

GnuCash教程与概念指南

教学文档

GnuCash文档团队

GnuCash教程与概念指南: 教学文档

由 GnuCash文档团队

本指南包含5.12使用教程，并介绍了GnuCash背后的概念原理。

版权 © 2009-2025 GnuCash文档团队

版权 © 2010-2011 Yawar Amin

版权 © 2010 Tom Bullock

版权 © 2010-2011 Cristian Marchi

版权 © 2006 Chris Lyttle

版权 © 2003-2004 Jon Lapham

版权 © 2002 Chris Lyttle

版权 © 2001 Carol Champagne 和 Chris Lyttle

版权 © 2023-2024 Xinhang Xu

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 [help:fdl] 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.

要报告有关程序或本文档的缺陷或提出相关建议，请按照GnuCash维基中Bugzilla页面 [https://wiki.gnucash.org/wiki/Bugzilla]所列指引使用GnuCash缺陷跟踪系统 [https://bugs.gnucash.org/]。建议首先在缺陷跟踪系统中通过浏览(browse) [https://bugs.gnucash.org/describecomponents.cgi]或搜索(search) [https://bugs.gnucash.org/query.cgi]寻找相关缺陷是否已被报告。程序相关的问题请选择产品GnuCash，本文档相关的任何问题请选择产品Documentation。

目录

| | |
|------------------------|-----|
| 关于作者 | xxi |
| 1. 概览 | 1 |
| 介绍 | 1 |
| 特点 | 2 |
| 易于使用 | 2 |
| 跟踪投资 | 3 |
| 国际化支持 | 3 |
| 企业功能支持 | 3 |
| 会计特性 | 4 |
| 关于本指南 | 4 |
| 获取帮助 | 6 |
| 上下文帮助 | 6 |
| GnuCash网站 | 6 |
| GnuCash维基 | 6 |
| GnuCash常见问题 | 6 |
| GnuCash在线帮助 | 6 |
| 主题搜索 | 7 |
| 安装 | 7 |
| I. 基础知识 | 8 |
| 2. 基础 | 11 |
| 会计概念 | 11 |
| 五类基本科目 | 11 |
| 会计等式 | 11 |
| 借贷记账法 | 12 |
| 复式记账法 | 13 |
| 数据记录概念 | 13 |
| 文件 | 13 |
| 科目 | 14 |
| 交易 | 14 |
| 运行GnuCash | 15 |
| 〔欢迎使用GnuCash〕对话框 | 15 |
| 建立科目结构 | 15 |
| 每日提示 | 16 |
| 科目树窗口 | 17 |
| 科目账簿窗口 | 18 |
| 工具栏按钮 | 19 |
| 标签栏 | 19 |
| 菜单项 | 20 |
| 菜单的访问键与快捷键 | 20 |
| 存储财务数据 | 20 |

| | |
|--|----|
| 概述 | 20 |
| 存储压缩及有关建议 | 21 |
| 存储格式对比表 | 21 |
| 创建文件 | 21 |
| 保存数据 | 22 |
| 打开数据 | 24 |
| 复制科目层次结构 | 24 |
| 备份与恢复数据 | 24 |
| 备份文件 (".gnucash") | 25 |
| 日志文件 (".log") | 25 |
| 锁定文件 (".LNK" 和 ".LCK") | 25 |
| 文件管理 | 26 |
| 迁移GnuCash数据 | 26 |
| 迁移财务数据 | 26 |
| 迁移首选项数据 | 26 |
| 步步为营 | 28 |
| 科目 | 34 |
| 基本会计概念 | 34 |
| GnuCash 科目 | 35 |
| 步步为营 | 39 |
| 交易 | 43 |
| 基本概念 | 43 |
| 科目账簿 | 43 |
| 简单交易还是分录交易? | 48 |
| 对账 | 53 |
| 计划交易 | 58 |
| 步步为营 | 68 |
| 3. Importing Data into GnuCash | 75 |
| Importing Transactions from Files | 75 |
| File Import Formats | 75 |
| Import Matcher | 75 |
| Import QIF | 77 |
| Import OFX/QFX | 78 |
| Import CSV | 78 |
| Import MT940 | 81 |
| Import MT942 | 81 |
| Import DTAUS | 81 |
| 4. Configuring GnuCash | 82 |
| Overview | 82 |
| Main Configuration Screens | 82 |
| Preferences: Accounting Period Tab | 82 |
| Preferences: Accounts Tab | 82 |
| Preferences: Online Banking | 82 |

| | |
|---------------------|-----|
| II. 常规使用 | 83 |
| 5. 活期账户与现金 | 87 |
| 创建科目 | 87 |
| 录入存款 | 87 |
| 录入取款 | 88 |
| 使用ATM机取出现金 | 88 |
| 科目对账 | 88 |
| 步步为营 | 89 |
| 期初余额 | 89 |
| 添加交易 | 90 |
| 期初余额 | 92 |
| 保存文件 | 95 |
| 报表 | 96 |
| 6. 费用 | 100 |
| 概念 | 100 |
| 创建科目 | 100 |
| 建立简单的费用科目 | 100 |
| 复杂的费用科目结构 | 100 |
| 录入费用交易 | 100 |
| 有关费用科目的其它注意事项 | 101 |
| 7. 信用卡 | 102 |
| 概念 | 102 |
| 创建科目 | 102 |
| 简易科目设置 | 102 |
| 完整科目设置 | 103 |
| 记录付款 | 103 |
| 记录还款 | 104 |
| 步步为营 | 106 |
| 打开GnuCash文件 | 106 |
| 购买 | 107 |
| 退款 | 108 |
| 缴纳利息 | 109 |
| 对账 | 110 |
| 还款 | 111 |
| 保存文件 | 113 |
| 报表 | 113 |
| 8. 贷款 | 117 |
| 基本概念 | 117 |
| 术语 | 117 |
| 创建科目 | 118 |
| 计算器 | 118 |
| 示例：每月还款 | 120 |
| 示例：贷款总期数 | 120 |

| | |
|-------------------------|-----|
| 进阶：计算详细过程 | 120 |
| 怎样：记录房屋按揭贷款 | 122 |
| 怎样：记录汽车贷款 | 125 |
| 怎样：记录给朋友的私人借款 | 126 |
| 借款约定 | 126 |
| 借款科目 | 127 |
| 借出钱款 | 127 |
| 收到第一笔还款 | 128 |
| 收到第二笔还款 | 129 |
| 怎样：为贷款账单对账 | 131 |
| 怎样：出售房屋或汽车 | 132 |
| 简单交易 | 132 |
| 更复杂的交易 | 133 |
| 9. 投资 | 135 |
| 基本概念 | 135 |
| 术语 | 135 |
| 投资类型 | 136 |
| 创建科目 | 137 |
| 使用预置的投资科目 | 137 |
| 手动创建投资科目 | 138 |
| 自建科目示例 | 138 |
| 计息科目 | 140 |
| 创建科目 | 140 |
| 示例 | 140 |
| 创建投资组合 | 143 |
| 创建股票和公募基金科目 | 144 |
| 股票科目示例 | 144 |
| 购入份额 | 148 |
| 录入已持有的份额 | 148 |
| 新购份额 | 149 |
| 设置份额价格 | 151 |
| 价格数据库初始设置 | 151 |
| 手动设置股票价格 | 152 |
| 配置自动获取报价 | 153 |
| 显示份额价值 | 153 |
| 报表中的“价格来源” | 154 |
| 卖出份额 | 156 |
| 卖出份额时手动计算资本损益 | 157 |
| 使用批次卖出份额并自动计算资本损益 | 164 |
| 分红 | 176 |
| 现金分红 | 176 |
| 红利再投 | 177 |
| 资本退回 | 178 |

| | |
|---|-----|
| 拆股与合股 | 178 |
| 简单的拆股 | 179 |
| 较为复杂的合股 | 181 |
| 10. Reports | 184 |
| Overview | 184 |
| Saved Reports Configurations Concepts | 184 |
| Standard Reports Overview | 185 |
| Transaction Report | 185 |
| Assets & Liabilities Group | 186 |
| Budget Group | 191 |
| Business Group | 192 |
| Income and Expense Group | 193 |
| Sample & Custom Group | 198 |
| Other Reports | 199 |
| Using the Transaction Report | 200 |
| 11. 资本利得 | 203 |
| 基本概念 | 203 |
| 估值 | 203 |
| 建立科目 | 204 |
| 示例 | 204 |
| 未实现损益 | 205 |
| 出售 | 206 |
| 缴税 | 208 |
| 12. 多货币 | 210 |
| 基本概念 | 210 |
| 手动记录外币交易 | 210 |
| 建立科目 | 210 |
| 记录并更新货币汇率 | 213 |
| 记录通过外币的购买 | 222 |
| 跟踪外币投资 | 228 |
| 对账外币账单 | 229 |
| 使用贸易科目自动记录外币交易 | 229 |
| 贸易科目的概念 | 230 |
| 使用贸易科目记录外币交易 | 230 |
| III. 高级功能 | 239 |
| 13. Business Features | 242 |
| Introduction to Business Features | 242 |
| Business Setup | 242 |
| Account Setup | 242 |
| Setting up Sales Tax Tables | 244 |
| Enter Company Information | 246 |
| Setting Business Preferences | 248 |
| Setting Billing Terms | 248 |

| | |
|---|-----|
| Accounts Receivable | 250 |
| System Components | 250 |
| Customers | 251 |
| Invoices | 252 |
| Customer Jobs | 256 |
| Process Payment | 257 |
| Changing the Invoice Appearance | 257 |
| Accounts Payable | 260 |
| System Components | 260 |
| Vendors | 261 |
| Bills | 263 |
| Vendor Jobs | 266 |
| Process Payment | 267 |
| Payroll | 267 |
| Basic Concepts | 267 |
| Account Setup | 268 |
| Payroll Protocol Sample | 269 |
| Business Payroll Example | 271 |
| 14. Budgets | 275 |
| Basic Concepts | 275 |
| Terminology | 275 |
| Creating a Budget | 276 |
| Choose Which Accounts To Budget For | 276 |
| Choosing a Budget Period | 276 |
| Getting Started | 276 |
| Entering Budget Values | 277 |
| Budget Reporting | 277 |
| Budgeted Income Statement | 277 |
| Budgeted Balance Sheet | 277 |
| 15. Other Assets | 278 |
| General Concepts | 278 |
| Other Assets Described | 278 |
| Current Assets | 278 |
| Long-term (Fixed) Assets | 278 |
| Current Assets | 279 |
| Short-term Receivables | 279 |
| Reimbursable Expenses | 279 |
| Travel Advances | 281 |
| Prepaid Premiums or Prepaid Rent | 281 |
| Suspense or Wash Accounts | 282 |
| Short or Long-term Assets | 283 |
| Long-term (Fixed) Assets | 284 |
| Land | 284 |

| | |
|---|-----|
| Buildings | 284 |
| Leasehold Improvements | 285 |
| Vehicles or Equipment | 285 |
| Intangibles | 285 |
| 16. Depreciation | 286 |
| Basic Concepts | 286 |
| Personal Finances | 286 |
| Business | 287 |
| Estimating Valuation | 288 |
| Depreciation Schemes | 288 |
| Account Setup | 290 |
| Example | 291 |
| 17. Python Bindings | 295 |
| 18. Importing Business Data | 296 |
| Importing Bills and Invoices | 296 |
| General | 296 |
| The format of the import file | 296 |
| Import your data | 298 |
| Feedback and statistics | 299 |
| What could go wrong? | 300 |
| Not supported invoice functionality | 302 |
| Importing Customers and Vendors | 302 |
| General | 302 |
| The format of the import file | 302 |
| Import your data | 304 |
| Feedback and statistics | 305 |
| What could go wrong? | 306 |
| Not supported customer/vendor functionality | 306 |
| IV. 附录 | 307 |
| GnuCash 术语表 | 309 |
| A. Migration Guide | 314 |
| Using Accounts vs. Categories | 314 |
| Importing Data | 314 |
| Import of Financial Data by QIF Files | 314 |
| Import of Business Data by CSV Files | 315 |
| Converting XML GnuCash File | 315 |
| B. Auxiliary File Formats | 317 |
| Check Format Files ("*.chk") | 317 |
| Overview | 317 |
| Example file | 317 |
| Field Descriptions | 318 |
| Creating Check Format Files | 332 |
| C. GNU Free Documentation License | 333 |

| | |
|---|-----|
| 0. PREAMBLE | 333 |
| 1. APPLICABILITY AND DEFINITIONS | 333 |
| 2. VERBATIM COPYING | 334 |
| 3. COPYING IN QUANTITY | 335 |
| 4. MODIFICATIONS | 335 |
| 5. COMBINING DOCUMENTS | 337 |
| 6. COLLECTIONS OF DOCUMENTS | 338 |
| 7. AGGREGATION WITH INDEPENDENT WORKS | 338 |
| 8. TRANSLATION | 338 |
| 9. TERMINATION | 339 |
| 10. FUTURE REVISIONS OF THIS LICENSE | 339 |
| Addendum | 339 |

