

GnuCash Tutorial and Concepts Guide

Carol Champagne

Chris Lyttle

GnuCash Documentation Team

<chris@wilddev.net>

Jon Lapham

GnuCash Documentation Team

<lapham@extracta.com.br>

This Guide contains a tutorial for using GnuCash and describes the concepts behind GnuCash.

Copyright © 2003-2004 Jon Lapham

Copyright © 2002 Chris Lyttle

Copyright © 2001 Carol Champagne and Chris Lyttle

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License (GFDL), Version 1.1 or any later version published by the Free Software Foundation with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. You can find a copy of the GFDL at this [link](#) or in the file COPYING-DOCS distributed with this manual.

This manual is part of a collection of GNOME manuals distributed under the GFDL. If you want to distribute this manual separately from the collection, you can do so by adding a copy of the license to the manual, as described in section 6 of the license.

Many of the names used by companies to distinguish their products and services are claimed as trademarks. Where those names appear in any GNOME documentation, and the members of the GNOME Documentation Project are made aware of those trademarks, then the names are in capital letters or initial capital letters.

DOCUMENT AND MODIFIED VERSIONS OF THE DOCUMENT ARE PROVIDED UNDER THE TERMS OF THE GNU FREE DOCUMENTATION LICENSE WITH THE FURTHER UNDERSTANDING THAT:

1. DOCUMENT IS PROVIDED ON AN "AS IS" BASIS, WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, WITHOUT LIMITATION, WARRANTIES THAT THE DOCUMENT OR MODIFIED VERSION OF THE DOCUMENT IS FREE OF DEFECTS MERCHANTABLE, FIT FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT. THE ENTIRE RISK AS TO THE QUALITY, ACCURACY, AND PERFORMANCE OF THE DOCUMENT OR MODIFIED VERSION OF THE DOCUMENT IS WITH YOU. SHOULD ANY DOCUMENT OR MODIFIED VERSION PROVE DEFECTIVE IN ANY RESPECT, YOU (NOT THE INITIAL WRITER, AUTHOR OR ANY CONTRIBUTOR) ASSUME THE COST OF ANY NECESSARY SERVICING, REPAIR OR CORRECTION. THIS DISCLAIMER OF WARRANTY CONSTITUTES AN ESSENTIAL PART OF THIS LICENSE. NO USE OF ANY DOCUMENT OR MODIFIED VERSION OF THE DOCUMENT IS AUTHORIZED HEREUNDER EXCEPT UNDER THIS DISCLAIMER; AND
2. UNDER NO CIRCUMSTANCES AND UNDER NO LEGAL THEORY, WHETHER IN TORT (INCLUDING NEGLIGENCE), CONTRACT, OR OTHERWISE, SHALL THE AUTHOR, INITIAL WRITER, ANY CONTRIBUTOR, OR ANY DISTRIBUTOR OF THE DOCUMENT

OR MODIFIED VERSION OF THE DOCUMENT, OR ANY SUPPLIER OF ANY OF SUCH PARTIES, BE LIABLE TO ANY PERSON FOR ANY DIRECT, INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES OF ANY CHARACTER INCLUDING, WITHOUT LIMITATION, DAMAGES FOR LOSS OF GOODWILL, WORK STOPPAGE, COMPUTER FAILURE OR MALFUNCTION, OR ANY AND ALL OTHER DAMAGES OR LOSSES ARISING OUT OF OR RELATING TO USE OF THE DOCUMENT AND MODIFIED VERSIONS OF THE DOCUMENT, EVEN IF SUCH PARTY SHALL HAVE BEEN INFORMED OF THE POSSIBILITY OF SUCH DAMAGES.

Feedback

To report a bug or make a suggestion regarding this package or this manual, follow the directions at the [GNOME Bug Tracking System](#).

Revision History	
Revision GnuCash Tutorial and Concepts Guide V1.8.4	11th April 2004
Jon Lapham < lapham@extracta.com.br >	
GnuCash Documentation Team	
Revision GnuCash Tutorial and Concepts Guide V1.8.3	Aug 2003
Jon Lapham < lapham@extracta.com.br >	
GnuCash Documentation Team	
Revision GnuCash Tutorial and Concepts Guide V1.8.2	Aug 2003
Jon Lapham < lapham@extracta.com.br >	
GnuCash Documentation Team	
Revision GnuCash Tutorial and Concepts Guide V1.8.1	May 2003
Chris Lyttle < chris@wilddev.net >	
GnuCash Documentation Team	
Revision GnuCash Tutorial and Concepts Guide V1.8.0	Jan 2003
Chris Lyttle < chris@wilddev.net >	
GnuCash Documentation Team	
Revision GnuCash User Guide V1.6.5	June 2002
Chris Lyttle < chris@wilddev.net >	
GnuCash Documentation Team	
Revision GnuCash User Guide V1.6.0	October 2001
Carol Champagne < carol@io.com >	
GnuCash Documentation Team	

Table of Contents

[1. Overview](#)

[Introduction](#)

[Features](#)

[Easy to Use](#)

[Tracks Your Investments](#)

[International Support](#)

[Business Support](#)

[Accounting Features](#)

[What's New in v1.8](#)

[About this Book](#)

[Top Ten Reasons to Use GnuCash](#)

[Installation](#)

[2. The Basics](#)

[Accounting Concepts](#)

[The 5 Basic Accounts](#)

[The Accounting Equation](#)

[Double Entry](#)

[Data Entry Concepts](#)

[Files, Accounts and Transactions](#)

[Double Entry](#)

[The GnuCash Interface](#)

[Create a new set of accounts](#)

[Tip of the Day](#)

[Account Tree Window](#)

[Account Register Window](#)

[Tool Bar Buttons](#)

[Menu Items](#)

[Menu Shortcuts](#)

[Setting Preferences](#)

[Accounts](#)

[Business](#)

[General](#)

[International](#)

[On-line Banking](#)

[Reconcile](#)

[Register](#)

[Register Colors](#)

[Scheduled Transactions](#)

[Getting Help](#)

[Help Manual](#)

[Web Access](#)

[Topic Search](#)

[Creating and Saving Files](#)

[Importing QIF Files](#)

[Backing Up and Recovering Data](#)

[Backup file \(.xac\)](#)

[Log file \(.log\)](#)

[Lock files \(.LNK and .LCK\)](#)

[File Management](#)

[Putting It All Together](#)

[3. Accounts](#)

[Basic Accounting Concepts](#)

[Balance Sheet Accounts](#)

[Income and Expense Accounts](#)

[GnuCash Accounts](#)

[Balance Sheet Accounts](#)
[Income and Expense Accounts](#)

[Putting It All Together](#)

[Simple Example](#)

[The Basic Top Level Accounts](#)

[Making Sub-Accounts](#)

[4. Transactions](#)

[Basic Concepts](#)

[The Account Register](#)

[Simple Transaction](#)

[Split Transaction](#)

[Features of the Account Register](#)

[Choosing a Register Style](#)

[Using Entry Shortcuts](#)

[Reconciliation](#)

[Reconcile Window](#)

[Scheduled Transactions](#)

[Creating from the Ledger](#)

[Creating from the Editor](#)

[Putting It All Together](#)

[Opening Balances](#)

[Additional Transaction Examples](#)

[5. Checkbook](#)

[Setting up Accounts](#)

[Entering Deposits](#)

[Entering Withdrawals](#)

[ATM/Cash Withdrawals](#)

[Reconciling Your Accounts](#)

[Putting It All Together](#)

[6. Credit Cards](#)

[Concepts](#)

[Setting Up Accounts](#)

[Simple Setup](#)

[Complete Setup](#)

[Entering Charges](#)

[Entering Payments](#)

[Putting It All Together](#)

[Purchases](#)

[Refund](#)

[Interest Charge](#)

[Reconciliation](#)

[Payment](#)

[7. Loans](#)

[Basic Concepts](#)

[Terminology](#)

[Setting Up Accounts](#)

[House Mortgage \(How-To\)](#)

[Calculations](#)

[Example: Monthly Payments](#)

[Example: Length of Loan](#)

[Advanced: Calculation Details](#)

[Missing Loan Documentation](#)

[8. Investments](#)

[Basic Concepts](#)

[Terminology](#)

[Types of Investments](#)

[Setting Up Accounts](#)

[Predefined Accounts](#)

[Custom Accounts Example](#)

[Interest Bearing Accounts](#)

[Account Setup](#)

[Example](#)

[Setup Stock Portfolio](#)

[Setup Preexisting Stocks](#)

[Example Stock Account](#)

[Buying New Investments](#)

[Setting Stock Price](#)

[Initial Price Editor Setup](#)

[Setting Stock Price Manually](#)

[Setting Stock Price Automatically](#)

[Displaying Stock Value](#)

[Making Stock Value Reports](#)

[Dividends](#)

[Selling Stocks](#)

[Example](#)

[To-do](#)

[9. Capital Gains](#)

[Basic Concepts](#)

[Estimating Valuation](#)

[Account Setup](#)

[Example](#)

[Unrealized Gains](#)

[Selling](#)

[Caution about Valuation](#)

[Taxation](#)

[10. Multiple Currencies](#)

[Basic Concepts](#)

[Account Setup](#)

[GnuCash Currency Support](#)

[Setting International Preferences](#)

[Recording Currency Exchange \(How-To\)](#)

[Recording Purchases in a Foreign Currency \(How-To\)](#)

[Tracking Currency Investments \(How-To\)](#)

[Reconciling Statements in a Foreign Currency \(How-To\)](#)

[Updating Exchange Rates \(How-To\)](#)

[Putting It All Together \(Examples\)](#)

[11. Depreciation](#)

[Basic Concepts](#)

[Estimating Valuation](#)

[Depreciation Schemes](#)

[Account Setup](#)

[Example](#)

[12. Accounts Receivable](#)

[Overview of Business Features](#)

[Basic Concepts](#)

[Initial Setup](#)

[Account Setup](#)

[Register Your Company](#)

[System Components](#)

[Customers](#)

[New](#)

[Find and Edit](#)

[Invoices](#)

[New](#)

[Edit](#)

[Post](#)

[Find](#)

[Print](#)

[Assign Starting Invoice Number](#)

[Customer Jobs](#)

[Process Payment](#)

[Changing the Invoice Appearance](#)

[13. Accounts Payable](#)

[Basic Concepts](#)

[Initial Setup](#)

[Account Setup](#)

[Register Your Company in GnuCash](#)

[System Components](#)

[Vendors](#)

[New](#)

[Find and Edit](#)

[Bills](#)

[New](#)

[Edit](#)

[Post](#)

[Find](#)

[Vendor Jobs](#)

[Process Payment](#)

[14. Payroll](#)

[Basic Concepts](#)

[Account Setup](#)

[Protocol](#)

[Step 1: Deductions list](#)

[Step 2: Create the Transaction Map](#)

[Step 3: Pay the Employee](#)

[Step 4: Pay the Government](#)

[Example](#)

[Build Protocol](#)

[Pay an Employee](#)

[Pay the Government](#)

[A. Migration Guide](#)

[Using Accounts vs. Categories](#)

[Organization of QIF Files \(Discussion\)](#)

[Common Duplication Issues \(Discussion\)](#)

[Checking QIF Data \(Discussion\)](#)

[Converting XML GnuCash File](#)

[B. Frequently Asked Questions](#)

[Sources of Information](#)

[Q: Where's the FAQ?](#)

[Q: Are there mailing lists for GnuCash?](#)

[Q: Is there a searchable archive for the mailing lists?](#)

[Q: Are there other means of obtaining support for GnuCash?](#)

[General Information](#)

[Q: Can I run GnuCash on Windows?](#)

[Q: I heard it is too hard to compile GnuCash!](#)

[Q: Is there a batch mode \(non-interactive\) available for GnuCash, for building reports, etc?](#)

[Q: Can multiple people access the same datafile in GnuCash?](#)

[Q: Why is GnuCash written in C?](#)

Q: Why don't you rewrite GnuCash in programming language xyz so that I can contribute easily?

Q: I really want feature XYZ but GnuCash doesn't have it. How do I get it added?

Q: Is there a web interface available for GnuCash?

Q: How can I provide security for GC data using CFS, etc.)

Q: How can I contribute to the GnuCash project?

Q: I think I found a bug. How do I report it?

Using GnuCash

Q: How can I move the transactions from account A into account B, thus combining them?

Q: Is it possible to merge two gnuCash files?

Q: How can I save a template of my account structure?

Q: When I search for customers (or anything else for that matter), how can I return a list of everything?

Q: How can I record a transaction on different dates (actual date and bank date)?

Accounting

Q: How do I treat taxes? As an account payable or as an expense?

C. Contributed Account Trees

UK Vat

D. GNU Free Documentation License

0. PREAMBLE

1. APPLICABILITY AND DEFINITIONS

2. VERBATIM COPYING

3. COPYING IN QUANTITY

4. MODIFICATIONS

5. COMBINING DOCUMENTS

6. COLLECTIONS OF DOCUMENTS

7. AGGREGATION WITH INDEPENDENT WORKS

8. TRANSLATION

9. TERMINATION

10. FUTURE REVISIONS OF THIS LICENSE

Addendum

List of Tables

- 7.1. [Buying a House Split Transaction](#)
- 8.1. [Selling Stock Split Transaction Scheme](#)
- 9.1. [Turning an Accrued Gain into a Realized Gain](#)
- 11.1. [Linear Depreciation Scheme Example](#)
- 11.2. [Geometric Depreciation Scheme Example](#)
- 11.3. [Sum of Digits Depreciation Scheme Example](#)
- 14.1. [Transaction Map](#)
- 14.2. [Transaction Map for Employee 1](#)

Chapter 1. Overview

Table of Contents

[Introduction](#)

[Features](#)

[Easy to Use](#)

[Tracks Your Investments](#)

[International Support](#)

[Business Support](#)

[Accounting Features](#)

[What's New in v1.8](#)

[About this Book](#)

Introduction

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.

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.

Features

Easy to Use

GnuCash has been designed from the ground up to be 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 with 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 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 one of the following export methods is supported by most banks. 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 is 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 is 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 can automatically create and enter, or remind you of scheduled or recurring transactions. When you are reminded of scheduled transactions, you can enter or postpone them. Recurring transactions can be configured to stop after a certain period.
- *Mortgage and Loan Repayment Druid*: 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) for GnuCash v1.8.x is: GNU/Linux (x86, Sparc, PPC), FreeBSD (x86), OpenBSD (x86), Solaris (Sparc), and MacOS X (PPC). Previous versions of GnuCash have been known to work with, SGI IRIX (MIPS), IBM AIX 4.1.5 (RS/6000), Unixware 7 (Intel), and SCO OpenServer 5.0.4 (Intel), but their current status is unknown.

Tracks Your Investments

GnuCash has a number of investment features. From simple certificates of deposit, to publicly traded stocks, GnuCash will track all your investments.

- *Stock/Mutual Fund Portfolios*: Track stocks individually (one per account) or in a portfolio of accounts (a group of accounts that can be displayed together).
- *On-line Stock and Mutual Fund Quotes*: With GnuCash, you no longer need look up your stock values one at a time. The process can be automated, to always present you with the latest value of your stocks.

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 fully translated into 12 languages: Chinese, Czech, Dutch, English, French, German, Italian, Portuguese, Russian, Slovak, Spanish and Swedish. More than 13 other languages are partially supported.
- *Multiple Currencies and Currency Trading*: Multiple currencies are supported and can be bought and sold (traded). Currency movements between accounts are fully balanced when double-entry is enabled.
- *New Multi-Currency Transaction Handling*: GnuCash no longer requires separate currency exchange accounts to handle multiple currency transfers.
- *International Date Handling*: GnuCash understands that different countries display the date differently. You are able to work with the date format you are accustomed to.

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.
- *Payroll*: GnuCash can track company payroll.
- *Depreciation*: GnuCash can track depreciation of capital assets.
- *Reports*: GnuCash offers a wide variety of ready-to-use business reports.

Accounting Features

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

- *Double Entry*: When enabled, 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 Ledger*: 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.

What's New in v1.8

- *Scheduled Transactions*

GnuCash now has the ability to automatically create and enter transactions, remind you when these transactions are due, giving you the choice of entering, postponing or removing the automated transaction.
- *Mortgage and Loan Repayment Druid*

Used to setup a variable payment loan scheduled transaction.
- *Small Business Accounting Features*

GnuCash can now track Customers and Vendors, Invoicing and Bill Payment, and can use different Tax and Billing Terms in a small business.
- *Open Financial Exchange (OFX)*: GnuCash is the first free software application to support the Open Financial Exchange protocol. Many financial institutions are beginning to support this format.
- *Home Banking Computer Interface (HBCI)*: GnuCash is the first free software application to support the German Home Banking Computer Interface protocol. The protocol includes statement download, initiates bank transfers, and makes direct debits possible.

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 an 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 Chapter 2, each successive chapter builds on the previous chapter's tutorial.

This manual is organized into 3 main parts:

- *Getting Started*
- *Managing Your Personal Finances*
- *Managing Your Business Finances*

Getting Started provides you with the most basic information needed to begin using GnuCash. Use this part as a quick-start guide. Its chapters will get you up and running:

- Chapter 1: Overview
- Chapter 2: The Basics
- Chapter 3: Accounts
- Chapter 4: Transactions

Managing Your 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:

- Chapter 5: Checkbook
- Chapter 6: Credit Cards
- Chapter 7: Loans
- Chapter 8: Investments
- Chapter 9: Capital Gains
- Chapter 10: Multiple Currencies

Managing Your Business Finances discusses the use of GnuCash in business accounting:

- Chapter 11: Depreciation
- Chapter 12: Accounts Receivable
- Chapter 13: Accounts Payable
- Chapter 14: Payroll

This manual also has an appendix, which contains extra information you might want to know:

- Appendix A: Migration Guide - Guide for former Quicken®, MS Money or other QIF users
- Appendix B: Frequently Asked Questions
- Appendix C: Contributed Account Trees
- Appendix D: GNU Free Documentation License

Last, but not least, a glossary and index help you quickly locate topics.

Top Ten Reasons to Use GnuCash

We've already discussed some of the general advantages of using GnuCash. Here are some specific features offered by GnuCash that may not exist in other programs.

1. Simple user interface

GnuCash is as easy to use as a checkbook register. It's simpler than a paper register, because auto-completion and other entry shortcuts, not only do work for you, but reduce data entry errors.

2. Easy import

GnuCash allows you to import data from on-line bank statements and software packages using QIF (Quicken® Interchange Format), OFX and HBCI files. An easy-to-use “druid” walks you through reviewing resulting changes and actually importing them into GnuCash.

3. Statement reconciliation

Reconcile monthly statements quickly by entering the statement ending balance and checking off transactions. GnuCash helps you catch any discrepancies between your data and statements you receive.

4. Investment tracking

GnuCash offers a host of ways to track your investment portfolio. Special investment accounts simplify data entry, and on-line tools allow you to update prices of your holdings as the markets change. Reports complete the picture, allowing you to analyze your investment decisions.

5. Multiple currency support

GnuCash allows you to track multiple currencies. If you have bank accounts, investments or other financial data in different currencies, use GnuCash to monitor them.

6. Customized reports and graphs

Reports and graphs give you valuable information for filing taxes, budgeting, or simply figuring out where your money goes. GnuCash offers a variety of easy-to-use reports and graphs to help analyze your financial position. It gives you the freedom to customize your own reports to suit your unique needs.

7. Double entry

To provide complete records, GnuCash uses the double entry method of bookkeeping. Double entry simply means that money doesn't just appear or disappear - an equal amount must come from one location and go to another location. By tracking the transaction in both locations, GnuCash will give you detailed reports from the perspective of either account.

8. Sources of help

A Tip of the Day dialog gives helpful tips to new users about GnuCash features. Within the program, a searchable Help menu guides you to information or connects to the GnuCash web page for further assistance. GnuCash also has strong, helpful developer and user communities who provide help through mailing lists.

9. Shortcuts

GnuCash offers many shortcuts to help you enter data. Type the first few characters of a common entry and GnuCash will automatically fill in the rest! You can also use copy, paste and duplicate functions to save typing time. Keyboard shortcuts let you quickly choose a menu option or to enter numerical data. Many numeric entry fields can act as a calculator: enter "92.18+33.26" and watch GnuCash input the corresponding sum for you!

10. Open source

GnuCash doesn't hide its methods. If you wonder how GnuCash computed a number, you can easily find it out. In addition, you can set preferences that tell GnuCash how much information to display to you. There is no "secret code" used in GnuCash - it continues to be an open-source program.

These are only a few of the advantages you'll discover when you start using GnuCash to track your financial information. Now get ready to enjoy the benefits of GnuCash for yourself!

Installation

Installation of GnuCash can be simple, as most modern linux distributions come with GnuCash precompiled and ready to go. In fact, most likely, GnuCash is already installed. If you do not have GnuCash installed, the simplest method for installing it is to insert the installation CD that came with your distribution and follow the instructions.

The [GnuCash home page](#) contains detailed instructions on how to install GnuCash in the event your current distribution does not include it or you want to install a different version. You can also read the README file in the source code.

Chapter 2. The Basics

Table of Contents

[Accounting Concepts](#)

[The 5 Basic Accounts](#)

[The Accounting Equation](#)

[Double Entry](#)

[Data Entry Concepts](#)

[Files, Accounts and Transactions](#)

[Double Entry](#)

[The GnuCash Interface](#)

[Create a new set of accounts](#)

[Tip of the Day](#)

[Account Tree Window](#)

[Account Register Window](#)

[Tool Bar Buttons](#)

[Menu Items](#)

[Menu Shortcuts](#)

[Setting Preferences](#)

[Accounts](#)

[Business](#)

[General](#)

[International](#)

[On-line Banking](#)

[Reconcile](#)

[Register](#)

[Register Colors](#)

[Scheduled Transactions](#)

[Getting Help](#)

[Help Manual](#)

[Web Access](#)

[Topic Search](#)

[Creating and Saving Files](#)

[Importing QIF Files](#)

[Backing Up and Recovering Data](#)

[Backup file \(.xac\)](#)

[Log file \(.log\)](#)

[Lock files \(.LNK and .LCK\)](#)

[File Management](#)

[Putting It All Together](#)

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. The chapters which follow this will begin to show you hands on examples.

Accounting Concepts

GnuCash is easy enough to use that you do not have to have a complete understanding of accounting principals to find it useful. However, you will find that some basic accounting knowledge will prove to be invaluable as GnuCash was designed using these principals 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

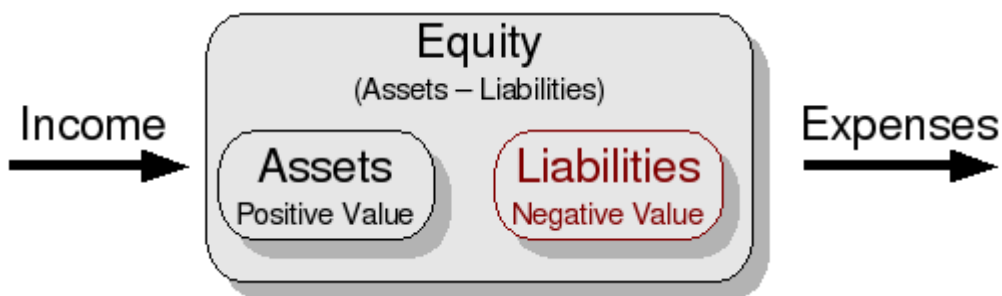
With the 5 basic accounts defined, what is the relationship between them? How does one type of account affect the others? Firstly, equity is defined by assets and liability. That is, your net worth is calculated by subtracting your liabilities from your assets:

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

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.



A graphical view of the relationship between the 5 basic accounts. Net worth (equity) increases through income and decreases through expenses. The arrows represent the movement of value.

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.

Double entry accounting serves two purposes. The first is to create an accounting trail, money always has to come from somewhere and go to somewhere. Additionally, double entry accounting historically served to double check the math of an accountant. Because the numbers are entered into multiple accounts simultaneously, there are multiple places to check to make sure the totals match. Of course, with the

advent of computers, the chances of a mathematical problem are low, but it is good to know that the concept still exists!

Double entry accounting has been around since the late 15th century, when it was described by an Italian friar, Luca Pacioli. Traditional double entry accounting involves recording each transaction in a book called a ledger, then copying each part of the transaction to separate books called journals. This method is still used in businesses today as a way to avoid entry errors and to track the source of those errors. GnuCash simplifies this traditional accounting by copying part of each transaction for you, so it may not catch some of the entry errors that would show up in traditional accounting. But it will flag transactions that are not balanced, and it will let you know when an account name is missing.

Note

Calling this double-entry bookkeeping is a bit misleading; it would be somewhat more accurate to call it multiple-entry bookkeeping, since a transaction can affect more than two accounts. Unfortunately, there's 700 years of history of use of the term, which sufficiently discourages changing it.

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, Accounts and Transactions

GnuCash uses *files* for storing information. GnuCash provides three types of files: data files, backup files, and log files. The main file that you will use to store your data is a data file. You will probably have only one data file that you use for your home data, but GnuCash will automatically save a backup copy for you each time you save that data file. GnuCash also provides log files which can be used to help reconstruct data. Backup and log files are described later in this chapter.

An *account* is a place for keeping track of what you own, owe, spend or receive. Although you only have one main data file, that file will contain many accounts. You probably already think of money you own or owe as being in an account. For example, at some point you opened checking and savings accounts at a particular bank, and that bank sends you monthly statements showing how much money you *own* in these accounts. Credit card accounts also send you statements showing what you *owe* to a credit card company, and the mortgage company may send you periodic statements showing how much you still *owe* on your loan.

In GnuCash, accounts are also used to categorize money you receive or spend, even though these are not physical accounts that receive statements. As we will cover more in Chapter 3, income type accounts are used to categorize money received (like a paycheck), and expense type accounts are used to categorize money spent (for pizza, to pay a bill, etc.) These accounts function much like categories in some other financial programs, with a few advantages discussed in Chapter 3.

A *transaction* represents the movement of money from one account to another account. Whenever you spend or receive money, or transfer money between accounts, that is a transaction. In GnuCash, as we will see in the next section, transactions always involve at least two accounts. Examples of transactions are: paying a bill, transferring money from savings to checking, buying a pizza, withdrawing money, and depositing a paycheck. Chapter 4 goes more in depth on how to enter transactions in GnuCash.

Double Entry

You've probably heard the saying, "Money doesn't grow on trees". It means that money must come from

somewhere - it doesn't just “appear”. *Double entry accounting* is a method of record-keeping that lets you track just where your money comes from and where it goes. Using double entry means that money is never gained nor lost - an equal amount is always transferred from one place to another. When you withdraw cash, you are transferring money from your bank account to your wallet. When you write a check to the grocery store, you are transferring money from your checking account to the grocery store. And when you deposit a paycheck, you are transferring money from your source of income to your bank account.

In GnuCash, these transfers are known as transactions, and each transaction requires at least two accounts. To enter the cash withdrawal, for example, you would enter a transfer of money from a bank account to a cash account. You would record the grocery check as a transfer from a checking account to a groceries expense account. And the paycheck deposit is recorded as a transfer from an income account to a bank account.

Unlike traditional accounting packages, GnuCash makes it very easy to enter your double entry transactions. Chapter 4 gives more detail on entering transactions, but for now let's take a general look at how they work. For a basic transaction like writing a check, you first create two accounts: a checking account and an account for the expense. For example, if you write a check to pay for groceries, then you need both a checking account and a groceries account. (See Chapter 3 for more detail on creating accounts.) To record the check, you simply enter a transaction to transfer money from the checking account to the groceries account. In this example, the GnuCash transactions look like this when viewed from the register windows of the checking account:

The screenshot shows the GnuCash interface with a menu bar (File, Edit, View, Actions, Reports, Tools, Help) and a toolbar with icons for Close, Enter, Cancel, Delete, Duplicate, Schedule, Split, and Blank. Below the toolbar, summary statistics are displayed: Present: \$950.00, Future: \$950.00, Cleared: \$1,000.00, Reconciled: \$0.00, and Projected. The main register table has columns for Date, Description, Transfer, R, Deposit, Withdrawal, and Balance. The first row shows an opening balance of 1,000.00. The second row shows a transaction on 01 September 2003 for a payment of 50.00 to the Grocery Store, with a balance of 950.00. A third row is highlighted in yellow, showing the input fields for a new transaction: Date (01), Description (Grocery Store), Transfer (Assets:Checking), R (Expenses:Groceries), Deposit (50.00), Withdrawal, and Balance (950.00).

Date	Description	Transfer	R	Deposit	Withdrawal	Balance
3	Opening Balance	Equity:Starting Balances	c	1,000.00		1,000.00
3	Grocery Store	Expenses:Groceries	n		50.00	950.00
01	Description	Transfer	n	Deposit	Withdrawal	Balance

Monday 01 September 2003

This image shows entering the date, description and transfer account for a payment of \$50 to the Grocery Store.

In this transaction, a check is written to Grocery Store for \$50. Since this is a double entry transaction, at least two accounts are affected and must be part of the transaction. GnuCash automatically enters the current account name (Assets:Checking) for you so you only enter the other account name affected (Expenses:Groceries).

Note

In this example for double entry accounting we used a checking account to pay for the groceries. But, notice that concept is the same no matter what method you use to pay for the groceries. If instead you used a credit card, the double entry accounting would simply involve your Credit Card account instead of the Checking account.

What about your paycheck? You can see that money goes into a bank account, but where does it come from? In double-entry, the money has to come from somewhere. Technically, that money comes from your employer, but you don't care about tracking your employer's accounts. So how do you account for money coming in? In GnuCash, you create a special income type account to track your incoming paychecks. (See Chapter 3 for more information on creating income accounts.) To record the paychecks, you simply enter a transaction to transfer money from the income account to a bank account.

Here's what a paycheck deposit looks like in GnuCash.

Num	Description	Transfer	R	Deposit	Withdrawal	Balance
	Opening Balance	Equity:Starting Balances	c	1,000.00		1,000.00
	Grocery Store	Expenses:Groceries	n		50.00	950.00
	Employers R Us	Income:Salary	n	600.00		1,550.00

Friday 31 January 2003

This image shows entering the date, description and transfer account for a \$600 Salary payment from Employers R Us.

In this example, \$600 is transferred from a Salary income account to the Checking Account. Because of the special nature of income accounts, discussed in Chapter 3, this transaction increases both the checking account balance and the income account balance by the amount transferred.

The main principle to remember is that there are at least two parts to every transaction, and the total amount transferred from a set of accounts must equal the total amount transferred to another set of accounts. When a transaction transfers an equal sum from accounts to other accounts, that transaction is said to be *in balance*. In GnuCash, as in double-entry accounting, you want to have all of your transactions in balance.

Why is this important to you? If all of your transactions are in balance, then your money is all accounted for. GnuCash has a record of where that money came from and where it was used. By storing the names of all accounts involved in each transaction, you provide data that can be sorted and viewed in report form later. Reports allow you to see things like how much money you made for the year and where it all went, what your net worth is, and what your taxes might be for the year. The more information you provide when entering transactions, the more detailed your reports will be.

Accounting Terminology

Accountants use the terms *debit* and *credit* to describe whether money is being transferred *to* or *from* an account. Money is recorded in the debit column, which is always the left column, when it is being transferred *to* an account. Money is recorded in the credit column, which is always the right column, when it is being transferred *from* an account. Money always flows from the right column of one account to the left column of another account.

