**Accounting Period** An accounting period is the period with reference to which accounting books of an entity are prepared. Common accounting periods include month, quarter, and year.



**AqBanking** A library which implements the German Home Banking Computer Interface (HBCI) and Electronic Banking Internet Communication Standard (EBICS), OFX Direct Connect and Paypal. It is the successor of openHBCI.

**Asset** An asset is something you own. Anything tangible or intangible that can be owned or controlled to produce value and that is held to have positive economic value is considered an asset.

**Balance Sheet** A Balance Sheet is a summary of the financial balances of an individual or organization. It summarizes a company's assets, liabilities and ownership equity at a specific point in time.

**Book** A book is a record of all transactions for an individual or organization. In GnuCash, each file contains a book.

**Capital Gain/Loss** The difference between the purchase and selling prices of an investment at the time the investment is sold. Also known as *Realized Gain/Loss*. Before an investment is sold, the difference in value is referred to as *Unrealized Gain/Loss*.

See also Realized Gain/Loss.

See also Unrealized Gain/Loss.

**Capital stock** See Stock.

**Chart of Accounts** A listing of all the accounts available in the data file. The chart of accounts consists of balance sheet accounts (assets, liabilities, stockholders' equity) and income statement accounts (revenues, expenses, gains, losses).

**Commission** The fee paid to a broker to buy or sell securities.

**Commodity** A commodity is something of value that is easily tradeable or sellable; for example, currencies, stocks, bonds, grain, copper, and oil are all commodities.

**Common stock** See Stock.

**Compounding** The concept that the reinvested interest can later earn interest of its own (interest on interest). This is often referred to as *compound interest*.

**Cost Basis** Cost basis is the original cost of property, adjusted for various factors. Its primary use is for tax purposes. Factors that affect the cost basis include: stock splits, dividends, depreciation and return of capital distributions. Cost basis is used to determine the *capital gain*.

**CSV** Stands for Comma Separated Values. CSV files are used to store data in plain text. Each line of the file is a record and each record can be comprised of multiple fields separated by commas. CSV is one import format that GnuCash supports.

**Dividends** Dividends are cash payments a company makes to shareholders. The amount of this payment is usually determined as some amount of the profits of the company. Not all common stocks give dividends.

**Equities** Equities are investments in which the investor becomes part (or whole) owner in something.

**Financial Transaction Services (FinTS)** A banking protocol used by German banks. Developed by the German Central Banking Committee ZKA (Zentraler Kredit-Ausschuss). The standard is used only by German banks. Prior 2002, it was called Home Banking Computer Interface (HBCI). FinTS is one import format that GnuCash supports.

See also Home Computer Banking Interface (HBCI).

**GSettings** Since version 2.5 the tool that stores user configuration data. It uses the native data store of the operating system:

**Linux** DConf since Gnome 3.0 dropped GConf

**MacOS** Defaults. Use the command-line defaults(1) to modify prefs when GnuCash isn't running.

**Windows** Registry.

**Home Computer Banking Interface (HBCI)** A bank-independent online banking protocol used by German banks. Home Banking Computer Interface (HBCI) was developed by the German Central Banking Committee ZKA (Zen-

traler Kredit-Ausschuss). Since 2002, it has been called Financial Transaction Services (FinTS). HBCI is one import format that GnuCash supports.

See also Financial Transaction Services (FinTS).

**Interest** What a borrower pays a lender for the use of their money. Normally, this is expressed in terms of a percentage of the principal per year. For example, a savings account with 1% interest will pay you \$1 for every \$100 you keep deposited per year.

**Liability** A liability is a debt or obligation that an individual or organization owes.

**LibOFX** An open source library for OFX. It was created by one of the GnuCash developers, originally to be the OFX interface for GnuCash.

See also OFX.

**Liquidity** A measure of how easily convertible an investment is to cash. Money in a savings account is very liquid, while money invested in a house has low liquidity because it takes time to sell a house.

**Lot** A lot is a means of grouping a commodity so that you later identify that the item bought in one transaction is the same as one sold in a different transaction. Lots are often implemented with stocks, where capital gain can depend on which item is being sold at a given time.

**MT940** A financial information standard defined by SWIFT and used by several European banks. It is also used internally in HBCI. Unfortunately, you can't download the MT940 standard, but some banks publish it on their websites. MT940 is one import format that GnuCash supports.

**OFX** The Open Financial eXchange format. This is a financial information exchange standard used by many institutions. OFX is one import format that GnuCash supports.

**Price Database** The price database contains a store of price quotes for stocks, mutual funds, and currencies.

**Principal** The original amount of money invested or borrowed.

**QFX** The Quicken Financial eXchange format is a proprietary financial information exchange standard promoted by Quicken and used by many institutions. QFX is one import format that GnuCash supports.

**QIF** The Quicken Interchange Format (QIF) is an open specification for reading and writing financial data to files. This is an older format that is still used by many institutions. QIF is one import format that GnuCash supports.

**Realized Gain/Loss** The difference between the purchase and selling prices of an investment at the time the investment is sold. Also known as *Capital Gain/Loss*. Before an investment is sold, the difference in value is referred to as *Unrealized Gain/Loss*.

See also Capital Gain/Loss.

See also Unrealized Gain/Loss.

**Reconciliation** Reconciliation is a verification process in which the user compares their records with those of the financial institution to ensure that the records in each system agree at some particular point in time. During reconciliation, the user checks individual transactions in their file against the institution's statement; upon completion, the reported balances of the institution and the reconciled account will match at that point in the register.

**Return** The total income plus capital gains or losses of an investment.

See also Yield.

**Reversing Transaction** In formal accounting, a transaction is never deleted. Therefore, when a mistake is made, the original transaction remains in the ledger, and a transaction is added to the ledger that reverses the original. This *reversing transaction* duplicates the original transaction, but with debit and credit amounts reversed. This removes the effect of the erroneous transaction from the books. After entering the reversing transaction, a corrected transaction can then be entered. Reversing transactions are not commonly used in personal accounting.

**Risk** The probability that the return on investment is different from what was expected. Investments are often grouped on a scale from low risk (e.g., savings accounts or government bonds) to high risk (e.g., common stocks or junk bonds). As a general rule of thumb, the higher the risk, the higher the possible return.

**Scheduled Transaction** Scheduled transactions provide a framework for remembering information about transactions that are set to occur in the future, either once or periodically.

See also **Yield**.

**Scheme** Scheme is a functional programming language based on a dialect of LISP. Reports in GnuCash use the Scheme programming language; thus, users wishing to customize reports must write Scheme.

**Shareholder** A shareholder is a person who holds common stock in a company.

**Split** A split, or Ledger Entry, is the fundamental accounting unit. Each split consists of an amount, the value of that amount expressed in a (possibly) different currency, a Memo, a pointer to the parent Transaction, a pointer to the debited Account, a reconciled flag and timestamp, an action field, and a key-value frame which can store arbitrary data.