插图清单

| | |
|--|----|
| 1.1. 包含多种货币和投资类型的科目表 | 1 |
| 2.1. 基本科目关系 | 12 |
| 2.2. 每日提示 | 17 |
| 2.3. 基本科目关系 | 17 |
| 2.4. 活期账户的账簿 | 18 |
| 2.5. 选择 [xml] 或 [sqlite3] 时的保存对话框 | 22 |
| 2.6. 选择 [mysql] 或 [postgres] 时的保存对话框 | 23 |
| 2.7. 建立科目结构: 介绍 | 28 |
| 2.8. 建立科目结构: 账簿选项 | 29 |
| 2.9. 建立科目结构: 货币选择 | 30 |
| 2.10. 建立科目结构: 科目选择 | 31 |
| 2.11. 建立科目结构: 科目设置 | 32 |
| 2.12. 建立科目结构: 完成 | 33 |
| 2.13. 教程: 初始示例文件的科目页面 | 34 |
| 2.14. 基本科目间的关系 | 34 |
| 2.15. 默认收入科目 | 38 |
| 2.16. 默认费用科目 | 38 |
| 2.17. 创建资产科目 | 40 |
| 2.18. 基本顶级科目 | 41 |
| 2.19. 基本科目表 | 42 |
| 2.20. 在资产:活期科目中查看购买三条牛仔裤的交易 | 44 |
| 2.21. 在费用:服装科目中以基本账簿模式查看购买三条牛仔裤的交易 | 45 |
| 2.22. 在费用:服装科目中以交易日志模式查看购买三条牛仔裤的交易 | 46 |
| 2.23. 资产:活期科目中的薪水 | 49 |
| 2.24. 收入:薪资科目中的薪水 | 50 |
| 2.25. 添加期初余额交易后的活期科目 | 51 |
| 2.26. 添加购买杂货交易后的活期科目 | 52 |
| 2.27. 录入一笔分录交易 | 53 |
| 2.28. 初始的对账窗口 | 54 |
| 2.29. 对账窗口 | 56 |
| 2.30. 推迟对账后的活期存款科目 | 58 |
| 2.31. 从账簿创建计划交易的第1步 | 59 |
| 2.32. 从账簿创建计划交易的第2步 | 59 |
| 2.33. 从账簿创建计划交易的第3步 | 60 |
| 2.34. 填写计划交易的参数 | 61 |
| 2.35. 空白的计划交易标签页 | 62 |
| 2.36. [编辑计划交易]窗口的[概览]标签页 | 63 |
| 2.37. [编辑计划交易]窗口的[频率]标签页 | 64 |
| 2.38. [编辑计划交易]窗口的[交易模板]标签页 | 66 |
| 2.39. 最终录入的计划交易 | 67 |

| | |
|--|-----|
| 2.40. 计划交易弹出提醒 | 68 |
| 2.41. 步步为营的起点 | 69 |
| 2.42. 设置期初余额后的科目表 | 70 |
| 2.43. 添加多项交易后的科目表 | 72 |
| 2.44. 三月份的现金流量表 | 73 |
| 2.45. 三月份期间活期科目的交易汇总表 | 74 |
| 2.46. 三月份期间费用科目的交易汇总表 | 74 |
| 5.1. 科目表 | 89 |
| 5.2. 活期科目账簿 | 90 |
| 5.3. 在填入更多支付记录后的活期科目账簿 | 91 |
| 5.4. 录入了ATM机取款的活期科目账簿 | 92 |
| 5.5. 一份示例银行账单 | 92 |
| 5.6. 对账窗口 | 93 |
| 5.7. 补充服务费后的活期科目 | 94 |
| 5.8. 已对账的活期科目 | 95 |
| 5.9. 科目表 | 96 |
| 5.10. 3月的现金流量表 | 97 |
| 5.11. 3月份期间资产科目的交易汇总表 | 98 |
| 5.12. Transaction Report For The Expenses During March | 99 |
| 7.1. 对冲购买交易后的信用卡科目 | 105 |
| 7.2. 对冲购买交易后的费用科目 | 106 |
| 7.3. 跟踪信用卡账目前的初始科目结构 | 107 |
| 7.4. 初始的信用卡购买 | 108 |
| 7.5. 信用卡退款交易 | 109 |
| 7.6. 信用卡收取的利息 | 110 |
| 7.7. 信用卡初始对账窗口 | 110 |
| 7.8. 存在差额的对账主窗口 | 111 |
| 7.9. 对账和还款后的信用卡科目 | 112 |
| 7.10. 对账和还款后的科目表 | 113 |
| 7.11. 3月份的现金流量表 | 114 |
| 7.12. 3、4月间Visa科目的交易汇总表 | 115 |
| 7.13. 4月期间费用科目的交易汇总表 | 116 |
| 8.1. 贷款还款计算器 | 119 |
| 8.2. 收到贷款前的科目 | 123 |
| 8.3. 房屋按揭贷款分录交易 | 124 |
| 8.4. 房屋按揭贷款科目 | 125 |
| 8.5. 计算私人借款的详细信息 | 127 |
| 8.6. 借出钱款后的科目表 | 128 |
| 8.7. 第一笔还款 | 129 |
| 8.8. 第二笔还款 | 130 |
| 8.9. 第二笔还款后的科目表 | 131 |
| 9.1. 只包含默认投资科目的新文件 | 138 |
| 9.2. 投资存单后的科目表 | 141 |

| | |
|--|-----|
| 9.3. 6个月后的存单科目 | 142 |
| 9.4. 6个月后的储蓄存款科目 | 142 |
| 9.5. 6个月后的科目表 | 143 |
| 9.6. 浪潮信息的“新建科目”窗口 | 145 |
| 9.7. [选择证券]窗口 | 146 |
| 9.8. “新建证券”窗口 | 147 |
| 9.9. 创建第一个股票科目后的科目表 | 148 |
| 9.10. 录入第一笔购入后的浪潮信息科目账簿 | 149 |
| 9.11. 第一次买入股票后科大讯飞科目的交易账簿 | 150 |
| 9.12. 第一次买入股票并支付佣金后的科大讯飞科目 | 151 |
| 9.13. 向数据库录入价格 | 152 |
| 9.14. 列出全部已知商品的价格数据库。 | 153 |
| 9.15. 在科目表中显示股票价值 | 154 |
| 9.16. 在报表中设置股票价格或货币汇率的来源 | 155 |
| 9.17. 使用“时间临近”价格来源的资产柱状图报表 | 156 |
| 9.18. 卖出份额、合并获利、毛价格 | 159 |
| 9.19. 获利卖出份额后的科目树 | 160 |
| 9.20. 在交易日志视图中使用净价格卖出份额并获利 | 161 |
| 9.21. 卖出份额获利，分离卖出份额与资本利得交易，使用毛价格，以[交易日志]视图显示 | 163 |
| 9.22. 卖出份额获利，分离卖出份额与资本利得交易，使用净价格，以[交易日志]视图显示 | 164 |
| 9.23. 卖出份额 - 资本利得 - 科目批次窗口 | 165 |
| 9.24. 卖出金额 - 资本得利 - 对消单个批次前的证券账簿 | 168 |
| 9.25. 卖出份额 - 资本得利 - 对消单个批次前的批次列表 | 169 |
| 9.26. 卖出份额 - 资本利得 - 手动关联分录后，对消单个批次前的分录列表。 | 169 |
| 9.27. 卖出份额 - 资本利得 - 手动关联并对消单个批次后的科目账簿 | 170 |
| 9.28. 卖出份额 - 资本利得 - 对消整个科目前的科目账簿 | 171 |
| 9.29. 卖出份额 - 资本利得 - 对消整个科目前的批次列表 | 172 |
| 9.30. 卖出份额 - 资本利得 - 对消整个科目后的批次列表 | 172 |
| 9.31. 卖出份额 - 资本利得 - 对消整个科目后的账簿 | 173 |
| 9.32. 卖出份额 - 资本利得 - 对消整个科目后的账簿 | 174 |
| 9.33. 卖出份额 - 资本利得 - 对消整个科目后的批次0 | 174 |
| 9.34. 卖出份额 - 资本利得 - 对消整个科目后的批次1 | 175 |
| 9.35. 卖出份额 - 资本利得 - 对消单个批次后的账簿 | 176 |
| 9.36. 现金分红交易示例 | 177 |
| 9.37. 红利再投交易示例 | 177 |
| 9.38. 资本退回交易示例 | 178 |
| 9.39. 启动“股票拆分”向导 | 179 |
| 9.40. 在“股票拆分”向导中选择股票科目 | 179 |
| 9.41. “股票拆分”向导中的拆分详情页面。 | 180 |
| 9.42. “股票拆分”向导中的“现金代付” | 181 |
| 9.43. 简单拆股后的重庆燃气科目 | 181 |

| | |
|--|-----|
| 9.44. 在“股票拆分”向导中选择股票科目 | 182 |
| 9.45. “股票拆分”向导中的拆分详情页面 | 183 |
| 9.46. 较为复杂的股票拆分后的中国北车科目 | 183 |
| 10.1. The Investment Lots Report | 189 |
| 11.1. 购入资产后的主窗口 | 205 |
| 11.2. 记录未实现损益后的主窗口 | 206 |
| 11.3. 记录已实现损益后的主窗口 | 207 |
| 12.1. 最初建立的多货币银行科目 | 212 |
| 12.2. 最初建立的多货币银行科目 | 214 |
| 12.3. 价格数据库窗口目前仍然为空 | 215 |
| 12.4. 设定欧元汇率 | 216 |
| 12.5. 设定欧元与人民币汇率后的价格数据库窗口 | 217 |
| 12.6. 设定欧元与人民币汇率后的科目表 | 218 |
| 12.7. 获取在线报价前的价格数据库 | 219 |
| 12.8. 获取在线报价后的价格数据库 | 220 |
| 12.9. 获取在线报价后的科目表 | 221 |
| 12.10. “外币转账至越南” | 223 |
| 12.11. 编辑汇率 | 223 |
| 12.12. 购入船只前的科目表 | 224 |
| 12.13. 购入船只后的科目表 | 224 |
| 12.14. 适用跨境股票交易的科目表 | 225 |
| 12.15. 境外证券 | 226 |
| 12.16. 资金转账 | 227 |
| 12.17. 买入的境外股票 | 227 |
| 12.18. 多货币银行科目初始结构 | 231 |
| 12.19. 价格数据库初始记录 | 231 |
| 12.20. 录入人民币、港元和欧元间汇率后的初始多货币科目结构 | 232 |
| 12.21. 创建从人民币到港元交易的转账对话框 | 233 |
| 12.22. 转账后显示在中国银行科目账簿的交易 | 234 |
| 12.23. 添加兑汇手续费分录后的交易 | 234 |
| 12.24. 完成从人民币到港元转账后的科目标签页余额。 | 235 |
| 12.25. 从香港银行到中国银行的资金转账交易 | 236 |
| 12.26. 资金转回中国银行科目后的价格数据库 | 236 |
| 12.27. 转回资金后的科目标签页 | 237 |
| 12.28. 进行上述交易后的资产负债表 | 238 |
| 13.1. Sales Tax Tables Editor | 244 |
| 13.2. New Sales Tax Table Entry | 245 |
| 13.3. Entering Company Information | 247 |
| 13.4. Billing Terms Editor | 249 |
| 13.5. New Billing Term | 249 |
| 13.6. New Customer Registration Window | 251 |
| 13.7. Find Customer Window | 252 |
| 13.8. Creating a New Invoice | 253 |

| | |
|--|-----|
| 13.9. Edit Invoice Window | 254 |
| 13.10. Post Invoice Window | 254 |
| 13.11. Invoice Print Output | 255 |
| 13.12. New Customer Job | 256 |
| 13.13. Process Payment From Customer | 257 |
| 13.14. New Style Sheet Window | 258 |
| 13.15. HTML Style Sheet Example Window | 259 |
| 13.16. HTML Style Sheets Example Output | 260 |
| 13.17. New Vendor Registration Window | 262 |
| 13.18. Find Vendor Window | 263 |
| 13.19. New Bill Registration Window | 264 |
| 13.20. Edit Bill Window | 265 |
| 13.21. Post Bill Window | 265 |
| 13.22. New Vendor Job | 266 |
| 13.23. Process Payment To Vendor | 267 |
| 13.24. Payroll Example: Initial Setup | 271 |
| 13.25. Payroll Example: Employee Split Transaction | 272 |
| 13.26. Payroll Example: Accounts After Salaries Paid | 273 |
| 13.27. Payroll Example: Accounts After Paying Government | 274 |
| 16.1. Asset Accounts Before Depreciation | 292 |
| 16.2. Asset Depreciation In The Register | 293 |
| 16.3. Accounts Tab After Asset Depreciation | 293 |