The main rule of accounting is this: *For every transaction, total debits must equal total credits*. This is just another way of repeating the double entry rule, that for each transaction, the amount of money transferred *from* accounts must equal the amount transferred *to* other accounts

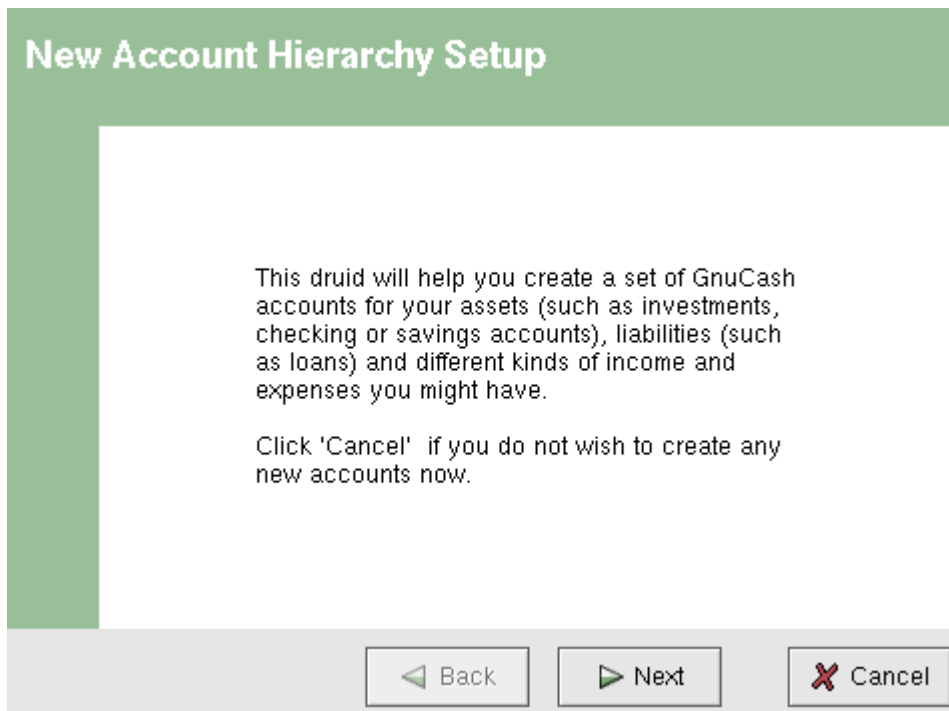
You don't have to use the terms debit and credit to use GnuCash, however. GnuCash account registers default to common column headings such as deposit and withdrawal - if you are more comfortable with those headings, use them. If you prefer the credit and debit headings, you can change the column headings to use accounting labels from the menu item Edit -> Preferences General (see section 2.5 for more detail on setting preferences).

The GnuCash Interface

The very first time you open GnuCash, you will see the Welcome to GnuCash! dialog. From there, GnuCash provides other tools to help you easily find what you are looking for. Let's take a look at some of the common screens and dialog boxes you will see.

Create a new set of accounts

The New Account Hierarchy Setup druid allows you to create several accounts at once. When you open GnuCash the first time, you will get this dialog:



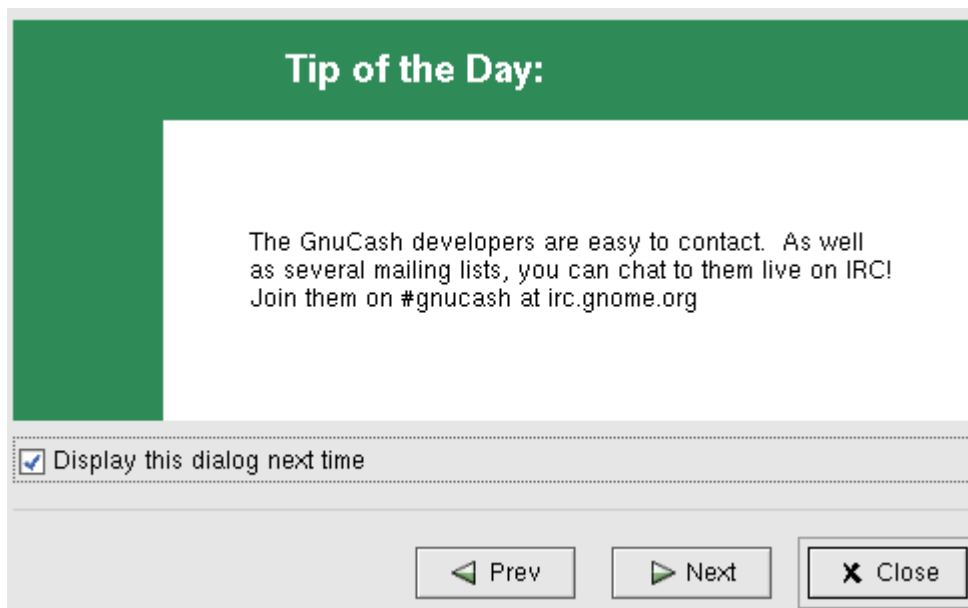
This image shows the first screen of the New Account Hierarchy Setup Druid.

Click Next and follow the instructions provided in the dialog. The dialog provides a default set of accounts with predefined account types, and you simply select the accounts you want. For more information on account types, see section 3.2.

If you intend to import your data from another program and you want to keep the same account names you used in that program, you may want to delay setting up default accounts at this time. You can simply import the data and the account names into GnuCash from the other program (Section 2.7 explains how to do this).

Tip of the Day

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

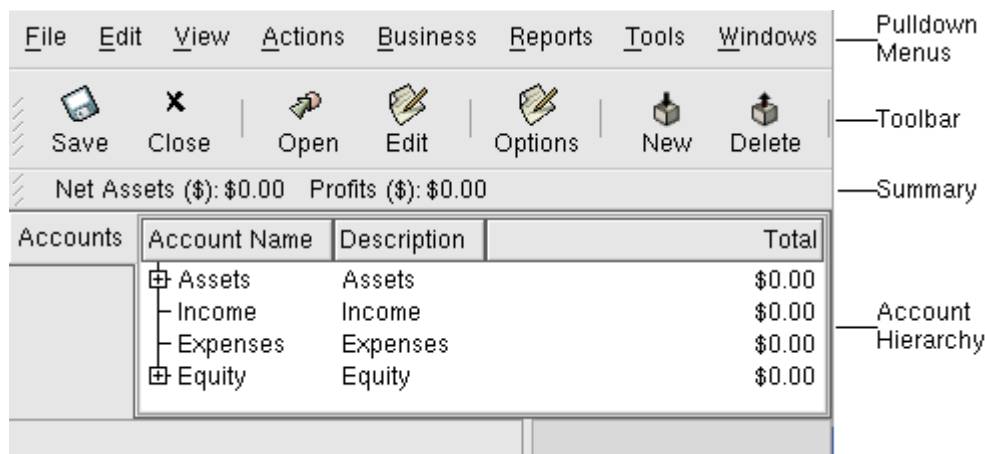


This image shows the Tip of the Day.

These tips provide useful information for beginning users. To view more of the tips, click Next to continue. If you do not wish to see this dialog box on startup, deselect the box next to Display this dialog next time. When you have finished viewing the helpful tips, click Close to close the Tip dialog.

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 "Simple Checkbook" accounts are shown.



This image shows the Accounts window.

The Account Tree window 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 any of the accounts. The individual account windows are called account registers. GnuCash allows you to have as many account registers open as you wish.

At the top of this window is the title bar, which displays the file name (once you have saved the file.) Below that is the menu bar. You can access the menu options by either clicking on these menu headings or by using shortcut key combinations, covered later in this chapter. Next is the tool bar, which contains buttons for the most common functions. Below that is the status bar, which tells you information about what you own (Net Assets) and how much money you have made (Profits).

The account tree appears below the status bar. 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 Options button.

Account Register Window

The Account Register windows are used to enter and edit your account data. As the name suggests, they look similar to a checkbook register. When you double-click an account name in the Accounts window, the Register window for that account will open.

Nun	Description	Transfer	R	Deposit	Withdrawal	Balance
	Opening Balance	Equity:Starting Balances	c	1,000.00		1,000.00
101	Grocery Store	Expenses:Groceries	n		50.00	950.00
	Employers R Us	Income:Salary	n	600.00		1,550.00
102	January Rent	Expenses:Rent	n		350.00	1,200.00
	Employers R Us		n			1,200.00
103	February Rent	Expenses:Rent	n		350.00	850.00

This image shows The Checking Account - Register with several transactions.

Chapter 4 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 title bar at the top contains the account name. Below that, the menu bar contains menu options related to the account register. Tool bar buttons simplify common data entry functions. The status bar appears below the tool bar, and it displays four types of account balances covered in Chapter 4. At the bottom of the account register window, information appears about the current location of the cursor.

Tool Bar Buttons

Both the account tree window and the account register window contain tool bar 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 - Use this function often to save the current file to disk.
- Close - Use this function to close the current notebook page.
- Open, Edit, New and Delete - These are functions related to accounts. They are discussed in Chapter 3.
- Options - This button edits the account view options.
- Exit - Closes your GnuCash session.

Register buttons are discussed in Chapter 4.

Menu Items

The account tree window and the account register window both contain menu headings in a menu bar. 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. 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, described next.

Menu Shortcuts

Most of the menu items have keyboard shortcuts, and these 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. A few of the menu items use **Ctrl** key and function (for example **F3** to open a new file) key shortcuts as well, and these are listed next to the items.

GnuCash, as a Gnome application, also allows you to define your own menu shortcuts. To define a shortcut, click on the menu header and move the mouse pointer over the menu item. While the menu item is highlighted, press a key or key combination to set the shortcut. You should now see your shortcut choice next to the item in the menu. To delete a shortcut, press the **Delete** key while the menu item is highlighted.

Setting Preferences

GnuCash allows you to customize your session by setting several preference options. From the menu select Edit -> Preferences, then make your desired changes using the list of items shown on the left. Once you have finished making your preference changes, select Apply or OK to apply the changes to your session. (Selecting Apply applies the changes while leaving the Preferences window still open. Selecting OK applies the changes and closes the Preferences window.) At any time, you can click on the Default button to return the options to the default preference setting.

Accounts

The Accounts item lets you set the following options:

- Account Separator - The account separator is the character that separates a parent account from its sub-account, for example Utilities:Electric. The default is a :(Colon), but you can also select /(Slash), \ (Backslash), -(Dash) or .(Period). Parent accounts and sub-accounts are discussed in section 3.4
- Reversed-balance accounts types - This option lets you determine whether account balances will display as positive or negative numbers:
 - None shows all credit balances as negative and all debit balances as positive. (See Accounting Terminology note in this chapter for more information on debits and credits.)
 - Credit Accounts (default) displays a positive balance for account types that would normally carry a credit balance (income, credit, liability, equity). See Chapter 3 for more information on these account types.
 - Income & Expense assigns a positive credit balance to income account balances and a negative debit balance to expense account balances. See Chapter 3 for more information on these account types.
- Use accounting labels - Select this option if you want column headings to refer to debits and credits instead of the default informal headings. (See Accounting Terminology note in this chapter for more information on debits and credits.)

Business

The Business item lets you set the following options:

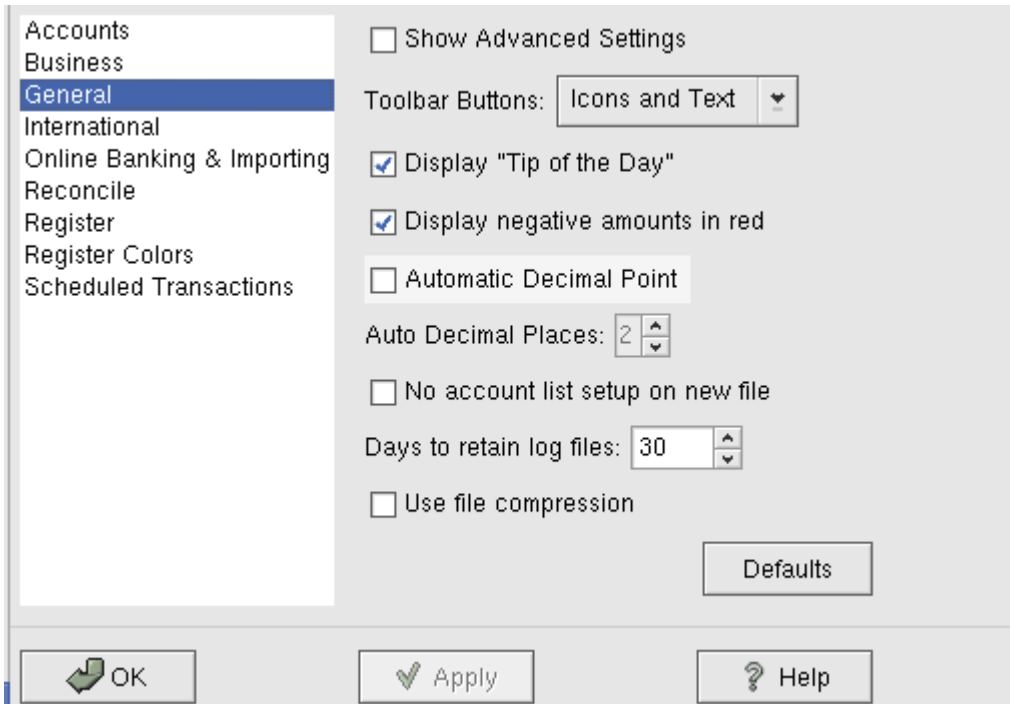
- Number of Rows - Lets you select the default number of register rows to display in invoices, bills

and expense vouchers.

- Invoice (or Bill) Tax Included? - Determines whether taxes specified in the Tax Table are already included in the value of the item. Set this if you're in a locale (like Australia) where the taxes (e.g. GST) is included in the price of your goods. In the US, where the tax is added to the value, you do not want to set Tax Included.
- Notify Bills Due? - Lets you set whether you want to be notified of when a bill is soon to be due. If selected, you can specify the number of days in advance to notify with the Bills Due Days: option.

General

In General, you will see the default settings shown here:



This image shows the GnuCash Preferences dialog where the options of how GnuCash looks and acts are setup.

- Show Advanced Settings - When this is selected, GnuCash displays the Advanced item in the GnuCash Preferences dialog. You must click on "Apply" before the advanced page will be displayed.
- Tool Bar Buttons - Select whether to display icons, text, or both on the tool bar.
- Display "Tip of the Day" - You can turn on or off the option to display the "Tip of the Day" dialog on startup.
- Display negative amounts in red - If you turn off this option, GnuCash will display negative numbers in black.
- Automatic Decimal Point - This option will automatically insert a decimal point into numbers you type in.
- Auto Decimal Places - If you select the Automatic Decimal Point option, this option allows you to set the number of decimal places to be used.
- No account list setup on new file - This option turns off the display of the New Account Hierarchy Setup druid when File -> New File is selected.
- Days to retain log files - This option deletes log files (and backup files) after the number of days set here.
- Use file compression - This option determines whether the GnuCash data file will be compressed

or not.

International

The International preferences item allows you to control the country specific behavior of GnuCash.

- Date Format - This item controls the appearance of the date. The available options are US, UK, Europe, ISO, and Locale. Choosing "Locale" will automatically determine the date format based on your operating system settings.
- New Account Default Currency - This item determines which currency will be selected by default when creating new accounts.
- Default Report Currency - This item determines which currency will be used by default when creating reports.
- Use 24-hour time format - This item toggles the use of 24 hour time format.
- Enable Euro support - If selected, enables support for the European Union EURO currency.

On-line Banking

The On-line Banking & Importing item contains only one option, QIF Verbose Documentation. The first time you use the importer (discussed later in this chapter), you may notice that the importer has detailed instructions on how to import a file. Once you have become familiar with using the importer, you might want to turn off this option. Turning off the option gives you less detail in the importer dialogs.

Reconcile

The Reconcile preferences item affects the behavior of GnuCash when reconciling accounts. More information about reconciling accounts can be found in "Chapter 4: Entering Transactions" of this guide.

- Automatic interest transfer - If selected, prior to reconciling an account which charges or pays interest, prompt the user to enter a transaction for the interest charge or payment. Currently only enabled for Bank, Credit, Mutual, Asset, Receivable, Payable, and Liability accounts.
- Automatic credit card payments - If selected, after reconciling a credit card statement, prompt the user to enter a credit card payment.
- Check off cleared transactions - If selected, automatically check off cleared transactions when reconciling.

Register

The Register preferences item affects the behavior of the transaction register windows. More information about the transaction register can be found in "Chapter 4: Entering Transactions" of this guide.

- Default Register Style - you can choose from 3 different styles for register windows, Basic Ledger (default), Auto-Split Ledger, and Transaction Journal. The basic style is a standard one-line register where everything appears on a single line. The auto-split style is similar to the basic style except that the current transaction is split-expanded. The journal style has every transaction split-expanded.
- Double Line Mode - If selected, show two lines of information for each transaction instead of one.
- 'Enter' moves to blank transaction - If selected, move to the blank transaction after the user presses 'enter', otherwise move down on row.
- Confirm before changing reconciled - If selected, use a dialog to confirm a change to a reconciled transaction.

- Register font: - The font to use in the transaction register window.
- Register hint font: - The font to use to show hints in the transaction register window.

Register Colors

The Register Colors preferences item changes the appearance of the register window. To change the color of one of the items listed, simply click on the color box and choose the new color. More information about the transaction register can be found in "Chapter 4: Entering Transactions" of this guide.

If selected, the item Double mode colors alternate with transactions configures the register window to alternate between the primary and secondary colors with each transaction, instead of each row.

Scheduled Transactions

The Scheduled Transactions preferences item affects the behavior of GnuCash for scheduling transactions. More information about scheduled transactions can be found in "Chapter 4: Entering Transactions" of this guide.

- Run on GnuCash start - If selected, the "Since-Last-Run" window will appear on GnuCash startup.
- Auto-Create new Scheduled Transactions by default - If selected, new Scheduled Transactions will have the 'Auto Create' flag set by default.
- Notify on new, auto-created Scheduled Transactions - If selected, new Scheduled Transactions will have the 'Auto Create' flag set to 'Notify' by default.
- Default number of days in advance to create - The default number of days in advance to create new Scheduled Transactions.
- Default number of days in advance to remind - The default number of days in advance to remind on new Scheduled Transactions.
- Template Register Lines: - Number of lines in the template register to display.

Getting Help

GnuCash offers help in many ways. We have already covered the Tip of the Day dialog that gives you helpful hints upon startup of your GnuCash session. GnuCash also offers an extensive help manual.

Help Manual

Once you have opened GnuCash, you will see the Account Tree window Help menu heading, which opens the on-line manual. The on-line manual is organized by topic, and you can expand each topic into its subtopics.

Topics are listed on the left side. To select a topic or subtopic, click on it, and you should see the text for that topic appear on the right. Use the Back and Forward buttons to navigate through your topic choices, and print any text using the Print button.

Web Access

The GnuCash Help window also acts as a simple web browser, so you can pull up a web site for additional information. You can open any web site under this window by clicking the Open tool bar button and then typing in the URL. Use the Back, Forward, Reload, Stop, and Print buttons as you would in a standard browser.

The [GnuCash](#) web site contains helpful information about the program and about any updates to it. It also contains links to the GnuCash mailing lists for developers and users, and you can search the archives of

these mailing lists for discussions on a particular topic. If you don't find the answers you are looking for, you can post your question to the user list, and someone on the list will attempt to answer you.

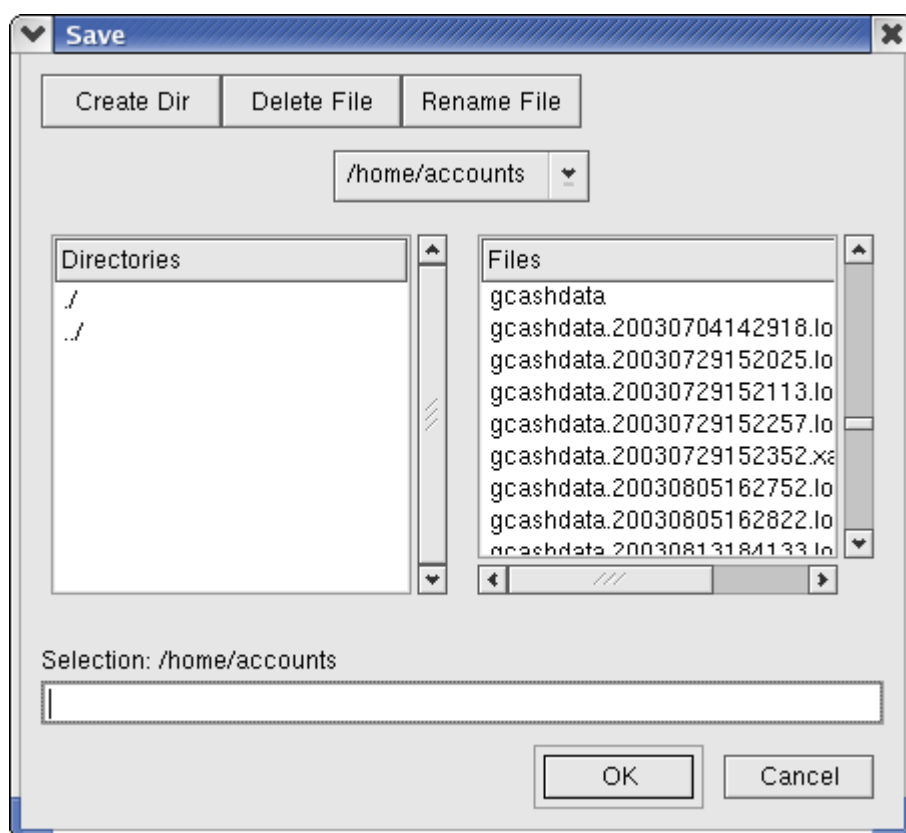
Topic Search

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.

Creating and Saving Files

If you are using GnuCash for the first time, it will automatically open a new file when you begin. Follow these steps to save the file under a new name:

1. Choose File -> Save As... from the menu bar or select the Save toolbar button. GnuCash will bring up this window:



This image shows the Save dialog.

2. In the text box below Selection type your chosen file name (e.g. gcashdata).
3. Click the OK 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. To create a new file:

1. From the main menu, choose File -> New File.
2. Follow the steps above for saving a file under a new name.

Before ending each GnuCash session, be sure to save your data changes using File -> Save or the Save toolbar button. In fact, it is a good idea to save your data changes often, since GnuCash does not permanently record your changes until you save them. To open an existing file, select File -> Open and choose your file from the list.

Importing QIF Files

Note

This section tells you how to initially import data from other financial programs that use QIF (Quicken® Interchange Format). Section 11.5 addresses importing QIF data from online bank statements.

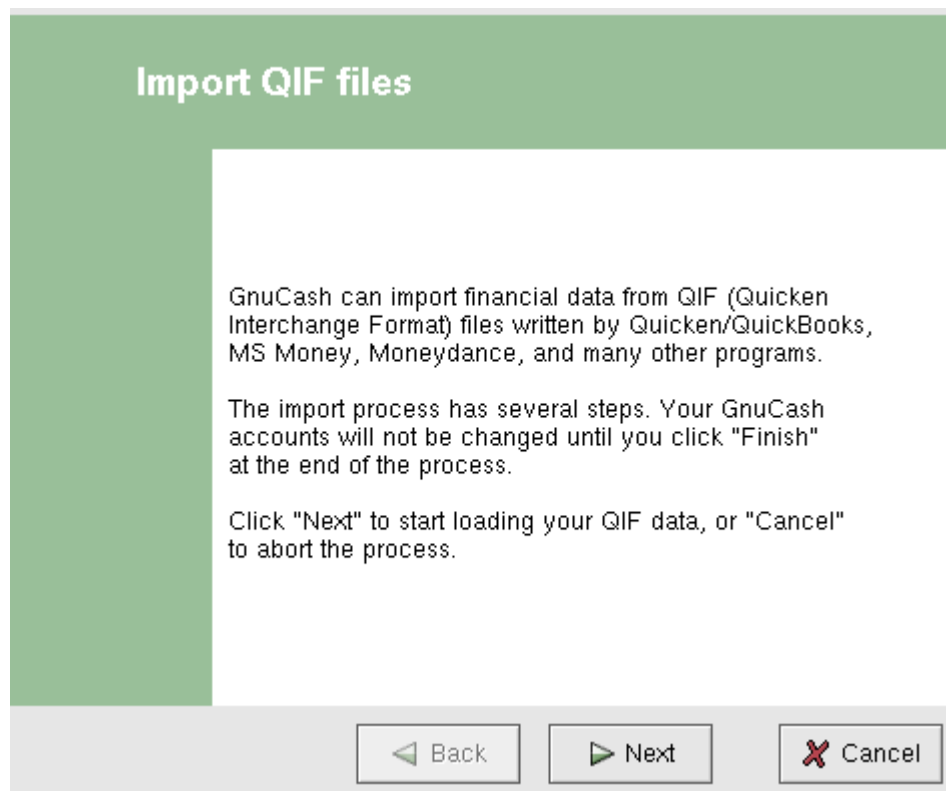
To import data from Quicken®, MS Money, or other programs that use QIF(Quicken® Interchange Format), you must first export your data to a QIF file. One way to do this is to export each account as a separate QIF file. An easier way, available in Quicken® 98 and beyond, is to export all accounts at once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

Once your data is in QIF form, follow the easy 3-step import process described below. Before you get started, though, please note that part of the import process involves checking for duplicate transactions, so it is very important that you load *all* of your QIF files before importing. Duplicate transactions result when more than one account is involved. For example, if one of your transactions transferred money from savings to checking, you would end up with this same transaction recorded twice - once in the savings account and once in the checking account.

To import QIF files:

1. 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 Druid.

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.

2. 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 (see section 3.1 for more on this). 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 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.

3. 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.

Backing Up and Recovering Data

GnuCash creates several types of files to help ensure that your data is not lost. You may see files with the following file extensions: xac, log, LCK, LNK in the same directory as your principal data file. What each of these files does is presented below.

Backup file (.xac)

Each time you save your data file, a backup copy will also be saved with the extension xac. 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 `gcashdata.20010404185747.xac` indicates this is a backup copy of the file `gcashdata` saved in the year 2001, April 4, at 6:57:47 p.m.

To restore an old backup file, simply open the .xac file with the date to which you wish to return. Be sure and save this file with a different file name, do not use a .xac extension for you regular file name.

Log file (.log)

Each time you open a file in GnuCash, a .log is created and saved with the same name format as the .xac backup files. As you make changes to the open data file, the log file saves only those changes. 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:

- Open the last saved GnuCash file.
- 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 wreck 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.).

Note

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 - data files do not have an automatic file extension. It's a good idea to keep some of the more recent .xac files, but you can safely delete the .log files since they are not complete copies of your data. You should also delete any .LCK and .LNK files that you see after closing GnuCash. If you decide to manually back up your data file to another disk, you only need to back up the main data file - not the .xac files.

Note

GnuCash will automatically delete any .xac, .log files that are older than 30 days. You can change this behavior in the GnuCash preferences.

Putting It All Together

Note

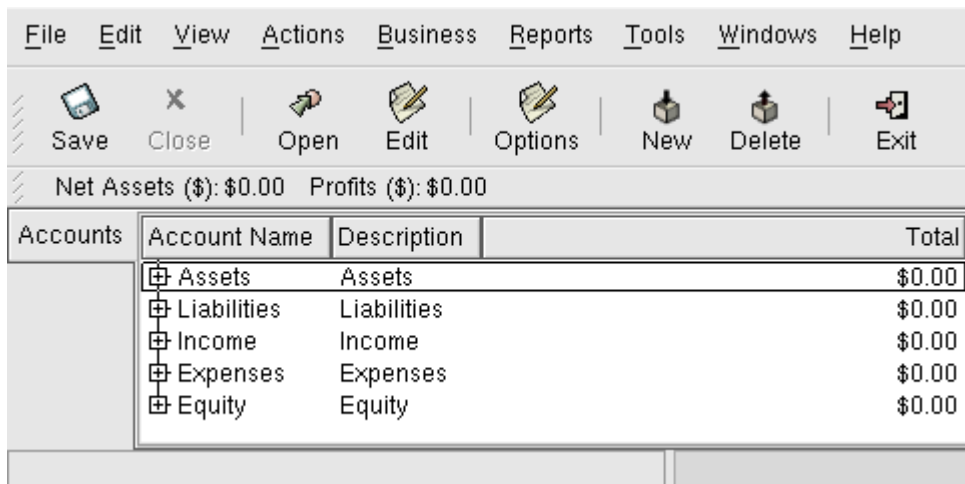
This section begins a tutorial that will continue throughout this book. At the end of each chapter, you will see a Putting It All Together section that walks you through examples to illustrate concepts discussed in that section. Each Putting It All Together section builds on the previous one, so be sure 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 go through the New Account Hierarchy Setup dialog. You will create a file and accounts to be used for your home finances, so choose the accounts that pertain to you. If you are not sure what accounts you will need, click the Select All option to create all of the accounts provided. When you have completed the setup, save

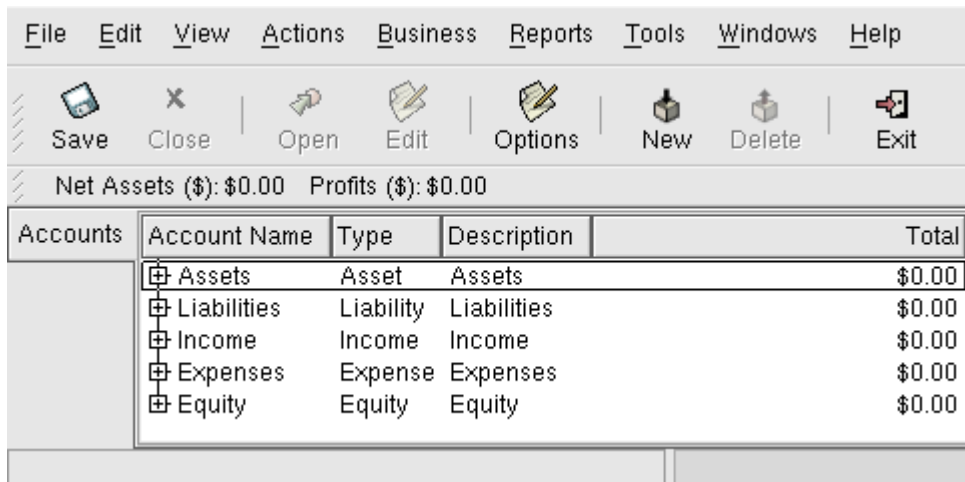
this file with a filename you want to use for your home data.

- Next, create a file to store test data. This is the file that will be used in the tutorials throughout this manual. Click File -> New to create a new file, and name it `gcashdata`. Your main window should now look something like this:



This image shows the Account Tree window with a set of accounts loaded from the druid.

- Now, adjust some preferences in the account tree window. Select Edit -> Preferences -> General and change the tool bar buttons so that they display only text, not icons. Click Apply and look at the tool bar buttons in the main window. Return to the Preferences dialog and change the tool bar buttons back to the default setting by clicking Default. From the tool bar, select Options and select the "Type" item in the Account Fields to display option. The main window should now look like this:



This image shows the effect of changing the display options.

Save your file.

- If you plan to import QIF data from another program, let's try a practice file first. (If not, skip to the next chapter on account setup.) First, create a new file called `gcashimport`. From the main menu, select File -> Import -> Import QIF... to start the import. When you are prompted for a filename, select (???GnuCash should have a sample QIF file available for this tutorial - can that be done easily?) Load the file and look at the results of the account match. Complete the import and take a look at the results. The main window should now look like this: (?Insert a screen shot here that corresponds to an imported test file) Save your file.

Chapter 3. Accounts

Table of Contents

[Basic Accounting Concepts](#)

[Balance Sheet Accounts](#)

[Income and Expense Accounts](#)

[GnuCash Accounts](#)

[Balance Sheet Accounts](#)

[Income and Expense Accounts](#)

[Putting It All Together](#)

[Simple Example](#)

[The Basic Top Level Accounts](#)

[Making Sub-Accounts](#)

This chapter will discuss 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.

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.

Balance Sheet Accounts

The three so-called *Balance Sheet Accounts* are Assets, Liabilities, and Equity. Balance Sheet Accounts are used to track the changes in value of 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.

Expenses refer to money you spend to purchase goods or services provided by someone else. 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.

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 reside 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 13 account types in GnuCash. These 13 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 7 asset accounts (*Cash*, *Bank*, *Stock*, *Mutual Fund*, *Currency*, *Accounts Receivable*, and *Asset*), 3 liability accounts (*Credit Card*, *Accounts Payable*, and *Liability*), 1 equity account (*Equity*), 1 income account (*Income*), and 1 expense account (*Expense*).

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.
5. **Currency** If you trade other currencies as investments, you can use this type of account to keep track of them. The register is similar to the stock register, except that you enter exchange rates instead of prices.
6. **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 in these funds arriving.
7. **Asset** For personal finances, use this type of account to track "big-ticket" item purchases that significantly impact your net worth. Generally, you can think of these as things you insure, such as a house, vehicles, jewelry, and other expensive belongings.