**Stock** A security that represents a certain fractional ownership of a company. This is what you buy when you “buy stock” in a company on the open market. This is also sometimes known as *capital stock*, or *common stock*.

**Stock split** Occurs when a company offers to issue some additional multiple of shares for each existing stock. For example, a “2 for 1” stock split means that if you own 100 shares of a stock, you will receive an additional 100 shares at no cost to you. The unit price of the shares will usually be adjusted so there is no net change in the value, so in this example the price per share will be halved. Note that such transactions affect the cost basis per share owned.

**Transaction** A transaction consists of a date, a description, an ID number, a list of two or more splits, and a key-value frame. Transactions embody the notion of “double entry” accounting.

**Unrealized Gain/Loss** Unrealized gain or loss is the difference in value between a purchase price and the current value of a given asset. These gains remain unrealized until the asset is sold, at which point they become *Realized gains or losses*.

See also **Capital Gain/Loss**.

See also **Realized Gain/Loss**.

**Valuation** The process of determining the market value or the price the investment would sell at in a “reasonable time frame”.

**XML** The eXtensible Markup Language is an international markup standard. GnuCash stores its data by default in an XML data structure.

**Yield** A measure of the amount of money you earn from an investment (i.e., how much income you receive from the investment). Typically, this is reported as a percentage of the principal amount. Yield does not include capital gains or losses (see **Return**). For example, a stock that sells for \$100 and gives \$2 in dividends per year has a yield of 2%.

## 2.21 Migration Guide

This appendix is to help current users of other financial software packages in their migration to GnuCash. We address the conceptual differences between the layout of GnuCash accounts versus other software packages.

## 2.22 Using Accounts vs. Categories

If you are familiar with other personal finance programs, you are already accustomed to tracking your income and expenses as categories. Since GnuCash is a double-entry system (see [Accounting Concepts](#)), income and expenses are tracked in accounts. The basic concept is the same, but the account structure allows more consistency with accepted

business practices. So, if you are a business user as well as a home user, GnuCash makes it easy to keep track of your business as well as your personal accounts.

Income and expense accounts give you the same information you would get with categories, but they also give you more flexibility in entering your transactions. In GnuCash, you have the option to enter transactions directly into income and expense accounts through their account registers. Other programs that use categories do not offer this option, because there is no “account register” for a category.

You also have the option in GnuCash to treat income and expense accounts exactly as you would treat categories, if you are more comfortable with that method. In Quicken and similar programs, transactions require an account and a category. Substitute an income or expense account name in GnuCash where you would normally enter a category name in the other programs, and the result should be the same. We will discuss transaction entry in *Transactions* in greater detail.

## 2.23 Importing Data

If you want to import data from your previous application, you should distinguish between financial data and other data. Probably the best way to import financial data is the “Quicken Interchange Format” QIF. It is a specific format for financial data, which most financial applications for the private sector know and can use for export.

For other data we suggest the use of the “Comma Separated Value” CSV format.

### 2.23.1 Import of Financial Data by QIF Files

See ‘<[https://en.wikipedia.org/wiki/Quicken\\_Interchange\\_Format](https://en.wikipedia.org/wiki/Quicken_Interchange_Format)>’\_\_ for it’s details.

Some GnuCash users collected their knowledge about the best use in our ‘<[https://wiki.gnucash.org/wiki/Quicken\\_Migration](https://wiki.gnucash.org/wiki/Quicken_Migration)>’\_\_.

For other formats and more details see *Importing Data into GnuCash*.

### 2.23.2 Import of Business Data by CSV Files

If you want to import customers and vendors or bills and invoices see *Importing Business Data*.

## 2.24 Converting XML GnuCash File

The GnuCash XML data file can be transformed to almost any other data format (e.g., QIF, CSV...) quite easily if one is familiar with the “Extensible Stylesheet Language Transformations” *XSLT*. The GnuCash data file is well-formed XML, and it can therefore be run through an XSLT parser with an associated stylesheet. This allows one to transform the file to just about any format that can be designed, given a properly written stylesheet.

A few steps need to be followed. The writing of a stylesheet is a task for a different time, but if you can get one written, here’s what you need to do:

1. Copy the GnuCash XML data file to a working file.

---

**Important:** If the file was last modified by a version of GnuCash older than 2.0, then before you continue to the next step you will need to modify the working file’s <gnc-v2> tag to read something like this:

```
<gnc-v2  xmlns:cd="http://www.gnucash.org/XML/cd"  xmlns:book="http://www.gnucash.org/XML/book"
xmlns:gnc="http://www.gnucash.org/XML/gnc"          xmlns:cmdty="http://www.gnucash.org/XML/cmdty"
xmlns:trn="http://www.gnucash.org/XML/trn"          xmlns:split="http://www.gnucash.org/XML/split"
```

---

```

xmlns:act="http://www.gnucash.org/XML/act"      xmlns:price="http://www.gnucash.org/XML/price"
xmlns:ts="http://www.gnucash.org/XML/ts"        xmlns:slot="http://www.gnucash.org/XML/kvpslot"
xmlns:cust="http://www.gnucash.org/XML/cust"    xmlns:entry="http://www.gnucash.org/XML/entry"
xmlns:lot="http://www.gnucash.org/XML/lot"      xmlns:invoice="http://www.gnucash.org/XML/invoice"
xmlns:owner="http://www.gnucash.org/XML/owner"  xmlns:job="http://www.gnucash.org/XML/job"
xmlns:billterm="http://www.gnucash.org/XML/billterm"  xmlns:bt-days="http://www.gnucash.org/XML/bt-days"
xmlns:sx="http://www.gnucash.org/XML/sx"        xmlns:fs="http://www.gnucash.org/XML/fs"
xmlns:addr="http://www.gnucash.org/XML/custaddr">

```

---

**Note:** You can put pretty much anything you want behind the equal signs, but a unique URL is what is typically used.

---

2. Create an XSLT stylesheet containing the transformation your desire, or obtain one that's already written.
  - in our repository: '<https://github.com/Gnucash/gnucash/tree/maint/contrib/xslt>'\_\_
  - in our wiki: '[https://wiki.gnucash.org/wiki/De/export\\_to\\_excel\\_xls\\_transform](https://wiki.gnucash.org/wiki/De/export_to_excel_xls_transform)'\_\_
3. Install an XSLT processor such as Saxon ('[https://en.wikipedia.org/wiki/Saxon\\_XSLT](https://en.wikipedia.org/wiki/Saxon_XSLT)'\_\_) or Xalan ('[https://en.wikipedia.org/wiki/Apache\\_Xalan](https://en.wikipedia.org/wiki/Apache_Xalan)'\_\_). Any conforming processor will do, really... .
4. Run the work file and the stylesheet through the processor according to the processor's instructions.
5. You will now have a file in the desired output format. An enterprising individual could go so far as to write a stylesheet to transform the GnuCash data file to an OpenOffice spreadsheet (or vice-versa, for that matter). Such things as QIF ought to be a little less work.