表格清单

| | |
|--|-----|
| 2.1. 借与贷对五类科目类型余额的影响 | 12 |
| 2.2. 存储格式对比 | 21 |
| 2.3. 应用设置的存储位置 | 27 |
| 2.4. 已保存报告的存储位置 | 27 |
| 2.5. 网上银行设置的存储位置 | 27 |
| 8.1. 购买房屋的分录交易 | 123 |
| 8.2. 给张三的个人借款 | 127 |
| 8.3. 获利出售资产（房屋） | 132 |
| 8.4. 亏损出售资产（房屋） | 133 |
| 8.5. 获利出售资产（房屋）之一 | 133 |
| 8.6. 获利出售资产（房屋）之二 | 134 |
| 9.1. 卖出份额分录方案：合并卖出份额与资本损益交易、毛价格 | 158 |
| 9.2. 卖出份额分录示例：合并卖出与获利、毛价格 | 159 |
| 9.3. 卖出份额分录方案：合并卖出份额与资本损益交易、净价格 | 160 |
| 9.4. 卖出份额分录示例：合并卖出与获利、净价格 | 161 |
| 9.5. 卖出份额分录方案：分离卖出份额与资本损益交易——卖出份额交易 | 161 |
| 9.6. 卖出份额分录方案：分离卖出份额与资本损益交易——资本损益交易 | 162 |
| 9.7. 卖出份额分录示例：卖出份额交易、毛价格 | 162 |
| 9.8. 卖出份额分录示例：资本利得交易、毛价格 | 162 |
| 9.9. 卖出份额分录方案：分离卖出份额与资本损益交易——卖出份额交易 | 163 |
| 9.10. 卖出份额分录方案：分离卖出份额与资本损益交易——资本损益交易 | 163 |
| 9.11. 卖出份额分录示例：卖出份额交易、净价格 | 164 |
| 9.12. 卖出份额分录示例：资本利得交易、净价格 | 164 |
| 9.13. 第一项交易处理收到的金额与份额的减少 | 167 |
| 9.14. 第二项交易记录资本损益（本例中为亏损） | 168 |
| 10.1. Sample Chart of Accounts for Income and GST Statement | 197 |
| 10.2. Income and GST Statement | 198 |
| 10.3. Sample Chart of Accounts for the Transaction Report | 200 |
| 11.1. 将未实现损益调整为已实现损益——估计准确的情况 | 206 |
| 11.2. 将未实现损益调整为已实现损益——高估的情况 | 207 |
| 11.3. 将未实现损益调整为已实现损益——低估的情况 | 208 |
| 12.1. 使用分录交易方案卖出外币 | 229 |
| 13.1. Sample Tax Table Entries for EU country (e.g. 21% / 6% / 0% Belgium, 20% / 5% / 0% UK etc.) (2017) | 245 |
| 13.2. Sample Tax Table Entries for Australia (2017) | 246 |
| 13.3. Sample Tax Table Entries for Cook County, Illinois (2017) | 246 |
| 13.4. Payroll Transaction Map | 269 |
| 13.5. Payroll Transaction Map for Employee 1 | 272 |
| 16.1. Linear Depreciation Scheme Example | 289 |
| 16.2. Geometric Depreciation Scheme Example | 289 |

| | |
|---|-----|
| 16.3. Sum of Digits Depreciation Scheme Example | 290 |
| B.1. Overall Page Description Fields | 318 |
| B.2. Multiple Checks Per Page Fields | 322 |
| B.3. Individual Check Item Fields | 324 |
| B.4. Individual Check Item Types | 328 |

范例清单

| | |
|--------------------------------------|-----|
| 8.1. 用于记录给张三借款的科目 | 127 |
| 9.1. 卖出份额获利：手动计算损益、合并卖出与获利、毛价格 | 158 |
| 9.2. 卖出份额获利：手动计算损益、合并卖出与获利、净价格 | 160 |
| 9.3. 卖出份额获利：手动计算损益、分离卖出与获利、毛价格 | 162 |
| 9.4. 卖出份额获利：手动计算损益、分离卖出与获利、净价格 | 163 |
| 9.5. 手动创建并关联批次 | 168 |
| 9.6. 自动创建资本损益交易 | 170 |
| 9.7. 自动创建资本损益交易，一次进行两笔卖出。 | 171 |
| 9.8. 自动创建资本利得交易 - 简单的拆股之后 | 173 |
| 9.9. 将孤立的收益-CCC更改为损益科目 | 175 |

公式清单

| | |
|-----------------------|----|
| 2.1. 静态会计等式 | 11 |
| 2.2. 动态会计等式 | 12 |
| 2.3. 重新排列后的会计等式 | 12 |

关于作者

GnuCash文档团队由一群自告奋勇的GnuCash用户和开发者组成，他们自愿编写本文档，为更广泛的用户群体带去便利。我们欢迎有志于加入文档团队的用户在GnuCash邮件列表（<gnucash-users>和<gnucash-devel>）中表达自己的兴趣，并获得下一步的指引。

目前文档团队的贡献者有： Yawar Amin , J. Alex Aycinena , Tom Bullock , Carol Champagne , Frank H. Ellenberger , Mike Evans , Chris Good , Dave Herman , Geert Janssens , Jon Lapham , Chris Lyttle , Cristian Marchi , Brent McBride , John Ralls , Robert Ratliff , Christian Stimming , David T. , Bengt Thuree .

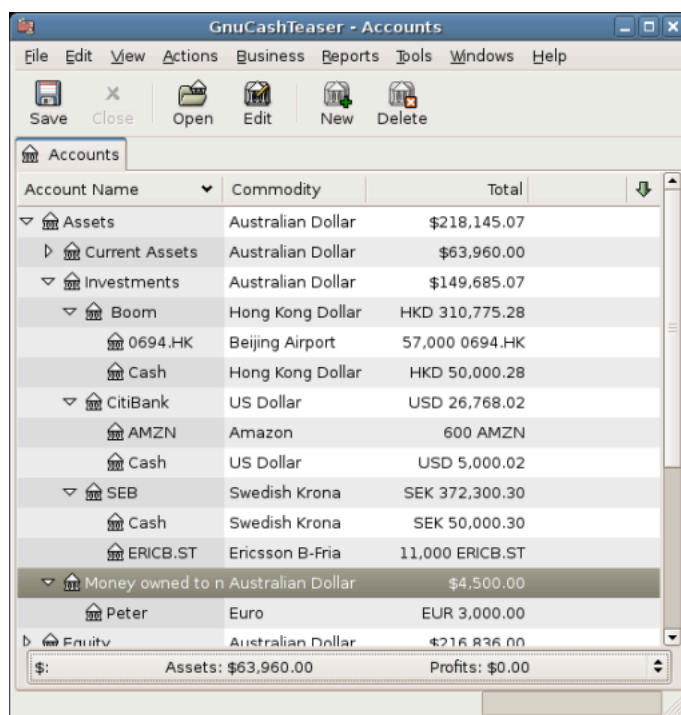
第 1 章 概览

介绍

GnuCash是为每个人设计的个人财务软件包。从简单的个人财务数据到复杂的企业财务管理，它都是跟踪财务信息的全能选手。它是为数不多的几款支持全球货币的财务软件之一，也是其中唯一的开源软件。不过，GnuCash最大的优点，还是易于学习使用。

所以，GnuCash可以帮你哪些忙呢？它可以跟踪你的个人财务数据，想要多么详细都可以。如果你刚开始使用，可以使用GnuCash跟踪你的日常花销。随后，你可能决定使用GnuCash跟踪现金收支和信用卡记录看看钱到底花哪儿了。在你开始投资理财后，可以使用GnuCash监控你的投资组合。买车或者买房子了？GnuCash会帮你安排投资计划，跟踪还款记录。如果你的财务记录跨越多个国家，GnuCash也提供你所需的多货币功能。

1.1.



| GnuCashTeaser - Accounts | | |
|--|---------------------|-----------------|
| File Edit View Actions Business Reports Tools Windows Help | | |
| Save Close Open Edit New Delete | | |
| Accounts | | |
| Account Name | Commodity | Total |
| Assets | Australian Dollar | \$218,145.07 |
| Current Assets | Australian Dollar | \$63,960.00 |
| Investments | Australian Dollar | \$149,685.07 |
| Boom | Hong Kong Dollar | HKD 310,775.28 |
| 0694.HK | Beijing Airport | 57,000 0694.HK |
| Cash | Hong Kong Dollar | HKD 50,000.28 |
| CitiBank | US Dollar | USD 26,768.02 |
| AMZN | Amazon | 600 AMZN |
| Cash | US Dollar | USD 5,000.02 |
| SEB | Swedish Krona | SEK 372,300.30 |
| Cash | Swedish Krona | SEK 50,000.30 |
| ERICB.ST | Ericsson B-Fria | 11,000 ERICB.ST |
| Money owned to n | Australian Dollar | \$4,500.00 |
| Peter | Euro | EUR 3,000.00 |
| Equity | Australian Dollar | \$218,145.07 |
| \$: | Assets: \$63,960.00 | Profits: \$0.00 |

此图展示了包含多种货币和投资类型的科目表。

尽管GnuCash十分适合个人财务管理，但对企业用户而言也是相当强大。从内置的应收账款与应付账款科目系统，到税务报表生成，GnuCash同样包含许多企业功能。你会发现GnuCash提供的许多企业功能强大而易用。

特点

易于使用

几分钟之内，你就可以录入你的个人财务数据并生成展示财务状况的美观图表。要是你会使用支票簿后面的账簿，那你就会使用GnuCash！直接向账簿输入内容，在不同列间跳转，还可以使用快速填充自动补全交易记录。上述操作方式也可以在软件内自行修改。

- 易用的菜单： GnuCash的菜单符合GNOME人机交互指南，也就是说，其菜单简单易用，并与其它GNOME软件外观相近。
- 文档： GnuCash内置帮助手册，以及内容更为丰富的用户指南。
- 数据导入方法： 除了手动输入，GnuCash还支持许多自动导入交易记录的方式。要是你可以在线访问银行账户，这一特性就特别实用，因为多数银行和信用卡公司都支持下述导入方式中的至少一种。如此一来，少一些时间录入数据，就可以多一些时间分析统计结果。
- Quicken导入文件（QIF）： Quicken QIF 风格的文件。这是一种在许多流行的个人财务商业软件间流行的文件格式。
- 开放式金融交换（OFX）： GnuCash是第一款支持开放式金融交换协议的自由软件。许多金融机构正在迁移至这一格式。
- 家庭银行业务计算机接口（HBCI）： GnuCash是第一款支持德国国家家庭银行业务计算机接口协议的自由软件。该协议可用于下载账单、发起银行转账，并使自动扣款成为可能。
- 报表： GnuCash预置超过30款报表，包括科目汇总表、收入表、费用表、交易汇总表、资产负债表、损益表、投资组合估值，以及许多其它报表。报表能够绘出数据图表，包括饼图、条形图和散点图。报表可以导出为HTML文件，也可以方便地自行修改。
- 计划交易： GnuCash现在支持自动创建并录入交易，或提醒你什么时候将要创建这些交易，并由你决定录入、推迟还是删除自动计划交易。
- 贷款还款助手： 用于计算各式各样的贷款偿还计划交易。
- 轻松实现科目对账： GnuCash内置对账功能，将GnuCash科目同银行账单对账现在简单又高效。
- 多平台兼容： 在多种平台和操作系统上都可以使用GnuCash。完全支持的操作系统和平台有：

GnuCash \geq 3.0

Linux (x86, x86_64)、FreeBSD (x86, x86_64)、OpenBSD (x86, x86_64)、macOS (64-bit Intel), 和 Windows (Windows 7 或更高版本).