Tip

For all GnuCash asset accounts, a *debit* (left-column value entry) increases the account balance and a *credit* (right-column value entry) decreases the balance. (See note later in this chapter.)

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.
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.

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 only a single GnuCash equity account, called naturally enough, Equity.

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 Chapter 2, 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. (Insert a chart that shows common accounts and recommended account types for these)

More on Debits and Credits

Remember the terms debit and credit discussed in Chapter 2? Contrary to popular belief and even some dictionary definitions, accounting debits and credits do not mean decrease and increase. The only constant definition of debits and credits is that debits are left-column entries and credits are right-column entries. In fact, debits and credits each increase certain types of accounts and decrease others. In asset and expense type accounts, debits increase the balance and credits decrease the balance. In liability, equity and income type accounts, credits increase the balance and debits decrease the balance.

For example, debits *increase* your bank account balance and credits *decrease* your bank account balance. Wait a minute, you might say, a *debit* card *decreases* the balance in my checking account, because I take money out of it. And when the bank gives me money back on something, they *credit* my account. So why is this reversed in accounting?

Banks report transactions from *their* perspective, not yours. Their perspective is exactly opposite to yours. To you, your bank account represents an asset, something you own. To the bank, your bank account represents a loan, or liability, because they owe you that money. As explained in this chapter, asset and liability accounts are exact opposites in the way they behave. In a liability account, debits *decrease* the balance and credits *increase* the balance.

When you take money out of your bank account, the balance in your account decreases. To you, this is a decrease in an asset, so you *credit* your bank account. To the bank, this is a decrease in a liability, so they *debit* your bank account.

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

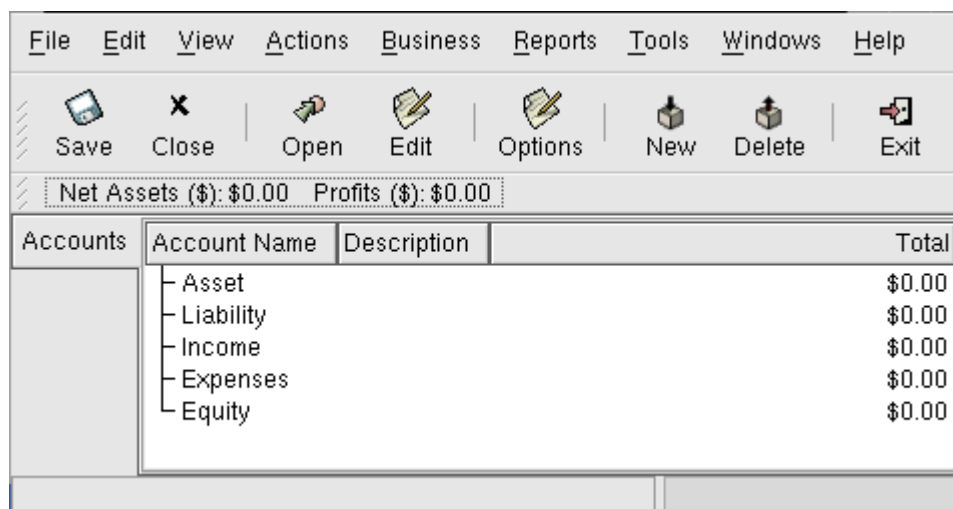
groceries, rent, electricity, phone, and taxes (Federal, Social Security, Medicare) on your salary. Simple, isn't it?

The Basic Top Level Accounts

Now, we must decide how you want to group these accounts. Most likely, you want your assets groups 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 (do not select any predefined accounts) and build this basic starting account structure (File -> New Account...).

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



The screenshot shows the GnuCash application window. The menu bar includes File, Edit, View, Actions, Business, Reports, Tools, Windows, and Help. The toolbar contains icons for Save, Close, Open, Edit, Options, New, Delete, and Exit. Below the toolbar, the status bar displays "Net Assets (\$) : \$0.00" and "Profits (\$) : \$0.00". The main window shows a table of accounts with the following columns: Accounts, Account Name, Description, and Total.

Accounts	Account Name	Description	Total
	- Asset		\$0.00
	- Liability		\$0.00
	- Income		\$0.00
	- Expenses		\$0.00
	- Equity		\$0.00

This image shows the basic top-level accounts.

Making Sub-Accounts

We 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).

1. Account name Checking (account type Bank, parent account Asset)
2. Account name Savings (account type Bank, parent account Asset)
3. Account name Visa (account type Credit Card, parent account Liabilities)
4. Account name Salary (account type Income, parent account Income)
5. Account name Phone (account type Expenses, parent account Expenses)
6. Account name Electricity (account type Expenses, parent account Expenses)
7. Account name Rent (account type Expenses, parent account Expenses)
8. Account name Groceries (account type Expenses, parent account Expenses)
9. Account name Taxes (account type Expenses, parent account Expenses)
10. Account name Federal (account type Expenses, parent account Expenses:Taxes)

11. Account name Social Security (account type Expenses, parent account Expenses:Taxes)
12. Account name Medicare (account type Expenses, parent account Expenses:Taxes)
13. Account name Starting Balances (account type Equity, parent account Equity)

Accounts	Account Name	Description	Total
	[-] Asset		\$0.00
	[-] Checking		\$0.00
	[-] Savings		\$0.00
	[-] Liability		\$0.00
	[-] Visa		\$0.00
	[-] Income		\$0.00
	[-] Salary		\$0.00
	[-] Expenses		\$0.00
	[-] Electricity		\$0.00
	[-] Groceries		\$0.00
	[-] Phone		\$0.00
	[-] Rent		\$0.00
	[-] Taxes		\$0.00
	[-] Federal		\$0.00
	[-] Medicare		\$0.00
	[-] Social Security		\$0.00
	[-] Equity		\$0.00
	[-] Starting Balances		\$0.00

This image shows a simple chart of accounts.

Save this chart of accounts somewhere, as we will continue to use it 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.

Chapter 4. Transactions

Table of Contents

[Basic Concepts](#)

[The Account Register](#)

[Simple Transaction](#)

[Split Transaction](#)

[Features of the Account Register](#)

[Choosing a Register Style](#)

[Using Entry Shortcuts](#)

[Reconciliation](#)

[Reconcile Window](#)

[Scheduled Transactions](#)

[Creating from the Ledger](#)

[Creating from the Editor](#)

[Putting It All Together](#)

[Opening Balances](#)

[Additional Transaction Examples](#)

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.

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 passing value to the "to" account. Accountants call these parts of a transaction *Ledger Entries*. In GnuCash, they are called *Splits*.

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

The *account register* is the GnuCash window which allows you to view or edit preexisting transactions, or add new transactions for a particular account. To enter the account register, simply highlight the account name in the account tree window and double-click to open it. You can also click the Open button in the tool bar or use the right mouse button menu to open the account. GnuCash will display the account register window.

Simple Transaction

For example, starting with the chart of accounts we created in the previous chapter, 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: June 1, 2003), a description (eg: "Opening Balance"), click on the Transfer pop-up menu and select Equity:Starting Balances, add a deposit value of \$1000, and press the Enter key. The account register should now appear similar to this figure:

Num	Description	Transfer	R	Deposit	Withdrawal	Balance
	Opening Balance	Equity:Opening Balances	n	1,000.00		1,000.00
<i>Num.</i>	<i>Description</i>	<i>Transfer</i>	<i>n</i>	<i>Deposit</i>	<i>Withdrawal</i>	<i>Balance</i>

This image shows Asset:Checking - Register after inserting a starting value transaction.

Setting the starting balances of an account is an example of a simple two account transaction. In this case, affecting the Asset:Checking and the Equity:Starting 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 Asset:Checking account, you would set Transfer to Expenses:Groceries. The account register should now appear:

Num	Description	Transfer	R	Deposit	Withdrawal	Balance
	Opening Balance	Equity:Opening Balances	n	1,000.00		1,000.00
	Grocery Store	Expenses:Groceries	n		45.21	954.79

Monday 01 September 2003

This image shows Asset:Checking - Register after adding a transaction for groceries.

Split Transaction

The need for 3 or more account transactions 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 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"). In the "split" line below this, the the gross total of your paycheck (eg: \$1000) as a withdrawal transfer from Income:Salary. Now, individually insert deposits to Asset:Checking (eg: \$670), Expenses:Taxes:Federal (eg: \$180), Expenses:Taxes:Medicare (eg: \$90), and Expenses:Taxes:Social Security (eg: \$60). The final split should look like this:

Num	Description	Transfer	R	Deposit	Withdrawal	Balance
	Opening Balance	Equity:Opening Balances	n	1,000.00		1,000.00
	Grocery Store	Expenses:Groceries	n		45.21	954.79
	Employers R Us		n	670.00		1,624.79
		Assets:Checking Account	n	670.00		
		Expenses:Taxes:Federal	n	180.00		
		Expenses:Taxes:Social Se	n	60.00		
		Expenses:Taxes:Medicare	n	90.00		
		Income:Salary	n		1,000.00	

Employers R Us

This image shows Asset:Checking - Register after adding a split transaction for salary.

Features of the Account Register

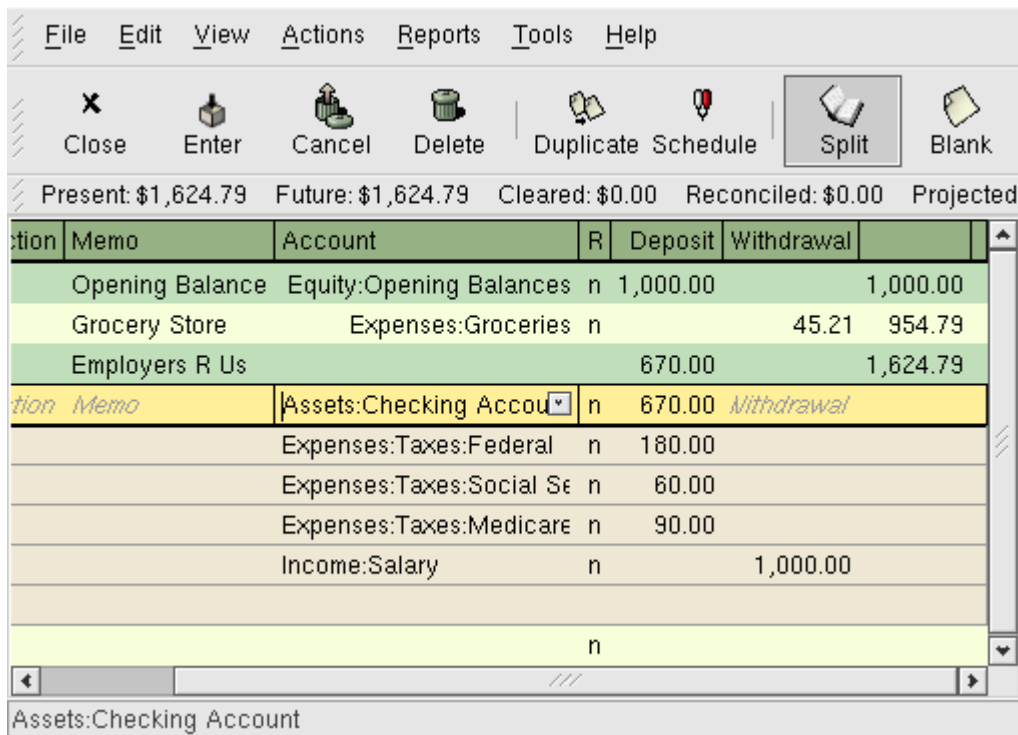
The title bar of the account register displays the account name, which in this case is Asset:Checking. Below the title bar, the menu bar displays the menu items available within the account register, and the

tool bar contains handy buttons that help you work with the account register.

Although each transaction has at least two splits, all you see in the basic register is a summary of the splits affecting the current account. In the Transfer column, you can see the other account from which money is *transferred* into or out of this account. If the transaction affects more than two accounts, GnuCash displays -- Split Transaction -- to show it is a multiple-split transaction. You can see the individual splits of each transaction by clicking the Split button in the tool bar 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 (Total 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.

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:



This image shows how split headings change.

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 section 2.1, 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 tool bar.

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

Num	Description	t Charge	Tot Income	Balance
	Employers R Us	t Charge	1,000.00	1,000.00
	Assets:Checking Account	n 670.00		
	Expenses:Taxes:Federal	n 180.00		
	Expenses:Taxes:Social Security	n 60.00		
	Expenses:Taxes:Medicare	n 90.00		
	Income:Salary	n 1,000.00		

This image shows a jump to the Income:Salary account.

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.

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 by selecting View -> Style from the menu. The 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. The Transaction Journal style is more like an accounting journal, in which all splits are shown for all transactions.

All styles permit you to view your data in either single-line or double-line format. In the View -> Style menu item, check the box next to Double Line, and you will see your transaction line expand to two register lines.

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.

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

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.

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.

Most of the menu items have keyboard shortcuts, 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-T** brings up the Transaction menu, then typing **S** will split the transaction. A few of the menu items use **Ctrl** key shortcuts instead, and these 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 **ShiftTab** moves to the previous box
- Space bar toggles the status between reconciled and not reconciled
- ↑ and ↓ navigate through the entries within the current box

Reconciliation

Transactions are typically checked against bank statements - a process known as "reconciliation". GnuCash keeps track of the reconciliation status of each transaction.

The reconciliation status of a transaction is shown by the reconciliation (R) field. '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. You can toggle the reconciliation status between 'n' and 'c' by clicking in the (R) field; you can set it to 'y' by using the Reconciliation Window.

At the bottom of the account window, there are two running totals: 'Balance', and 'Cleared'. 'Cleared' should correspond to how much money the bank thinks you have in your account, while 'Balance' includes outstanding transactions.

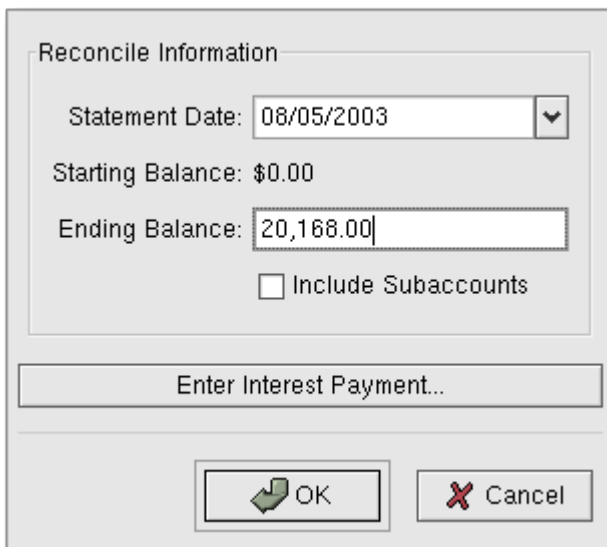
Reconcile Window

The Reconciliation Window is 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.

At the bottom of the account window, there are two running balances: the "cleared and reconciled" balance, and the "total" balance. The "cleared and reconciled" balance should correspond to how much money the bank thinks you have in your account, and the "total" balance 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 think 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 and reconciled" balance will change to include this amount. When the bank statement arrives, you can then compare it to what you've recorded in GnuCash in the Reconciliation window. There, you will be able to change the R field to y (reconciled). You cannot reconcile in the register window, you must use the reconciliation window. Once a transaction has been marked "reconciled", it can no longer be easily changed.

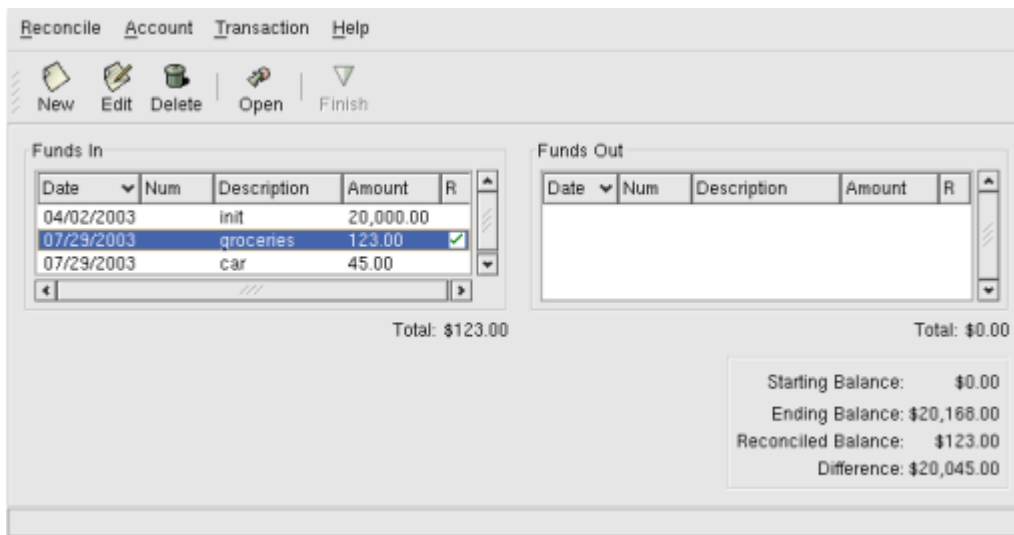
To use the Reconciliation Windows, select an account from the account tree and click on Actions -> Reconcile. A window will appear in which you can enter the reconcile information.



The initial reconcile window.

- *Reconcile Info: Statement Date* - this is the date of the statement you will be reconciling against.
- *Reconcile Info: 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.
- *Reconcile Info: Ending Balance* - this is the ending balance as it appears in the statement.
- *Reconcile Info: Include Sub-accounts* - select this if you want to include the sub-accounts of the currently selected account in this reconciliation.
- *Enter Interest Payment* - click here to enter an interest payment to XXX (??).

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



The transactions listing in the reconcile window.

The two lists called Funds In and Funds Out lists all the unreconciled transactions. The 'R' columns show whether the transactions have been cleared or reconciled.

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

If you cannot find one, then perhaps you forgot to enter a transaction, or did not know that the transaction had happened. You can use the New button on the tool bar, or the New menu item in the Transaction menu, to open a register window and enter the missing transaction. The new item will appear in the Reconcile window when you press enter 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 tool bar, or the Edit menu item in the Transaction menu, to open a register window and correct the transaction.

If the amounts agree, click on the item in the Reconcile window. The 'R' column will change to 'y' (reconciled) from 'n' or 'c'.

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.

At the bottom of the "Reconcile" window 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.)

When you have marked off all the items on the bank statement, and when the difference is \$0.00, press the Finish button on the tool bar or select Finish from the Reconcile menu (the latter method will work even if your "Difference" isn't \$0.00). The Reconcile window will close. In the Register window, the R field of the reconciled transactions will change to 'y'.

Scheduled Transactions

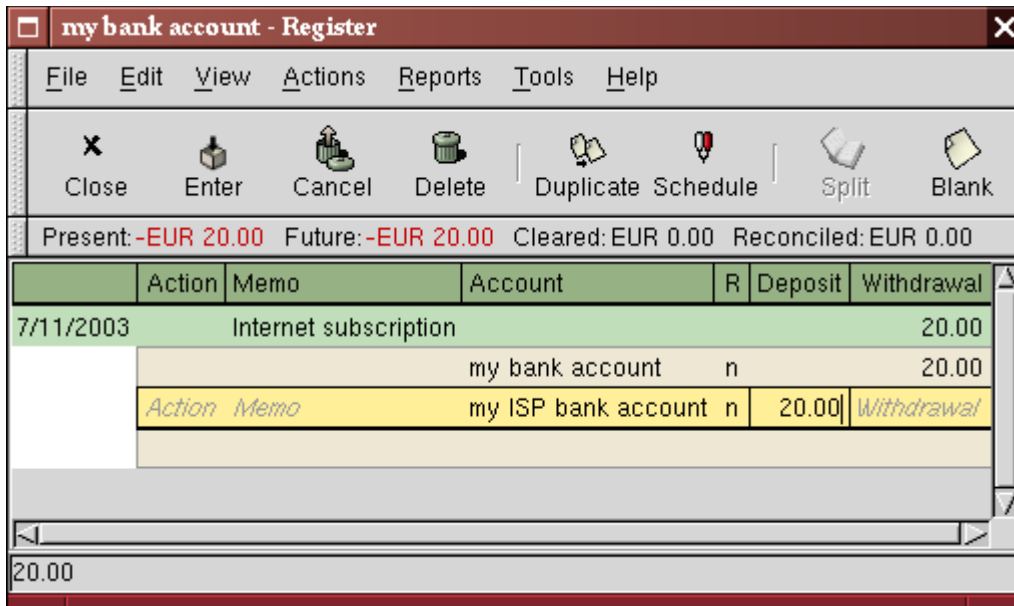
Scheduled transactions is made to help entering repetitive money operations, like subscriptions, insurances or taxes. By using scheduled transactions, you only have to enter the concerned transaction once, set a few parameters like start date, frequency and a little description, and then GnuCash will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

In this howto, we'll take a monthly Internet subscription of 20 EUR as example, which is taken on the first day 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.

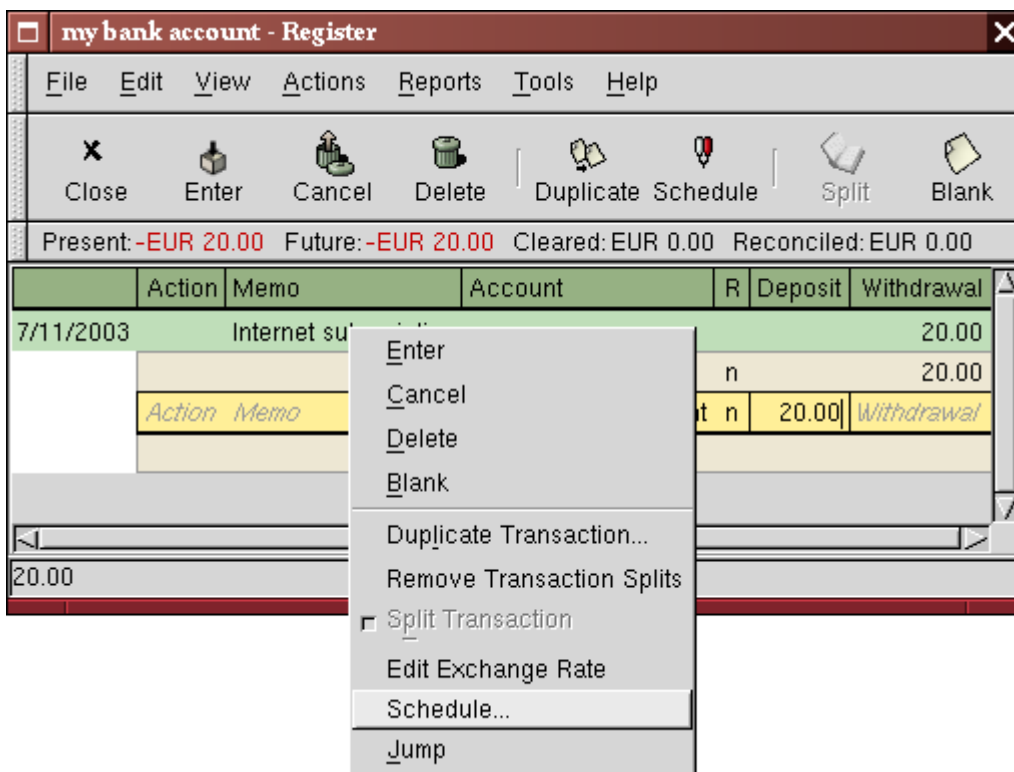


The screenshot shows a window titled "my bank account - Register" with a menu bar (File, Edit, View, Actions, Reports, Tools, Help) and a toolbar with icons for Close, Enter, Cancel, Delete, Duplicate, Schedule, Split, and Blank. Below the toolbar, summary statistics are displayed: Present: -EUR 20.00, Future: -EUR 20.00, Cleared: EUR 0.00, Reconciled: EUR 0.00. The main area is a ledger table with columns: Action, Memo, Account, R, Deposit, and Withdrawal. The first entry is dated 7/11/2003 with the memo "Internet subscription" and a withdrawal of 20.00. The account is "my bank account" with a reference "n". Below this, an "Action Memo" entry shows a withdrawal of 20.00 from "my ISP bank account" with a reference "n".

Action	Memo	Account	R	Deposit	Withdrawal
7/11/2003	Internet subscription				20.00
		my bank account	n		20.00
	Action Memo	my ISP bank account	n	20.00	Withdrawal

Step one creating scheduled transaction from the ledger

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

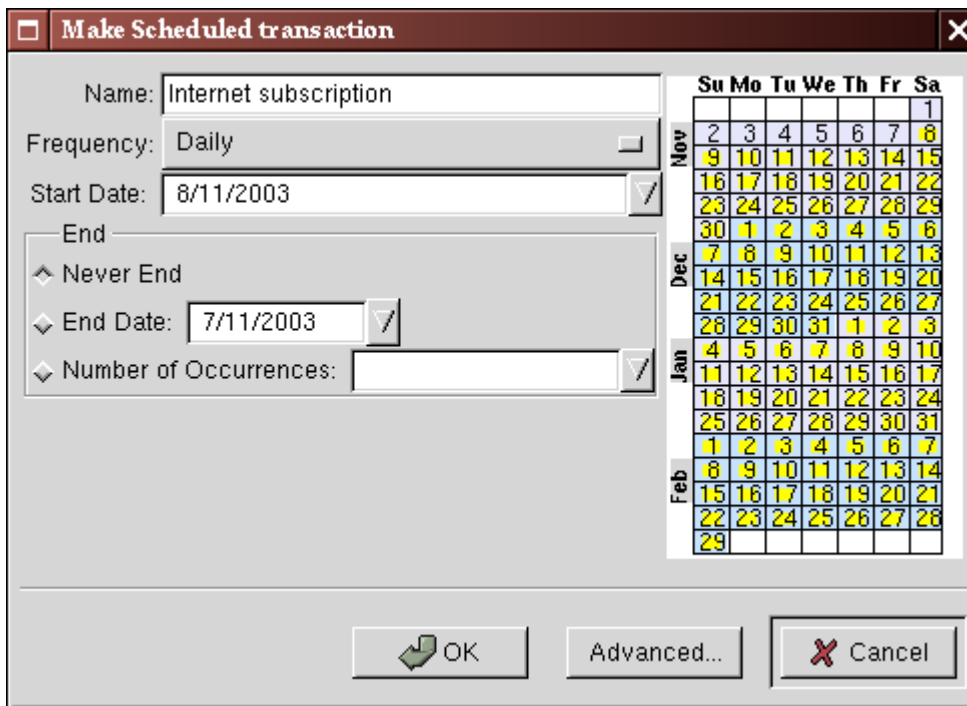


The screenshot shows the same window as above, but with a context menu open over the "Internet subscription" entry. The menu options are: Enter, Cancel, Delete, Blank, Duplicate Transaction..., Remove Transaction Splits, Split Transaction (highlighted), Edit Exchange Rate, Schedule..., and Jump.

Action	Memo	Account	R	Deposit	Withdrawal
7/11/2003	Internet subscription				20.00
		my bank account	n		20.00
	Action Memo	my ISP bank account	n	20.00	Withdrawal

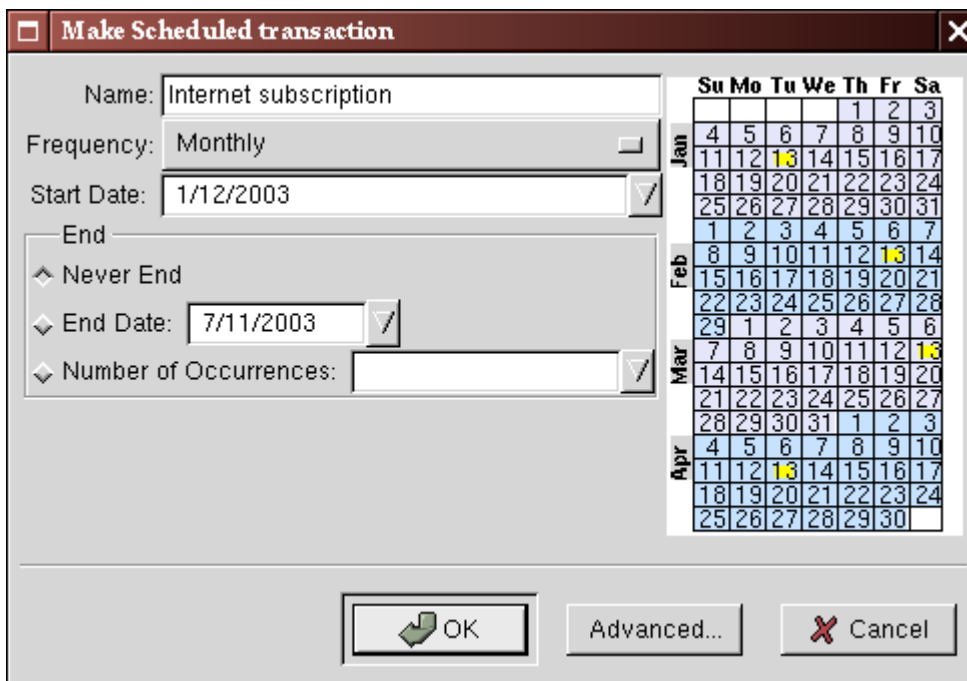
Step two creating scheduled transaction from the ledger

A window like this will appear:



Step three creating scheduled transaction from the ledger

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



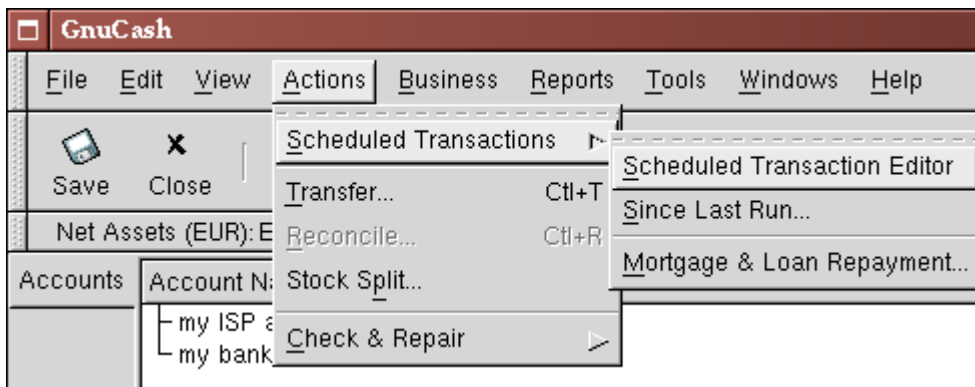
Filling in data to a scheduled transaction

Click the OK button, and the transaction will be scheduled. GnuCash now has memorized this scheduled transaction and on the first 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

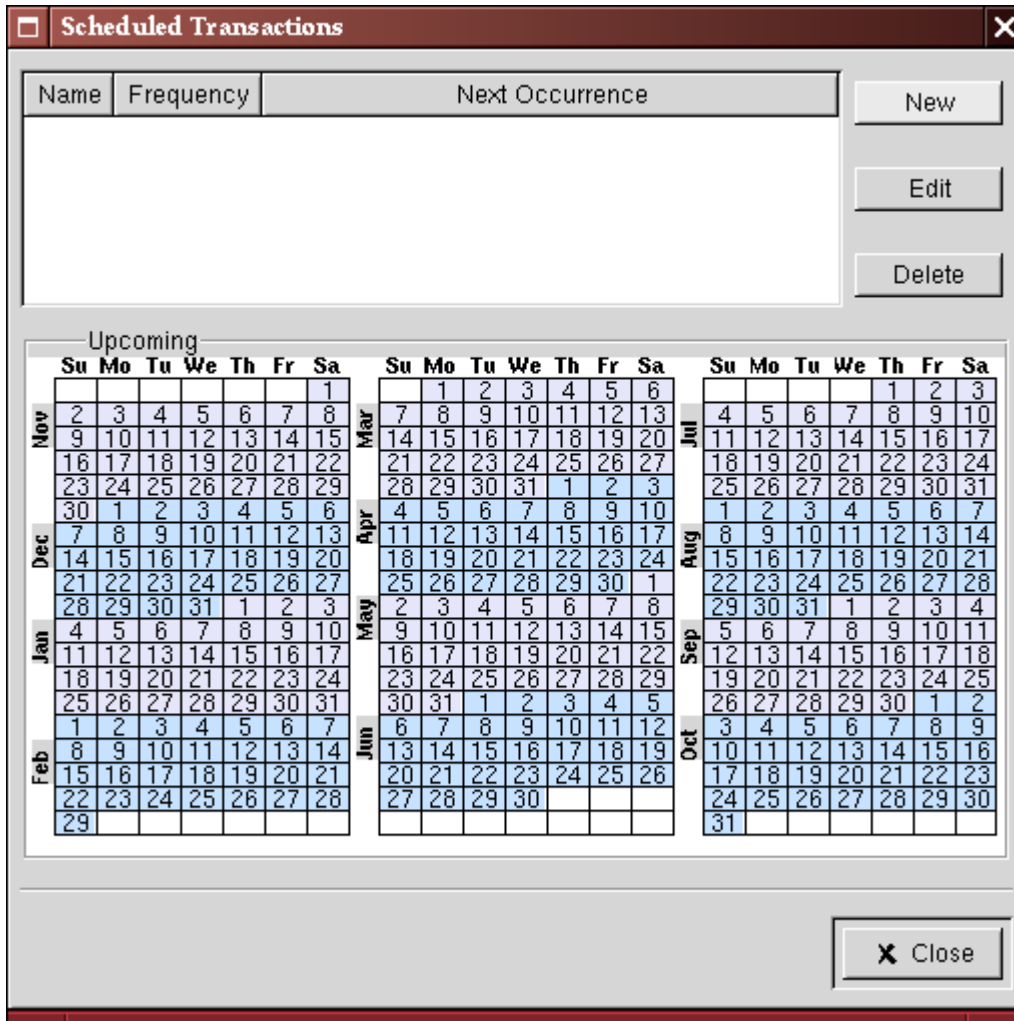
Another way of entering a scheduled transaction is from the Scheduled Transaction Editor, it may be faster if we have several scheduled transactions to create at once.