Benefits are that you don't need to write a Scheme module or a new C routine to do this transformation. Anyone who knows or can learn XML and XSLT can perform this task. Not much harder, really, than writing a Web page... .

Anyhow, I just wanted this tidbit to be captured somewhere permanently. The process works on 3.10 datafiles, and ought to work on earlier versions, too.

## 2.25 Frequently Asked Questions

This is a list of questions asked on the mailing lists for which there really is no section in the documentation covering the subject.

## 2.26 Sources of Information

**Q:** Where is the FAQ?

**A:** You're looking at an very old copy of it. The most up-to-date copy can be found within the [GnuCash Wiki](#).

**Q:** Are there mailing lists for GnuCash?

**A:** Yes. Go to '<https://lists.gnucash.org/mailman/listinfo/gnucash-user>'\_\_ and '<https://lists.gnucash.org/mailman/listinfo/gnucash-devel>'\_\_ to subscribe.

**Q:** Is there a searchable archive for the mailing lists?

**A:** Yes, you can search the mail list archives at '<http://news.gmane.org/gmane.comp.gnome.apps.gnucash.devel>'\_\_ and '<http://news.gmane.org/gmane.comp.gnome.apps.gnucash.user>'\_\_ (and '<http://news.gmane.org/gmane.comp.gnome.apps.gnucash.german>'\_\_ if you speak German).

**Q:** Are there other means of obtaining support for GnuCash?

**A:** Yes. Many of the developers hang out on icq in the #gnucash discussion on ‘<[irc://irc.gnome.org/gnucash](http://irc.gnome.org/gnucash)>‘ \_\_. Also, there is a [wiki](#) online.

## 2.27 General Information

**Q:** Can I run GnuCash on Windows?

**A:** Yes. Starting with release 2.2.0, GnuCash is also available on Windows.

**A:** Other related options would be colinux, VMWare and a windows-based X-server hosting a remote GnuCash session.

**Q:** I heard it is too hard to compile GnuCash!

**A:** This was probably true at the time when 1.6.0 was released. It is no longer true today, as almost every distribution ships with all the necessary libraries. However, by default, distributions won’t install the development packages of the required libraries, so you might need to start your distribution’s installer program and tell it to install not only the library packages but also the -devel packages. In general, it was noted that this problem concerns many applications in the gnome domain, and this also boils down to the fact that there is no such thing as “one monolithic gnome package”.

**Q:** Is there a batch mode (non-interactive) available for GnuCash, for building reports, etc?

**A:** No, for now GnuCash must be run interactively.

**Q:** Can multiple people access the same datafile in GnuCash?

**A:** You can have multiple people with access to the same datafile, but they cannot use the data file simultaneously.

To setup multi-person access, all the people must have read/write access to the directory containing the file (to read the other’s created files, and to create new files). One way to do this is by creating a user group and setting the data directory to be owned by the shared group and set to mode 2775. The “2” makes the directory setgid which copies the permissions to all files.

**Q:** Why is GnuCash written in C?

**A:** The core functionality of GnuCash is written in C, but do not forget that much of this can be accessed through Guile (scheme). There are a number of reasons for why GnuCash is written in C. The first is historical, GnuCash was started in 1996 (or maybe even earlier!) and many of the OOP (C++, Java, Python) compilers were not yet mature and standardized enough on the variety of platforms considered at that time, so C was the only option at that time. A second reason is because the standard GUI GnuCash uses is GTK, which is written in C.

**Q:** Why don’t you rewrite GnuCash in programming language xyz so that I can contribute easily?

**A:** The quick answer is “We won’t”.

**A:** The longer answer is complex but still amounts to “We won’t”. GnuCash is a large body of code maintained by a small group of developers who are comfortable in C and Scheme (Guile). Actually, 80% of it is in C and approx. 13% is in Scheme/Lisp. There is no valid reason that would justify rewriting this amount of existing code in a newer language. Also, creating language bindings to recent languages such as Python or Ruby or (insert your favourite language here) is labor intensive, and we’re already stretched pretty thin maintaining and developing the existing code.

**A:** Having said that, this is an open source project and you’re free to do with it or contribute what you want. Just don’t expect much support if the reason for your changes is that you’re not willing to learn C or Scheme. Also, GnuCash used to have SWIG bindings (‘<<http://www.swig.org>>‘ \_\_) which have been used for some perl programming code. According to a list discussion, these SWIG bindings might still be a way to include other languages into GnuCash, but currently they are unused and unmaintained.

**Q:** I really want feature XYZ but GnuCash doesn’t have it. How do I get it added?

**A:** Ask nicely. :-) You can file an enhancement request at ‘<<https://bugs.gnucash.org/>>‘\_\_\_. Please bear in mind to describe your proposed enhancement as verbosely as possible. The trick here is to learn how to give the best information to the programmers about what your proposed new feature should do. If you want to speed up development significantly, consider donating some money as described on GnuCashDevelopment.

**Q:** Is there a web interface available for GnuCash?

**A:** No

**Q:** How can I provide security for GnuCash data using CFS, etc.)

**A:** Unanswered

**Q:** How can I contribute to the GnuCash project?

**A:** We’re working on a more formal process, but for now you should subscribe to the mailing list at ‘<<https://lists.gnucash.org/mailman/listinfo/gnucash-user>>‘\_\_ and ‘<<https://lists.gnucash.org/mailman/listinfo/gnucash-devel>>‘\_\_ and discuss what you can contribute with the participants on the lists. Please be aware that GnuCash is a large body of code written in C and Scheme (see “#appendixb\_general\_WhyC” above, if you want to know why). If these are languages that you are not willing to work with, consider contributing in other ways.

**Q:** I think I found a bug. How do I report it?

**A:** First of all, try to verify that it is indeed a bug and that it has not been reported before. Search the mail list archives (see FAQ above). Then search the [GnuCash Bugzilla](#) database.

If you feel you have indeed found a bug, you can then report it at ‘<<https://bugs.gnucash.org/>>‘\_\_\_. Please bear in mind to report your bug as verbosely as possible. The trick here is to learn how to give the best information to the programmers about how to reproduce bugs. A Programmer will usually only be able to fix a bug they can see, if you can’t make the programmer see your bug, it won’t get fixed!

## 2.28 Using GnuCash

**Q:** How can I move the transactions from account “A” into account “B”, thus combining them?

**A:** At present, GnuCash does not offer a way to move groups of splits from one account to another. You will need to move them one at a time. Open the register for account “A” and select the pulldown menu item View > Transaction Journal to expose all the splits. For every split where the “Account” field shows account “A” reset it to account “B”. To do this quickly and safely, first use Ctrl+C to copy the destination account name (“account B”) to the clipboard. Then highlight each reference to account “A” by double clicking on it and use Ctrl+V to paste the destination account name. Pressing Enter after each paste, silently moves the transaction out of the register.

**Caution:** If you inadvertently set the “Account” field to an unintended location, you will need to search through all your accounts to find the lost transaction to correct your mistake.