GnuCash \leq 2.6.21

macOS (32-bit Intel, PPC) 和 Windows (XP and Vista).

GnuCash ≤ 2.2.9
Windows (2000).

更早版本的GnuCash

目前已知可以在SGI IRIX (MIPS)、IBM AIX 4.1.5 (RS/6000)、Unixware 7 (Intel)、SCO OpenServer 5.0.4 (Intel) 和 Solaris (SPARC) 上使用，但各平台目前的情况不确定。

跟踪投资

GnuCash包含许多投资功能，可以帮助你跟踪每一笔投资。使用GnuCash，你可以独立地（一支股票对应一个科目），或是以科目组合的形式（可以一并显示的一组科目）跟踪股票。

GnuCash支持在线获取股票和基金报价。这样一来，就再也不必逐支股票地查询报价。此过程可以自动执行，这样你就可以看到最新的股价了。

国际化支持

GnuCash是一款可以为来自全世界用户提供帮助的软件。其中内置的许多功能能够方便你在当下国际化的世界中游刃有余。

- 本地语言：GnuCash已被翻译为超过55种语言。被完整翻译的有，繁体中文、克罗地亚语、荷兰语、希伯来语、意大利语、葡萄牙语和乌克兰语。一些语言的翻译工作进展同样可喜，已有九成内容得到翻译，有，加泰罗尼亚语、简体中文、德语、印度尼西亚语、日语、葡萄牙语（巴西）和西班牙语。¹
- 国际化格式处理：GnuCash能够识别和处理不同国家书写日期与数字的格式。你可以使用你习惯的日期与数字格式。
- 多货币兑换：GnuCash支持ISO 4217 [https://en.wikipedia.org/wiki/ISO_4217]中列出的所有货币，既可以购汇也可以结汇（兑换）。如果“贸易科目”功能开启，科目间货币的兑换会完全保持平衡。
- 在线汇率：使用GnuCash，你再也不必一次次地查询货币汇率。此过程可自动执行，这样一来，GnuCash总是可以为你展示最新汇率下以你常用货币计算的科目余额。

企业功能支持

GnuCash中的许多功能满足企业会计需求。

- 应收账款与应付账款科目：GnuCash内置应收账款与应付账款科目系统。可在小型企业中用于跟踪客户、供应商、发票和账单缴付，并自定义不同的税费和计费期限。
- 折旧：GnuCash能够管理固定资产折旧。

¹如果你想开展或完善当地语言的翻译工作，可见 <https://wiki.gnucash.org/wiki/Translation>.

- 报表：GnuCash能够随时生成各式各样、能够直接拿来使用的企业级报表。

会计特性

了解会计工作的用户可能想要了解下列GnuCash会计特性。

- 复式记账： 每一项交易必须在一些科目中记借并在其它科目中以相同的金额记贷。这一规则确保了会计平衡——收入与费用间的差额与所有银行存款、股票和其它资产的总额确切相等。
- 分录交易： 一项交易可被分拆为几部分，用于记录税费、服务费和其它分录的组合。
- 科目表： 主科目可以下设子科目以构成层次结构。这一特性可以将现金、银行、股票等相似的科目类型编组为“资产”这样的主科目。
- 通用日记账： 这是一种可以同时显示多个科目的账簿窗口。该功能使查出录入错误更加简单。它也可以一同显示由许多股票组成的投资组合下的所有交易，以便于浏览。
- 收入、花费科目类型： 这几种科目类型将你的现金流归类，当合理使用复式记账特性时能够给出准确的损益表。

关于本指南

本指南将尽快帮助你上手使用GnuCash。

每一章都以清晰明了的结构展开。每一章开始是对“概念”的讨论，介绍整章涉及的核心主题和术语。接着，是“怎么做”小节，介绍具体的操作过程。末尾，是“步步为营”小节，给出详尽具体的操作示例。

新用户会发现“概念”小节特别实用。这些小节会展示跟踪个人财务的优秀实践，也会简要介绍财务背景知识和术语。有经验的用户可以直接跳到“怎么做”小节快速浏览操作步骤。这两个小节简明扼要地列出了完成具体任务的步骤。“步步为营”小节以操作说明的形式给出了实际场景下的使用示例。在按第 2 章 基础所述创建文件之后，后一章都在前一章操作说明的基础上更进一步。

本手册主要分为三大部分：

- 第 I 部分 “基础知识”
- 第 II 部分 “常规使用”
- 第 III 部分 “高级功能”

第 I 部分 “基础知识” 提供开始使用GnuCash所需的最基本信息。这部分章节介绍使用GnuCash的核心概念。GnuCash的新用户应当熟悉这部分章节讲解的内容，以尽快上手并达到熟练使用GnuCash的水平。

- 第 1 章 概览 ——（也就是你在读的这一章）大致介绍了GnuCash的基本情况。

- 第 2 章 基础 —— 这一章简要地介绍了会计准则，并介绍了GnuCash如何组织其数据。另外也说明了GnuCash的基本界面构成。最后，此章解释了GnuCash如何存储与管理你的数据。

- “科目”一节 —— 更具体地介绍了科目的概念，以及组织科目的方法。
- “交易”一节 —— 介绍了交易的基本概念，以及录入交易的方法。

第 II 部分 “常规使用” 更加详细地介绍了GnuCash的常见功能与应用。其中提供了更详尽的使用示例，这些示例都是根据日常场景中应用GnuCash时常常会遇到的问题给出的。下面是这一部分涉及的功能与应用：

- 第 5 章 活期账户与现金
- 第 7 章 信用卡
- 第 8 章 贷款
- 第 9 章 投资
- 第 11 章 资本利得
- 第 12 章 多货币

第 III 部分 “高级功能” 讨论了在企业会计事务中使用GnuCash

- 第 13 章 Business Features
- 第 14 章 Budgets
- 第 15 章 Other Assets
- 第 16 章 Depreciation
- 第 17 章 Python Bindings
- 第 18 章 Importing Business Data

手册也包含几节附录，或许涉及你想要了解的其它信息。

- GnuCash 术语表 —— GnuCash中使用的术语列表
- 附录 A, Migration Guide —— 针对前 Quicken®、MS Money 或其它 QIF用户的指南。
- 附录 B, Auxiliary File Formats
- 附录 C, GNU Free Documentation License

最后值得一提的是，术语表和索引有助于快速查找你想要了解的主题。

获取帮助

GnuCash提供多种形式的帮助指引，包括〔每日提示〕、上下文帮助、网站、维基、邮件列表和IRC聊天频道。

上下文帮助

上下文帮助为使用GnuCash中的菜单、窗口和按钮提供了详细的指引。要打开帮助，选择〔帮助〕→〔内容〕。

〔GnuCash帮助〕窗口也能用作简单的网络浏览器，因此你可以转到网站进一步查阅信息。点击打开 工具栏按钮并输入URL，就可以在此窗口中打开任何网站。你可以就像在标准的网络浏览器中一样使用后退、前进、刷新、停止和打印按钮。

GnuCash网站

GnuCash网站 [<https://www.gnucash.org/>] 包含有关程序及其软件更新的帮助信息。其中也包含其它在线资源的链接。

GnuCash维基

数目可观的不那么正式的文档，GnuCash本身及其维护开发工作有关的内容都可以在GnuCash维基 [<https://wiki.gnucash.org/wiki/>]上找到。

GnuCash常见问题

GnuCash常见问题 [<https://wiki.gnucash.org/wiki/FAQ>]（FAQ）页面也位于GnuCash维基上。每当你在使用GnuCash时遇到困难，都应当先查阅常见问题。现在它也被翻译为德语和简体中文。如果你想要访问翻译页面，请查看GnuCash常见问题 [<https://wiki.gnucash.org/wiki/FAQ>]页面顶部的标签页。

GnuCash在线帮助

邮件列表

最重要的用户支持就来自于用户邮件列表 [<mailto:gnucash-user@gnucash.org>]。你如果更想使用论坛的形式，可以通过Nabble [<http://gnucash.1415818.n4.nabble.com/GnuCash-f1415818.subapps.html>] 使用。即使是使用Nabble，发表讨论也需要先订阅。

IRC

几位开发者持续关注irc.gimp.net上的 #gnucash 频道 [<irc://irc.gimp.net/gnucash>]。他们通常忙于其它事务，当然也不是时时刻刻坐在电脑前。请登录、询问你的问题、保持登录。过几个小时，你的问题就会被留意到并得到回应。

· 如果你失去连接，请检查IRC记录 [<https://lists.gnucash.org/logs/>]确认你没有遗漏相关消息。

· IRC维基页面 [<https://wiki.gnucash.org/wiki/IRC>] 介绍了前提要求、发言准则和其它详细信息。

GnuCash网站 [<https://www.gnucash.org/>]详细介绍了这些频道。也能在上面找到指向GnuCash维基、软件缺陷跟踪系统等更多有用资源的链接。

主题搜索

在线手册也提供了搜索功能。要搜索特定的主题，点击帮助窗口底部的〔搜索〕标签页，并在输入框中输入要搜索的主题。点击搜索按钮执行搜索。搜索结果会出现在下方的列表中，点击其中一项搜索结果就会在右侧显示相应的帮助内容。

安装

GnuCash安装起来一般很简单。

GnuCash下载页 [<https://www.gnucash.org/download.phtml>]为每一款支持的操作系统提供了详细的安装指引。

部分 I. 基础知识

目录

| | |
|-----------------------------|----|
| 2. 基础 | 11 |
| 会计概念 | 11 |
| 五类基本科目 | 11 |
| 会计等式 | 11 |
| 借贷记账法 | 12 |
| 复式记账法 | 13 |
| 数据记录概念 | 13 |
| 文件 | 13 |
| 科目 | 14 |
| 交易 | 14 |
| 运行GnuCash | 15 |
| 〔欢迎使用GnuCash〕对话框 | 15 |
| 建立科目结构 | 15 |
| 每日提示 | 16 |
| 科目树窗口 | 17 |
| 科目账簿窗口 | 18 |
| 工具栏按钮 | 19 |
| 标签栏 | 19 |
| 菜单项 | 20 |
| 菜单的访问键与快捷键 | 20 |
| 存储财务数据 | 20 |
| 概述 | 20 |
| 存储压缩及有关建议 | 21 |
| 存储格式对比表 | 21 |
| 创建文件 | 21 |
| 保存数据 | 22 |
| 打开数据 | 24 |
| 复制科目层次结构 | 24 |
| 备份与恢复数据 | 24 |
| 备份文件（".gnucash"） | 25 |
| 日志文件（".log"） | 25 |
| 锁定文件（".LNK" 和 ".LCK"） | 25 |
| 文件管理 | 26 |
| 迁移GnuCash数据 | 26 |
| 迁移财务数据 | 26 |
| 迁移首选项数据 | 26 |
| 步步为营 | 28 |
| 科目 | 34 |
| 基本会计概念 | 34 |
| GnuCash 科目 | 35 |

| | |
|--|----|
| 步步为营 | 39 |
| 交易 | 43 |
| 基本概念 | 43 |
| 科目账簿 | 43 |
| 简单交易还是分录交易? | 48 |
| 对账 | 53 |
| 计划交易 | 58 |
| 步步为营 | 68 |
| 3. Importing Data into GnuCash | 75 |
| Importing Transactions from Files | 75 |
| File Import Formats | 75 |
| Import Matcher | 75 |
| Import QIF | 77 |
| Import OFX/QFX | 78 |
| Import CSV | 78 |
| Import MT940 | 81 |
| Import MT942 | 81 |
| Import DTAUS | 81 |
| 4. Configuring GnuCash | 82 |
| Overview | 82 |
| Main Configuration Screens | 82 |
| Preferences: Accounting Period Tab | 82 |
| Preferences: Accounts Tab | 82 |
| Preferences: Online Banking | 82 |

第 2 章 基础

本章将介绍使用GnuCash时需要了解的基本概念。推荐你在上手使用GnuCash前通读本章。

会计概念

GnuCash简单易用，你不必完全理解会计准则就可以上手使用。然而，掌握一些基本的会计知识还是值得的，因为GnuCash正是基于这些准则设计的。强烈建议你在上手使用前理解指南的这一部分。

五类基本科目

基本会计规则将财务上的所有事项分为五类基本“科目”类型。也就是说，会计要处理的一切事项均属于这五类科目之一：

资产

你自己拥有的东西

负债

你欠别人的东西

所有者权益

总净值

收入

科目中增加的价值

费用

科目中减少的价值

很明显，你可以将财务中一切事项分为五组。例如，银行账户现金是资产，房屋贷款是负债，工资是收入，昨天晚饭的开销是费用。

会计等式

现在有了五类基本科目，但它们之间的关系是什么？其中一类科目怎样影响其它科目？首先，定义一个包含资产和负债的等式：净值等于资产减去负债。

2.1.