From the main accounts windows, go in the Actions menu, select Scheduled Transactions, a submenu will open in which you'll click Scheduled transaction Editor.



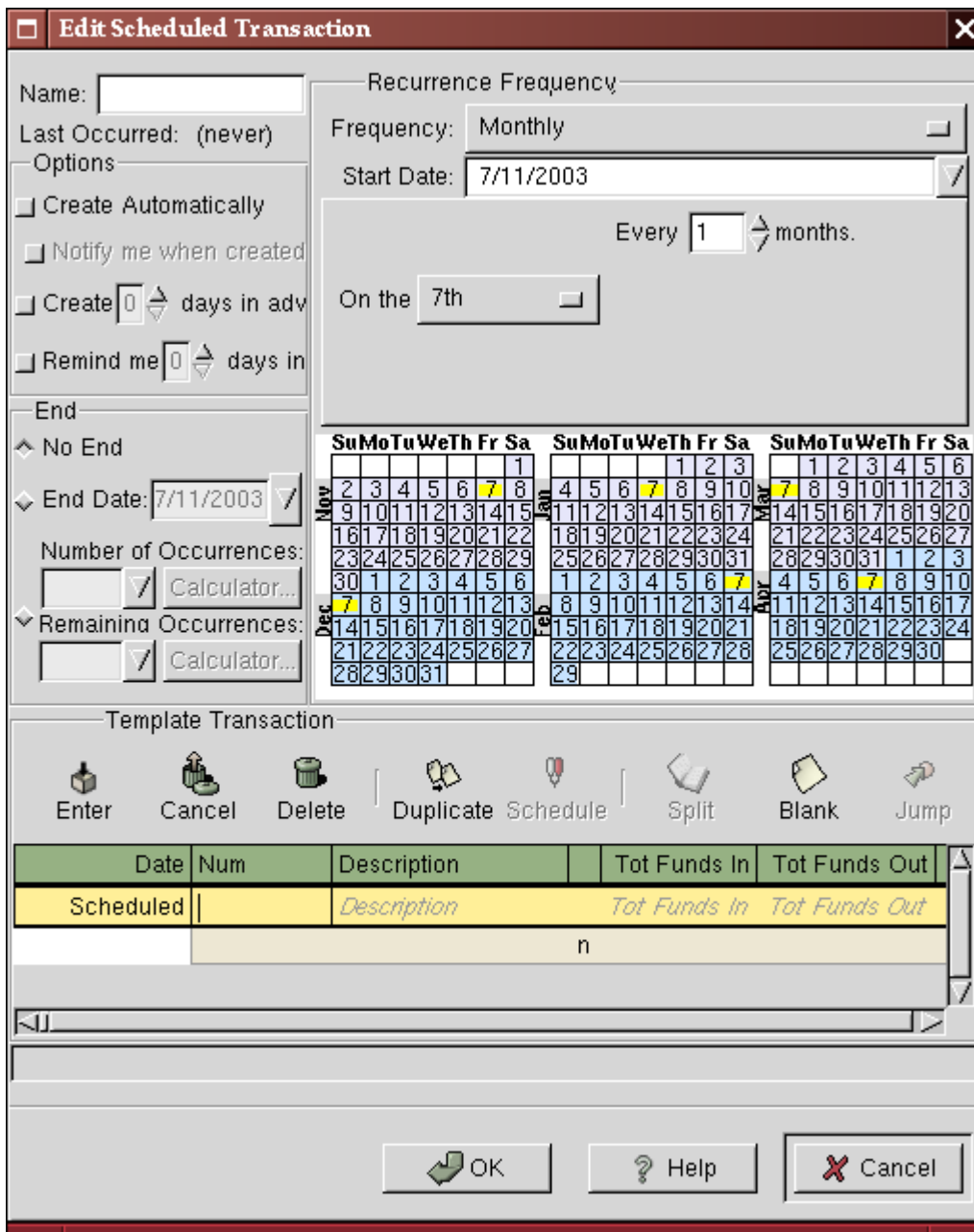
Step one creating scheduled transaction from the editor

A window like this will pop up :



Step two creating scheduled transaction from the editor

This window contains a list, now empty, of all the scheduled transactions. Let's create a new one by clicking on the "New" button, another window will pop up :



Step three creating scheduled transaction from the editor

This window may seem complicated but it's fairly easy to understand. First, let's enter a name for this new scheduled transaction, this name will only identify it in the Scheduled Transaction Editor, it will never appear in the ledger.

Then we'll enter the transaction's parameters. We know that the subscription is taken on the first day of each month, so the start date will be Nov 1, 2003 (assuming November is the next month), the frequency will be "Monthly", it will be taken every month on the 1st.

In the "Options" tab you have three options :

- "Create automatically" will insert this transaction in the ledger without asking you before (see below)
- "Create n days in advance" is explicit
- "Remind me n days in advance" is explicit and used for example when you have to pay something by check, GnuCash can remind you one week before so you send your check before the deadline.

The End tab 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.

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.

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

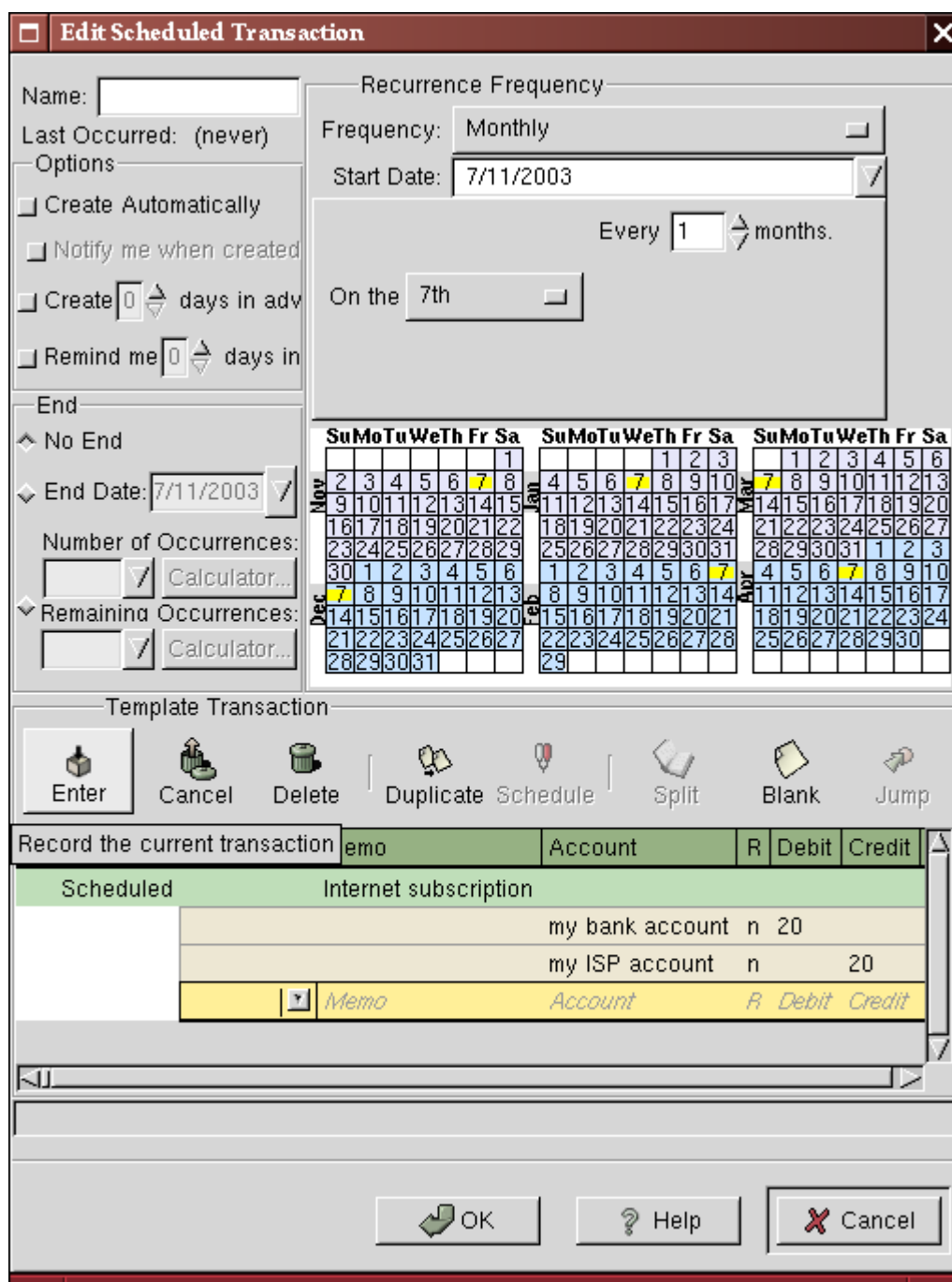
Note

Warning: don't use the ENTER key like in the ledger, as it will activate the default button of the whole window : Cancel. Instead, use TAB to go to the next field, and when you have finished entering your transaction, click on the "Enter" button to validate it.

Note

Note: In the version I have (1.8.5) the "Debit Formula" and "Credit Formula" are swapped, so I typed 20 as debit for my bank account, but you'll see below that it will be recorded as funds in. Look in your version if the bug is still here, and remember that the debit column is the right one, and the credit the left one.

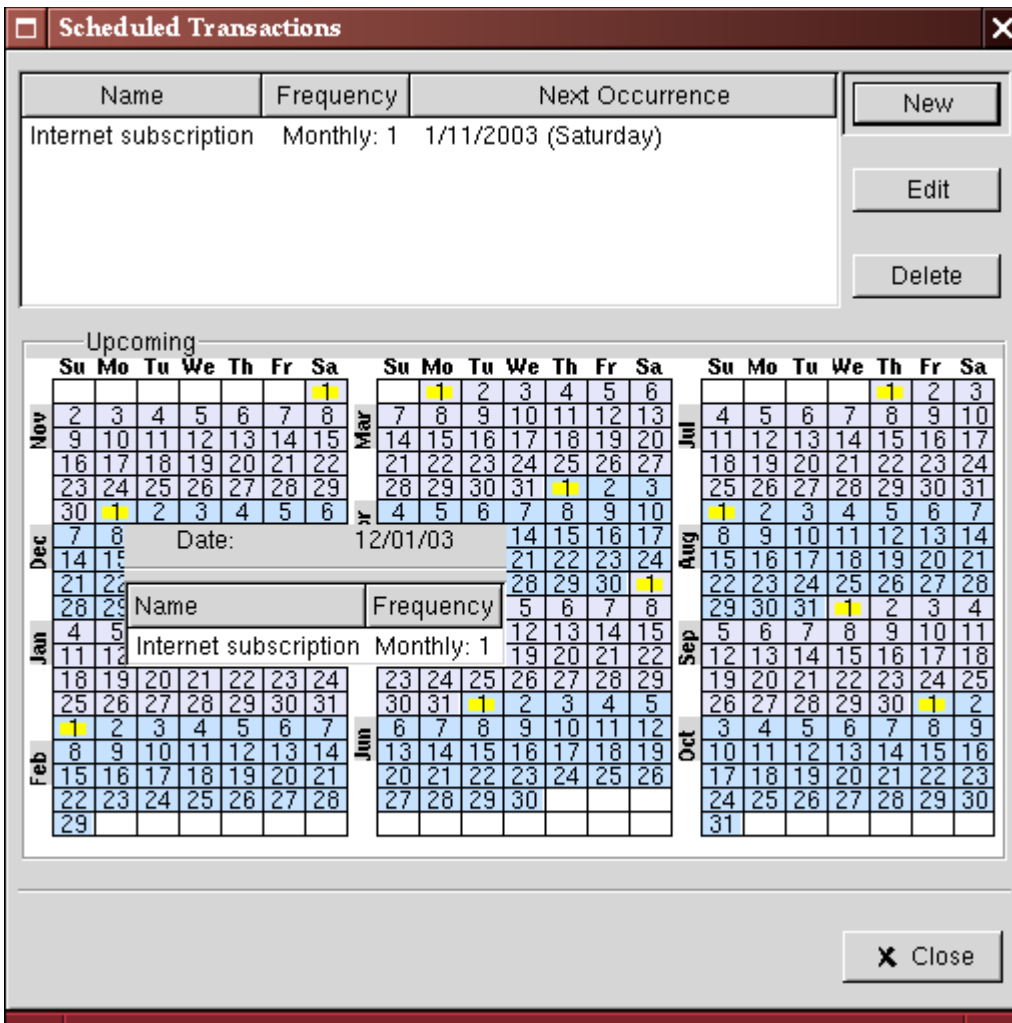
Now, you should have a window like this:



Validate a scheduled transaction from the editor

You have to validate the transaction you just entered, like in the ledger, before clicking the OK button. Many users reported problems creating scheduled transactions because they were not validating the template, and the scheduled transaction was not created.

Now click OK, it takes you to the previous window, now with one item in the list :

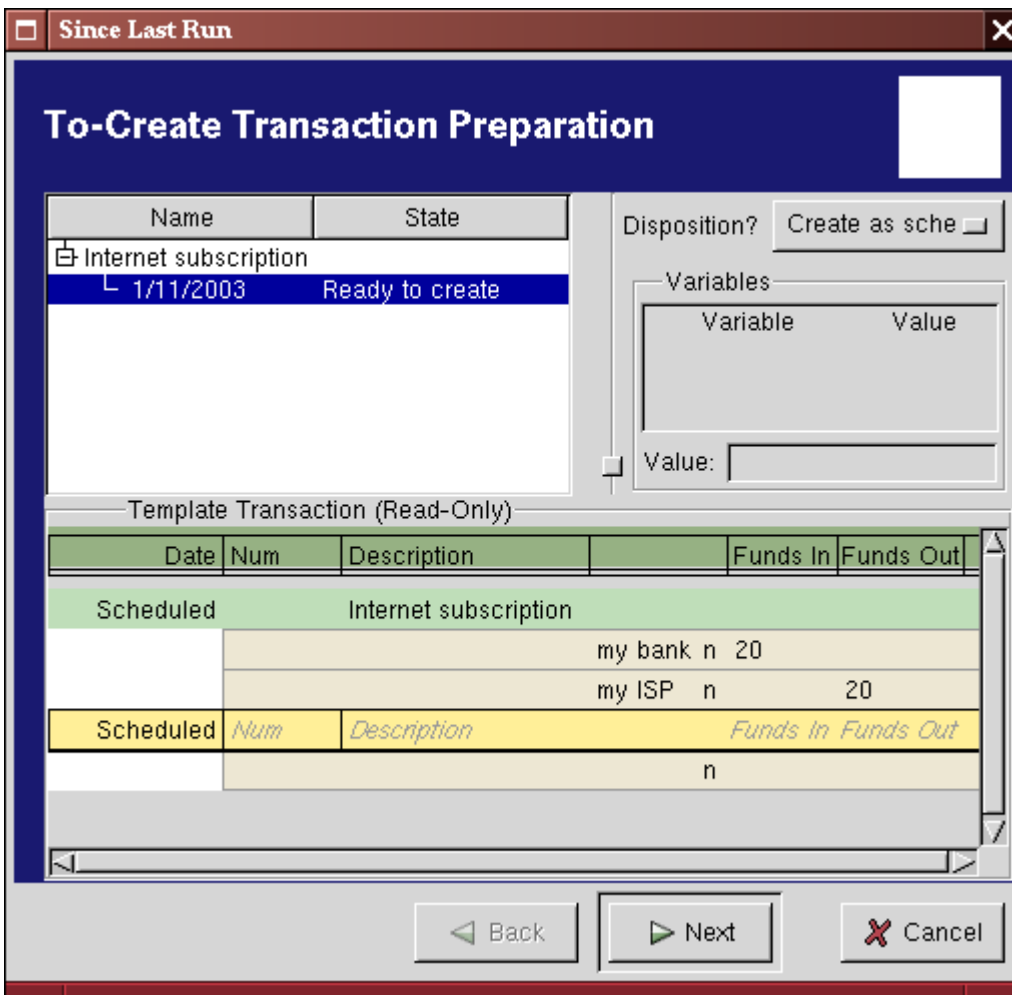


Validate a scheduled transaction from the editor

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.

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

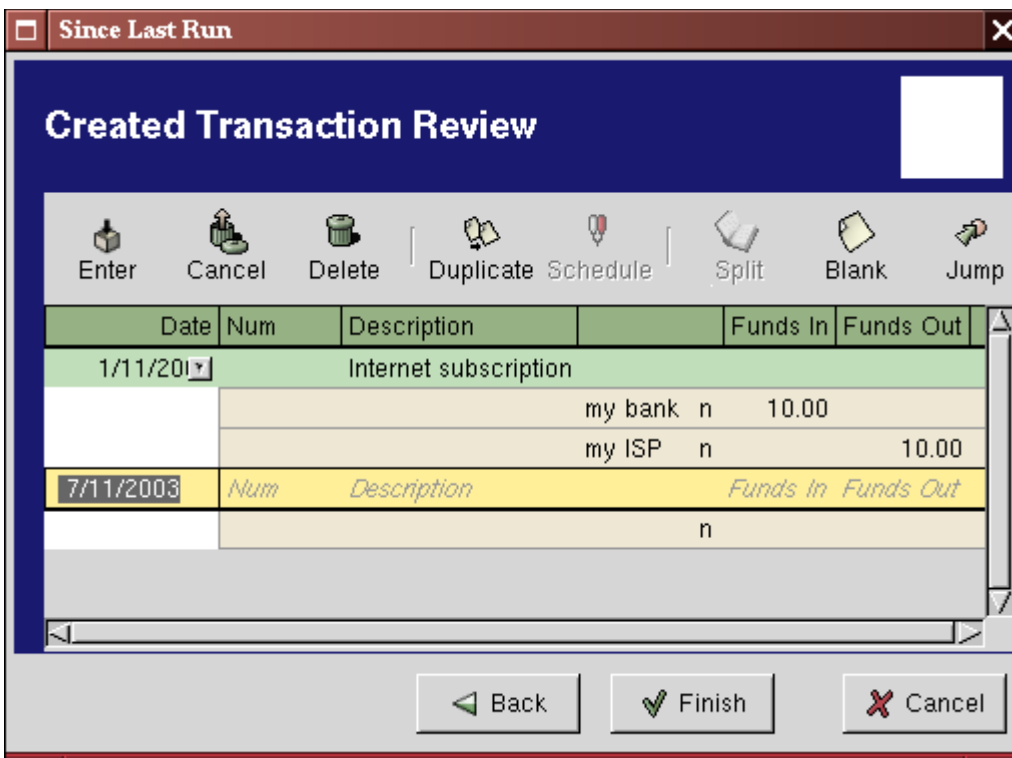
From now on, on each first day of month (or the next time you launch GnuCash after this day) a windows will pop up to remind you to create this transaction :



Scheduled transaction popup reminder

You can click on the list entry to see the detailed transaction in the lower part, then just click on Next to advance.

When you click Next, a window summarizing all transactions to be created will appear :



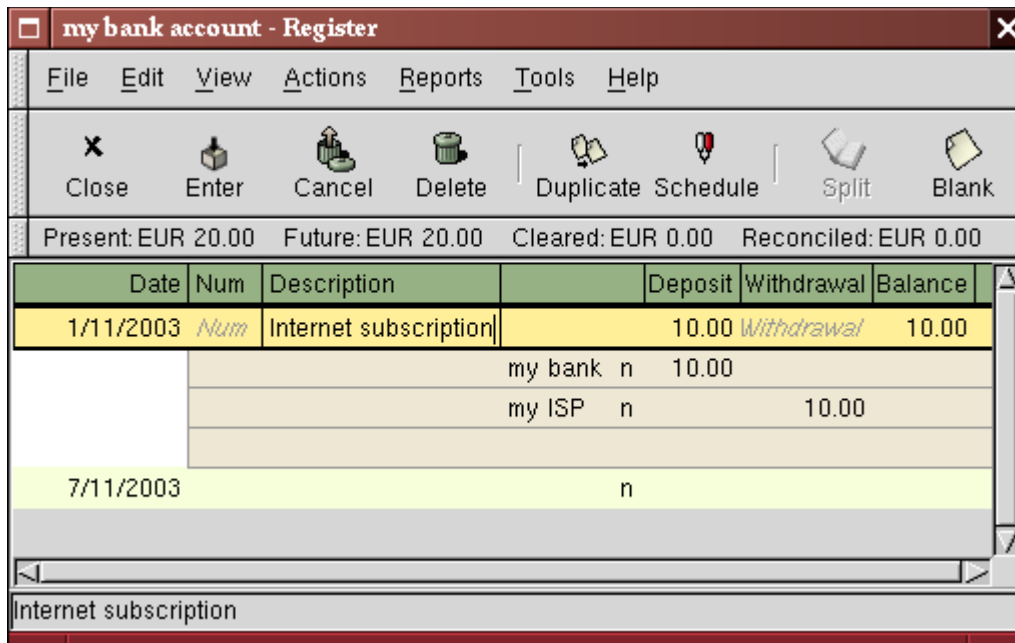
Scheduled transaction summary

In this window you can adjust the transactions that will be created. For example, this month my ISP made

a special offer and my subscription only costed 10EUR, so I just modified the amount from 20 to 10.

When you have finished with this window, just click "Finish". The window will disappear and the transactions will be created.

Then if you take a look in your bank account, you'll see the transaction has been created :

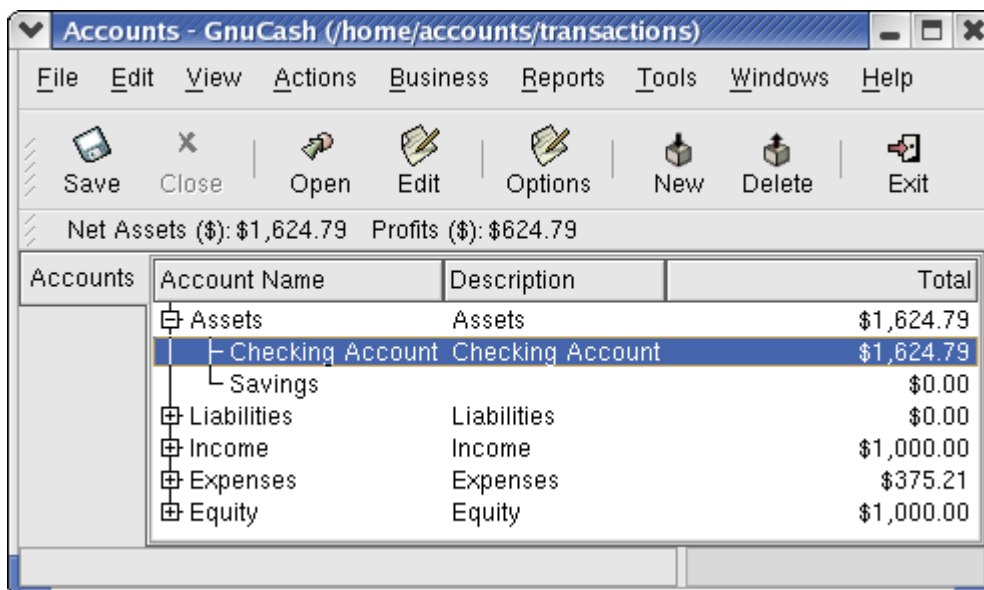


Scheduled transaction created transaction

Putting It All Together

In the previous sections of this chapter the concepts and mechanics of working with transactions in GnuCash has been discussed. This section will expand upon the chart of accounts initially built in the previous chapter.

Start by opening your chart of accounts file. Assuming you have been following the progress of this example chart of accounts, you should have 3 transactions already saved within the checking account. The main window should look something like this:



This image shows the starting point for this section.

Opening Balances

As shown earlier with the Asset:Checking account, the starting balances in an account are typically

assigned to a special account called Equity:Starting Balances. 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 -> Style 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 Asset:Savings account register window, enter a basic 2 account transaction to set your starting balance to \$1000, transferred from Equity:Starting Balances. 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 Asset:Checking account register window, confirm that you have already set the starting balance to \$1000, transferred from Equity:Starting Balances. This should already have been done previously from the example earlier in this chapter, but if not, do so now.
4. From the Liabilities:Visa account register window, enter a basic 2 account transaction to set your starting balance to \$500, transferred from Equity:Starting Balances. This is done by entering the \$500 as a "decrease", 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.

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. 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: June 30, 2003). Enter a description (eg: Light Company) and the check number (eg: 101). 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: June 30, 2003). Enter a description (eg: Phone Company Name) and the check number (eg: 102). 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: June 30, 2003). Enter a description (eg: June Rent) and the check number (eg: 101). The Transfer account should be Assets:Checking.

Duplicate this transaction using the Duplicate button in the tool bar. 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: July 31, 2003) and set the check number (eg: 102), 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. FIXME, add scheduled transaction stuff.

4. To transfer money from you 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.

Chapter 5. Checkbook

Table of Contents

[Setting up Accounts](#)

[Entering Deposits](#)

[Entering Withdrawals](#)

[ATM/Cash Withdrawals](#)

[Reconciling Your Accounts](#)

[Putting It All Together](#)

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 Chapter 4, 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.

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 Section 3.3. For instructions on importing your accounts from another program, refer to Section 2.4.

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 druid, 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.

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 in the Account Information Dialog 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 Chapter 7.

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.

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.

Reconciling Your Accounts

GnuCash makes reconciliation of your bank account with your monthly bank statement much easier. Section 4.6 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 next chapter, 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.

Putting It All Together

In Chapter 4, you entered some transactions in your checking account. In this chapter, we will add more transactions and then reconcile them. So, let's get started by opening the gcashdata file you saved in the last chapter. Your chart of accounts should look like this: FIXME:insert chartacct2.png

The screenshot shows the 'Accounts' window in GnuCash. The title bar reads 'Accounts - GnuCash (/home/accounts/gcashdata)'. 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, Options, New, Delete, and Exit. A status bar shows 'Net Assets (\$): \$0.00' and 'Profits (\$): \$0.00'. The main area is a table with columns: Accounts, Account Name, Type, Description, and Total.

Accounts	Account Name	Type	Description	Total
	Assets	Asset	Assets	\$0.00
	Liabilities	Liability	Liabilities	\$0.00
	Income	Income	Income	\$0.00
	Expenses	Expense	Expenses	\$0.00
	Equity	Equity	Equity	\$0.00

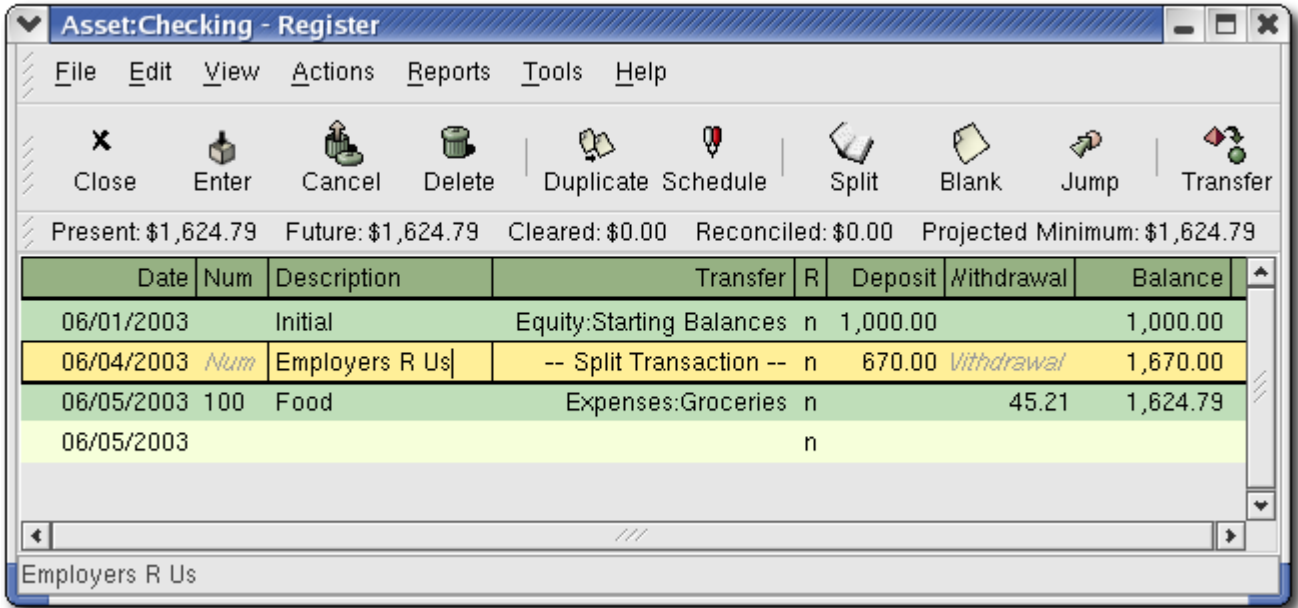
This image shows the Checking Account Register.

1. Open the Savings account and enter an opening balance transaction for \$1000. To do this, you will enter the transfer account as Opening Balances and the \$1000 in the Deposit column:
FIXME:insert savings.png

The screenshot shows the 'Asset:Checking - Register' window. The title bar reads 'Asset:Checking - Register'. The menu bar includes File, Edit, View, Actions, Reports, Tools, and Help. Below the menu is a toolbar with icons for Close, Enter, Cancel, Delete, Duplicate, Schedule, Split, Blank, Jump, and Transfer. A status bar shows 'Present: \$1,624.79', 'Future: \$1,624.79', 'Cleared: \$0.00', 'Reconciled: \$0.00', and 'Projected Minimum: \$1,624.79'. The main area is a table with columns: Date, Num, Description, Transfer, R, Deposit, Withdrawal, and Balance.