**Q:** Is it possible to merge two GnuCash files?

**A:** At present this is not possible.

**Q:** How can I save a template of my account structure?

**A:** This is available from the menu: File > Export > Export Accounts

**Q:** When I search for customers (or anything else for that matter), how can I return a list of everything?

**A:** Enter a search criteria of matches regex, and place a single dot “.” in the text field area. Then, click Find. The regular expression “.” means to match anything.

**Q:** How can I record a transaction on different dates (actual date and bank date)?

**A:** You record the transaction on the date you write the check or initiate the transaction. When it “clears” the bank, you can click in the “Reconciled” field to “clear” the transaction (change the “n”on-reconciled to “c”leared).

## 2.29 Accounting

**Q:** How do I treat taxes? As an account payable or as an expense?

**A:** This is a loaded question, and you should really talk to your accountant. How you treat taxes really depends on what kind of taxes they are, and how you WANT to treat them.. In some cases they are expenses, in some cases they are liabilities.

## 2.30 Contributed Account Trees

This is an ancient example. For recent account trees see ‘<<https://github.com/Gnucash/gnucash/tree/maint/data/accounts>>’, select your language and if existing region.

If you want to contribute your new or improved templates, read ‘<[https://wiki.gnucash.org/wiki/Translation#How\\_to\\_translate\\_the\\_files\\_containing\\_the\\_new\\_account\\_hierarchies](https://wiki.gnucash.org/wiki/Translation#How_to_translate_the_files_containing_the_new_account_hierarchies)>’.

## 2.31 UK Vat

Account types (only shown if different to parent type)

- [E] Expense
- [I] Income
- [A] Asset
- [L] Liability
- [Q] Equity
- [B] Bank accounts
- [C] Credit Cards
- [R] Accounts Receivable
- [P] Accounts Payable

(Box n) refers to VAT form box number (I actually have these as descriptions to the account to remind me)

Add all the (Box n -part) together to get the whole (Box n) The VAT shows you liability - if its negative they owe you.

Capital Equipment (Box 7 - part) and (Box 6 - part) is the value of all \*additions\* (purchases) made over the VAT return period - not the absolute value, nor the difference in value unless that difference is wholly due to new purchases. Depreciation, losses (e.g a write off of faulty item) and other reductions in capital value are not included. If you sell a capital item then that sale and its VAT is recorded under Income. The asset is “converted to cash”, so the “net of VAT” increase in your bank account, when the invoice is paid, is matched by a decrease in capital.

```
Bank Accounts [B]
|___ Main Account
|___ Reserve Account

Cash [A]
```

(continues on next page)



(continued from previous page)

Assets [A]  
   |\_\_\_ Capital Equipment       (Box 7 - Part) - additions only, not absolute value  
   |                  |\_\_\_ Computers       Can be depreciated to zero this year  
   |                  |\_\_\_ EU reverse VAT purchase   (Box 6 - Part) create sub-accounts if \_  
↪needed  
   |\_\_\_ Other

Receivable [R] Customers to whom you give credit - (business section)

Cards [C]  
   |\_\_\_ Card 1

Liabilities [L]  
   |\_\_\_ Owed Corp Tax  
   |\_\_\_ Owed Fees  
   |\_\_\_ Owed Tax / NI  
   |\_\_\_ Other

VAT [L]   Net (Box 5)  
   |\_\_\_ i/p [A]   purchases (Box4)  
   |\_\_\_ o/p [L]       (Box3)  
                   |\_\_\_EEC   on reverse VAT purchases (Box 2)  
                   |\_\_\_Sales   all including zero rate UK/ EU   and World (Box1)

Payable [P]   Suppliers who give you credit (business section)

Equity [Q]  
   |\_\_\_ Corp Tax  
   |\_\_\_ Director's Loan  
   |\_\_\_ Dividends  
   |                  |\_\_\_ Director1  
   |                  |\_\_\_ Director2  
   |                  |\_\_\_ Shareholder 1  
   |\_\_\_ Grants (and stuff that does not count as income)  
   |\_\_\_Opening Balances

Income [I]   (Box 6 - part)  
   |\_\_\_ Interest  
   |\_\_\_ Misc  
   |\_\_\_ Sales  
                   |\_\_\_ EU  
                   |          |\_\_\_ goods   (Box 8) (sub accounts as needed)  
                   |          |\_\_\_ services includes software (sub accounts as needed)  
                   |\_\_\_ UK  
                   |\_\_\_ World

Expenses [E]  
   |\_\_\_Depreciation  
   |\_\_\_ Emoluments  
   |                  |\_\_\_ Directors Fees  
   |                  |\_\_\_ NI Employer  
   |                  |\_\_\_ Employee 1  
   |                  |\_\_\_NI  
   |                  |\_\_\_Net Salary  
   |                  |\_\_\_Stakeholder

(continues on next page)

(continued from previous page)

```

|          |___Tax
|___ Other Non VAT Expenses
|___ VAT Purchases      (Box 7 - part)
|       |___ Accountancy
|       |___ Bank Charges
|       |___ Consumables
|       |___ EU reverse VAT purchases (Box 6 - Part)
|       |___ goods (Box 9) (sub accounts as needed)
|       |___ services includes software (sub accounts as
↪needed)
|       |___ Office
|       |___ Phone and Internet
|       |___ Software
|       |___ Subscriptions
|       |___ Sundry
|       |___ Travel / Accom

```

## 2.32 Auxiliary File Formats

These are the formats of some auxiliary files used by GnuCash.

## 2.33 Check Format Files (\* .chk)

### 2.33.1 Overview

The check format file is used to tell GnuCash how to print a check or checks onto a page of paper. This file first describes the overall layout of a page (number of checks, orientation, etc) and then describes the layout of the specific items on a single check. The file is organized as a typical Key/Value file used by many Linux applications. Keys/values pairs are grouped into sections that begin with the group name enclosed in square brackets.

GnuCash looks for check format files in two different locations when you bring up the check printing dialog. The first location is typically `/usr/share/gnucash/checks`, where check files distributed with the application can be found. The second location is the user private `~/ .local/share/gnucash/checks1` directory. Users may add check formats at any time (even while GnuCash is running) simply by dropping a new `*.chk` file in this directory. The next time the check printing dialog is opened the new check format will appear in the list of available check formats.

---

**Note:** Printing functions differently depending on the version of GTK that is installed on your system. When GnuCash is using a version of GTK prior to 2.10 all offsets are measured from the lower left corner of the page or check. When using GTK 2.10 or later, all offsets are measured from the upper left corner of the page or check.

---

### 2.33.2 Example file

A typical GnuCash check file is presented below. The contents of this file will be described in the next sections.