资产 - 负债 = 所有者权益

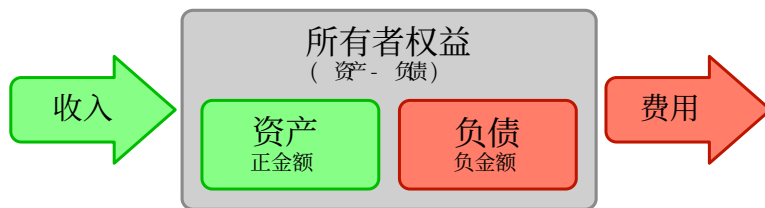
另外，收入可以增加净值，费用可以减少净值。这当然有道理，当你得到工资你就变得“更富有”，当你为晚饭付钱你就变得“更贫穷”。这就以数学的方式表达了一个被人们称为会计等式的关系：

2.2.

$$\text{资产} - \text{负债} = \text{所有者权益} + (\text{收入} - \text{费用})$$

此等式应当永远成立，这意味着只有同时变动多个科目的金额时才能满足要求。例如，如果以收入的形式得到了一笔钱，则必须等额地增加资产科目的金额。又如，如果负债科目的金额增加，则应当同时增加资产科目的金额。

2.1.



五类基本科目关系图。净值（所有者权益）随收入增加和费用减少而增加。箭头代表了金额的转移。

借贷记账法

初次接触会计工作的人常常不明白会计中“借”与“贷”的用法，及其对各类科目的作用。上文介绍的会计等式是理解哪类科目在什么情况下记借或记贷的关键。首先，展开并重新排列会计等式，使资产位于等号的左边，并将任何带有负号的科目移到另一侧，这样可以得到：

2.3.

$$\text{资产} + \text{费用} = \text{负债} + \text{所有者权益} + \text{收入}$$

此形式的会计等式中，等号左边的科目称为借方科目，记借的分录会增加这些科目的余额。相反，记贷的分录会减少这些科目的余额。

类似地，在等号右边的科目也被称为贷方科目，即这些科目的余额由记贷的分录增加。同样地，这些科目中记借的分录会减少科目的余额。

借与贷是专用的会计术语，这两个字在会计工作中的含义与日常生活中的用法并不一致。

2.1.

| | (Dr) | (Cr) |
|----|------|------|
| 资产 | 增加 | 减少 |

| | | |
|-------|------|------|
| | | |
| | (Dr) | (Cr) |
| 费用 | | |
| 负债 | 减少 | 增加 |
| 所有者权益 | | |
| 收入 | | |

复式记账法

会计等式是复式记账会计制度的核心。当会计等式中一个科目的余额变动时，必须对等地变动另一个科目。这一概念也被称为平衡原则，它是理解GnuCash和其它复式记账会计制度的基础。当使用GnuCash时，为了保持会计等式的平衡，总要同时关注至少两个科目。

账户间的平衡变动（或资金转移）是通过借记一个账户同时贷记另一个账户来完成的。会计上的借与贷，并不意味着“减少”与“增加”。正如前一部分所述，借与贷都会在增加某些科目的余额的同时，减少其它科目的余额。在资产和费用科目中，借会增加余额，贷会减少余额。在负债、所有者权益和收入科目中，贷会增加余额，借会减少余额。

在传统的复式记账中，账簿左边的一列记借，右边的一列记贷。会计师会在记借的一侧（左侧）记录资产和费用科目的增加，在记贷的一侧（右侧）记录负债、收入和所有者权益的增加。GnuCash遵循这一使用账簿记账时的惯例。

这些会计术语可能会令新用户感到困惑，因此GnuCash默认使用更常见的术语〔存入〕和〔取出〕。如果你希望使用正式的会计术语，可以在〔首选项〕（更多有关设置首选项的信息请查阅GnuCash帮助手册）中的〔科目〕选项卡进行设置。

在日常生活中，借与贷这两个字的用法和会计工作中的用法并不一致。在日常使用中，贷通常有增加的意思；在会计工作中，贷意味着影响某一科目分类账表的右列。这与资产和费用科目的减少相关，与收入、负债和所有者权益科目的增加相关。

详细信息可参阅https://zh.wikipedia.org/wiki/Debits_and_credits。

数据记录概念

在GnuCash中录入数据时，你应当了解GnuCash用于分隔数据的三级组织结构：文件、科目和交易。这几个层级按其复杂性排列出，一个文件包含许多科目，一个科目包含许多交易。认识这种分隔方法是理解怎样使用GnuCash的基础。

文件

GnuCash以文件形式存储最高层级的信息。一个文件以单个XML文件（在所有版本的GnuCash上）或SQL数据库（2.4或更高版本的GnuCash上）的形式存储在你的计算机上。

SQL发音为“sequel”，所常写作和念作“SQL数据库”。

使用XML文件格式时，GnuCash会在单个XML文件中存储你的数据，文件通常使用压缩格式（此行为可以在GnuCash〔首选项〕下的〔通用〕选项卡中进行修改）。

使用SQL存储时，GnuCash会在你选择的数据库软件（SQLite3、MySQL或PostgreSQL）中把数据存储至SQL数据库中。

你需要为你正在处理的一组科目准备主文件或数据库。要了解怎样创建和管理GnuCash文件，参见“存储财务数据”一节。

你如果认为一组科目不能满足需求，则应在处理账目前咨询专业会计师。大多数用户仅需一个数据文件。

备份文件和日志文件由GnuCash适时自动生成。备份文件和日志文件会在“备份与恢复数据”一节详细阐述。

科目

科目跟踪你拥有的、你欠别人的、你花费的或你得到的东西。每个GnuCash文件可包含任意数量的科目，每个科目又可包含许多子科目，层数不限。正如下文所述，这个简单的特性使GnuCash能够管理你的财务的方方面面。

例如，科目可以包括：活期账户、储蓄账户、信用卡账户、房屋贷款和其它贷款。每个GnuCash科目跟踪对应的“实际”账户的活动，便于了解其状态。

此外，科目也可以把你收到或花出的钱分类。例如，你可以创建费用科目记录为公共服务和食品杂货付的钱。即便这些科目并不对应于能收到账单的实际账户，也仍可使你了解在各个方面花了多少钱。

本指南会在“科目”一节中更详细地介绍科目。

交易

交易代表资金在科目间的转移。任何你花出或收到钱，或在账户间转账的时候，都会产生交易。

例如，交易可以是：支付手机话费、把钱从储蓄账户转账到活期账户、买烧饼、取出现金和存入工资。“交易”一节会更深入地介绍怎样录入交易。

在复式记账中，交易总是至少涉及两个科目——一个付款科目，一个收款科目。根据这一规则，GnuCash会在交易中为每个受影响的科目各插入一行，并在每一行中记录涉及的金额。交易中记录涉及的科目与金额的行称为分录。每项交易包含的分录数量不限。

交易分录会在“分录交易示例”一节详细介绍。

运行GnuCash

可以选择桌面主菜单中的相应菜单项运行GnuCash。

也可以在命令行中执行gnucash来运行。

在启动过程中，GnuCash会显示欢迎屏幕，列出一些有关程序（版本号、构建号等）和加载过程的信息。

〔欢迎使用GnuCash〕对话框

最初打开GnuCash你会看到〔欢迎使用GnuCash〕对话框，其中包含三个选项：

- 〔创建一组新科目〕 —— 运行〔建立科目结构〕向导（参考“建立科目结构”一节）。如果你想GnuCash的帮助下创建一组新科目，请选择此选项。
- 〔导入我的QIF文件〕 —— 运行〔导入QIF文件〕向导（参考“Import QIF”一节）。如果你已经拥有Quicken文件（“.qif”文件）并想要将其导入GnuCash，请选择此选项。
- 〔打开新用户教程〕 —— 打开GnuCash教程与概念指南。如果你刚开始接触GnuCash和会计概念，请选择此选项。

离开此对话框后，你仍然可以使用其中列出的各项功能，但〔欢迎使用GnuCash〕不会再次显示。要创建一组新的科目，参见“建立科目结构”一节。要导入QIF文件，参见“Import QIF”一节。

建立科目结构

建立科目结构向导能够帮助你创建一组新的GnuCash科目。如果在〔欢迎使用GnuCash〕中选择创建一组新的科目，或选择〔文件(F)〕→〔新建(N)〕（**Ctrl** + **N**）时，该向导会启动。你也可以在〔科目〕页面中选择〔操作(A)〕→〔建立科目结构…〕启动向导往现有账簿添加一系列科目。

该向导会指引你创建会计科目表。此向导包含几个步骤，简要介绍如下：

1. 第一个页面简要介绍了该向导的功能。
2. 〔新建账簿选项〕只在创建新账簿时出现。你可以在此设置许多账簿属性。此页面包含四个选项卡：〔科目〕、〔预算〕、〔企业〕和〔计数器〕。这些条目会在本指南的其它部分中展开介绍，可以随时使用〔文件(F)〕→〔属性(T)〕（**Ctrl** + **Enter**）修改。
3. 〔选择货币〕设置根科目和新建科目的本位币。该选项的默认值根据计算机的区域设置确定。请认真设置此选项，因为它不仅会设定新建科目的默认货币，也会成为整个账簿的主货

币，还会是所有股票价格和外币汇率的默认货币。尽管可以通过〔操作〕菜单重新运行〔建立科目结构〕向导进行更改，但这样操作后你已经创建的交易可能会损坏。

4. 〔选择要创建的科目〕提供了多种可添加到账簿的会计模板。这里主要有两个面板：左侧是模板列表，每一项旁边有一个复选框；右侧显示将要由选中的模板添加到账簿的科目。

在〔模板〕列表上方，是两个下拉框，其中一个是语言代码，另一个是国家代码。初始选中项同样根据计算机区域设置确定。你可以根据需要选择任何语言和国家的组合。

〔建立科目结构〕向导中，不能一次同时选择多个区域下的模板，因此如果你需要从多个区域中选择多项（可能是因为打算在首选区域的模板之上使用来自其它国家的模板进行拓展），则可以从〔操作〕菜单多次运行此向导。

5. 〔设置选择的科目〕会以树状结构列出由在〔选择要创建的科目〕中选中的模板所给出的科目。根据你是否正在创建新的账簿，列表可能有四列或五列。

- 〔科目名称〕可编辑，只需点击内容文本，或选中该行按下〈Enter〉。
- 〔类型〕不可编辑。此项指示科目类型。
- 〔占位符〕科目。占位符科目指只能在其下建立子科目而自身不能添加交易的科目。你可以勾选此项使科目成为占位符，或取消勾选使科目为常规科目。
- 〔期初余额〕可以为非所有者权益的非占位符科目设定余额，因为已经存在的账户早就存在余额，占位符科目不能存储交易，且所有者权益科目没有期初余额。

〔期初余额〕输入框只有点击后才可见。你必须点击别处将光标移出输入框，或在点击前进按钮之前按〈Tab〉或〈Enter〉，否则输入的内容会丢失。

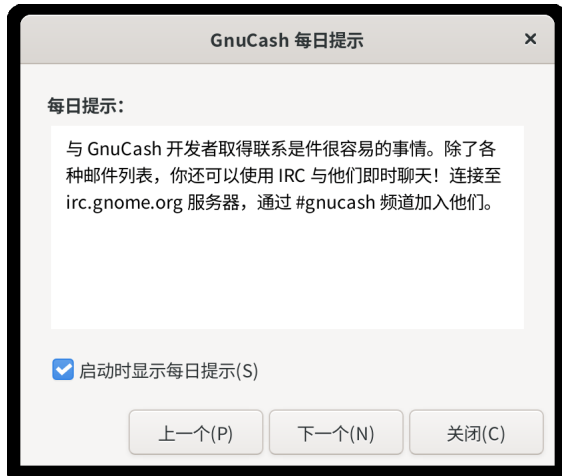
- 〔使用已有科目〕只会在从〔操作〕菜单启动〔建立科目结构〕向导时才会出现。它指示科目是否已经存在。此项不可编辑。
6. 〔完成科目设置〕是最后一个页面，给你最后的机会取消此过程，回头更改一些设置，或点击应用确认你的选择。如果你正在新建账簿，GnuCash会弹出对话框，提示选择存储格式，可以用以保存账簿的文件，另外如果GnuCash内含存储至数据库的功能且已经安装了必要的软件依赖，也可以输入数据库服务器地址、数据库名和认证信息以创建MySQL或PostgreSQL数据库。

你如果选择取消，则至此做出的任何选择都会丢失。

每日提示

GnuCash的〔每日提示〕屏幕会给出有助于使用GnuCash的提示。

2.2.