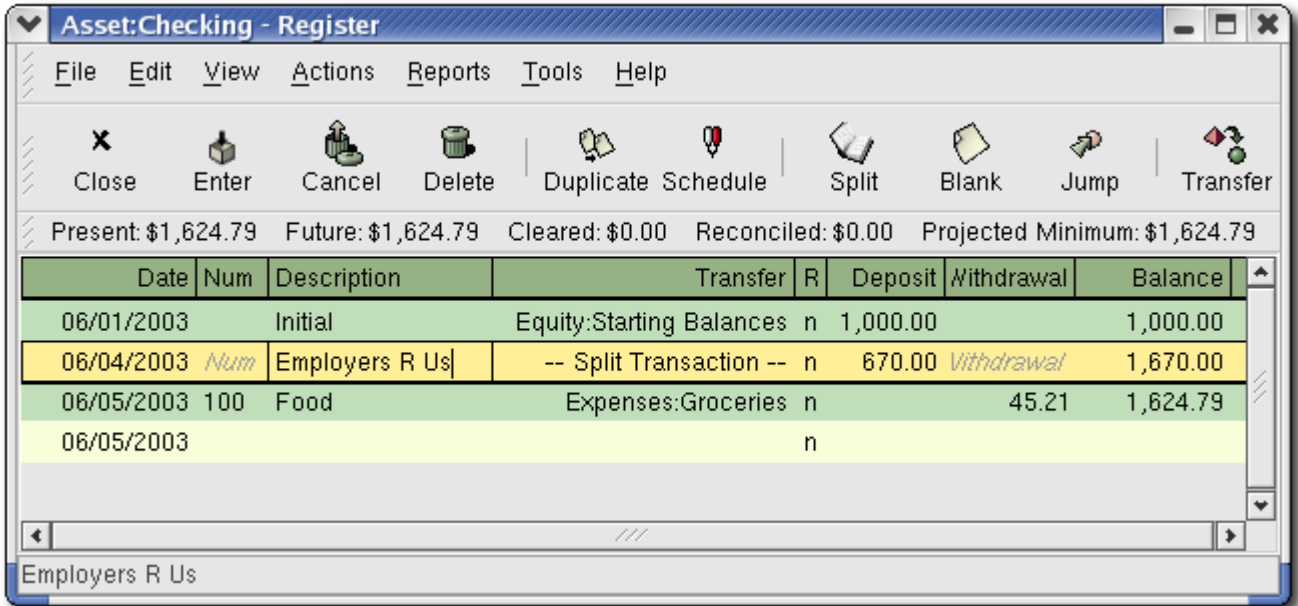
Date	Num	Description	Transfer	R	Deposit	Withdrawal	Balance
06/01/2003		Initial	Equity:Starting Balances	n	1,000.00		1,000.00
06/04/2003	Num	Employers R Us	-- Split Transaction --	n	670.00	Withdrawal	1,670.00
06/05/2003	100	Food	Expenses:Groceries	n		45.21	1,624.79
06/05/2003				n			

This image shows the Checking Account Register.

- 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 into Checking. 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
06/01/2003		Initial	Equity:Starting Balances	n	1,000.00		1,000.00
06/04/2003	Num	Employers R Us	-- Split Transaction --	n	670.00	Withdrawal	1,670.00
06/05/2003	100	Food	Expenses:Groceries	n		45.21	1,624.79
06/05/2003				n			

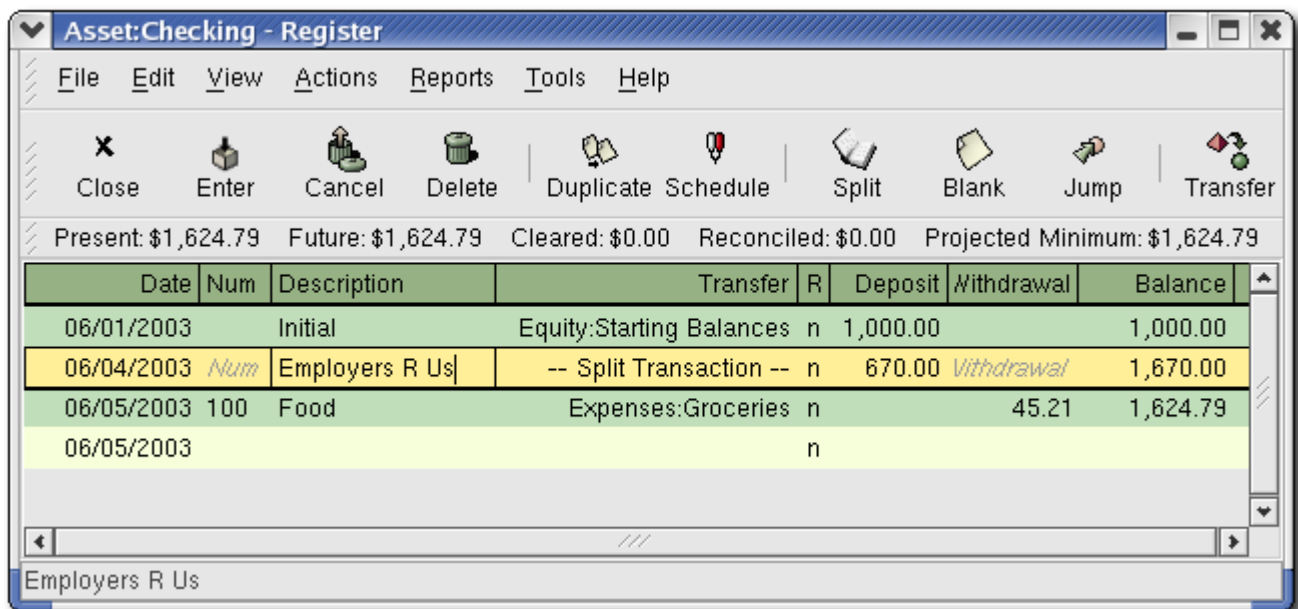
This image shows the Checking Account Register.

- Now let's write some checks on this account. First, write a check to HomeTown Grocery for \$75. Your transfer account is Groceries, since all of this money is going to buy groceries. Next, write a check to ABC Hardware for \$100, 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
06/01/2003		Initial	Equity:Starting Balances	n	1,000.00		1,000.00
06/04/2003	Num	Employers R Us	-- Split Transaction --	n	670.00	Withdrawal	1,670.00
06/05/2003	100	Food	Expenses:Groceries	n		45.21	1,624.79
06/05/2003				n			

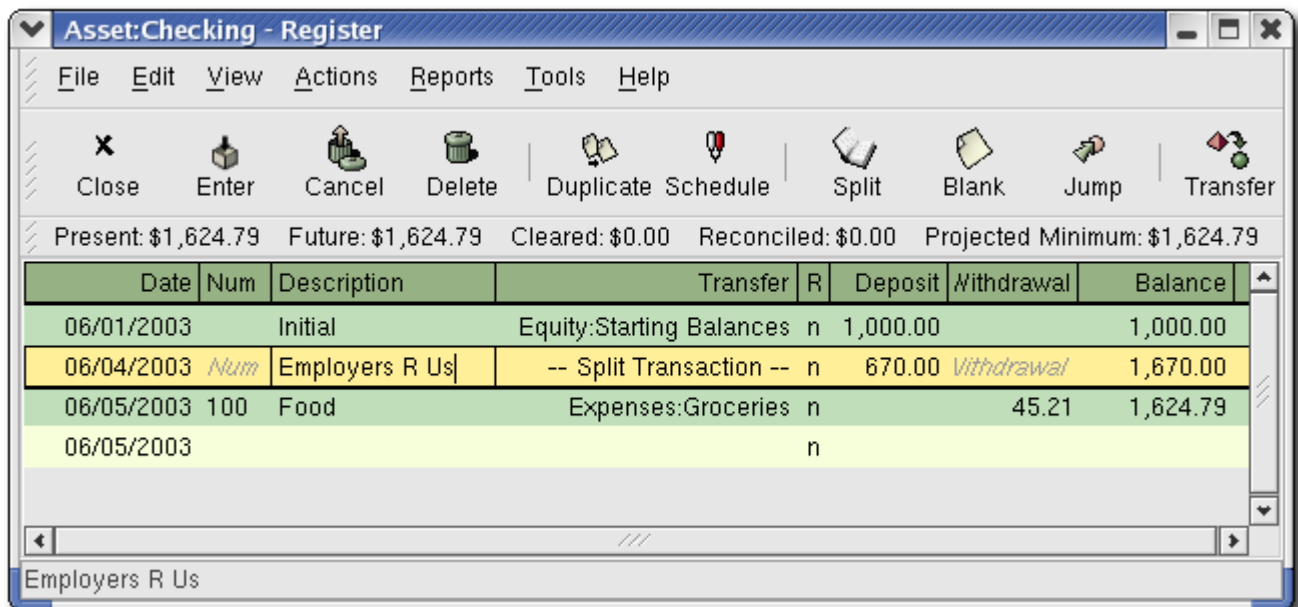
This image shows the Checking Account Register.

- 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 cash. From your Checking account register, enter an ATM type withdrawal to transfer \$100 from Checking to Cash. 



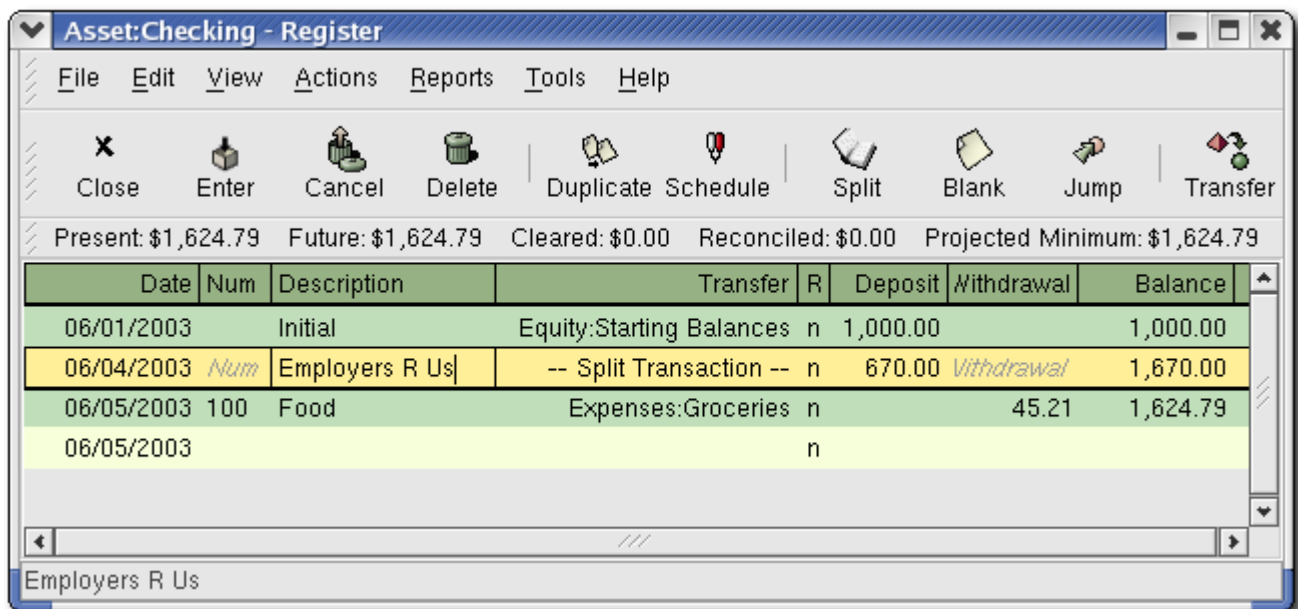
This image shows the Checking Account Register.

- Now we are ready to reconcile this Checking account, using this sample bank statement: FIXME:insert a picture of a sample bank statement here



This image shows the Checking Account Register.

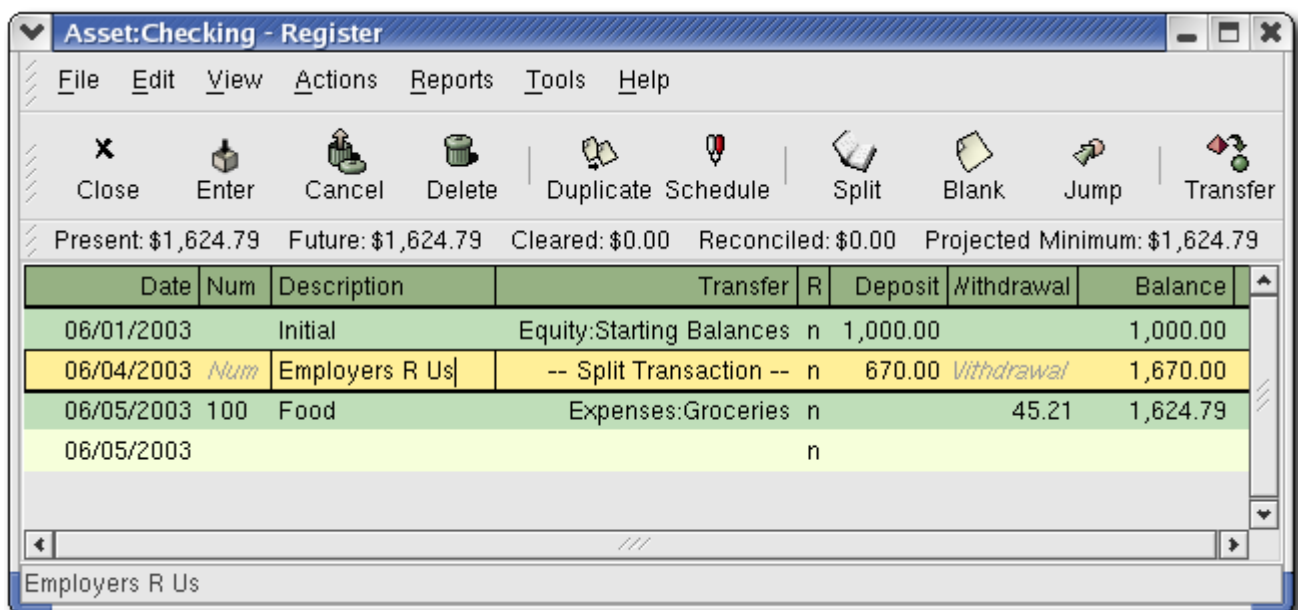
Select Account|Reconcile... from the menu, and fill in the Closing balance as \$2620. Click OK to begin reconciling the account. Check off the entries as they appear on the sample statement. When you have checked off all your entries, the reconcile window should look like this: FIXME:insert reconexamp.png



This image shows the Checking Account Register.

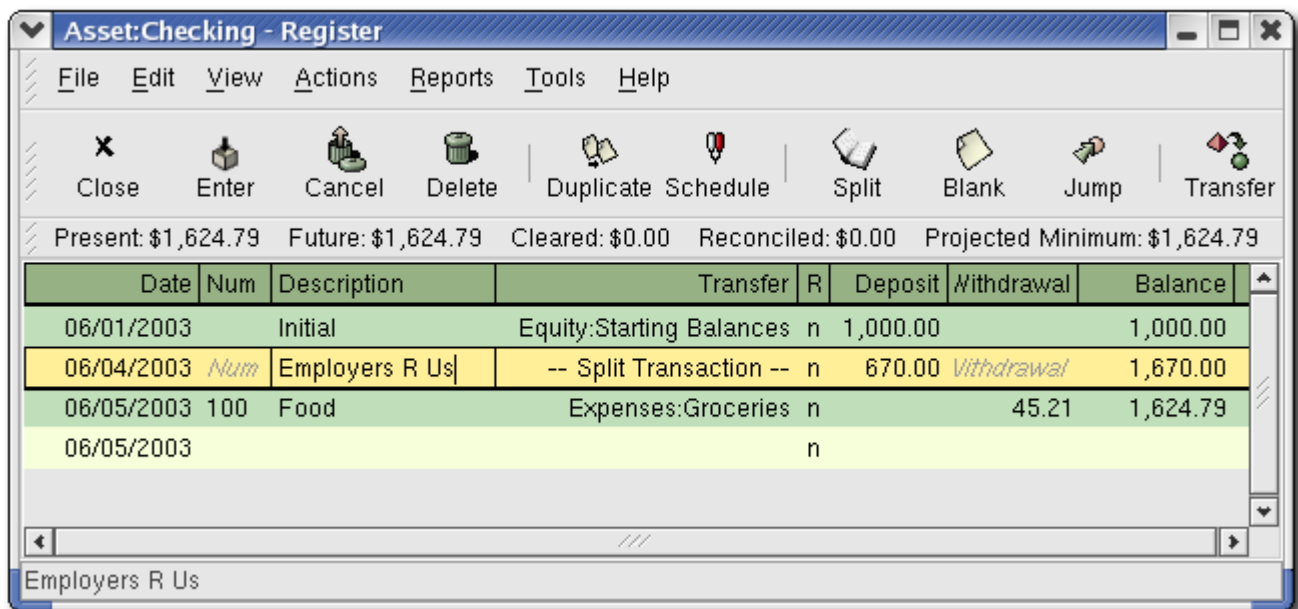
Notice that your reconciled balance differs from your ending balance by \$5.00. If you look at the sample bank statement, you will see there is a \$5.00 service charge that has not been added to your Checking account.

- 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:



This image shows the Checking Account Register.

- 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 tool bar to complete the reconciliation. The Reconcile R column in your Checking register should now show y for each transaction you just reconciled.
- Go back to the main window and save your file. Your chart of accounts is steadily growing, and it should now look like this:



This image shows the Checking Account Register.

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.

Chapter 6. Credit Cards

Table of Contents

[Concepts](#)

[Setting Up Accounts](#)

[Simple Setup](#)

[Complete Setup](#)

[Entering Charges](#)

[Entering Payments](#)

[Putting It All Together](#)

[Purchases](#)

[Refund](#)

[Interest Charge](#)

[Reconciliation](#)

[Payment](#)

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

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.

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 setup 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 setup 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 setup and maintain.

Complete Setup

If you want to track your expenses more completely, you should setup 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.

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 swipe that credit card through the machine at a store, you are transferring money from your credit card company to that merchant. This transaction increases the amount of money you owe the credit card company. If you keep track of these types of transactions, you can reconcile your credit card account with your monthly statement, and you should quickly spot any transactions that look suspicious. So how should you account for charges to your credit card?

One way is to manually enter all your receipts, either throughout the month or all at once. This can be time-consuming, but it is probably the best way to make sure that all of your charges are authorized. If you consistently enter all your receipts, you can quickly catch any errors or unauthorized transactions during your monthly reconciliation.

Another way is to import a credit card statement directly into your account. If your credit card company supports a QIF file format for its statements, you have the option of directly importing the statement. This type of import is covered in the chapter The Basics, and it is a convenient way to track your purchases without having to manually enter all your receipts.

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

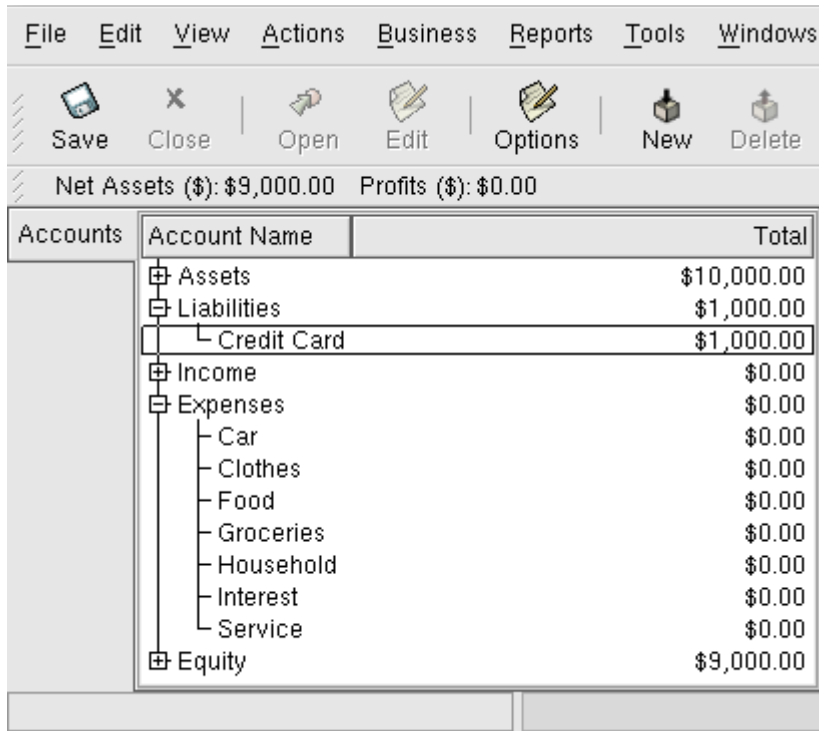
A common mistake is to enter a refund as income. It is not income, but rather a "negative expense". That is why you must transfer money from the expense account to your credit

card when you receive a refund.

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, make a partial payment to the credit card, get charged interest on the unpaid balance, and finally make a complete payoff of the credit card.

Start with an account layout similar to that described earlier in this chapter, shown here as it would appear in GnuCash. Notice that we are starting with \$10000 in the bank and a balance of \$1000 on the credit card.



The screenshot shows the GnuCash Accounts window. The menu bar includes File, Edit, View, Actions, Business, Reports, Tools, and Windows. The toolbar contains icons for Save, Close, Open, Edit, Options, New, and Delete. Below the toolbar, the status bar displays "Net Assets (\$): \$9,000.00" and "Profits (\$): \$0.00". The main area is a table with columns for "Accounts", "Account Name", and "Total".

Accounts	Account Name	Total
	Assets	\$10,000.00
	Liabilities	\$1,000.00
	└─ Credit Card	\$1,000.00
	Income	\$0.00
	Expenses	\$0.00
	├─ Car	\$0.00
	├─ Clothes	\$0.00
	├─ Food	\$0.00
	├─ Groceries	\$0.00
	├─ Household	\$0.00
	├─ Interest	\$0.00
	└─ Service	\$0.00
	Equity	\$9,000.00

Starting account structure for tracking a credit card in the putting it all together example.

Purchases

Let's make some purchases. 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. The register window for the credit card liability should look like this:

Date	Num	Description	Transfer	R	Payment	Charge	Balance
01/01/2003		Initial Value	Equity:Opening Balances	n		1,000.00	1,000.00
02/01/2003		Greasy Spoon Cafe	Expenses:Food	n		25.00	1,025.00
02/03/2003		Faux Pas Fashions	Expenses:Clothes	n		100.00	1,125.00
02/15/2003		Premium Gas	Expenses:Car	n		25.00	1,150.00
02/20/2003		Groceries R Us	-- Split Transaction --	n		125.00	1,275.00
02/25/2003		CheapMart	Expenses:Household	n		60.00	1,335.00

Present: \$1,335.00 Future: \$1,335.00 Cleared: \$0.00 Reconciled: \$0.00 Projected Minimum: \$1,335.00

Enter a description of the transaction

Initial credit card purchases.

Refund

Now suppose that on February 27th you return the clothes you bought on February 3rd 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	Num	Description	Transfer	R	Payment	Charge	Balance
01/01/2003		Initial Value	Equity:Opening Balances	n		1,000.00	1,000.00
02/01/2003		Greasy Spoon Cafe	Expenses:Food	n		25.00	1,025.00
02/03/2003		Faux Pas Fashions	Expenses:Clothes	n		100.00	1,125.00
02/15/2003		Premium Gas	Expenses:Car	n		25.00	1,150.00
02/20/2003		Groceries R Us	-- Split Transaction --	n		125.00	1,275.00
02/25/2003		CheapMart	Expenses:Household	n		60.00	1,335.00
02/27/2003		Faux Pas Fashions	Expenses:Clothes	n	100.00		1,235.00

Present: \$1,235.00 Future: \$1,235.00 Cleared: \$0.00 Reconciled: \$0.00 Projected Minimum: \$1,235.00

Friday 29 August 2003

Returning clothes to Faux Pas Fashions, refund to credit card.

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 February because of the balance you carried from the previous month. This gets entered into the credit card account as an expense.

Date	Num	Description	Transfer	R	Payment	Charge	Balance
01/01/2003		Initial Value	Equity:Opening Balances	n		1,000.00	1,000.00
02/01/2003		Greasy Spoon Cafe	Expenses:Food	n		25.00	1,025.00
02/03/2003		Faux Pas Fashions	Expenses:Clothes	n		100.00	1,125.00
02/15/2003		Premium Gas	Expenses:Car	n		25.00	1,150.00
02/20/2003		Groceries R Us	-- Split Transaction --	n		125.00	1,275.00
02/25/2003		CheapMart	Expenses:Household	n		60.00	1,335.00
02/27/2003		Faux Pas Fashions	Expenses:Clothes	n	100.00		1,235.00
02/28/2003		interest	Expenses:Interest	n		20.00	1,255.00

Interest charge.

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 "Transactions" chapter, but we will step through the process here as well. For this example, let's assume that the credit card statement is dated February 1st, with a final balance of \$955. Enter these values in to the initial Reconcile window as shown here.

Reconcile Information

Statement Date:

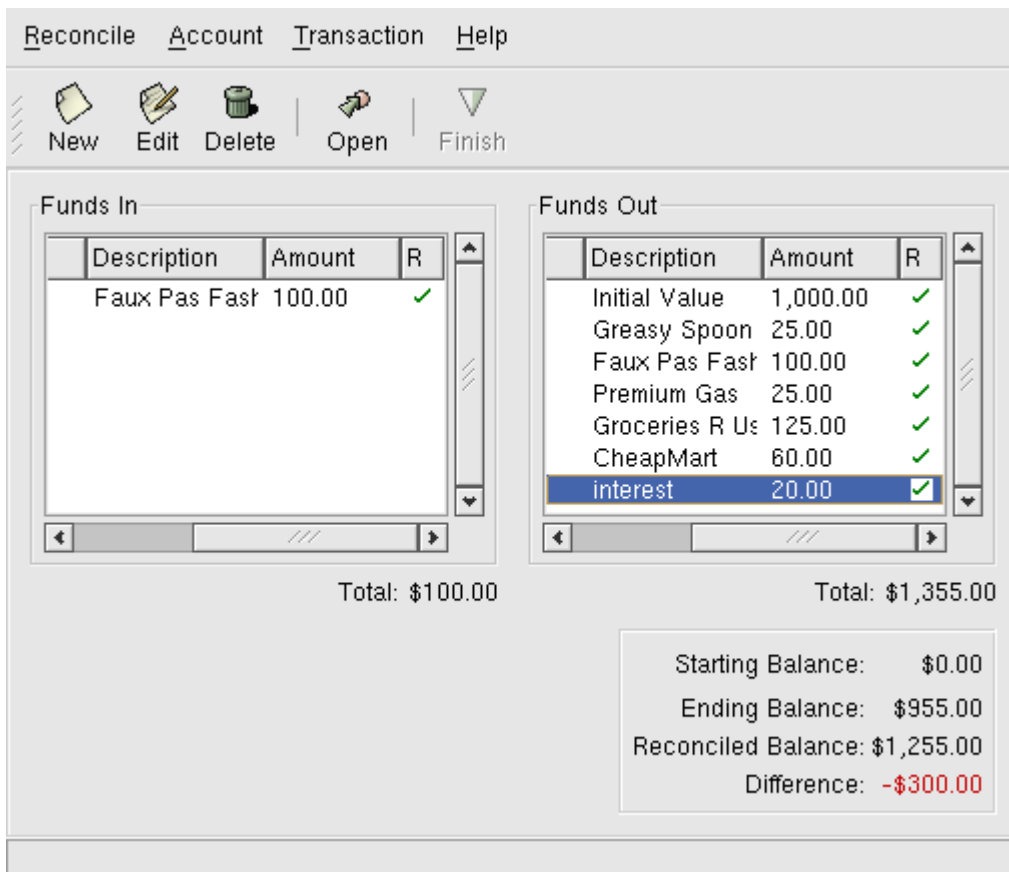
Starting Balance: \$0.00

Ending Balance:

Include Subaccounts

Initial account reconciliation window.

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.



Main account reconciliation window, demonstrating a discrepancy of \$300.

Some investigation uncovers that you forgot to record a payment you made to the credit card company for \$300 received on February 5th, you must enter this payment transaction from your bank account to the credit card. Now the credit card statement and your GnuCash account are reconciled, with a balance of \$955.

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 \$955 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 \$655. Your credit card account register should now appear like this:

Date	Description	Transfer	R	Payment	Charge	Balance
11/01/2003	Initial Value	Opening Balances	y		1,000.00	1,000.00
12/01/2003	Greasy Spoon Ca	Expenses:Food	y		25.00	1,025.00
12/03/2003	Faux Pas Fashior	Expenses:Clothes	y		100.00	1,125.00
12/05/2003	payment	s:Savings Account	y	300.00		825.00
12/15/2003	Premium Gas	Expenses:Car	y		25.00	850.00
12/20/2003	Groceries R Us	Split Transaction --	y		125.00	975.00
12/25/2003	CheapMart	Expenses:Household	y		60.00	1,035.00
12/27/2003	Faux Pas Fashior	Expenses:Clothes	y	100.00		935.00
12/28/2003	interest	Expenses:Interest	y		20.00	955.00
13/02/2003	payment	s:Savings Account	n	300.00		655.00

Account register after account reconciliation and payment.

Your main GnuCash main window should now appear like this:

Accounts	Account Name	Total
	Assets	\$9,400.00
	Current Assets	\$9,400.00
	Liabilities	\$655.00
	Credit Card	\$655.00
	Income	\$0.00
	Expenses	\$255.00
	Car	\$25.00
	Clothes	\$0.00
	Food	\$25.00
	Groceries	\$85.00
	Household	\$100.00
	Interest	\$20.00
	Service	\$0.00

GnuCash main window after account reconciliation and payment.

Chapter 7. Loans

Table of Contents

[Basic Concepts](#)

[Terminology](#)

[Setting Up Accounts](#)

[House Mortgage \(How-To\)](#)

[Calculations](#)

[Example: Monthly Payments](#)

[Example: Length of Loan](#)

[Advanced: Calculation Details](#)

[Missing Loan Documentation](#)

This chapter explains how to manage your loans with GnuCash.

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.

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
 - Bank Account
 - Asset Purchased
- Liability
 - Loan
- Expenses
 - Loan Interest
 - Loan Admin 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 File -> New File and select the loan types in which you are interested.

House Mortgage (How-To)

A house mortgage can be setup using the account structure present in the previous section.

As an example, assume you have \$60k in you 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 GnuCash interface with a menu bar (File, Edit, View, Actions, Business, Reports, Tools, Windows) and a toolbar (Save, Close, Open, Edit, Options, New, Delete). Below the toolbar, it displays 'Net Assets (\$): \$60,000.00' and 'Profits (\$): \$0.00'. The main window shows a tree view of accounts with the following structure:

Accounts	Account Name	Total
	Assets	\$60,000.00
	Bank	\$60,000.00
	House	\$0.00
	Liabilities	\$0.00
	Home Mortgage	\$0.00
	Expenses	\$0.00
	Mortgage Adm Fees	\$0.00
	Mortgage Interest	\$0.00
	Equity	\$60,000.00
	Opening Balances	\$60,000.00

Accounts Before Receiving Loan

The purchase of the house is recorded with a split transaction in the Asset: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.

Table 7.1. Buying a House Split Transaction

Account	Increase	Decrease
Assets:House	\$150,000	

Assets:Bank		\$50,000
Liabilities:Mortgage		\$103,000
Expenses:Mortgage Adm Fees	\$3000	

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

Date	Description		Tot Increase	Tot Decrease	Balance
	Notes				
10/02/2003	Buy House		150,000.00	<i>Tot Decrease</i>	150,000.00
	Notes				
	Total Value of House	Assets:House	n	150,000.00	
	Closing Costs	Expenses:Mortgage Adm Fees	n	3,000.00	
	Down Payment	Assets:Bank	n		50,000.00
	Loan & Closing Costs	Liabilities:Home Mortgage	n		103,000.00

Mortgage Split Transaction

Which will give account totals like this:

The screenshot shows the GnuCash interface with a menu bar (File, Edit, View, Actions, Business, Reports, Tools, Windows) and a toolbar (Save, Close, Open, Edit, Options, New, Delete). Below the toolbar, it displays 'Net Assets (\$) : \$57,000.00' and 'Profits (\$) : -\$3,000.00'. The main window shows a tree view of accounts with the following totals:

Accounts	Account Name	Total
[-]	Assets	\$160,000.00
	[-] Bank	\$10,000.00
	[-] House	\$150,000.00
[-]	Liabilities	\$103,000.00
	[-] Home Mortgage	\$103,000.00
[-]	Expenses	\$3,000.00
	[-] Mortgage Adm Fees	\$3,000.00
	[-] Mortgage Interest	\$0.00
[-]	Equity	\$60,000.00
	[-] Opening Balances	\$60,000.00

Mortgage Account

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 Financial Calculator. To access the calculator, go to Tools -> Financial Calculator.

The GnuCash Financial Calculator.

The Financial 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 -473.30.

Answer: You must make monthly payments of 473.30.

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 set 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 button next to the Payment Periods area, you should calculate 48.

Answer: You will pay off the loan in 4 years (48 months).