---

<sup>1</sup> Up to GnuCash 2.6.21 it was `~/ .gnucash/checks`

```

[Top]
Guid = 67b144d1-96a5-48d5-9337-0e1083bbf229
Title = Quicken/QuickBooks (tm) US-Letter
Rotation = 0.0
Translation = 0.0;4.0
Show_Grid = false
Show_Boxes = false

[Check Positions]
Height = 252.0
Names = Top;Middle;Bottom

[Check Items]
Type_1 = PAYEE
Coords_1 = 90.0;102.0;400.0;20.0

Type_2 = AMOUNT_WORDS
Coords_2 = 90.0;132.0

Type_3 = AMOUNT_NUMBER
Blocking_Chars_3 = true
Coords_3 = 500.0;102.0

Type_4 = DATE
Coords_4 = 500.0;67.0

Type_5 = NOTES
Coords_5 = 50.0;212.0

```

### 2.33.3 Field Descriptions

#### Top Group

This section of the check file describes the overall layout of a page of checks (or check) that goes into the printer.

Table 27: Overall Page Description Fields

Name	Type	Re- quired	Description
Guid	string	mandatory	The guid is used to uniquely identify a check format to GnuCash. It must be unique across the entire set of application supplied and user supplied check formats. If you copy an application check file as the basis of your own check format, you must change this value. The <i>uuidgen</i> program may be used to generate these identifiers.
Title	string	mandatory	The title is used to uniquely identify a check format to the user. This value is presented verbatim in the check format list of the check printing dialog. If you copy an application check file as the basis of your own check format, you should change this value. The title may be any utf-8 string.
Font	string	optional	If supplied, this is the default font used to print all text items on this check. This field can contain any string that is acceptable by gtk as a font specifier. If this field is omitted, the default font is the font specified in the GnuCash preferences dialog. A typical string would be "sans 12".
Blocking_Chars	boolean	optional	If supplied, this is the default used when printing all <i>TEXT</i> items on this check. When set to true, will print *** before and after each text field on the check. Blocking characters are printed to protect check fields from alteration. For example, the amount field may be printed as * ***100.00***
Date-Format	boolean	optional	If supplied, this is the default used when printing all <i>DATE</i> items on this check. When set to true, will print the format of the DATE in 8 point type, centered and below the actual DATE. For example DDMMYYYY.
Rotation	double	optional	This value specified the rotation of the entire page (in degrees) around the origin point. For gtk versions prior to 2.10, the origin point is in the lower left corner of the page and rotation values increase in the counter-clockwise direction. For gtk version 2.10 and later, the origin point is in the upper left corner of the page and rotation values increase in the clockwise direction. Rotation of the page is applied before translation.
Translation	list of 2 doubles	optional	These values specify the x and y translation of the entire page (in points) relative to the origin point. For gtk versions prior to 2.10, the origin point is in the lower left corner of the page and translation values increase moving up and to the right. For gtk version 2.10 and later, the origin point is in the upper left corner of the page and translation values increase moving down and to the right. Rotation of the page is applied before translation.
Show_Grid	boolean	optional	If this value is set to <i>true</i> then GnuCash will draw a grid on the page, starting at the origin with the lines spaced every 50 points. This can be helpful when creating a check format file.
Show_Boxes	boolean	optional	If this value is set to <i>true</i> then for each item where the width and height have been specified, GnuCash will draw a box showing location and maximum size of that item. This can be helpful when creating a check format file.

**Note:** The Blocking\_Chars and DateFormat options are defined for all check formats in Edit->Preferences->Printing. It is recommended that these global options be set to false (the default), and that the options be set for individual Check Items as described below.

## Check Positions Group

This group of items specifies how multiple checks are laid out on the same sheet of paper, and gives names to each of these check locations so that a user can specify which check location that GnuCash should print. This entire group of key/value pairs is optional, and should be omitted if the format file only specifies a single check per page of paper.

Table 28: Multiple Checks Per Page Fields

Name	Type	Re- quired	Description
Height	double	mandatory	This field specifies the height of a single check on the page. If there are multiple checks per page then this item is mandatory. If there is only a single check per page, this entire section should be omitted.
Names	list of strings	mandatory	This field specifies the names of the check locations that can be printed on each page. These names represent the check positions starting from the top of the page and moving downward. The names are presented verbatim in the check position list of the check printing dialog. A typical value for this field is “Top;Middle;Bottom”, but it could also be “First;Second;Third” or any other set of strings that clearly identify the check locations. If there are multiple checks per page then this item is mandatory. If there is only a single check per page, this entire section should be omitted.

### Check Items Group

This section specifies the individual items that are printed on the check. There is no limit to the number of items that may be present in this section, and any given type of item can be repeated multiple times. This allows for the printing of checks that have a side stub, or for the one-per-page business checks that have both the check and multiple check stubs on the same page. For example, to print the payee name on a business check and on both stubs, simply specify three payee items with differing print coordinates.

Each key names in this section explicitly includes the item number to which it applies. E.G. The key named Type\_1 applies to the first item to be printed, and the key Coords\_3 applies to the third item to be printed. Item numbers start at one and increase sequentially. Any gap in the numbering sequence is interpreted by GnuCash as the end of the item list. Items are printed in the order of their item numbers, not in the order in which they appear in the file.

Each item specified must include a type declaration. The rest of the parameters for that item depend upon the particular type of that item. See *table\_title* for a list of valid item types and their required parameters.

Table 29: Individual Check Item Fields

Name	Type	Re- quired	Description
Type_	nstring	mandatory	This field specifies the type of a single item to be printed on a check. See <i>table _title</i> for a list of valid item types.
Co- ords_	list nof 2 or 4 dou- bles	mandatory	This field specifies the coordinates where the item should be placed on a check, and optionally also specifies the width and height of the item. The numbers in order are the X and Y offset of the lower left corner of the item, and optionally the width and height of the item. If the width is supplied then the height must also be supplied, so this field will always contain two or four numbers. For gtk versions prior to 2.10, the origin point is in the lower left corner of the page and translation values increase moving up and to the right. For gtk version 2.10 and later, the origin point is in the upper left corner of the page and translation values increase moving down and to the right.  <b>Note:</b> Regardless of whether the origin is at the top or the bottom of the page, the coordinates always specify the lower left point of the item.
Font_	nstring	optional	If supplied, this is the font used to print this specific text item. This field can contain any string that is acceptable by gtk as a font specifier. If this field is omitted, the default font is the font specified in the <i>Top</i> section of the check description file, or if that was omitted the font specified in the GnuCash preferences dialog. This field is only recognized when using gtk version 2.10 or later.
Align_	nstring	optional	If supplied, this is the alignment used to print this specific text item. This field must contain one of the strings “left”, “center” or “right”. If this field is omitted, the text will be left aligned. This field is only recognized when using gtk version 2.10 or later.
Text_	nstring	optional	This field is only used when the item type is <i>TEXT</i> . It specifies the utf-8 text that should be printed on the check.
File- name_	string n	optional	This field is only used when the item type is <i>PICTURE</i> . It specifies the filename of the image that should be printed on the check. The string may specify either an absolute path name or as a relative path name. If a relative path name is specified, GnuCash first looks in the application check format folder (typically <code>/usr/share/g_nucash/checks</code> ) for the image file, and if it is not found there then it looks in the user private <code>~/.local/share/g_nucash/checks</code> directory for the image. This field is only recognized when using gtk version 2.10 or later.
Blocki- ng_Chars_	boolean n	optional	If supplied, this will set the print <i>* Blocking_Chars*</i> option for this item.
Da- teFor- mat_	boolean n	optional	If supplied, this will set the print <i>DateFormat</i> option for this item.