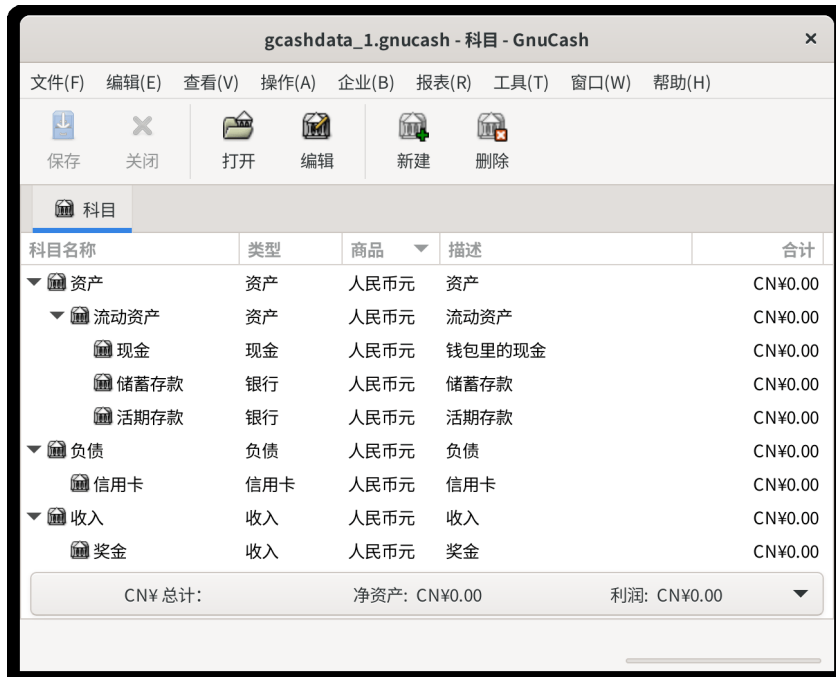
此图展示了〔每日提示〕屏幕。

这些提示对初学者很有帮助。可单击前进继续浏览更多提示。如果不希望在启动时看到此屏幕，请取消勾选旁边的〔启动时显示每日提示〕复选框。浏览完这些有用的提示后，点击关闭按钮关闭〔每日提示〕屏幕。

科目树窗口

你现在应该看到了〔科目〕窗口，如下所示。科目树的具体结构依赖于你在〔建立科目结构〕中选择了哪些默认科目。此处示例列出的是〔常用科目〕。

2.3.



此图展示了〔科目〕窗口。

科目树窗口（也称为科目表）汇总了当前文件所含数据，包含科目名称列表和其当前余额。

在此窗口中，你可以双击科目名称，或者右击科目名称并从菜单中选择〔打开账户〕，以及使用工具栏上的打开按钮打开科目账簿。可以在GnuCash中打开任意数量的科目账簿。参见“科目账簿窗口”一节了解使用科目账簿的更多信息。

如果科目下存在子科目，可以点击科目名称左侧的小三角展开显示其子科目树。

窗口顶端是标题栏，保存文件后就会显示此组科目保存的文件名。下方是菜单栏，可以点击菜单标题或使用快捷键（参见“菜单的访问键与快捷键”一节）访问菜单项。再下面是工具栏，包含调用常用功能的按钮。

科目树显示在工具栏下方。在你创建科目后，科目名称就会显示在科目树中。你可以使用科目树上方最右侧的小箭头设置要显示的表头。

底部是状态栏，告知有关你拥有多少财产（净资产）和你赚到了多少钱（利润）的信息。

科目账簿窗口

科目账簿窗口用于输入和编辑科目数据。恰如其名，科目账簿窗口看起来就像是纸制记账本。

2.4.

*gcashdata_4.7.gnucash - 活期存款 - GnuCash

文件(F) 编辑(E) 查看(V) 交易(N) 操作(A) 企业(B) 报表(R) 工具(T) 窗口(W) 帮助(H)

保存

关闭

创建副本

删除

确认

取消

转到空行

展开分录

科目

活期存款

| 日期 | 编号 | 描述 | 转账 | 对 | 借方 | 贷方 | 余额 |
|------------|-----|----------|--------------|---|----------|--------|----------|
| 2024-03-01 | | | 所有者权益:期初余额 | n | 1,000.00 | | 1,000.00 |
| 2024-03-05 | 101 | 超市采购 | 费用:食品杂货 | n | | 45.21 | 954.79 |
| 2024-03-05 | 105 | 家滋味便利店 | 费用:食品杂货 | n | | 75.00 | 879.79 |
| 2024-03-06 | | 定期存款到期 | 资产:流动资产:储蓄存款 | n | 100.00 | | 979.79 |
| 2024-03-06 | 106 | 甲乙丙五金店 | -- 复合分录 -- | n | | 100.00 | 879.79 |
| 2024-03-14 | | 发放工资 | -- 复合分录 -- | n | 670.00 | | 1,549.79 |
| 2024-03-24 | | 定期存款提前支取 | 资产:流动资产:储蓄存款 | n | 500.00 | | 2,049.79 |
| 2024-03-25 | 编号 | ATM取现 | 资产:流动资产:现金 | n | 借方 | 100.00 | 1,949.79 |
| 2024-03-28 | 102 | 国家电网 | 费用:水电费:电费 | n | | 78.00 | 1,871.79 |
| 2024-03-28 | 103 | 中国移动 | 费用:电话费 | n | | 45.00 | 1,826.79 |
| 2024-03-28 | 104 | 费用:房租 | 费用:房租 | n | | 350.00 | 1,476.79 |

现在: CN¥1,456.79 将来: CN¥1,106.79 已结清: CN¥0.00 已对账: CN¥0.00 预计最低: CN¥1,106.79 排序: 标准

此图展示了已经录入几笔交易后的活期账户账簿。

“交易”一节将进一步介绍科目账簿窗口和向里面录入数据的方法。这里，请注意科目账簿窗口的有些区域类似于科目树窗口。顶部的标题栏显示科目名。下方，菜单栏包含与科目账簿相关的菜单项。使用工具栏按钮可以快捷地访问常用的数据录入功能。窗口底部的状态栏显示当前科目余额，会在“交易”一节详细介绍。科目账簿窗口底部显示有关当前光标位置的信息。

在账簿窗口中，可以调整GnuCash显示的各列尺寸，但要注意〔描述〕和〔余额〕列的调整方法与其它列不同。

〔描述〕列设计为自动扩展以填充所有未使用的水平空间。因此在设置〔描述〕列宽之前应当设置所有其它列的宽度。

工具栏按钮

科目树窗口和科目账簿窗口均提供工具栏按钮。借助这些按钮可以快速执行像是科目树窗口中的保存、打开，以及科目账簿窗口中的保存、删除等常用功能。如果你不清楚一个按钮会执行什么操作，可以将鼠标指针移动到按钮上，就会弹出对此功能的描述。

下面总结了科目树窗口中各按钮的功能：

保存

将当前文件保存到磁盘。

关闭

关闭当前账簿页。

打开、编辑、新建、删除

这些功能与科目相关，将在“科目”一节讨论。

有关科目账簿的按钮将在“交易”一节介绍。

标签栏

GnuCash使用标签式的设计，允许同时打开多个科目账簿和报表。每个打开的窗口（可以是科目账簿、报表或计划交易窗口，等等）都会在标签栏中分配一个标签页，可以点击标签页查看对应窗口。可以在〔首选项〕中设置标签栏沿GnuCash窗口的任意一边显示。

将鼠标指针悬停在标签上可以查看标签页的全名。如果打开的标签比整个窗口能显示的要多，部分标签可能不会显示。

1. 点击标签栏两侧的箭头可以浏览全部标签页。

2. 也可以通过〔首选项〕窗口中〔窗口〕标签页 [https://www.gnucash.org/docs/v5/C/gnucash-manual/set-prefs.html#prefs-window]下的〔标签页位置〕更改标签栏的对齐方式。
3. 可以右击标签栏查看标签页的完整列表，然后点击选择任一标签页。

菜单项

科目树窗口和科目账簿窗口在菜单栏处均包含相应的菜单标题。点击菜单标题会弹出与之对应的菜单项。

你可以点击科目树菜单标题，再将鼠标指针移动到菜单项上了解相应的功能。在鼠标指针移动到菜单项上时，菜单项的功能描述会出现在窗口的左下角（在状态栏内）。点击可以选择菜单项。

你也可以在窗口中的任何地方右击鼠标，访问最常用的菜单项。在科目树窗口中，会弹出一列科目相关的菜单项。在科目账簿窗口中，会弹出一列与交易相关的菜单项。

访问菜单项的其它方式还有快捷键和访问键，下面将介绍。

菜单的访问键与快捷键

所有菜单项都分配了访问键，由菜单名称中的下划线字符标出。按下〈Alt〉键和菜单标题中标有下划线的字符，就会弹出菜单标题相应的菜单项。菜单项弹出后，在键盘上按下菜单项中标有下划线的字符以选择菜单项。例如，在主窗口中按下〈Alt〉 + 〈F〉会弹出〔文件〕菜单，再按下〈S〉会保存文件。访问键是固定的，无法由用户更改。

部分更为常用的菜单项也分配了快捷键。使用快捷键，无需在菜单中层层寻找即可直接执行命令。虽说快捷键可以是任意按键的组合，但这些快捷键往往会使用〈Ctrl〉键。菜单快捷键显示在每个菜单项的右端。

存储财务数据

概述

GnuCash提供了几种格式存储财务数据。默认的文件存储格式是XML，而SQL存储在SQLite、MySQL和PostgreSQL格式中可用。对于新文件，用户可以通过〔文件〕→〔保存〕对话框选择文件格式；对于存在的文件，则通过〔文件〕→〔另存为…〕对话框选择文件格式。

XML存储格式是一种默认压缩存储的文本文件，是否压缩可以通过〔编辑〕→〔首选项〕+〔常规〕+〔压缩文件〕设置。也可以使用SQLite存储，像XML格式一样，它将你的数据存储为系统中的单个文件。然而，在程序内部，SQLite文件会作为数据库进行管理。MySQL和PostgreSQL存储选项需要连接到MySQL或PostgreSQL数据库服务器，以及在计算机上额外安装的数据库驱动程序。

用户可以随时使用〔文件〕→〔另存为…〕更改存储格式。这会以选定的格式创建数据文件的副本。

存储压缩及有关建议

每种存储格式各有优缺点，用户需要结合自身的需求和能力进行考虑。参见下文的存储格式对比了解详细信息。

XML格式是最为稳定成熟的，因此建议大多数用户选择此格式。SQL格式于2.4版添加，在用户中成为了越发受欢迎的选择。SQLite格式让用户无需花功夫安装并管理完整的数据库管理软件（DBMS）就可以享受SQL存储的优势。MySQL和PostgreSQL选项分别要求安装MySQL和PostgreSQL数据库管理软件（DBMS），这些数据库管理软件最好由经验丰富的数据库管理员维护。

GnuCash对SQL存储后端的支持意味着GnuCash现在完全兼容数据库管理软件（DBMS）中的许多特性，包括多用户和增量数据操作。然而，GnuCash当前并没有实现这些功能，不过这是开发组的长期目标。

存储格式对比表

2.2.

| | XML | SQLite | MySQL | PostgreSQL |
|-------|---------|---------------------|----------------|------------|
| 可用性 | 内置 | 依赖于软件包 ^a | | |
| 文件拓展名 | gnucash | | 无 ^b | |
| 额外软件 | 无 | | MySQL | PostgreSQL |
| 额外技能 | 无 | | 数据库管理员 | |
| 压缩 | gzip | 无 | | |
| 文件保存 | 执行保存命令时 | 提交时 | | |
| 多用户支持 | 不支持 | 不支持 | 不支持 | 不支持 |

^aSQL后端依赖额外的软件包。这些软件包已内置于在 macOS 和 Microsoft Windows 下的安装包中，会随GnuCash的安装程序自动安装。Linux发行版用户可能需要使用包管理器专门安装所需的额外软件包。对于SQLite3、MySQL和Postgresql分别查找包名类似libdbd-sqlite3、libdbd-mysql或libdbd-pgsql的软件包进行安装。