Advanced: Calculation Details

In order to discuss the mathematical formulas used by the Financial 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 == quaterly
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 Interst: $i = CF * [(1 + ieff)^{(PF/CF)} - 1]$

Continuous Interest: $i = ln[(1 + ieff)^{PF}]$

Note

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

Example: Monthly Payments

Let's recalculate the examples shown above, this time using the mathematical formulas rather than the Financial 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: $n = (30*12) = 360$, $PV = 100000$, $PMT = \text{unknown}$, $FV = 0$, $i = 4$, $CF = PV = 12$, $X = 0$ (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 $ieff = (1 + 4/12)^{(12/12)} - 1$, thus $ieff = 1/3 = 0.3333333$.

Now we can calculate A and B. $A = (1+i)^n - 1 = (1 + 1/3)^{360} - 1 = 9.504848 \times 10^{44}$. $B = (1+iX)/i = (1 + 0.33333*0)/0.33333 = 3$.

With A and B, we can calculate PMT. $PMT = -[FV + PV*(A + 1)]/[A*B] = -[0 + 100000*(9.504848 \times 10^{44} + 1)] / [9.504848 \times 10^{44} * 3] = -9.504848 \times 10^{49} / 2.851454 \times 10^{45} = -33333.3$.

Missing Loan Documentation

-Automobile Loan (How-To)

-Entering a Payment Schedule

- Monthly Payments (How-To)
- Final Payment (How-To)

-Selling a House or Automobile (How-To)

-Reconciling with the Loan Statements (How-To)

-Putting It All Together

Chapter 8. Investments

Table of Contents

[Basic Concepts](#)

[Terminology](#)

[Types of Investments](#)

[Setting Up Accounts](#)

[Predefined Accounts](#)

[Custom Accounts Example](#)

[Interest Bearing Accounts](#)

[Account Setup](#)

[Example](#)

[Setup Stock Portfolio](#)

[Setup Preexisting Stocks](#)

[Example Stock Account](#)

[Buying New Investments](#)

[Setting Stock Price](#)

[Initial Price Editor Setup](#)

[Setting Stock Price Manually](#)

[Setting Stock Price Automatically](#)

[Displaying Stock Value](#)

[Making Stock Value Reports](#)

[Dividends](#)

[Selling Stocks](#)

[Example](#)

[To-do](#)

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.

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* is 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* is the fee you pay to a broker to buy or sell securities.
- *Common stock* is 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* is 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* are cash payments a company makes to shareholders. The amount of this payment is usually determined as some amount of the profits the company. Note that not all common stocks give dividends.
- *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* is 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* is 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* is 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* is the total income plus capital gains or losses of an investment. See yield.
- *Risk* is 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* 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 share will be adjusted so there is no net change in the value, so in this example the price per share will be halved.
- *Valuation* is the process of determining the sell value of an investment.
- *Yield* is 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 saved. 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.

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 low risk investment. Examples are certificates of deposit or some government bonds.

- *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 high risk, as you have no guarantee what the future price of a stock could be.

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*

An asset 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 would not receive this tax break on an expensive painting.

Fixed asset investments are discussed in the chapter on "Depreciation and Capital Gains".

Typically, there is not much to do in terms of accounting for fixed asset investments except record the buying and selling transactions.

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.

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.

Predefined Accounts

To use the predefined investment account hierarchy, you must create a new GnuCash file. This will run the New Account Hierarchy Setup druid. After choosing the default currency to use, you will be asked to Choose accounts to create. At this point, choose the "Investment Accounts" option (along with any others you are interested in). This will create an account hierarchy as shown below.

- Assets

- Investments
 - Brokerage Accounts
 - Mutual Funds
 - Market Index
 - Stock
 - Bond
- Income
 - Interest Income
 - Bond Interest
 - Dividend Income
- Expenses
 - Commissions

You will probably at least want to add a *Bank* account to the Assets and probably an *Equity:Starting Values* account, as we have done in previous chapters.

Custom Accounts Example

You can also manually setup your own investment account hierarchy. 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

There really is no unique 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.

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.

- Asset
 - Bank
 - CD
 - Savings
- Expenses
 - Bank
 - Charges
- Income
 - Bank
 - Interest

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 1 year 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:

Accounts	Account Name	Description	Total
	Asset		\$10,000.00
	Bank		\$10,000.00
	CD		\$5,000.00
	Savings		\$5,000.00
	Income		\$0.00
	Bank		\$0.00
	Interest		\$0.00
	Expenses		\$0.00
	Bank		\$0.00
	Charges		\$0.00
	Equity		\$10,000.00
	Starting Balances		\$10,000.00

This image the creation of the interest investment account

Now, during the course of the next year, 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 12 months, the register window for the CD should look like this:

Date	Num	Description	Transfer	R	Increase	Decrease	Balance
01/01/2000		Purchase CD	Asset:Bank:Savings	n	5,000.00		5,000.00
01/31/2000			Income:Bank:Interest	n	8.33		5,008.33
02/29/2000			Income:Bank:Interest	n	8.35		5,016.68
03/31/2000			Income:Bank:Interest	c	8.36		5,025.04
04/30/2000			Income:Bank:Interest	n	8.38		5,033.42
05/31/2000			Income:Bank:Interest	n	8.39		5,041.81
06/30/2000			Income:Bank:Interest	n	8.40		5,050.21
/16/2003		<i>Num Description</i>	<i>Transfer</i>	<i>n</i>	<i>Increase</i>	<i>Decrease</i>	<i>Balance</i>

This image the creation of the interest investment account

And this is the main GnuCash window:

Accounts	Account Name	Description	Total
	Asset		\$10,063.24
	Bank		\$10,063.24
	CD		\$5,050.21
	Savings		\$5,013.03
	Income		\$75.24
	Bank		\$75.24
	Interest		\$75.24
	Expenses		\$12.00
	Bank		\$12.00
	Charges		\$12.00
	Equity		\$10,000.00
	Starting Balances		\$10,000.00

This image shows the creation of the interest investment account

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.

Setup Stock Portfolio

Now that you have built an account hierarchy in the previous section, this section will show you how to populate the accounts with your stock portfolio. During this initial setup of your portfolio, you may have stocks purchased from before you started using GnuCash. For these stocks, follow the instructions in the *Setup Your Pre-existing Stocks* section below. If you have just purchased your stocks, then use the *Buying New Stocks* section.

Setup Preexisting Stocks

This section will show you how to add stocks to GnuCash assuming that you purchased these stocks from before you started using GnuCash. In this section, we will assume you are using the basic predefined 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 + (plus) signs next to the account names until the tree is fully opened. 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. So, for example, you could name your accounts *AMZN* and *IBM* for your Amazon and IBM stocks respectively. Below is a schematic model of the layout (only showing the Assets sub-accounts).

- Assets
 - Investments
 - Brokerage Accounts
 - Mutual Funds

- Market Index
- Stock
 - AMZN
 - IBM
- Bond

Example Stock Account

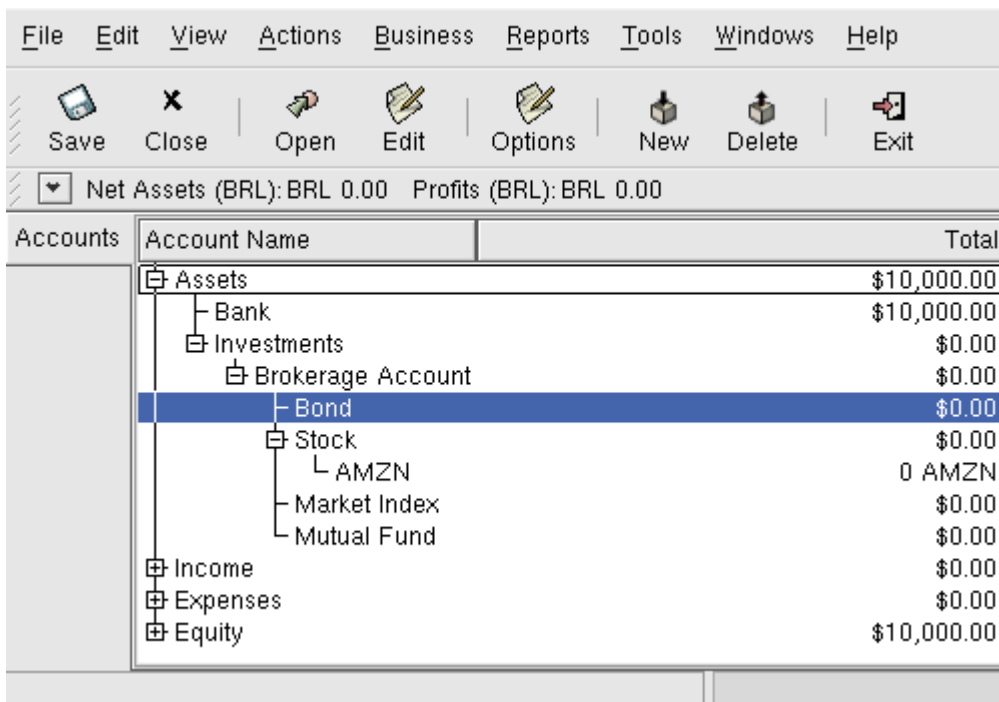
As an example, let's assume that you currently own 100 shares of Amazon stock. First, create the stock account AMZN by selecting the Stock account and click on the menu File -> New Account. The New Account dialog will appear, follow the step below to setup your new stock account.

1. *Name the Account* - Usually, use the stock ticker abbreviation, IE: "AMZN"
2. *Create the New Commodity* - To use a new stock, you must create the stock as a new commodity
 - a. *Select Commodity* - Click on the "Select" button next to the Commodity line. We must change the commodity from the default (your default currency) to this specific stock. This will bring up the "Select currency/security" dialog.
 - b. *Commodity Type* - Change the type from CURRENCY to (in this example) NASDAQ.
 - c. *Create the Commodity* - Click on the New... button and enter the appropriate information for this stock. The Full name: is "Amazon.com Inc", the Symbol/abbreviation: is "AMZN", the Type: should already be "NASDAQ", the CUSIP or other code is where you can enter some other coding number or text (leave it blank in this example), and finally the Fraction traded should be adjusted to the smallest fraction of this commodity which can be traded, usually 1/100 or 1/10000. Below is what this window should look like when finished:

New Security Window

- d. *Save Commodity* - Click on the OK button to save this new commodity
3. *Select the Commodity* - you should now see the newly create commodity available in the pull down menu for Currency/security. Select it (it is probably already selected) and click on OK.
4. *Finished* - You should now have been automatically returned to the New Account dialog, with the Commodity: line set to "AMZN (Amazon.com Inc)". Click on OK to save this new stock account.

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



This image shows the creation of the first stock account

Open the account register window for this AMZN stock account (double click on it). Here you see the Portfolio view. This gives you an overview of your stock portfolio including the number of shares bought and sold. Obviously, we have not bought or sold any stocks yet, so the Portfolio view should not contain any transactions.

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 2001), a description (eg: Initial Purchase), transfer from *Equity: Opening Balances*, Shares (eg: 100), and Price (eg: \$20). You do not need to fill in the Buy column, as it will be calculated for you. Your AMZN Portfolio view should now appear like this:

Date	Num	Description	Transfer	R	Shares	Price	Buy	Sell	Balance
01/01/2003		Initial	Equity	n	100.00	20.00	2,000.00		100.00
01/01/2003				n		1.00			100.00

This image shows the creation of the first stock account

Notice that the Balance 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 Investments

The only difference between setting up a new investment 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: Starting Balances* account, you use your *Asset: Bank* account.

Setting Stock Price

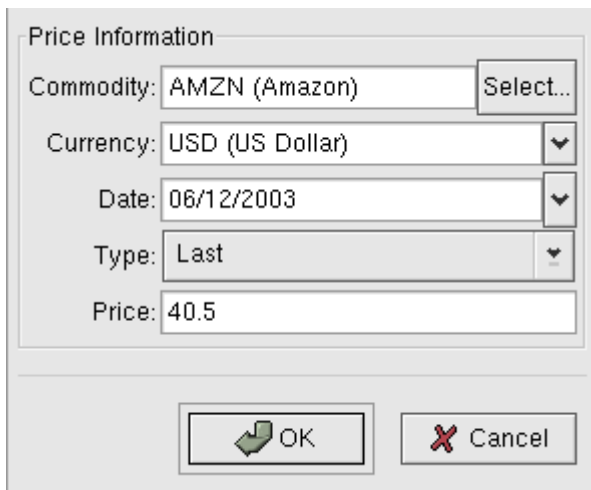
The value of a commodity, such as a stock, must be explicitly set. The stock accounts track the quantity of stocks you own, but the value of the stock is stored in the *Price Editor*. The values set in the Price

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. 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 5 fields in the New Commodity window:

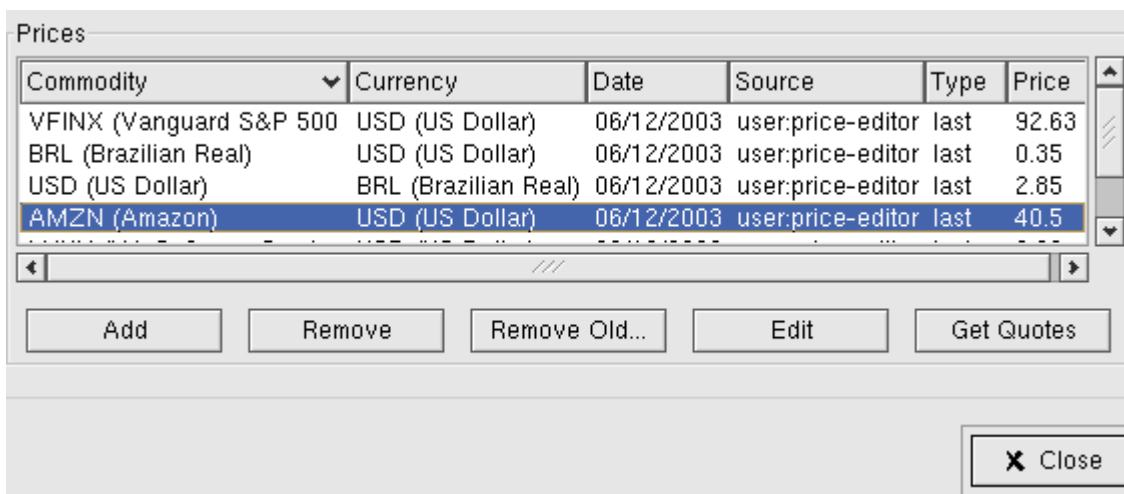
- Commodity - the name of the commodity, must be chosen from the Select... list
- Currency - the currency the in which the Price is expressed.
- Date - date that the price is valid
- Type - one of: Bid (the market buying price), Ask (the market selling price), Last (the last transaction price), Net Asset 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.



The image shows a dialog box titled "Price Information". It contains five input fields: "Commodity" with a dropdown menu showing "AMZN (Amazon)" and a "Select..." button; "Currency" with a dropdown menu showing "USD (US Dollar)"; "Date" with a dropdown menu showing "06/12/2003"; "Type" with a dropdown menu showing "Last"; and "Price" with a text input field containing "40.5". At the bottom, there are two buttons: "OK" with a green checkmark icon and "Cancel" with a red X icon.

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.



The image shows a dialog box titled "Prices". It contains a table with the following data:

Commodity	Currency	Date	Source	Type	Price
VFINX (Vanguard S&P 500	USD (US Dollar)	06/12/2003	user:price-editor	last	92.63
BRL (Brazilian Real)	USD (US Dollar)	06/12/2003	user:price-editor	last	0.35
USD (US Dollar)	BRL (Brazilian Real)	06/12/2003	user:price-editor	last	2.85
AMZN (Amazon)	USD (US Dollar)	06/12/2003	user:price-editor	last	40.5

Below the table are five buttons: "Add", "Remove", "Remove Old...", "Edit", and "Get Quotes". At the bottom right, there is a "Close" button with a red X icon.

The main price editor window, showing the list of all known commodities.

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.

Setting Stock Price Automatically

If you have many commodities, you will tire of having to update their prices constantly. GnuCash has the ability to automatically determine 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 look 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

To install Finance::Quote begin by first closing any GnuCash applications you have running. Second, open a root shell and run the command “update-finance-quote” (without the quotation marks). 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 update-finance-quote 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 "dump-finance-quote" test program 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 (gnucash-user@gnucash.org) for help or come to the GnuCash IRC channel on irc.gnome.org. You can also leave out this step and manually update your stock prices.

Configuring Finance::Quote

With Finance::Quote installed and functioning correctly, you must configure your GnuCash stock accounts to use this feature to obtain updated price information automatically. For stock accounts that have already been setup, edit the account and select the Get Online Quotes box. You should now be able to modify the popup menus The source for price quotes and The timezone for these quotes. When finished editing the stock account, 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.xac" can be used to fetch the current prices of your stocks. This can be automated by creating a crontab entry. For example, to update your file every Friday evening after the Wall Street markets close (modify the time accordingly for your time zone), you could add the following to your personal crontab:

```
0 16 * * 5 gnucash --add-price-quotes $HOME/gnucash.xac > /dev/null 2>&1
```

Displaying Stock Value

The main account window will 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 account options, by clicking on the tool bar Options button, and select to display the account field “Total in Report Currency”. You will see a new column in the main window entitled Total (Report) that will express the value of all commodities in the report currency.

Accounts	Account Name	Total	Total (Report)
	Assets	\$12,000.00	\$12,000.00
	Bank	\$10,000.00	\$10,000.00
	Investments	\$2,000.00	\$2,000.00
	Brokerage Account	\$2,000.00	\$2,000.00
	Bond	\$0.00	\$0.00
	Stock	\$2,000.00	\$2,000.00
	AMZN	100 AMZN	\$2,000.00
	Market Index	\$0.00	\$0.00
	Mutual Fund	\$0.00	\$0.00
	Income	\$0.00	\$0.00
	Expenses	\$0.00	\$0.00
	Equity	\$10,000.00	\$10,000.00

Viewing the value of a stock commodity in the main window using the Total in Report Currency option.

Making Stock Value Reports

GnuCash offers three alternatives to determine stock prices in reports:

Report name: Amazon Stock Over Time

Stylesheet: Default

From: 20. 1. 2004 Today

To: 30. 1. 2004 Today

Step Size: Day

Report's currency: Weighted Average

Price Source: Nearest in time

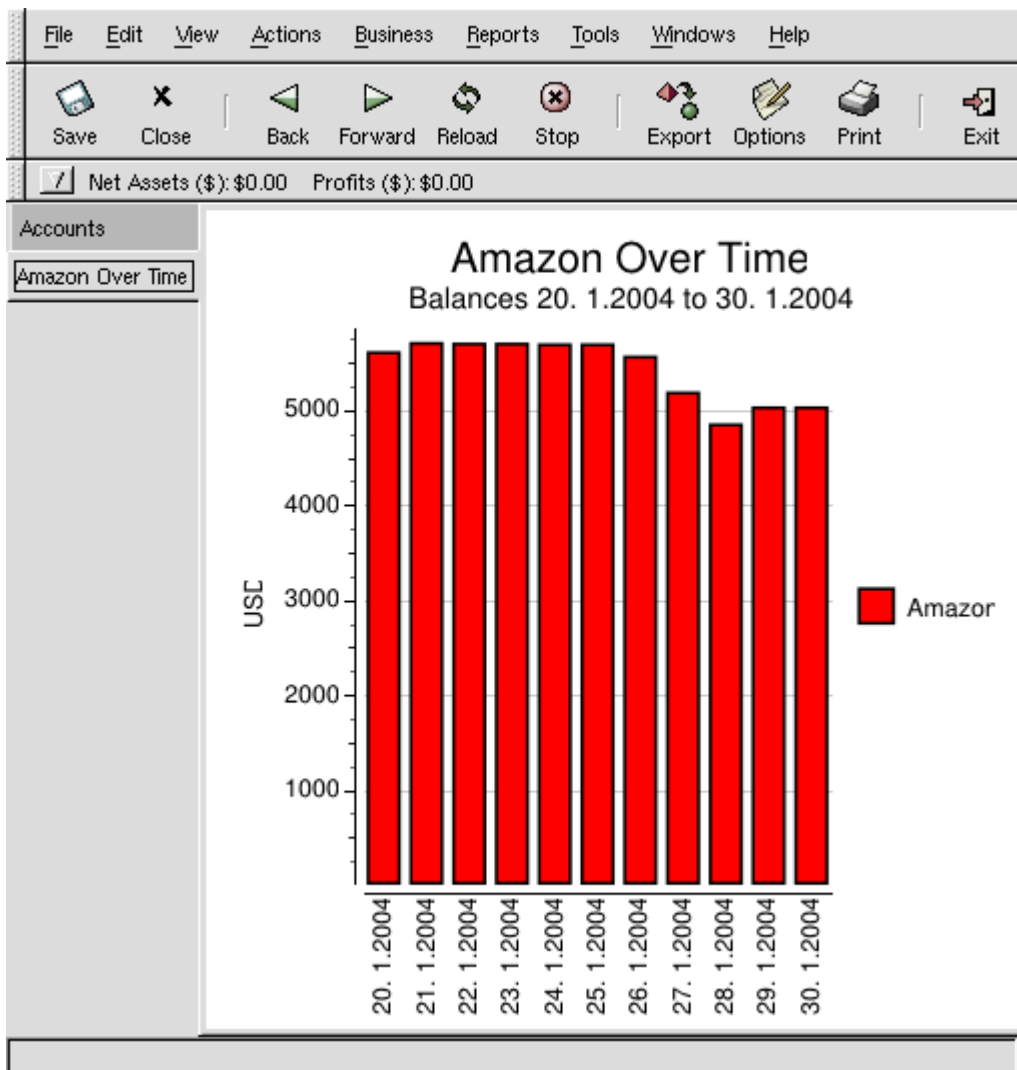
Buttons: OK, Apply, Help, Cancel

Determining the value of a stock commodity in a report by setting the Price Source option.

- Weighted Average - gives you a graph of the weighted price of all *transactions*. It will *not* take into account prices defined in the price editor.
- Most recent - displays the changing value of your stocks based solely on *the most recent* price

available from the price editor. The price figuring in your stock transactions is *not* considered.

- Nearest in time - the graph is exclusively based on the prices available from the price editor. The value of your stock at each step and point in time is calculated based on the nearest available price in the price editor.



Tracking what value your stocks would've had on the stock exchanges.

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.

If you receive the dividend in the form of an automatic reinvestment, the transaction to account for this should be handled within the stock or mutual fund account as income from "Income:Dividend" for the appropriate number of reinvested shares.

If the dividend is presented as cash, you should record the transaction directly in the bank account that received the money, as income from "Income::Dividends" with a note mentioning from which commodity it was derived. There is no simple way to show this transaction from within the stock or mutual fund account itself.

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.

Selling Stocks

Entering an investment you are selling is done in the same way as buying one except the total cost of the transaction is entered in the *Sell* column and the *Shares* column is entered as a negative amount. The proceeds from the sale should be transferred from the stock account to your bank account.

The proper recording of the stock sale *must* 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 Expense:Capital Loss). To balance this income, you will need to enter the stock asset twice in the split. Once to record the actual sale (using the correct number of shares and correct price per share) and once to balance the income profit (setting the number of shares to 0 and price per share to 0).

In the split transaction scheme presented below, the symbol NUM_SHARES is the number of shares you are selling, SELL_PRICE is the price for which you sold the shares, GROSS_TOTAL is the total price for which you sold shares, equal to NUM_SHARES*SELL_PRICE. PROFIT is the amount of money you made on the sale. COMMISSIONS are the brokerage commissions. NET_SALE is the net amount of money received from the sale, equal to GROSS_TOTAL - COMMISSIONS.

Table 8.1. Selling Stock Split Transaction Scheme

<i>Account</i>	<i>Number of Shares</i>	<i>Share Price</i>	<i>Total Buy</i>	<i>Total Sell</i>
Asset:Bank			NET_SALE	
Asset:Stock:SYMBOL	-NUM_SHARES	SELL_PRICE		GROSS_SALE
Asset:Stock:SYMBOL			PROFIT	
Income:Capital Gains				PROFIT
Expenses:Commissions			COMMISSION	

If you will be recording the sale of the stock as a capital gain (or loss), please see the chapter "Depreciation and Capital Gains" for more information on this topic.

Example

As an example, assume you bought 100 shares of a stock for \$20 per share, then later sell them all for \$30 per share. In the split transaction scheme above, PRICEBUY is \$20 (the original buying price), NUMSHARES is 100, TOTALBUY is \$2000 (the original buying price), TOTALPROFIT is \$1000, and finally TOTALSALE is \$3000.

Date	Num	Description	Shares	Tot Buy	Tot Sell
01/01/2001		Initial Purchase	100 20	2,000.00	
07/30/2003		Sell stock for profit	-- Split Transaction --		3,000.00
07/30/2003	Num	Sell stock for profit	-100	Tot Buy	100
		Net Sales Receipt	Assets:Bank	n	2,965.00
		Broker Commissions	Expenses:Commissions	n	35.00
		Stock Profit Balance	Assets:Investments:Brokerage Ac	n	1,000.00
Sell	Sell Stock	Assets:Investments:Brokerage Ac	n -100 30		3,000.00
		Profit income	Income:Capital Gains	n	1,000.00

1

An example of selling stock. You bought 100 shares of AMZN for \$20 per share, and sold for \$30.

To-do

These are the remaining documentation sections that originally appeared in the 1.8.4 documentation which require being written up.

- Recording Stock Splits and Mergers (How-To)
- Recording Employee Stock Plans (Discussion)
- Stock Options (How-To)
- Reconciling with the Brokerage Statement (How-To)

Chapter 9. Capital Gains

Table of Contents

[Basic Concepts](#)

[Estimating Valuation](#)

[Account Setup](#)

[Example](#)

[Unrealized Gains](#)

[Selling](#)

[Caution about Valuation](#)

[Taxation](#)

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.

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.

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.

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:ITEM1:Cost" accounts are balanced by the "Assets:Bank" account, the "Assets:ITEM?:Unrealized Gains" accounts are balanced by the "Income:Unrealized Gains" account.

- Assets
 - ITEM1
 - Cost
 - Unrealized Gain
 - ITEM2
 - Cost
 - Unrealized Gain
- Bank
- Income
 - Unrealized Gains
 - Realized Gains

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 than shown in the previous section, 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 \$100000 in the bank and then transferring that from your bank account to your "Asset:Degas" account (the asset purchase transaction). You should now have a main account window which appears like this:

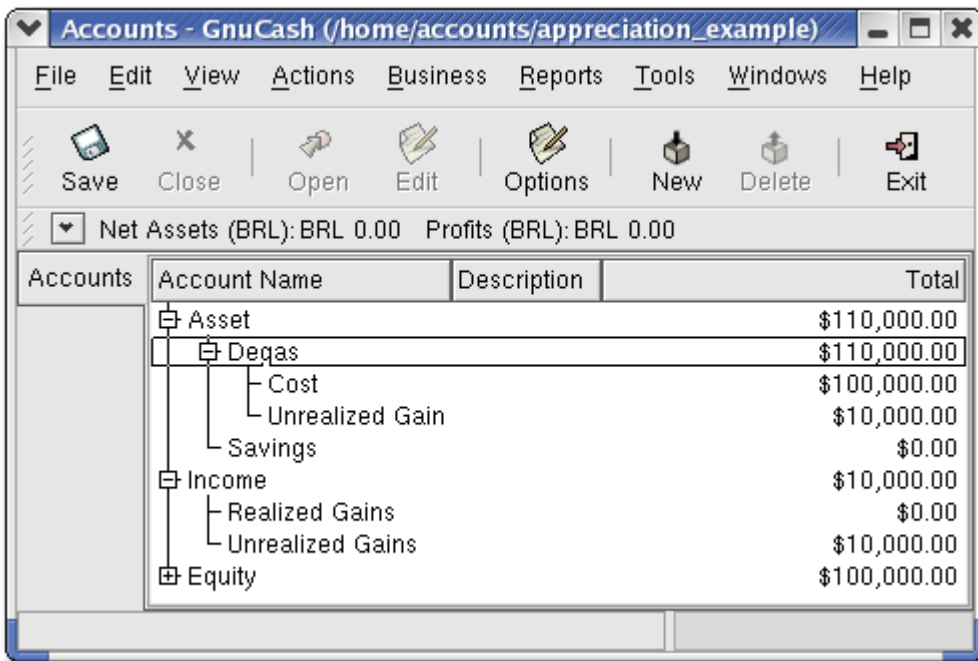
The screenshot shows the GnuCash Accounts window with the following account hierarchy and balances:

Accounts	Account Name	Description	Total
	Asset		\$100,000.00
	Degas		\$100,000.00
	Cost		\$100,000.00
	Unrealized Gain		\$0.00
	Savings		\$0.00
	Income		\$0.00
	Realized Gains		\$0.00
	Unrealized Gains		\$0.00
	Equity		\$100,000.00

The asset appreciation example main window

Unrealized Gains

A month later, you have reason to suspect that the value of your painting has increased by \$10000 (an unrealized gain). In order to record this you transfer \$10000 from your Accrued Gains income account to your asset Unrealized Gains account. Your main window will resemble this:



This image shows the asset appreciation main window

Selling

Let's suppose another month later prices for Degas paintings have gone up some more, in this case about \$20000, you estimate. You duly record the \$20000 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 "Asset:Degas" sub-accounts. Transfer the full "Asset:Degas:Cost" value into "Asset:Bank", and the full "Asset:Degas:Unrealized Gain" into "Asset:Bank".

These transactions should now appear as follows:

Table 9.1. Turning an Accrued Gain into a Realized Gain

<i>Account</i>	<i>Transfer to</i>	<i>Transaction Amount</i>	<i>Account Total</i>
Income:Unrealized Gains	Income:Realized Gains	\$30000	\$0
Asset:Degas:Cost	Asset:Bank	\$100000	\$0
Asset:Degas:Unrealized Gains	Asset:Bank	\$30000	\$0

This leaves the "Asset:Bank" 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 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.

Under construction

3. *Over estimation* of unrealized gain.

You manage to sell your painting for more than you thought in your wildest dreams. The extra value is, again, recorded as a gain, i.e. an income.

Under Construction

Caution about Valuation

As we see in this example, for non-financial assets, it may be difficult to correctly estimate the "true" value of an asset. It is quite easy to count yourself rich based on questionable estimates that do not reflect "money in the bank."

When dealing with appreciation of assets,

- Be careful with your estimation of values. Do not indulge in wishful thinking.
- Never, ever, count on money you do not have in your bank or as cash. Until you have actually sold your asset and got the money, any numbers on paper (or magnetic patterns on your hard disk) are merely that. If you could realistically convince a banker to lend you money, using the assets as collateral, that is a pretty reasonable evidence that the assets have value, as lenders are professionally suspicious of dubious overestimations of value. Be aware: all too many companies that appear "profitable" on paper go out of business as a result of running out of cash, precisely because "valuable assets" were not the same thing as cash.

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...

Chapter 10. Multiple Currencies

Table of Contents

[Basic Concepts](#)

[Account Setup](#)

[GnuCash Currency Support](#)

[Setting International Preferences](#)

[Recording Currency Exchange \(How-To\)](#)

[Recording Purchases in a Foreign Currency \(How-To\)](#)

[Tracking Currency Investments \(How-To\)](#)

[Reconciling Statements in a Foreign Currency \(How-To\)](#)

[Updating Exchange Rates \(How-To\)](#)

[Putting It All Together \(Examples\)](#)

This chapter will show you how to setup your GnuCash accounts to use multiple currencies.

Basic Concepts

GnuCash supports the use of multiple currencies in different accounts. For example, you can have a bank account setup in Euros, and another using British Pounds.

Some of the issues which arise when using multiple currencies is 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?

Account Setup

Your default account currency is set in Edit -> Preferences -> International -> New Account Default Currency. You should set this parameter correctly, as it will save you much time when building your account structure.

When you create a new account, you have the option to define the commodity. For currency accounts, you can specify any one of the hundreds of currencies supported by GnuCash by simply selecting it from the currency commodity list. You will notice that the default currency is always whatever you have defined in the preferences. So, if you mostly work with Euros, but have the occasional Ethiopian Birr account, be sure to set your preferences to Euro.