These are the individual items that can be printed on a check. All items require the coordinates on the page where the item should be printed. The majority of these items result in text being printed on the page, and these items may have individual font and alignments specified. For example, the numerical amount of a check could be printed right justified while everything else is printed left justified. Other types may have unique parameters.

Table 30: Individual Check Item Types

Name	Re- quired Fields	Op- tional Fields	Description
PAYEE	Co- ords	Font Align Block- ing_Chars	This type value tells GnuCash to print the check payee name at the specified coordinates.
DATE	Co- ords	Font Align Block- ing_Chars Date- Format	This type value tells GnuCash to print the check date at the specified coordinates.
NOTES	Co- ords	Font Align Block- ing_Chars	This type value tells GnuCash to print the transaction notes field at the specified coordinates.
CHECK NUMBER	Co- ords	Font Align Block- ing_Chars	This type value tells GnuCash to print the check number at the specified coordinates. The check number reflects the book option selection under File > Properties for number source (transaction number or anchor-split action - see <a href="#">Use Split Action Field for Number</a> in the Book Options section of the GnuCash Help Manual).
MEMO	Co- ords	Font Align Block- ing_Chars	This type value tells GnuCash to print the split memo field at the specified coordinates.
AC- TION	Co- ords	Font Align Block- ing_Chars	This type value tells GnuCash to print the split action field at the specified coordinates. However, the printed field reflects the book option selection under File > Properties for number source (transaction number or anchor-split action - see <a href="#">Use Split Action Field for Number</a> in the Book Options section of the GnuCash Help Manual). If number source for the book is specified as anchor-split action, this field will instead print the transaction number field.
AMOUNT WORDS	Co- ords	Font Align Block- ing_Chars	This type value tells GnuCash to print the check amount in words at the specified coordinates. The amount will appear similar to the string “One thousand, two hundred thirty four and 56/100”.
AMOUNT NUMBER	Co- ords	Font Align Block- ing_Chars	This type value tells GnuCash to print the check amount in numbers at the specified coordinates. The amount will appear similar to the number “\$1,234.56”.
AD- DRESS	Co- ords	Font Align Block- ing_Chars	This type value tells GnuCash to print the address at the specified coordinates.
SPLITS ACCOUNT	Co- ords	Font Align Block- ing_Chars	This type value tells GnuCash to print the account names for each split entry stating at the specified coordinates. See the note on splits printing.
SPLITS AMOUNT	Co- ords	Font Align Block- ing_Chars	This type value tells GnuCash to print the amount for each split entry stating at the specified coordinates. Amounts are printed with currency symbols. See the note on splits printing.
SPLITS MEMO	Co- ords	Font Align Block- ing_Chars	This type value tells GnuCash to print the memo text for each split entry stating at the specified coordinates. See the note on splits printing.
<b>2.33. Check Format Files (*.chk)</b>			<b>235</b>
TEXT	Co- ords, Text	Font Align Block-	This type value tells GnuCash to print an arbitrary string at the specified coordinates. The string to be printed is specified with the <i>Text_n</i> key.

---

**Note:** SPLIT items include all split entries for the transaction except for the split that applies to the current account register (referred to as the anchor-split). This is usually the last split listed when splits are displayed in the register. The coordinate location defines the lower left location for the split information.

---

### 2.33.4 Creating Check Format Files

Creating your own check format file is a fairly simple task. The easiest way to start is to copy an existing check format file from the application directory (typically `/usr/share/gnucash/checks`) to the directory `~/.local/share/gnucash/checks`. Make sure to change the guid so the new file will be accepted by gnucash, and change the title to something descriptive. Then change or add individual item fields as necessary. You can also create a new check file by clicking the Save Format button on the Custom format page of the check printing dialog.

---

**Note:** Key names are case sensitive. If you are having problems with a check format file, ensure that all key names have capital letters as documented above.

---

## 2.34 GNU Free Documentation License

### 2.35 0. PREAMBLE

The purpose of this License is to make a manual, textbook, or other written document “free” in the sense of freedom: to assure everyone the effective freedom to copy and redistribute it, with or without modifying it, either commercially or noncommercially. Secondly, this License preserves for the author and publisher a way to get credit for their work, while not being considered responsible for modifications made by others.

This License is a kind of “copyleft”, which means that derivative works of the document must themselves be free in the same sense. It complements the GNU General Public License, which is a copyleft license designed for free software.

We have designed this License in order to use it for manuals for free software, because free software needs free documentation: a free program should come with manuals providing the same freedoms that the software does. But this License is not limited to software manuals; it can be used for any textual work, regardless of subject matter or whether it is published as a printed book. We recommend this License principally for works whose purpose is instruction or reference.

### 2.36 1. APPLICABILITY AND DEFINITIONS

This License applies to any manual or other work that contains a notice placed by the copyright holder saying it can be distributed under the terms of this License. The “Document”, below, refers to any such manual or work. Any member of the public is a licensee, and is addressed as “you”.

A “Modified Version” of the Document means any work containing the Document or a portion of it, either copied verbatim, or with modifications and/or translated into another language.

A “Secondary Section” is a named appendix or a front-matter section of the *Document* that deals exclusively with the relationship of the publishers or authors of the Document to the Document’s overall subject (or to related matters) and contains nothing that could fall directly within that overall subject. (For example, if the Document is in part a textbook of mathematics, a Secondary Section may not explain any mathematics.) The relationship could be a matter



of historical connection with the subject or with related matters, or of legal, commercial, philosophical, ethical or political position regarding them.

The “Invariant Sections” are certain *Secondary Sections* whose titles are designated, as being those of Invariant Sections, in the notice that says that the *Document* is released under this License.

The “Cover Texts” are certain short passages of text that are listed, as Front-Cover Texts or Back-Cover Texts, in the notice that says that the *Document* is released under this License.

A “Transparent” copy of the *Document* means a machine-readable copy, represented in a format whose specification is available to the general public, whose contents can be viewed and edited directly and straightforwardly with generic text editors or (for images composed of pixels) generic paint programs or (for drawings) some widely available drawing editor, and that is suitable for input to text formatters or for automatic translation to a variety of formats suitable for input to text formatters. A copy made in an otherwise Transparent file format whose markup has been designed to thwart or discourage subsequent modification by readers is not Transparent. A copy that is not “Transparent” is called “Opaque”.

Examples of suitable formats for Transparent copies include plain ASCII without markup, Texinfo input format, LaTeX input format, SGML or XML using a publicly available DTD, and standard-conforming simple HTML designed for human modification. Opaque formats include PostScript, PDF, proprietary formats that can be read and edited only by proprietary word processors, SGML or XML for which the DTD and/or processing tools are not generally available, and the machine-generated HTML produced by some word processors for output purposes only.