^bMySQL 和 PostgreSQL 将数据存储在他们自己的存储系统中。

创建文件

要创建新的GnuCash文件，按如下步骤操作：

1. 从GnuCash菜单栏选择〔文件〕→〔新建〕。〔建立科目结构〕向导将启动。

如果是首次运行GnuCash，将显示〔欢迎使用GnuCash〕屏幕。此屏幕会在GnuCash帮助手册详细介绍。

2. 在向导中设置各选项。使用前进、取消和后退按钮在各步骤间跳转。

保存数据

按照下面的步骤以你喜欢的名称保存文件：

1. 从菜单栏选择〔文件〕→〔另存为…〕或选择工具栏上的保存按钮。GnuCash会弹出保存窗口。
2. 从下拉列表中选择文件的〔数据格式〕。默认选择的是〔XML〕，如果你已经设置了数据库后端可以更改为相应的格式。

根据你选择的〔数据格式〕，窗口会发生如下变化。

3. 如果你选择了〔xml〕或〔sqlite3〕，你会看到像这样的对话框：

2.5. 〔xml〕 〔sqlite3〕



此图展示了〔数据格式〕选为〔xml〕或〔sqlite3〕时的〔保存〕对话框。

在〔名称〕文本框中输入文件名，不必指定扩展名。GnuCash会自动向文件添加拓展名".gnucash"。

在下方区域中浏览目录选择文件保存路径。

点击创建文件夹按钮在选择的路径下以指定的名称创建文件夹。

· 如果你选择了〔mysql〕或〔postgres〕你会看到如下对话框：

2.6. 〔mysql〕 〔postgres〕



此图展示了〔数据格式〕选为〔mysql〕或〔postgres〕时的〔保存〕对话框。

在此窗口中输入〔数据库连接〕信息：〔主机〕、〔数据库〕、〔用户名〕和〔密码〕。

保存至〔mysql〕或〔postgres〕需要合适的数据库权限，即需要拥有使用给定的数据库名称创建新数据库的权限，或向具有给定数据库名称的现有数据库进行写入的权限。

4. 点击另存为按钮保存文件。

你如果正在跟踪家庭财务状况，则只需要一个文件。但如果你同时还在管理企业财务，或出于某种原因希望将数据分离存储，则不只需要一个文件。

在结束每次GnuCash会话前，确保使用〔文件〕→〔保存〕或保存 工具栏按钮保存数据更改。

经常保存数据以避免由任何原因导致的数据丢失是相当重要的，所以GnuCash能够每隔一段时间自动保存打开的文件。保存间隔时间可以在〔编辑〕→〔首选

项] (macOS上为[GnuCash] → [首选项]) 中的[通用]选项卡下设置。注意此选项只有在你以XML格式保存时才有意义。如果你正在使用数据库存储, 则做出的更改会即时保存, 所以保存按钮和[保存]菜单项呈灰色。

打开数据

要打开存在的文件或数据库, 从菜单中选择[文件] → [打开]。在弹出的窗口中, 选择[数据格式]。如果你选择了[文件], 则在下方的面板浏览文件夹选择你想打开的文件; 否则, 输入所需的[数据库连接]信息。

GnuCash会记录最近打开的文件列表。打开[文件]菜单, 可以看到菜单内列出了最近打开过的文件名称。点击就可以载入你想要打开的文件。

复制科目层次结构

某些情况下, 将现有数据文件的结构复制到新文件可能很有用。例如, 你可能会想在不破坏实际会计数据的情况下尝试新的会计技能, 或是需要根据会计准则在年末了结当前账簿并拿空白账簿开始新一年的会计工作。

GnuCash能够创建科目树的空白副本, 只需选择[文件] → [导出] → [导出科目]。选择此命令后, GnuCash会要求你提供新的空白文件的名称, 随后GnuCash会创建一个新的数据文件, 其中仅包含科目层次结构(即不含交易数据)。一旦保存, 这个新文件就能像其它GnuCash数据文件一样以上文所述的方法打开。

备份与恢复数据

GnuCash会创建多种类型的文件, 帮助你确保数据不会丢失。如果查看保存的文件的所在文件夹, 可能会看到由GnuCash生成的具有以下扩展名的其他文件: ".gnucash"、".log"、".LCK"、".LNK"。每种文件的作用如下:

下文仅当你以XML格式保存时才适用:

```
$ ls
myfile.gnucash
myfile.gnucash.20100414185747.gnucash
myfile.gnucash.20100414223248.log
myfile.gnucash.20100415114340.gnucash
myfile.gnucash.20100415154508.log
```

```
myfile.gnucash.20100415173322.gnucash
myfile.gnucash.20100415194251.log
myfile.gnucash.7f0982.12093.LNK
myfile.gnucash.LCK
```

备份文件（".gnucash"）

每当你保存数据文件时，一个扩展名为 ".YYYYMMDDHHMMSS.gnucash"的备份副本也会保存。此备份文件是先前数据文件的完整副本，文件命名格式涉及数据文件名称、年、月、日和备份时间。例如，文件名 "myfile.gnucash.20100414185747.gnucash"表明这是在2010年4月14日下午6:57:47保存的文件 "myfile"的备份副本。

要恢复之前备份的文件，只需打开具有希望回溯到的日期的 ".YYYYMMDDHHMMSS.gnucash"文件，再以不同的文件名保存该文件。

日志文件（".log"）

每次在GnuCash中打开并编辑文件，GnuCash就会创建日志文件，其中记录对数据文件做出的修改。日志文件使用与备份文件相似的命名格式： ".YYYYMMDDHHMMSS.log"。日志文件不是对数据文件的完整备份，而只是简单地记录在当前GnuCash会话中对数据文件作出的更改。

在你无意间退出GnuCash的情况下，可能是由于断电或操作系统崩溃，你可以使用日志文件恢复在最后一次保存GnuCash文件后的做出的大部分修改。过程如下：

1. 打开最近一次保存的GnuCash文件。
2. 点击〔文件〕→〔导入〕→〔重放 GnuCash .log 文件〕，选择一个与你刚才打开的文件日期相同的 ".log" 文件。确保你选择了正确的 ".log" 文件，否则可能会对你的科目数据造成巨大的损坏。

日志重放会恢复自最近一次保存后输入的任何影响科目余额的交易，包括由计划交易和企业功能（如开具发票、订单管理等）创建的交易。

不会恢复对计划交易、开具发票或订单管理功能自身的修改，并且由之创建的交易在恢复后可能不会恰当地与之关联，因此需要二次检查。特别是对于企业交易，你可能需要删除并重新创建部分交易。如果不这样做，尽管科目余额可能正确，但一些报表可能不会正确显示。

锁定文件（".LNK" 和 ".LCK"）

你可能会偶然看到 ".LNK"和 ".LCK"文件出现。它们不存储任何数据，但会被创建用于防止多个用户在同一时刻打开同一个文件。这些锁定文件是在打开数据文件时自动创建的，当关闭

GnuCash会话或打开另一个文件时，GnuCash会删除 ".LNK"和 ".LCK"文件以解锁前一个数据文件。

如果在已经打开了数据文件时，GnuCash崩溃了，则 ".LCK"和 ".LNK"文件不会被删除。下一次尝试打开GnuCash时，GnuCash会弹出警告消息提示文件已被锁定。之所以弹出警告消息是由于目录中仍然存在 ".LNK"和 ".LCK"文件。此时选择是打开文件是安全的，但你应当删除 ".LNK"和 ".LCK"文件。删除这些文件后，你将不再看到警告消息，除非GnuCash再次崩溃。

文件管理

所以到底应该保留哪些文件呢？当然是保留你的主数据文件。保留部分最近的 ".YYYYMMDDHHMMSS.gnucash"备份文件也是妥当的做法，不过你可以安全地删除 ".log"文件，因为它们不是数据的完整副本。

你也应该在关闭GnuCash后删除任何看到的 ".LCK"和 ".LNK"文件。如果你决定将数据文件手动备份到另一个磁盘，则只备份主数据文件就足够了，无需关心 ".YYYYMMDDHHMMSS.gnucash"备份文件。

一般情况下GnuCash会自动删除任何30天以前的 ".YYYYMMDDHHMMSS.gnucash"和 ".log"备份文件。可以在〔编辑〕→〔首选项〕（macOS 上的〔GnuCash〕→〔首选项〕）中的〔通用〕选项卡下更改此行为。

迁移GnuCash数据

有时，你可能需要将财务数据和GnuCash移动到另一台计算机。典型情况有，当你购置了一台新计算机的时候，或者在双引导计算机中想要在两个不同的操作系统间共用同一份设置的时候。

迁移财务数据

迁移GnuCash的财务数据很简单，如果你知道 ".gnucash"文件的保存位置，只需要使用文件管理器把它复制过去。你如果不记得把它保存到哪里了，可以直接在GnuCash中打开它，再在GnuCash中将它保存到你需要的路径下。

文件夹中其它所有文件是备份文件或日志文件。把它们一起复制过去不会有什么危害，但也没什么必要。

迁移首选项数据

首选项存储在三处不同的位置：一处用于GnuCash首选项，一处用于报告，一处用于网上银行设置。应用首选项由gsettings管理，报告由GnuCash自身管理，网上银行由aqbanking管理。如果你不使用网上银行功能，计算机中不会存在相应的文件夹。

GnuCash首选项的存储位置在不同的操作系统下不尽相同（参见表 2.3 “应用设置的存储位置”、表 2.4 “已保存报告的存储位置”和表 2.5 “网上银行设置的存储位置”）。要备份并迁移整个已安装的GnuCash，你必须一并复制这些首选项。

2.3.

| | |
|---------|--|
| | |
| Unix | GnuCash首选项存储在dconf中。可以在旧计算机上执行命令dconf dump /org/gnucash/，然后在新计算机上执行命令dconf load /org/gnucash/以迁移首选项。 |
| macOS | "~/Library/Preferences/gnucash.plist" |
| Windows | 首选项存储于Windows注册表 "HKEY_CURRENT_USER/software/GSettings" |

2.4.

| | |
|---------|--|
| | |
| Unix | "~/local/share/gnucash/" ^a |
| macOS | "~/Library/Application Support/gnucash" |
| Windows | "Documents and Settings/ Username/.gnucash" 或 "Users/ Username/.gnucash" |

^aGnuCash 2.6.21 及之前的版本位于 "~/gnucash/"

2.5.

| | |
|---------|--|
| | |
| Unix | "~/aqbanking" |
| macOS | "~/aqbanking" |
| Windows | "Documents and Settings/ Username/.aqbanking" |

在Unix和macOS上，这些文件夹通常不会显示在文件管理器中。为了看到这些文件夹，必须设置文件管理器显示隐藏的文件和文件夹。

在Unix和macOS上， "~"符号代表 "home"文件夹。

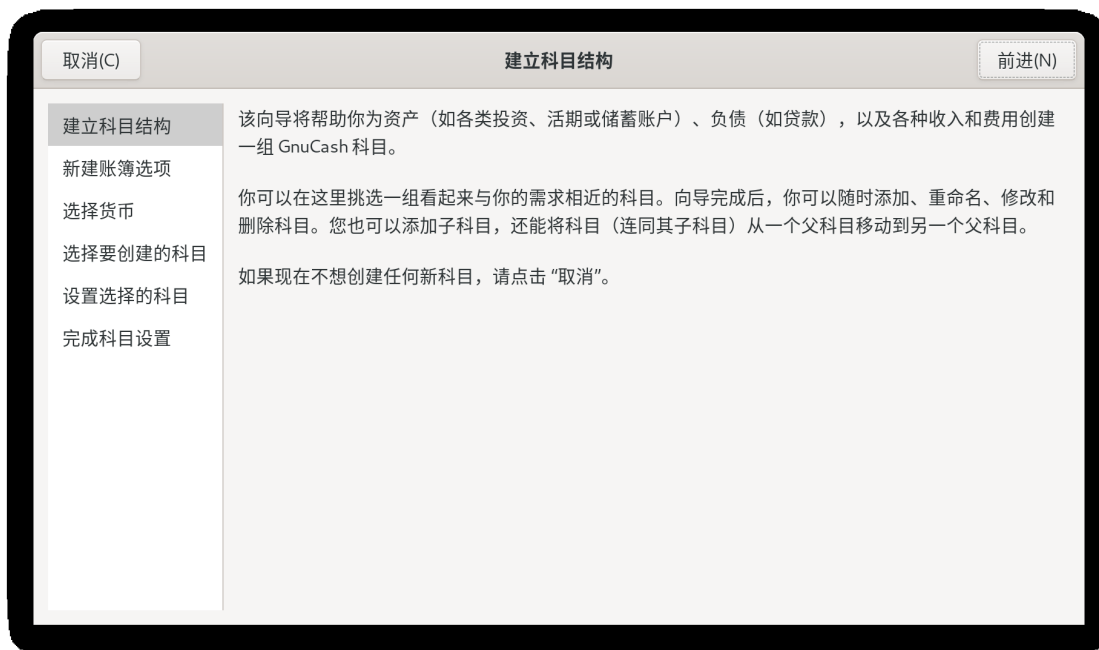
步步为营

这一部分开启了贯穿整部指南的教程。在每一章的结束，你都会看到〔步步为营〕一节，它会带领你通过各种例子形象地体会各章节介绍的概念。每节〔步步为营〕都会在前一部分的基础上更进一步，所以请将你的文件保存到方便访问的地方。

就从这里开始吧！

1. 首先，创建一个文件，用于保存你的数据。打开GnuCash并从菜单栏选择〔文件〕→〔新建〕，会启动〔建立科目结构〕向导，其中能够一次创建数个科目。

2.7.