As an example, let's set up a typical bank account scenario where you mostly work in US Dollars, but do also have a European bank account using the Euro currency. So, setup 2 bank accounts, one using the Euro currency, and another in dollars. One possible account structure for this would be:

- Assets (USD)
- Banks (USD)
 - US Bank (USD)
 - European Bank (EUR)
- Equity (USD)
 - Starting Balances USD (USD)
 - Starting Balances EUR (EUR)

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. To totals shown in the account tree window will always be converted to the currency of each particular account. Notice, we also setup 2 Starting Balances equity accounts, used to initially populate the 2 banks.

Note

You could also setup 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 explain in a later section.

Below you see the result of this example, in which we start with USD 10000 and EUR 10000 in the two bank accounts. Notice that the total of the parent accounts only show the value of the currency of subaccounts with matching currencies. In the future you can setup the exchange rates between the currencies, and the parent accounts will calculate the converted value of all subaccounts. See the later section on how to do this.

The screenshot shows the GnuCash interface with a menu bar (File, Edit, View, Actions, Business, Reports, Tools, Windows, Help) and a toolbar (Save, Close, Open, Edit, Options, New, Delete, Exit). Below the toolbar, it displays 'Net Assets (\$): \$10,000.00' and 'Profits (\$): \$0.00'. The main area is a table with columns 'Accounts', 'Account Name', 'Total', and 'Total (Report)'. The table content is as follows:

Accounts	Account Name	Total	Total (Report)
	Assets	\$10,000.00	\$10,000.00
	└ Banks	\$10,000.00	\$10,000.00
	└ European Bank	EUR 10,000.00	\$0.00
	└ US Bank	\$10,000.00	\$10,000.00
	Equity	\$10,000.00	\$10,000.00
	└ Starting Balances (EUR)	EUR 10,000.00	\$0.00
	└ Starting Balances (USD)	\$10,000.00	\$10,000.00

Initial setup of 2 bank account with different currencies. Notice that the "Total (Report)" column is being displayed. This is configured in the Options -> Account Fields to Display area.

GnuCash Currency Support

GnuCash supports over a hundred currencies, from the Andorran Franc to the Zimbabwe Dollar. You will also find supports for some lesser popular currencies like the US Dollar, Canadian Dollar, European Euro, and British Pound.

Setting International Preferences

For a full explanation of the International Preferences available, see the second chapter of this guide entitled "The Basics", specifically the Section "Setting Preferences".

Recording Currency Exchange (How-To)

In order to view your account values displayed in a currency other than their native currency, you will need to set up currency exchange rates. This is accomplished using the Price Editor.

Open the Price Editor by going to Tools -> Price Editor. Click on the Add button to add a new currency exchange. A window will appear to allow you to select the commodity you wish to define the exchange rate for. Click on Select and set the Type to ISO4217 and the Currency/security to EUR (Euro). This window should appear like this:

The screenshot shows the 'Price Editor Select Commodity Window'. It has two dropdown menus: 'Type:' set to 'CURRENCY' and 'Currency/security:' set to 'EUR (Euro)'. At the bottom, there are three buttons: 'OK' (with a green checkmark icon), 'New...', and 'Cancel' (with a red X icon).

Price Editor Select Commodity Window

Click Ok and in the Select Commodity window, set the exchange rate between the selected commodity and the selected currency. In this example, we will set the exchange rate to 1 EUR to 1 USD.

Price Information

Commodity: EUR (Euro)

Currency: USD (US Dollar)

Date: 08/06/2003

Type: Unknown

Price: 1

Price Editor Select Commodity Window

Prices

Commodity	Currency	Date	Source	Type	Price
EUR (Euro)	USD (US Dollar)	08/06/2003	user:price-editor	unknown	1

The Price Editor window after setting the exchange rate between Euros and US Dollars

File Edit View Actions Business Reports Tools Windows Help

Save Close Open Edit Options New Delete Exit

Net Assets (\$): \$10,000.00 Profits (\$): \$0.00

Accounts	Account Name	Total	Total (Report)
	Assets	\$20,000.00	\$20,000.00
	Banks	\$20,000.00	\$20,000.00
	European Bank	EUR 10,000.00	\$10,000.00
	US Bank	\$10,000.00	\$10,000.00
	Equity	\$20,000.00	\$20,000.00
	Starting Balances (EUR)	EUR 10,000.00	\$10,000.00
	Starting Balances (USD)	\$10,000.00	\$10,000.00

The main account window after setting the exchange rate between Euros and US Dollars

Recording Purchases in a Foreign Currency (How-To)

Tracking Currency Investments (How-To)

Reconciling Statements in a Foreign Currency (How-To)

Updating Exchange Rates (How-To)

Putting It All Together (Examples)

Chapter 11. Depreciation

Table of Contents

[Basic Concepts](#)

[Estimating Valuation](#)

[Depreciation Schemes](#)

[Account Setup](#)

[Example](#)

This chapter will present some of the techniques used to keep track of the changing values of assets, IE: depreciation, unrealized gains, capital gains.

Basic Concepts

This chapter will present some of the techniques used to keep track of the changing values of assets, IE: depreciation, unrealized gains, capital gains.

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.

Depreciation is the effect in which the value of an asset decreases with time. The often used example of an asset to which this is often applied is an automobile. An automobile holds retained value after the purchase date, but this value decreases with time. If you hold assets for business purposes, their depreciation can be treated as a deduction for tax purposes.

Depreciation is usually recognized as an ongoing (accrued) expense, gradually reducing the value of an asset toward zero.

Normally, depreciation is only calculated on assets used for professional or business purposes, because governments don't generally allow you to claim depreciation deductions on personal assets, and it's

usually pointless to bother with the procedure if it's not deductible. The only case where you may want to track depreciation for personal assets would be if you will sell the asset in the future and you want to track your potential personal worth.

Note

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.

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 lose value over time. For every asset which undergoes depreciation, you will need to decide on a depreciation scheme. Since depreciation of assets is very often driven by tax policies, the discussion of depreciation will focus in that direction, on some of the more common depreciation calculation schemes.

Once you have an idea of the future value of your asset, you must decide on a depreciation scheme (remember, this may be dictated by your local tax codes). This section will present 3 of the more popular depreciation schemes: *linear*, *geometric*, and *sum of digits*.

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 11.1. 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 11.2. 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 11.3. 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

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 Depreciation Expense account.

- Assets
 - ITEM1
 - Cost (Asset Cost account)
 - Depreciation (Accumulate Depreciation account)
 - ITEM2
 - Cost (Asset Cost account)
 - Depreciation (Accumulate Depreciation account)
- Bank
- Expenses
 - 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:ITEM1" account total shows you the current estimated value for item1, the "Assets:ITEM1:Cost" shows you what you originally paid for item1, "Assets:ITEM1:Depreciation" shows you your accrued depreciation for item1, and finally, "Expenses:Depreciation" demonstrates the total accrued depreciation of all your assets.

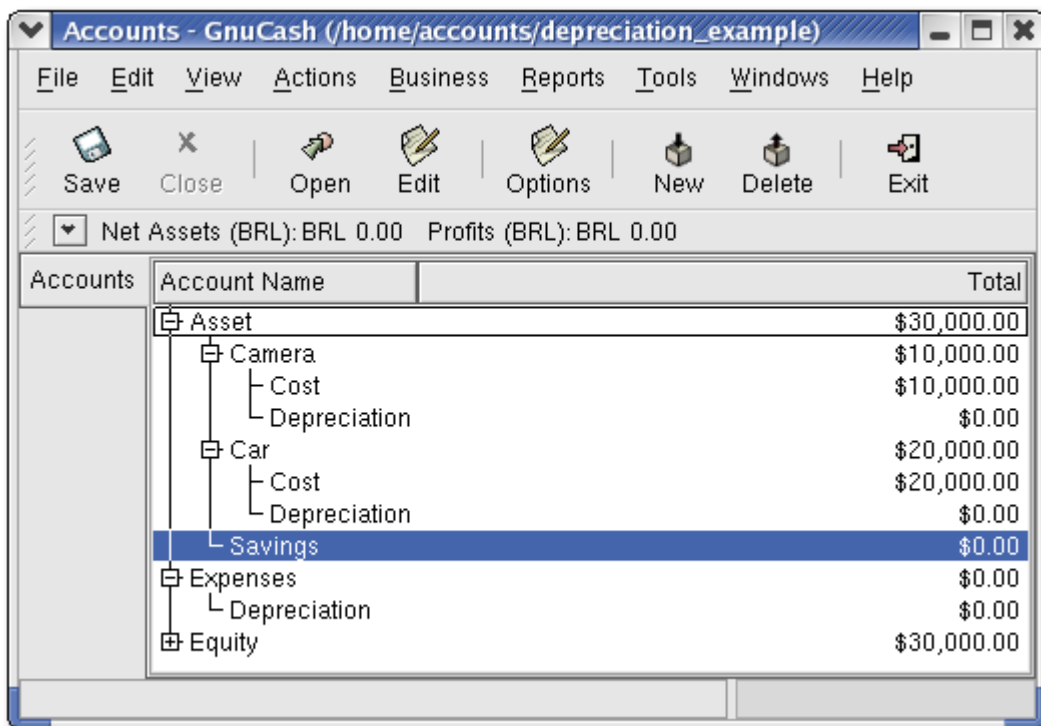
It is certainly possible to use a different account hierarchy. One popular account setup is to combine the *Asset Cost* and *Accrued Depreciation* asset accounts. This has the advantage of having fewer accounts cluttering your account hierarchy, but with the disadvantage that to determine some of the 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.

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.

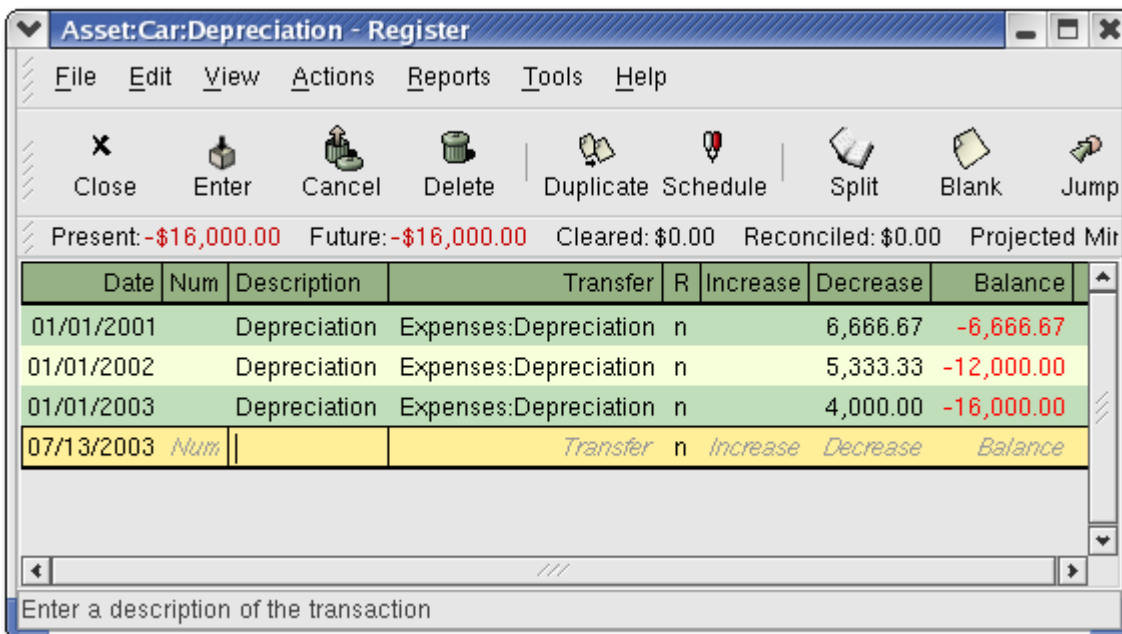
The first step is to build the account hierarchy (as shown in the previous section, replace ITEM1 and ITEM2 with "car" and "camera"). Now, record the purchase of your assets by transferring the money from your bank account to the appropriate *Asset Cost* accounts for each item (eg: the "Assets:Car:Cost" 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.



The asset depreciation example main window, before depreciation

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 "Asset: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.



The asset depreciation register window

Accounts	Account Name	Total
	Asset	\$6,000.00
	Camera	\$2,000.00
	Cost	\$10,000.00
	Depreciation	-\$8,000.00
	Car	\$4,000.00
	Cost	\$20,000.00
	Depreciation	-\$16,000.00
	Savings	\$0.00
	Expenses	\$24,000.00
	Depreciation	\$24,000.00
	Equity	\$30,000.00

The asset depreciation main window

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.

Chapter 12. Accounts Receivable

Table of Contents

[Overview of Business Features](#)

[Basic Concepts](#)

[Initial Setup](#)

[Account Setup](#)

[Register Your Company](#)

[System Components](#)

[Customers](#)

[New](#)

[Find and Edit](#)

[Invoices](#)

[New](#)

[Edit](#)

[Post](#)

[Find](#)

[Print](#)

[Assign Starting Invoice Number](#)

[Customer Jobs](#)

[Process Payment](#)

[Changing the Invoice Appearance](#)

Accounts Receivable (or A/R) refers to products or services provided by the company on credit for which

payment has not yet been received.

Overview of Business Features

The accounting needs of a business can be 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 some business oriented features to help facilitate these needs.

Accounts Receivable (A/R) and Accounts Payable (A/P) are used by businesses to record sales for which they are not immediately paid, or to record bills that they have received, but might not pay until later. These types of accounts are used primarily when you've got a lot of bills and receipts flowing in and out, and don't want to lose track of them just because you don't pay/get paid right away. For most home users, A/R and A/P are too complicated and confusing to be worth the effort.

Basic Concepts

Accounts Receivable (or A/R) refers to products or services provided by the company on credit for which payment has not yet been received. This is represented on the balance sheet as an asset, because the expectation is that you will receive payment soon.

Initial Setup

To setup GnuCash to work with handling accounts receivable for a company, there are 2 preliminary things that must be done. The first thing to do is to build an appropriate Account Hierarchy and the second is to register the company in GnuCash.

Account Setup

There are many different ways to setup a business account hierarchy. You can start with the Business Accounts setup which is available from the New Account druid, 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.

Most likely, the prebuilt Business Account hierarchy will exactly meet your needs, however it should be close enough that it is recommended you begin with it. You may need adjust some accounts manually for it to function well with your particular situation, but the defaults should be adequate for many situations.

Accounts Receivable (or A/R) refers to products or services provided by the company on credit for which payment has not yet been received. This is represented on the balance sheet as an asset, because the expectation is that you will receive payment soon. To use GnuCash's integrated accounts receivable system, you must first setup an account (usually a sub-account under Assets) defined with account type *A/Receivable*. It is within this account that the integrated A/R system will place transactions.

Basic A/R Account Hierarchy:

- Assets
 - Checking
 - Accounts Receivable
- Income
 - Sales

You will need to add additional asset, expense, and income accounts to this hierarchy for it to be useful. But, the most important aspects of this hierarchy are the use of an income account and the Accounts Receivable asset account, with account type set to *A/Receivable*.

Note

You do not need to create an individual A/R account for each customer. GnuCash will keep track of customers internally and provides reports per-customer based on the internal tracking.

Register Your Company

After you have built your account structure, you will want to register the GnuCash file as belonging to your company. To do so, add your Business Name, Address and GST number. Click on File -> Properties. Type the Business name and address, add an empty line and add the GST number: "GST: 1234-5678RT". Click OK.

System Components

GnuCash has an integrated accounts receivable system. The transactions generated by the A/R system will be placed within the Accounts Receivable account, as a record of what occurs, but you do not directly work within this account. Rather, you will principally be working with the 4 integrated GnuCash A/R application components, available from the sub-menu Business -> Customers. These 4 application 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, requesting payment. This invoice contains an itemized list of things you sold.
- Jobs (optional) is where you register Customer Jobs, a grouping mechanism by which you can group invoices to a particular customer.
- Process Payments is where you register money payments from a customer to your company.

In the next 4 sections of this chapter, the above 4 application components will be introduced.

Customers

Customers are people of 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 the appropriate billing address for the customer, such as Company Name, Address, Phone, Fax, etc. Below is a list of the other options:

- Identification - Customer Number - can be any number you would like to use to refer to this customer, or you may leave it blank and a number will be chosen automatically.
- Identification - Active - this check box is used to differentiate active customers from inactive ones, useful for when you have many past customers, and you want to see only the active.
- Billing Address - Name - this is the contact name of the person to receive the bill.
- Notes - you can enter any text you would like to in this area, additional commentaries about this customer.
- Billing Information - Currency - here you can specify what the default billing currency will be.
- Billing Information - Terms - here you can specify what the default billing terms will be for this customer. These terms must be preregistered using Business -> Billing Terms.

- Billing Information - Tax Included - this specifies whether tax has already been included in the invoice. You can choose Yes, No, or Use Global. Yes means that the tax is already included in the invoice, No means tax has not been included in the invoice, and Use Global means to use the global tax (Derek?).
- Billing Information - Tax Table - this allows you to specify a specific preregistered tax table to apply to invoices to this customer. Tax tables must be registered from the menu item Business -> Tax Table.
- Shipping Address - if the customers shipping address is different from the billing address, you can enter this information in the third and final windows tab entitled Shipping Address.

This is what the New Customer registration window looks like:

The screenshot shows a software window titled 'New Customer Registration Window'. It has three tabs: 'Customer', 'Billing Information', and 'Shipping Address'. The 'Billing Information' tab is selected. The window is divided into three main sections:

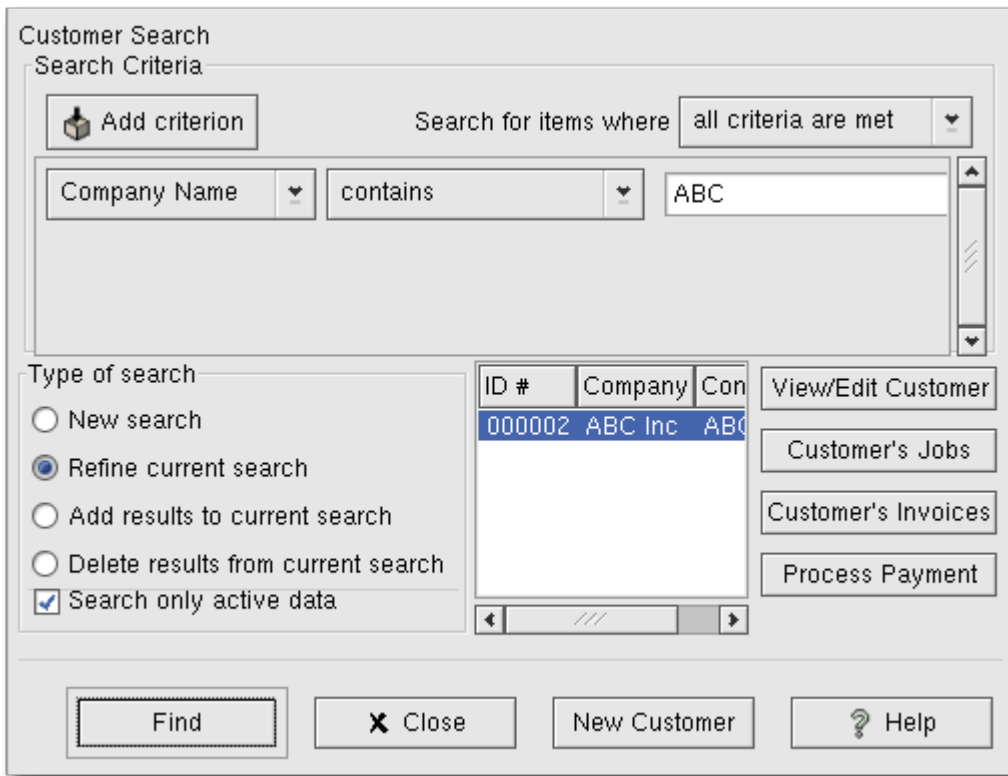
- Identification:**
 - Customer Number: [Empty text box]
 - Company Name: ABC Inc
 - Active
- Billing Address:**
 - Name: Bob McBob
 - Address: 123 First Av.
Somecity, SS 12345
 - Phone: 515-234-5678
 - Fax: 515-234-5679
 - Email: abc@abc.com
- Notes:**
 - Bob McBob, Sales Dept.

At the bottom of the window are three buttons: 'OK' (with a green arrow icon), 'Cancel' (with a red X icon), and 'Help' (with a question mark icon).

New Customer Registration Window

Find and Edit

To search for a preexisting customer, use the Business -> Customers -> Find Customers application. From the results of the search, you can select a customer to edit, or view.



Find Customer Window

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. Then, click Find. The regular expression "." means to match anything.

Invoices

An invoice is the paperwork you send to a customer requesting payment for a product or services rendered. GnuCash can be used to generate and track invoices.

New

To send an invoice to a customer you must first create a new invoice. This is accomplished using the application available at Business -> Customers -> New Invoice. The New Invoice window should be filled in appropriately:

- Invoice Info - Invoice ID - the identification number of this invoice (IE: your internal number for this invoice). If you leave it blank, a number will automatically be generated for it.
- Invoice Info - Date Opened - the date this Invoice was created.
- Billing Info - Customer - the customer who is to receive this invoice.
- Billing Info - Job (optional) - the customer job to associate with this invoice.
- Billing Info - Billing ID - this is the customer's PO Number or other "customer reference number" that you should use to identify your invoices to this customer (and job, if you have one).
- Billing Info - Terms - the payback terms agreement for this invoice, a list of preregistered terms is available within the pop up menu.

Invoice Information

Invoice ID

Date Opened 07/25/2003

Billing Information

Customer ABC Inc

Job

Billing ID

Terms None

Notes

Notes go here

Creating a New Invoice

Once you fill in all the information for the new invoice, you will be presented with the Edit Invoice window.

Edit

From the Edit Invoice window you can enter an itemized list of things you sold on this invoice in a manner similar to how the account register works.

File Edit View Actions Reports Tools Help

Close Edit Enter Cancel Delete Duplicate Blank Print Post Unpost

Invoice Information

Invoice ID 000003

Date Opened 07/25/2003

Active

Billing Information

Customer ABC Inc

Job

Billing ID

Terms None

Notes

Notes go here

Invoice Entries

Date	In	Description	Action	Income Account	Quantity	Unit Price	Dis	Dis	nt	Ta	Ta	le	Subtotal
07/25/2003	X	Nails		Income:Sales	1,000.00	0.10	%	<		X			100.00
07/25/2003	X	Hammer		Income:Sales	1.00	500.00	%	<		X			500.00
07/25/2003		Description	Action	Income Account	Quantity	Unit Price	%	<	nt	X	le		0.00

Total: \$600.00

Friday 25 July 2003

Edit Invoice Window

There are 15 columns in the Invoice Entries area:

- Date - The date this item was sold
- Invoiced? - X means the item is attached to this invoice, an empty box means the item is not attached to this invoice. If the box is empty you can attach the item to the invoice by first selecting the item row and then clicking in this box.
- Description (optional) - what the item is called
- Action (optional) - user defined field. Can place Cost Center info here, or use one of the 3 predefined actions, Hours, Material, or Project.
- Income Account - which income account is credited with this income
- Quantity - how many of the items you sold
- Unit Price - the unit price of the item
- Discount Type - type of discount, monetary value (\$) or percentage (%)
- Discount How - the discount can be computed after the tax is applied (>), before the tax is applied(<), or the discount and tax can be applied to the pretax value (=)
- Discount (optional) - total discount (in monetary units or percentage, depending on Discount Type. You can leave blank for no discount.
- Taxable? - is this item taxable? X means yes, a blank field means no.
- Tax Included? - has the tax already been included in the unit price? X means yes, a blank field means no. For example, if there is 1 item of \$100 with a tax of 5%, if Tax Included is empty then subtotal = \$100 and tax = \$5, whereas if Tax Included is true, then subtotal = \$95 and tax = \$5
- Tax Table (optional) - this is a pop up menu of all the available tax tables. If you set the item as taxable then this tax table is used to compute the amount of tax.
- Subtotal (uneditable) - computed subtotal for this item (less tax)
- Tax (uneditable) - computed tax for this item

When you have finished entering all the items, Post the invoice.

Post

When you have finished editing an invoice and are ready to print you must Post the invoice. You do not have to post the invoice immediately, and in fact you should only post the invoice when you are ready to print it. Posting an invoice will place the invoice transactions into an accounts receivable account. The Post Invoice window will appear asking you to enter the Post Date, the Due Date, a Description, and Post to Account (from which you can select from a list of A/Receivable accounts).

Do you really want to post the invoice?

Post Date 07/25/2003

Due Date 07/31/2003

Description

Post to Account Assets:Account Receivable New...

OK Cancel

Post Invoice Window

Find

To find a preexisting invoice, use the Business -> Customers -> Find Invoices application. From the results of the search, you can select an invoice to edit, or view.

Note

Before you can edit a preexisting posted invoice, you will need to unpost it.

Print

After you have posted an invoice, you should print it to send to your customer. This can be done from the Edit Invoice window using the Print button, or from within the Process Payment dialog, or from the main window: Reports -> Business Reports -> Printable Invoice. You can edit the invoice options to select the invoice you want to print. See the later section in this chapter on the Process Payment dialog for instructions on printing.

An example of the default GnuCash invoice print output is shown below.

Invoice #000003							
Invoice Date:		07/25/2003					
Due Date:		07/31/2003		July 25, 2003			
ABC Inc 123 First Av. Somecity, SS 12345							
Date	Description	Charge Type	Quantity	Unit Price	Discount	Taxable	Total
07/25/2003	Nails		1,000.00	\$0.10	0.00%		\$100.00
07/25/2003	Hammer	666	1.00	\$500.00	0.00%		\$500.00
Subtotal							\$600.00
Tax							\$0.00
Amount Due							\$600.00

Invoice Print Output

Note

You can modify the appearance of the invoice, IE: add a company logo, etc. To do so, see the section in this chapter entitled "Changing the Invoice Appearance".

Assign Starting Invoice Number

There is no easy way to assign an automatic starting invoice number. GnuCash will start with number 1 and increment from there. You can override the invoice number by typing an invoice number into the text box each time you create an invoice, but this is bound to get tiring and will sooner or later lead to duplicate numbers.

If, however, it is important that you be able to change the invoice starting number, there is a way, but unfortunately there is no UI interface to do it. You have to hand edit your XML data file (it should be right at the beginning of the file in the "counters" section).

Customer Jobs

You can use the optional GnuCash feature called Customer Jobs to associate invoices which fall under a single customer. This is useful for when you have multiple "jobs" for the same customer, and would like to view all the invoices for one job.

To use customer jobs, you must create them using the Business -> Customers -> New Jobs application. Once there, you will see the New Job window. The editable fields are:

- Job Info - Job Number (optional) - enter the number for this job. If left blank, a reasonable number will be chosen for you.
- Job Info - Job Name - the name you want to call this job.
- Owner Info - Customer - the customer associated with this job.
- Owner Info - Billing ID - the customer's reference to this job (e.g. their PO Number). It's the "Billing Identification" that they require on your invoices to them.
- Job Active - toggles this job being active or not. This is useful to use when you have lots of inactive jobs, it is easier to find all the active ones.

The screenshot shows a dialog box with the following fields and controls:

- Job Information:**
 - Job Number: [Empty text box]
 - Job Name: [Text box containing "Some Name"]
- Owner Information:**
 - Customer: [Dropdown menu showing "ABC Inc" and a "Select..." button]
 - Billing ID: [Empty text box]
- Active:**
 - Job Active
- Buttons:** OK, Cancel, Help

New Customer Job

To find a preexisting customer job, use the Business -> Customer -> Find Job application. From the results of the search, you can select a job to edit, or view.

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 -> Customers -> Process Payment.

The Process Payment application consists of:

- Payment Info - Customer - the customer who paid you.
- Payment Info - Date - the date you received payment, normally the current date.
- Payment Info - Amount - the amount of money received.
- Payment Info - Num - the check number.
- Payment Info - Memo - any comments about this payment.
- Post To - the A/Receivable account to post this transaction.
- Transfer Account - the account where the money will be deposited to (a checking account for

example) from this payment.

Below is the GnuCash Accounts Receivable payment window.

Payment Information

Customer	ABC Inc	Select...
Date	07/25/2003	▼
Amount	50.00	
Num	23	
Memo	finally, they paid	

Post To

Assets:Account Receivable ▼

Transfer Account

- Assets
 - Checking
 - Liabilities

OK Cancel

Process Payment Window

Note

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
- * you can search for invoices by company name

Or, if you prefer, you could

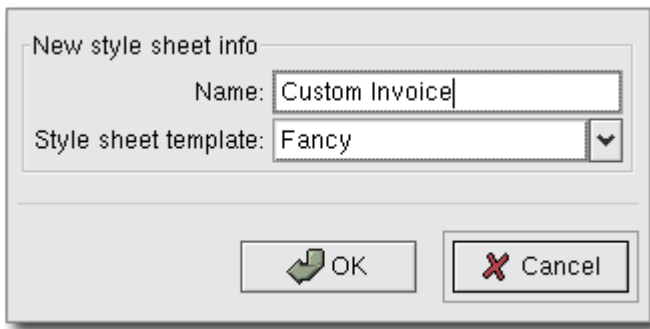
- * you can create a new invoice and then select the customer
- * you can search for a customer, view their invoices, then click "new invoice"

Changing the Invoice Appearance

The default Invoice style is fairly barren, as you can see in the example given in the "Invoices" section of this chapter. The default invoice style leaves the top part of the form blank, so that you print on company letterhead paper. However, there are some things you can do to change this default appearance.

If you go to File, Properties, then you can enter your company name and address, which will then be printed on the right hand side of the invoices.

To add a customized background, heading banner or logo to the invoices, modify the invoice style sheets. To do this, go to Edit -> Style Sheets and make a New... style sheet. You will see a window like this:

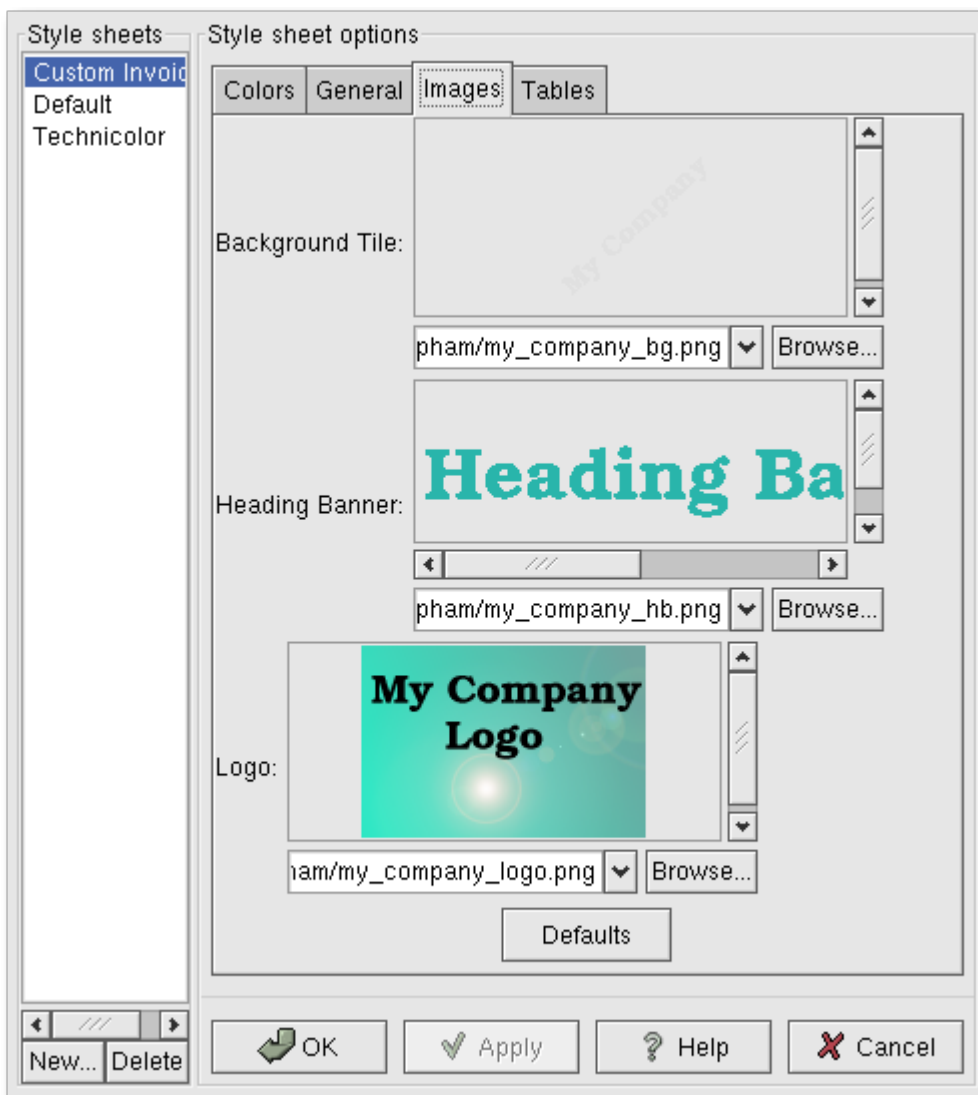


The New Style Sheet initial window.