The “Title Page” means, for a printed book, the title page itself, plus such following pages as are needed to hold, legibly, the material this License requires to appear in the title page. For works in formats which do not have any title page as such, “Title Page” means the text near the most prominent appearance of the work’s title, preceding the beginning of the body of the text.

## 2.37 2. VERBATIM COPYING

You may copy and distribute the *Document* in any medium, either commercially or noncommercially, provided that this License, the copyright notices, and the license notice saying this License applies to the *Document* are reproduced in all copies, and that you add no other conditions whatsoever to those of this License. You may not use technical measures to obstruct or control the reading or further copying of the copies you make or distribute. However, you may accept compensation in exchange for copies. If you distribute a large enough number of copies you must also follow the conditions in [section 3](#).

You may also lend copies, under the same conditions stated above, and you may publicly display copies.

## 2.38 3. COPYING IN QUANTITY

If you publish printed copies of the *Document* numbering more than 100, and the *Document*’s license notice requires *Cover Texts*, you must enclose the copies in covers that carry, clearly and legibly, all these Cover Texts: Front-Cover Texts on the front cover, and Back-Cover Texts on the back cover. Both covers must also clearly and legibly identify you as the publisher of these copies. The front cover must present the full title with all words of the title equally prominent and visible. You may add other material on the covers in addition. Copying with changes limited to the covers, as long as they preserve the title of the *Document* and satisfy these conditions, can be treated as verbatim copying in other respects.

If the required texts for either cover are too voluminous to fit legibly, you should put the first ones listed (as many as fit reasonably) on the actual cover, and continue the rest onto adjacent pages.

If you publish or distribute *Opaque* copies of the *Document* numbering more than 100, you must either include a machine-readable *Transparent* copy along with each *Opaque* copy, or state in or with each *Opaque* copy a publicly-accessible computer-network location containing a complete *Transparent* copy of the *Document*, free of added mate-

rial, which the general network-using public has access to download anonymously at no charge using public-standard network protocols. If you use the latter option, you must take reasonably prudent steps, when you begin distribution of Opaque copies in quantity, to ensure that this Transparent copy will remain thus accessible at the stated location until at least one year after the last time you distribute an Opaque copy (directly or through your agents or retailers) of that edition to the public.

It is requested, but not required, that you contact the authors of the *Document* well before redistributing any large number of copies, to give them a chance to provide you with an updated version of the Document.

## **2.39 4. MODIFICATIONS**

You may copy and distribute a *Modified Version* of the *Document* under the conditions of sections 2 and 3 above, provided that you release the Modified Version under precisely this License, with the Modified Version filling the role of the Document, thus licensing distribution and modification of the Modified Version to whoever possesses a copy of it. In addition, you must do these things in the Modified Version:

- **A.**  
Use in the *Title Page* (and on the covers, if any) a title distinct from that of the *Document*, and from those of previous versions (which should, if there were any, be listed in the History section of the Document). You may use the same title as a previous version if the original publisher of that version gives permission.
- **B.**  
List on the *Title Page*, as authors, one or more persons or entities responsible for authorship of the modifications in the *Modified Version*, together with at least five of the principal authors of the *Document* (all of its principal authors, if it has less than five).
- **C.**  
State on the *Title Page* the name of the publisher of the *Modified Version*, as the publisher.
- **D.**  
Preserve all the copyright notices of the *Document*.
- **E.**  
Add an appropriate copyright notice for your modifications adjacent to the other copyright notices.
- **F.**  
Include, immediately after the copyright notices, a license notice giving the public permission to use the *Modified Version* under the terms of this License, in the form shown in the Addendum below.
- **G.**  
Preserve in that license notice the full lists of *Invariant Sections* and required *Cover Texts* given in the *Document's* license notice.
- **H.**  
Include an unaltered copy of this License.
- **I.**  
Preserve the section entitled “History”, and its title, and add to it an item stating at least the title, year, new authors, and publisher of the *Modified Version* as given on the *Title Page*. If there is no section entitled “History” in the *Document*, create one stating the title, year, authors, and publisher of the Document as given on its Title Page, then add an item describing the Modified Version as stated in the previous sentence.

- **J.**

Preserve the network location, if any, given in the *Document* for public access to a *Transparent* copy of the Document, and likewise the network locations given in the Document for previous versions it was based on. These may be placed in the “History” section. You may omit a network location for a work that was published at least four years before the Document itself, or if the original publisher of the version it refers to gives permission.

- **K.**

In any section entitled “Acknowledgements” or “Dedications”, preserve the section’s title, and preserve in the section all the substance and tone of each of the contributor acknowledgements and/or dedications given therein.

- **L.**

Preserve all the *Invariant Sections* of the *Document*, unaltered in their text and in their titles. Section numbers or the equivalent are not considered part of the section titles.

- **M.**

Delete any section entitled “Endorsements”. Such a section may not be included in the *Modified Version*.

- **N.**

Do not retitle any existing section as “Endorsements” or to conflict in title with any *Invariant Section*.

If the *Modified Version* includes new front-matter sections or appendices that qualify as *Secondary Sections* and contain no material copied from the Document, you may at your option designate some or all of these sections as invariant. To do this, add their titles to the list of *Invariant Sections* in the Modified Version’s license notice. These titles must be distinct from any other section titles.

You may add a section entitled “Endorsements”, provided it contains nothing but endorsements of your *Modified Version* by various parties—for example, statements of peer review or that the text has been approved by an organization as the authoritative definition of a standard.

You may add a passage of up to five words as a *Front-Cover Text*, and a passage of up to 25 words as a *Back-Cover Text*, to the end of the list of *Cover Texts* in the *Modified Version*. Only one passage of Front-Cover Text and one of Back-Cover Text may be added by (or through arrangements made by) any one entity. If the *Document* already includes a cover text for the same cover, previously added by you or by arrangement made by the same entity you are acting on behalf of, you may not add another; but you may replace the old one, on explicit permission from the previous publisher that added the old one.

The author(s) and publisher(s) of the *Document* do not by this License give permission to use their names for publicity for or to assert or imply endorsement of any *Modified Version*.

## 2.40 5. COMBINING DOCUMENTS

You may combine the *Document* with other documents released under this License, under the terms defined in [section 4](#) above for modified versions, provided that you include in the combination all of the *Invariant Sections* of all of the original documents, unmodified, and list them all as Invariant Sections of your combined work in its license notice.

The combined work need only contain one copy of this License, and multiple identical *Invariant Sections* may be replaced with a single copy. If there are multiple Invariant Sections with the same name but different contents, make the title of each such section unique by adding at the end of it, in parentheses, the name of the original author or publisher of that section if known, or else a unique number. Make the same adjustment to the section titles in the list of Invariant Sections in the license notice of the combined work.

In the combination, you must combine any sections entitled “History” in the various original documents, forming one section entitled “History”; likewise combine any sections entitled “Acknowledgements”, and any sections entitled “Dedications”. You must delete all sections entitled “Endorsements.”

## 2.41 6. COLLECTIONS OF DOCUMENTS

You may make a collection consisting of the *Document* and other documents released under this License, and replace the individual copies of this License in the various documents with a single copy that is included in the collection, provided that you follow the rules of this License for verbatim copying of each of the documents in all other respects.

You may extract a single document from such a collection, and distribute it individually under this License, provided you insert a copy of this License into the extracted document, and follow this License in all other respects regarding verbatim copying of that document.

## 2.42 7. AGGREGATION WITH INDEPENDENT WORKS

A compilation of the *Document* or its derivatives with other separate and independent documents or works, in or on a volume of a storage or distribution medium, does not as a whole count as a *Modified Version* of the Document, provided no compilation copyright is claimed for the compilation. Such a compilation is called an “aggregate”, and this License does not apply to the other self-contained works thus compiled with the Document, on account of their being thus compiled, if they are not themselves derivative works of the Document. If the *Cover Text* requirement of [section 3](#) is applicable to these copies of the Document, then if the Document is less than one quarter of the entire aggregate, the Document’s Cover Texts may be placed on covers that surround only the Document within the aggregate. Otherwise they must appear on covers around the whole aggregate.

## 2.43 8. TRANSLATION

Translation is considered a kind of modification, so you may distribute translations of the *Document* under the terms of [section 4](#). Replacing *Invariant Sections* with translations requires special permission from their copyright holders, but you may include translations of some or all Invariant Sections in addition to the original versions of these Invariant Sections. You may include a translation of this License provided that you also include the original English version of this License. In case of a disagreement between the translation and the original English version of this License, the original English version will prevail.

## 2.44 9. TERMINATION

You may not copy, modify, sublicense, or distribute the *Document* except as expressly provided for under this License. Any other attempt to copy, modify, sublicense or distribute the Document is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

## 2.45 10. FUTURE REVISIONS OF THIS LICENSE

The [Free Software Foundation](#) may publish new, revised versions of the GNU Free Documentation License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns. See <http://www.gnu.org/copyleft/>.

Each version of the License is given a distinguishing version number. If the *Document* specifies that a particular numbered version of this License “or any later version” applies to it, you have the option of following the terms and conditions either of that specified version or of any later version that has been published (not as a draft) by the Free Software Foundation. If the Document does not specify a version number of this License, you may choose any version ever published (not as a draft) by the Free Software Foundation.

## 2.46 Addendum

To use this License in a document you have written, include a copy of the License in the document and put the following copyright and license notices just after the title page:

Copyright YEAR YOUR NAME.

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.1 or any later version published by the Free Software Foundation; with the *Invariant Sections* being LIST THEIR TITLES, with the *Front-Cover Texts* being LIST, and with the *Back-Cover Texts* being LIST. A copy of the license is included in the section entitled “GNU Free Documentation License”.

If you have no *Invariant Sections*, write “with no Invariant Sections” instead of saying which ones are invariant. If you have no *Front-Cover Texts*, write “no Front-Cover Texts” instead of “Front-Cover Texts being LIST”; likewise for *Back-Cover Texts*.

If your document contains nontrivial examples of program code, we recommend releasing these examples in parallel under your choice of free software license, such as the [GNU General Public License](#), to permit their use in free software.

*Getting Started* provides you with the most basic information needed to begin using GnuCash. The chapters in this section explain concepts that are fundamental to using GnuCash. New users of GnuCash should familiarize themselves with the information in these chapters to get you up and running:

- *Overview* - (this chapter) gives a general overview of GnuCash
- *The Basics* - gives users a very brief introduction to accounting principles, and then provides information about how GnuCash structures its data. There is also information on basic interface elements in GnuCash. Finally, this chapter explains how GnuCash stores and manages your data.
- *Accounts* - gives further information about accounts and how to organize them.
- *Transactions* - gives basic information about transactions and how to enter them.

*Managing Personal Finances* addresses common applications and features of GnuCash in greater detail. You will see more specific cases, based on frequently asked questions about applying GnuCash to everyday situations. Here are the applications and features covered in this part:

- *Checkbook*
- *Credit Cards*
- *Loans*
- *Investments*
- *Capital Gains*
- *Multiple Currencies*

*Managing Business Finances* discusses the use of GnuCash in business accounting:

- *Business Features*
- *Budgets*
- *Other Assets*
- *Depreciation*
- *Python Bindings*
- *Importing Business Data*

This manual also includes several appendices, which contains extra information you might want to know:

- *GnuCash Glossary* - Glossary of terms used in GnuCash
- *Migration Guide* - Guide for former Quicken, MS Money or other QIF users
- *Frequently Asked Questions*
- *Contributed Account Trees*
- *Auxiliary File Formats*
- *GNU Free Documentation License*

Last, but not least, a glossary and index help you quickly locate topics.

---

### Getting More Help

---

GnuCash offers help in many ways, including the *Tip of the Day*, context help, a website, a wiki, the mailing lists, and IRC chat.

#### 3.1 Context Help

The context help provides detailed instructions for using GnuCash's menus, windows, and controls. To open online help, select Help > Contents.

The GnuCash Help window also acts as a simple web browser, so you can pull up a website for additional information. You can open any website under this window by clicking the Open *Toolbar* button and then typing in the URL. Use the Back, Forward, Reload, Stop, and Print buttons as you would in a standard browser.

#### 3.2 GnuCash Website

The [GnuCash](#) website contains helpful information about the program and about any updates to it. It also contains links to other online resources.

#### 3.3 GnuCash Wiki

An immense amount of less-formal documentation, both of GnuCash itself and its maintenance and development may be found in the [GnuCash Wiki](#); the [Frequently Asked Questions](#) page should be a first stop whenever you encounter difficulty using GnuCash.

#### 3.4 GnuCash On-line Assistance

##### Mailing List

The primary source of user support is the [user mailing list](#). If you prefer a web forum-like presentation, you can use it via [Nabble](#). One must [subscribe](#) before posting, even if using Nabble.

### IRC

Several of the developers monitor the #gnucash channel at [irc.gnome.org](#). They're usually doing something else, too, and of course aren't always at their computers. Log in, ask your question, and stay logged in; it may be several hours before your question is noticed and responded to. To see if you missed anything [check the IRC logs](#).

The [GnuCash website](#) has more details on these channels. You will also find pointers there to additional useful resources such as the GnuCash wiki and bug tracking system.

## 3.5 Topic Search

The online manual also provides a search function. To search for a particular topic, click the Search tab at the bottom of the help window and type in your topic in the field provided. Click the Search button to complete your search. A list of choices should appear in the box below, clicking a choice will bring up its text on the right.



## CHAPTER 4

---

### Installation

---

Installation of GnuCash is usually simple.

The [GnuCash download page](#) contains detailed instructions on how to install GnuCash for each operating system supported.