You must select the Fancy style sheet template, and choose a name for your new style sheet. The new style sheet will appear in the list to the left. Select it, and now use the tabbed window to the right to make modifications. There are 4 tabs available, Colors, General, Images, and Tables. The Colors tab allows you to change the colors of the various sections of the invoice. The General tab allows you to set the Preparer and Prepared for information, and to enable links. The Images tab allows you to import graphics into the style sheet, such as your company logo. Finally, the Tables tab allows you to adjust the spacing around the tables which make up the invoice.

To create the graphical images, use your favorite graphics application such as The Gimp or OpenOffice Draw, and save the image in either GIF or PNG format. Then import them into the style sheet.

Below is presented a (hideous) example.



The HTML Style Sheets window with an example Background Tile, Heading Banner, and Logo.

The placement of the images in the invoice is 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.

My Company Logo

Heading Banner Text

Invoice #000005
 Invoice Date: 08/05/2003
 Due Date: 08/05/2003

ABC Inc
 123 First Av.
 Somecity, SS 12345

My Company Inc
 123 My Company Rd
 Somecity, SS, 12345
 August 5, 2003

Date	Description	Charge Type	Quantity	Unit Price	Discount	Taxable	Total
08/05/2003	Hammers		3.00	\$23.00	0.00 %		\$69.00
Subtotal							\$69.00
Tax							\$0.00
Amount Due							\$69.00

Thank you for your patronage

The hideous invoice which results from the graphics selected in the style sheet.

Chapter 13. Accounts Payable

Table of Contents

[Basic Concepts](#)

[Initial Setup](#)

[Account Setup](#)

[Register Your Company in GnuCash](#)

[System Components](#)

[Vendors](#)

[New](#)

[Find and Edit](#)

[Bills](#)

[New](#)

[Edit](#)

[Post](#)

[Find](#)

[Vendor Jobs](#)

[Process Payment](#)

Accounts Payable (or A/P) refers to the accounting of products or services which a company has bought and needs to pay off.

Basic Concepts

This chapter will focus on Accounts Payable (or A/P). A/P refers to the accounting of products or services that your company has bought, for which payment needs to be scheduled.

Initial Setup

To setup GnuCash to work with handling accounts payable for a company, there are 2 preliminary things that must be done. The first thing to do is to build an appropriate Account Hierarchy and the second is to register the company in GnuCash.

Account Setup

There are many different ways to setup a business account hierarchy. You can start with the Business Accounts setup which is available from the New Account druid, 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.

Most likely, the prebuilt Business Account hierarchy will not be perfect for your needs, you will need to perform some amount of manual adjustments for the accounts to function well with your situation. This chapter will present example account structures which you can add to your existing accounts.

Accounts Payable (or A/P) refers to products or services bought by the company for which payment has not yet been sent. This is represented on the balance sheet as a liability, because the expectation is that you will have to pay for it. To use GnuCash's integrated accounts payable system, you must first setup an account (usually a sub-account under Liabilities) defined with account type *A/Payable*. It is within this account that the integrated A/P system will place transactions.

Basic A/P Account Hierarchy:

- Assets
 - Bank
- Liabilities
 - Accounts Payable
- Expenses
 - AP Expenses

Note

Note that you will need to add real asset accounts and real expense accounts to this hierarchy. But, the most important aspects of this hierarchy are that you need an expense account and an Accounts Payable account, with account type set to *A/Payable*.

Register Your Company in GnuCash

After you have built your account structure, you will want to register the GnuCash file as belonging to your company. To do so, add your Business Name, Address and GST number. Click on File -> Properties. Type the Business name and address, add an empty line and add the GST number: "GST: 1234-5678RT". Click OK.

System Components

GnuCash has an integrated accounts payable system. The transactions generated by the A/P system will be placed within the Accounts Payable account, as a record of what occurs, but you do not directly work within this account. Rather, you will principally be working with the 4 integrated GnuCash A/P application components, available from the sub-menu Business -> Vendors. These 4 application components are:

- Vendors are people or companies from which you buy products or services on credit.
- Bills represent the physical bill you receive from a vendor, requesting payment. A bill contains an itemized list of things you purchased.
- Jobs (optional) is where you register Vendor Jobs, a grouping mechanism by which you can group bills to a particular vendor.
- Process Payments is where you register money payments to a vendor to whom you owe money.

In the next 4 sections of this chapter, the above 4 application components will be introduced.

Vendors

A vendor is a company or person from whom you purchase goods or services. The vendors must be register within the A/P system.

New

To register a new vendor, enter the menu Business -> Vendor -> New Vendor. Fill in the appropriate general information about the vendor, such as Company Name, Address, Phone, Fax, etc. Below is a list of the other options:

- Identification - Vendor Number - can be any number you would like to refer to this vendor, or you may leave it blank and a number will be chosen automatically.
- Identification - Active - this check box is used to differentiate active vendors from inactive ones.
- Payment Address - Name - this is the contact name of the person to receive the payment.
- Notes - you can enter any text you would like to in this area, additional commentaries about this vendor. For example, the names of contact people from within the vendor's company.
- Payment Information - Currency - here you can specify what the default payment currency will be.
- Payment Information - Terms - here you can specify what the default payment terms will be. These terms must be preregistered using Business -> Billing Terms.
- Payment Information - Tax Included - this specifies whether tax has already been included in the bill. You can choose Yes, No, or Use Global. Yes means that the tax is already included in the bill, No means tax has not been included in the bill, and Use Global means to use the global tax table specified in the "Tax Table" option below.
- Payment Information - Tax Table - this allows you to specify a specific preregistered tax table to apply to bills from this vendor. Tax tables must be registered from the menu item Business -> Tax Table.

This is what the New Vendor registration window looks like:

Vendor **Payment Information**

Identification

Vendor Number:

Company Name:

Active

Payment Address

Name:

Address:

Phone:

Fax:

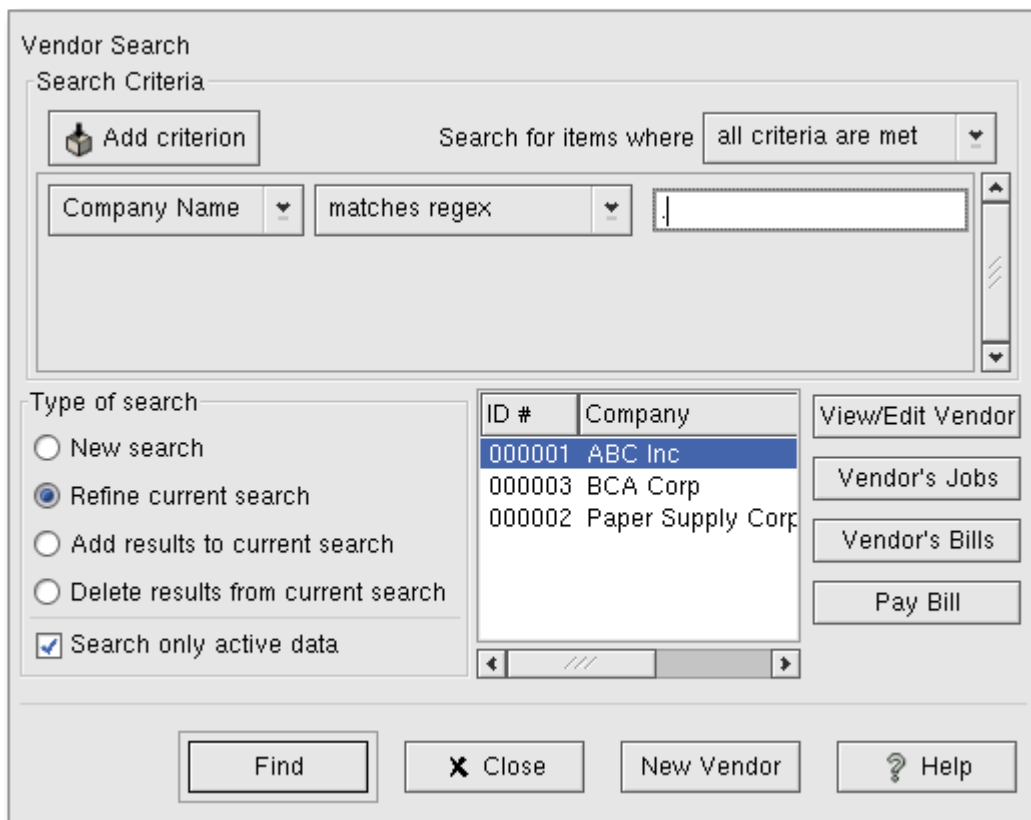
Email:

Notes

New Vendor Registration Window

Find and Edit

To search for a preexisting vendor, use the Business -> Vendors -> Find Vendors application. From the results of the search, you can select a vendor to edit, or view.



Find Vendor Window

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. Then, click Find. The regular expression "." means to match anything.

Bills

A bill is the paperwork you receive from a customer requesting payment for a product or services you purchased. GnuCash can be used to generate and track bills.

New

When you receive a bill from a vendor and want to enter it into GnuCash, you must create a new bill. This is accomplished using the application available at Business -> Vendors -> New Bill. The New Bill window should be filled in appropriately:

- Invoice Info - Invoice ID (optional) - the identification number of the invoice as emitted by the vendor (IE: the vendor's internal number for this invoice).
- Invoice Info - Date Opened - the date the Invoice was emitted by the vendor.
- Billing Info - Vendor - the issuing vendor.
- Billing Info - Job (optional) - the vendor job to associate with this bill.
- Billing Info - Billing ID (optional) - the vendor's ID for the bill (e.g.: their invoice #). Can be left blank.
- Billing Info - Terms - the payback terms agreement for this bill, a list of preregistered terms is available within the pop up menu.
- Chargeback Project - Customer (optional) - the customer to associate with this bill, for purposes of

charging the customer later.

- Chargeback Project - Job (optional) - the customer job to associate with this bill.
- Notes (optional) - any notes concerning the bill go here.

The screenshot shows a 'New Bill Registration Window' with the following fields and controls:

- Invoice Information:** Invoice ID (text box), Date Opened (07/25/2003, dropdown arrow).
- Billing Information:** Vendor (BCA Corp, Select... button), Job (text box, Select... button), Billing ID (text box), Terms (None, dropdown arrow).
- Default Chargeback Project:** Customer (text box, Select... button), Job (text box).
- Notes:** Text area containing 'notes go here', with up and down arrow buttons.
- Buttons:** OK (with a green arrow icon), Cancel (with a red X icon), Help (with a question mark icon).

New Bill Registration Window

Once you fill in all the information for the new bill, you will be presented with the Edit Bill window.

Edit

From the Edit Bill window you can enter an itemized list of things purchased on this bill, in a manner similar to how the account register works.

File Edit View Actions Reports Tools Help

Close Edit Enter Cancel Delete Duplicate Blank Print Post Unpost

Invoice Information: Invoice ID: 000002, Date Opened: 07/24/2003, Active:

Billing Information: Vendor: ABC Inc, Job: , Billing ID: , Terms: None

Notes:

Default Chargeback Project: Customer: , Job:

Invoice Entries

Date	In	Description	Action	Expense Account	Quantity	Unit Price	Taxable	Tax Included	Subtotal	Billable
07/24/2003		fly swatters	Action	enses:Office Supplies	2.00	12.00	X		24.00	
07/25/2003							X			

Total: \$24.00
fly swatters

Edit Bill Window

There are 11 columns in the Invoice Entries area:

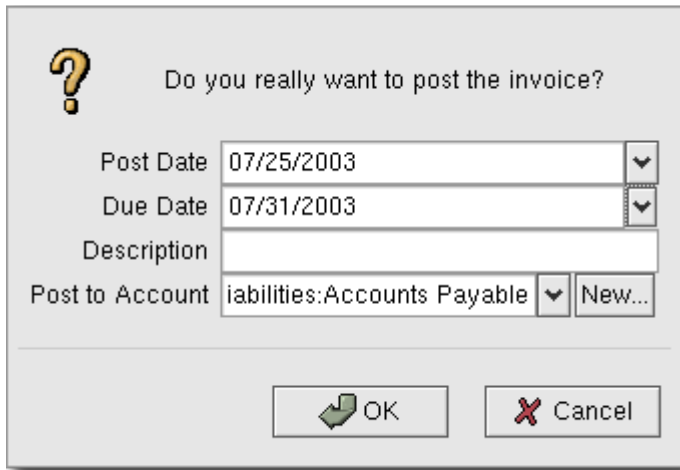
- Date - The date this item was sold
- Invoiced? - X means the item is attached to this invoice, an empty box means the item is not attached to this invoice. If the box is empty you can attach the item to the invoice by first selecting the item row and then clicking in this box.
- Description (optional) - what the item is called
- Action (optional) - user defined field. Can place Cost Center info here, or use one of the 3 predefined actions, Hours, Material, or Project.
- Expense Account - which expense account will be changed for this bill
- Quantity - how many of the items were sold to you
- Unit Price - the unit price of the item
- Taxable? - is this item taxable? X means yes, a blank field means no.
- Tax Included? - has the tax already been included in the unit price? X means yes, a blank field means no.
- Tax Table (optional) - this is a pop up menu of all the available tax tables. If you set the item as taxable and set that the tax has not been included in the unit price, then this tax table is used to compute the amount of tax.
- Subtotal (uneditable) - computed subtotal for this item (less tax)
- Billable - is this item billable to the chargeback customer/job?

When you have finished entering all the items, Post the bill.

Post

When you have finished editing a bill, you should Post the bill. You do not have to post the bill, you can simply close it and return to it later, however you need to eventually post it when you are finished editing.

Posting the bill will place the bill transactions into an accounts payable account. The Post Bill window will appear asking you to enter the Post Date, the Due Date, a Description, and Post to Account (from which you can select from a list of A/Payable accounts).



Do you really want to post the invoice?

Post Date 07/25/2003

Due Date 07/31/2003

Description

Post to Account liabilities:Accounts Payable New...

OK Cancel

Post Bill Window

Find

To find a preexisting bill, use the Business -> Vendors -> Find Bill application. From the results of the search, you can select a bill to edit, or view.

Note

Note, before you can edit a preexisting posted bill, you will need to unpost it.

Vendor Jobs

You can use the optional GnuCash feature called Vendor Jobs to associate bills which fall under a single vendor. This is useful for when you have multiple "jobs" for the same vendor, and would like to view all the bills for one job.

To use vendor jobs, you must create them using the Business -> Vendors -> New Jobs application. Once there, you will see the New Job window. The editable fields are:

- Job Info - Job Number (optional) - enter the number for this job. If left blank, a reasonable number will be chosen for you.
- Job Info - Job Name - the name you want to call this job.
- Owner Info - Vendor - the vendor associated with this job.
- Owner Info - Billing ID - the vendor's reference to this job (e.g. their PO Number). It's the "Billing Identification" that they require on your invoices to them.
- Job Active - toggles this job being active or not. This is useful to use when you have lots of inactive jobs, it is easier to find all the active ones.

The image shows a dialog box titled "New Vendor Job". It is divided into three main sections:

- Job Information:** Contains two text input fields. "Job Number" is empty, and "Job Name" contains the text "something".
- Owner Information:** Contains two fields. "Vendor" is a dropdown menu showing "ABC Inc" and a "Select..." button. "Billing ID" is an empty text input field.
- Active:** Contains a checked checkbox labeled "Job Active".

At the bottom of the dialog box, there are three buttons: "OK" (with a green arrow icon), "Cancel" (with a red X icon), and "Help" (with a question mark icon).

New Vendor Job

To find a preexisting vendor job, use the Business -> Vendors -> Find Job application. From the results of the search, you can select a job to edit, or view.

Process Payment

Eventually, you will need to pay your bills. To do so, use the Process Payment application found in Business -> Vendors -> Process Payment.

The Process Payment application consists of:

- Payment Info - Vendor - the vendor you wish to pay
- Payment Info - Date - the date you wish to make the payment, normally the current date.
- Payment Info - Amount - the amount of money to transfer in this payment.
- Payment Info - Num - the check number.
- Payment Info - Memo - any comments about this payment.
- Post To - the A/Payable account to post this transaction.
- Transfer Account - the account where the money will come from (a checking account for example) to make this payment.

Below is the GnuCash Accounts Receivable payment window.

Payment Information	
Vendor	ABC Inc <input type="button" value="Select..."/>
Date	07/25/2003 <input type="button" value="v"/>
Amount	0.00
Num	32
Memo	finally paying them off
Post To	
	Liabilities:Accounts Payable <input type="button" value="v"/>
Transfer Account	
<input type="checkbox"/>	Assets
<input type="checkbox"/>	└─ Checking
<input type="checkbox"/>	Liabilities

Process Payment Window

Chapter 14. Payroll

Table of Contents

[Basic Concepts](#)

[Account Setup](#)

[Protocol](#)

[Step 1: Deductions list](#)

[Step 2: Create the Transaction Map](#)

[Step 3: Pay the Employee](#)

[Step 4: Pay the Government](#)

[Example](#)

[Build Protocol](#)

[Pay an Employee](#)

[Pay the Government](#)

Payroll is a financial record of wages, net pay, paid vacations, and deductions for an employee. This chapter will demonstrate how to track payroll in 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. This is one of the more complex tasks in terms of accounting, simply because there are many different accounts, people, and agencies involved in the act of paying an employee's salary.

Payroll is typically accounted for as an expense. Sometimes, accountants will "store" some payroll expenses in a short term liability account. This is useful for things such as payroll taxes, which may be paid to the government at a different times from paying employee salaries. This chapter will present a methodology which expenses payroll immediately for salaries, but store taxes in a liability account.

Note

GnuCash does not (yet) have an integrated payroll system. What this means is that while you can track payroll expenses in GnuCash, the actual calculation of taxes and deductions will have to be done outside of GnuCash.

Account Setup

Some considerations that must be addressed when setting up your accounts is how your local tax laws work. Because there are so many different ways payroll taxes can be handled throughout the world, this section will present a simple structure. From this, you should be able to adapt your particular payroll deductions setup.

Let us assume that you must pay 2 taxes, Tax1 and Tax2 and that each has an employee contributed component and an employer contributed component.

The employee's salary and these two taxes will have to have expense accounts associated, and the taxes will have to have liability accounts. The tax liability account is the place where you accumulate the withheld taxes for all of your employees, which is 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
 - Taxes

Note

There is a temptation to create per-employee sub-accounts, to track each employee separately. However, this often leads to unmanageably large lists of accounts. Imagine how your account structure would look after a few years of employees coming and going in your company. It is much simpler to keep all of your employees payroll records within a single account (Expense:Salaries for example) and use reports to view per-employee information.

Protocol

GnuCash does not (yet) have an integrated payroll system. You can use GnuCash to track your payroll expenses, but you will 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, which you can use 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, which should include definitions and formulas for calculating each value. This step of the protocol will only need to be done once, and again in the future when the payroll laws or tax rates change. For our simple payroll situation with the two taxes, 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.

You will want to place the actual formulas for calculating each deduction in this list. Sometimes, these formulas can be quite complex, and sometimes the formula may simply say "look this number up in table XYZ of the tax codes."

Notice that you can calculate some interesting values using the above definitions, such as 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, it is done with a single split transaction. This split transaction will populate the appropriate expense and liability accounts. If, in the future, you want to discover any of the payroll details for a particular employee, simply open the split transaction.

With the deductions list from above in hand, an employee split transaction map can be generated which maps each of the items in the list to a GnuCash account.

Table 14.1. Transaction Map

Account	Increase	Decrease
Assets:Checking		<i>E_NET_SALARY</i>
Expenses:Salaries	<i>E_GROSS_SALARY</i>	
Liabilities:Tax1		<i>E_TAX1</i>
Liabilities:Tax2		<i>E_TAX2</i>
Expenses:Tax1	<i>C_TAX1</i>	
Liabilities:Tax1		<i>C_TAX1</i>
Expenses:Tax2	<i>C_TAX2</i>	
Liabilities:Tax2		<i>C_TAX2</i>

Step 3: Pay the Employee

Go to the account from which the employee will be paid, for example your Asset: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 will certainly want to make use of when entering employee payroll is duplicate transaction (use the Duplicate tool bar button). Using this will save you from

having to enter all the transaction splits for each employee. You will still need to change the amounts of money to match each employee's real payroll values, but at least you do not need to build the split each 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 will 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 will be between the checking account and the liability account, no expense account are involved.

Example

Using the account setup seen previously, let's go through an example. Assume that there are 2 employees (E1 and E2) which 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 employees gross salary.

Starting with \$50k in the bank, and before doing any payroll, the account hierarchy looks like this:

Account Name	Total
Assets	\$50,000.00
└─ Checking	\$50,000.00
Liabilities	\$0.00
└─ Tax1	\$0.00
└─ Tax2	\$0.00
Expenses	\$0.00
└─ Salaries	\$0.00
└─ Tax1	\$0.00
└─ Tax2	\$0.00
Equity	\$50,000.00
└─ Starting Balances	\$50,000.00

Payroll Initial Setup

Build Protocol

The deductions list for employee 1:

- *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*)

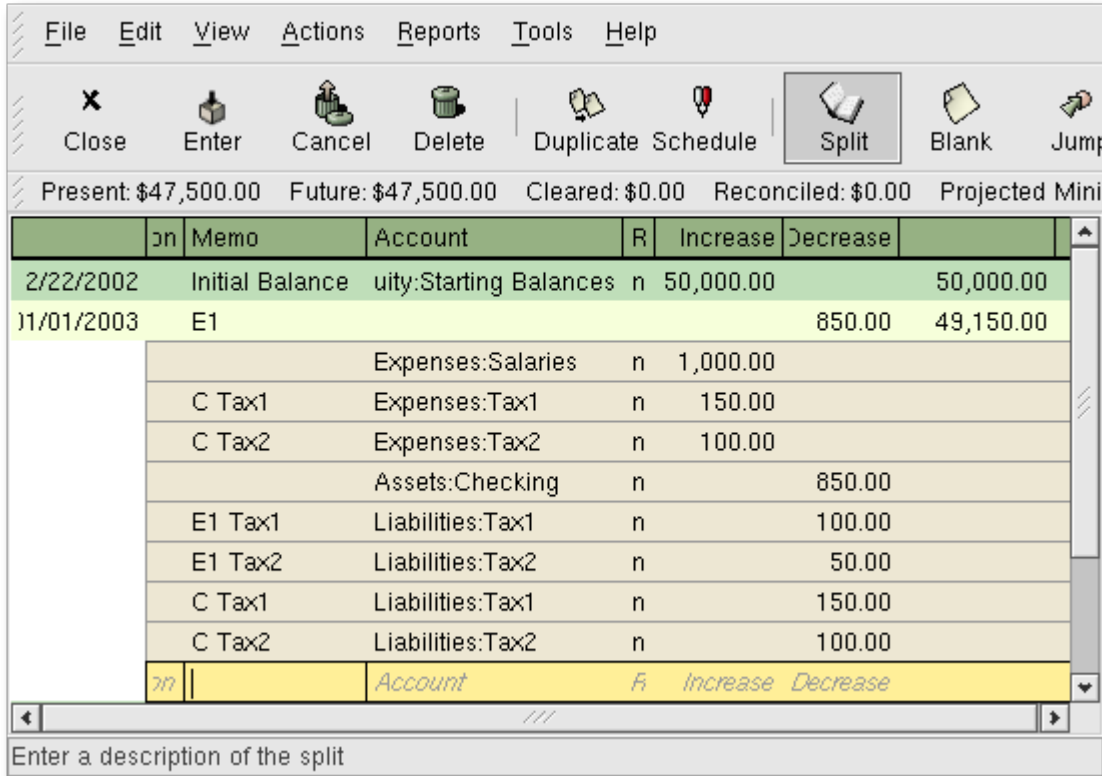
Table 14.2. Transaction Map for Employee 1

Account	Increase	Decrease
Assets:Checking		\$850 (<i>E_NET_SALARY</i>)
Expenses:Salaries	\$1000 (<i>E_GROSS_SALARY</i>)	
Liabilities:Tax1		\$100 (<i>E_TAX1</i>)
Liabilities:Tax2		\$50 (<i>E_TAX2</i>)

Expenses:Tax1	\$150 (C_TAX1)	
Liabilities:Tax1		\$150 (C_TAX1)
Expenses:Tax2	\$100 (C_TAX2)	
Liabilities:Tax2		\$100 (C_TAX2)

Pay an Employee

Now, enter the first split transaction for employee 1 in the checking account. The split transaction will look like this:



Employee 1 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 will automatically be made out to the employee correctly. 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:

Account Name	Total
Assets	\$48,300.00
Checking	\$48,300.00
Liabilities	\$800.00
Tax1	\$500.00
Tax2	\$300.00
Expenses	\$2,500.00
Salaries	\$2,000.00
Tax1	\$300.00
Tax2	\$200.00
Equity	\$50,000.00
Starting Balances	\$50,000.00

Pay the Government

The Liability:Tax1 and Liability:Tax2 accounts will continue to track how much you must pay to the government agencies responsible for each. When it is time to pay these agencies, make a transaction from the checking account to the liability accounts. No expense accounts are involved. The main account will then appear like this for this example:

Account Name	Total
[-] Assets	\$47,500.00
[-] Checking	\$47,500.00
[-] Liabilities	\$0.00
[-] Tax1	\$0.00
[-] Tax2	\$0.00
[-] Expenses	\$2,500.00
[-] Salaries	\$2,000.00
[-] Tax1	\$300.00
[-] Tax2	\$200.00
[-] Equity	\$50,000.00
[-] Starting Balances	\$50,000.00

Accounts After Paying Government

Appendix A. Migration Guide

Table of Contents

[Using Accounts vs. Categories](#)

[Organization of QIF Files \(Discussion\)](#)

[Common Duplication Issues \(Discussion\)](#)

[Checking QIF Data \(Discussion\)](#)

[Converting XML GnuCash File](#)

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.

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 (refer to section 2.1), incomes 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 Chapter 4 in greater detail.

Organization of QIF Files (Discussion)

Common Duplication Issues (Discussion)

Checking QIF Data (Discussion)

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 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 to a working file. Modify the working file's `<gnc-v2>` tag to read something like what's below. Note that you can pretty much put anything you want in the `'="..."` part; I used the URL because it's traditional (if such can be said about such a young technology!).

```
<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:comdty="http://www.gnucash.org/XML/comdty"
  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:addr="http://www.gnucash.org/XML/custaddr">
```

2. Create an XSLT stylesheet containing the transformation your desire, or obtain one that's already written (AFAIK, there aren't any, but I'm working on a CSV one).
3. Install an XSLT processor such as Saxon (<http://saxon.sourceforge.net/>) or Xalan-J (<http://xml.apache.org/>). 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. I know the process works on 1.8.4 and CVS HEAD datafiles, and ought to work on earlier 1.8.x versions, too. Haven't mucked with 1.6.x in a while, but it *should* work there, too...

Appendix B. Frequently Asked Questions

Table of Contents

Sources of Information

[Q: Where's the FAQ?](#)

[Q: Are there mailing lists for GnuCash?](#)

[Q: Is there a searchable archive for the mailing lists?](#)

[Q: Are there other means of obtaining support for GnuCash?](#)

General Information

[Q: Can I run GnuCash on Windows?](#)

[Q: I heard it is too hard to compile GnuCash!](#)

[Q: Is there a batch mode \(non-interactive\) available for GnuCash, for building reports, etc?](#)

[Q: Can multiple people access the same datafile in GnuCash?](#)

[Q: Why is GnuCash written in C?](#)

[Q: Why don't you rewrite GnuCash in programming language xyz so that I can contribute easily?](#)

[Q: I really want feature XYZ but GnuCash doesn't have it. How do I get it added?](#)

[Q: Is there a web interface available for GnuCash?](#)

[Q: How can I provide security for GC data using CFS, etc.\)](#)

[Q: How can I contribute to the GnuCash project?](#)

[Q: I think I found a bug. How do I report it?](#)

Using GnuCash

[Q: How can I move the transactions from account A into account B, thus combining them?](#)

[Q: Is it possible to merge two gnuCash files?](#)

[Q: How can I save a template of my account structure?](#)

[Q: When I search for customers \(or anything else for that matter\), how can I return a list of everything?](#)

[Q: How can I record a transaction on different dates \(actual date and bank date\)?](#)

Accounting

[Q: How do I treat taxes? As an account payable or as an expense?](#)

This is a list of questions asked on the mailing lists for which there really is no section in the documentation covering the subject.

Sources of Information

Q: Where's the FAQ?

A: You're looking at it. The definitive copy can be found within the GnuCash Guide <http://www.gnucash.org/docs/v1.8/C/gnucash-guide/>

Q: Are there mailing lists for GnuCash?

A: Yes. Go to <http://www.gnucash.org/cgi-bin/mailman/listinfo/gnucash-user> and <http://www.gnucash.org/cgi-bin/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 #gncash discussion on irc.gnome.org. Also, there is a wiki online at <http://gnomesupport.org/wiki/index.php/GnuCash>.

General Information

Q: Can I run GnuCash on Windows?

A: No, GnuCash is not developed on (or for) Windows.

If you must run GnuCash from a Windows machine, your best bet is to use VNC to run GnuCash remotely from a UNIX server. Another possibility would be to install VMWare (www.vmware.com) on your windows machine and run a copy of Linux within VMWare. We don't have many (any?) windows developers, so nobody is really working hard on a port to windows....

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 (except g-wrap, which means there is in fact "one" extra library to be installed before compiling GnuCash). 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". 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.

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 http://bugzilla.gnome.org/enter_bug.cgi?product=GnuCash. 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 GC 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 <http://www.gnucash.org/cgi-bin/mailman/listinfo/gnucash-user> and <http://www.gnucash.org/cgi-bin/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 the FAQ above, "Why is GnuCash written in C?" 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 [Gnome Bugzilla](#) database.

If you feel you have indeed found a bug, you can then report it at http://bugzilla.gnome.org/enter_bug.cgi?product=GnuCash. 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!

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 -> Style -> 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.

Be careful! 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 -> 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).

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.

Appendix C. Contributed Account Trees

Table of Contents

[UK Vat](#)

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 payed, is matched by a decrease in capital.

Bank Accounts [B]

- Main Account
- Reserve Account

Cash [A]

Assets [A]

- Capital Equipment (Box 7 - Part) - additions only, not absolute value
- Computers Can be depreciated to zero this year
- EEC 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/ EEC 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
 - EEC
 - 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
 - Tax
- Other Non VAT Expenses
- VAT Purchases (Box 7 - part)
 - Accountancy
 - Bank Charges
 - Consumables
 - EEC 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

Appendix D. GNU Free Documentation License

Version 1.1, March 2000

Copyright © 2000 Free Software Foundation, Inc.

Free Software Foundation, Inc. 59 Temple Place,

Suite 330, Boston, MA
02111-1307 USA

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

Table of Contents

[0. PREAMBLE](#)

[1. APPLICABILITY AND DEFINITIONS](#)

[2. VERBATIM COPYING](#)

[3. COPYING IN QUANTITY](#)

[4. MODIFICATIONS](#)

[5. COMBINING DOCUMENTS](#)

[6. COLLECTIONS OF DOCUMENTS](#)

[7. AGGREGATION WITH INDEPENDENT WORKS](#)

[8. TRANSLATION](#)

[9. TERMINATION](#)

[10. FUTURE REVISIONS OF THIS LICENSE](#)

[Addendum](#)

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.

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. 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.

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 material, 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.

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](#).

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.”

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.

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.

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.

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.

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.

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.