此图展示了〔建立科目结构〕向导的第一个页面。

向导的第一个页面描述了其功能。点击前进按钮进入下一个页面。

2. 在第二个页面，通过不同的选项卡设置〔新账簿选项〕，然后点击前进按钮。你也可以稍后使用〔文件〕→〔属性〕修改这些选项。要详细了解这些选项，参见GnuCash用户手册中“自定义GnuCash：账簿选项”一章。

2.8.

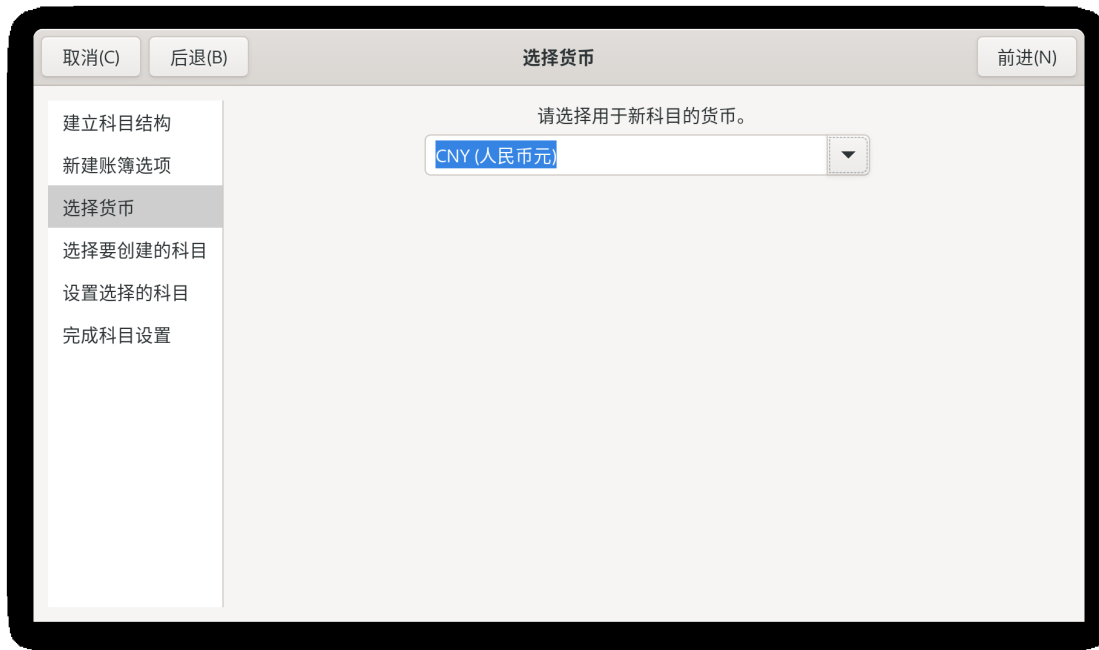


此图展示了〔建立科目结构〕向导的第二个页面，此处可调整〔账簿选项〕。

3. 在第三个页面，从下拉列表中选择要用于新科目的货币，然后点击前进按钮。

此处选择的货币会被指定到所有由此向导创建的科目。

2.9.

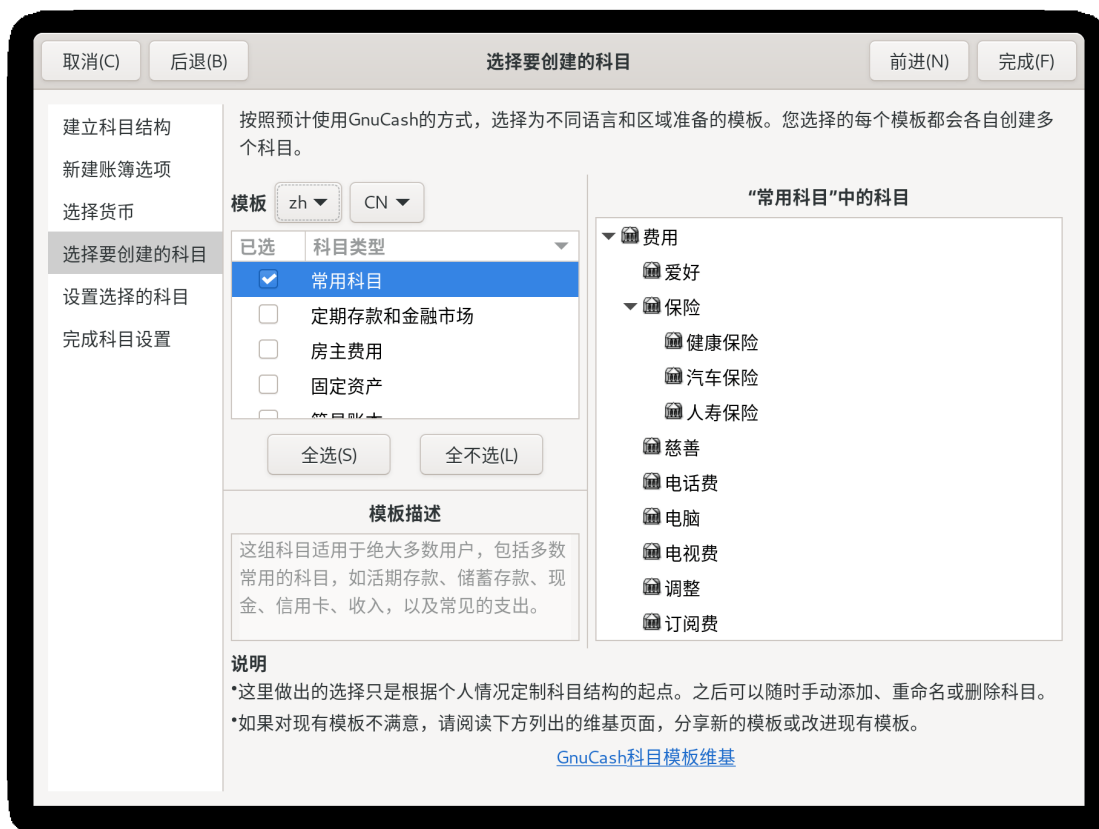


此图展示了〔建立科目结构〕向导的第三个页面，此处可选择〔货币〕。

4. 在第四个页面中，勾选〔模板〕面板下〔常用科目〕科目组。再点击前进按钮继续。

若需要，可在此选择更多预定义的科目组。有关科目类型的更多信息可见“GnuCash 科目”一节。

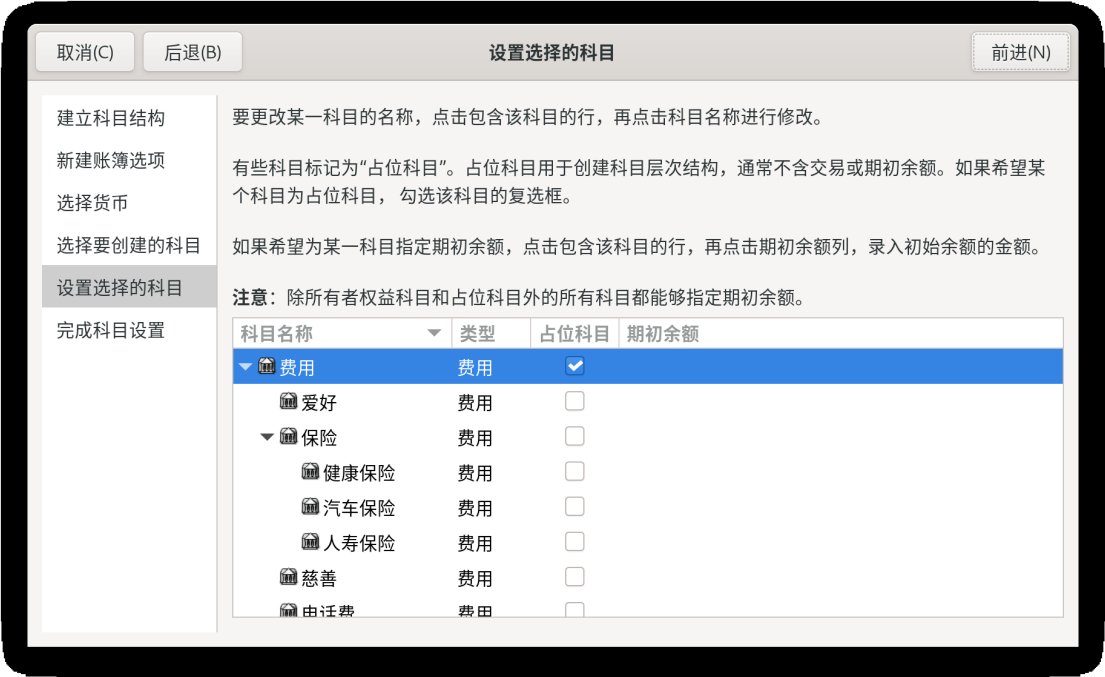
2.10.



此图展示了〔建立科目结构〕向导的第四个页面，此处可选择各种〔科目〕。

5. 在第五个页面，可以设置每一科目的〔期初余额〕，并指定科目是否为〔占位符〕。这些功能将在下一章详细介绍，这里先不更改这些选项。下面点击前进按钮进入此向导的最后一个页面。

2.11.



此图展示了〔建立科目结构〕向导的第五个页面，此处可设置〔期初余额〕。

6. 在向导的最后一个页面中，点击应用创建所有科目并结束向导。

2.12.



此图展示了〔建立科目结构〕向导的最后一个页面。

7. 在之前的窗口点击应用按钮后，会弹出保存对话框。选择XML〔数据格式〕，文件〔名称〕为 "gcashdata_1"，选择要保存文件的文件夹（请记住这里选择的保存位置，因为这份数据文件会在整份指南后续教程中继续使用），最后点击保存按钮。

现在，主窗口应该如下图所示：

2.13.



科目

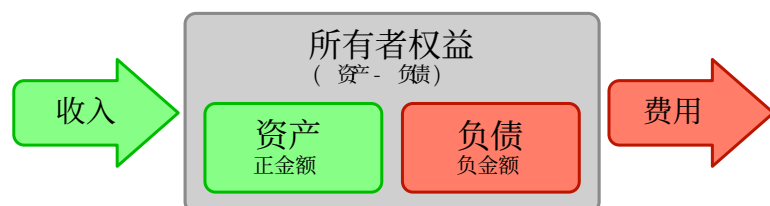
这一章会介绍一些用于组织科目的实用概念。自从GnuCash不再要求固定的科目结构之后，你可以通过任何希望的方式自由地设计科目结构。然而，在设计尽量实用的科目结构时，建议你遵循几个基本会计概念。

基本会计概念

正如前文所述，会计工作围绕五类基本科目类型展开：〔资产〕、〔负债〕、〔所有者权益〕、〔收入〕和〔费用〕。我们现在更深入地理解这些科目类型，并展示它们在GnuCash中的表现形式。不过首先，我们将它们分为两组：资产负债科目和损益科目。恰如其名，资产负债科目可以汇总在某一时间点处你的资产和负债金额，而损益科目汇总在损益表报告中，损益表报告显示了一段时间内的净值变动，例如一财年内的变动。

在深入理解科目类型之前，先简单回顾一下会计等式：资产 - 负债 = 所有者权益 + (收入 - 费用)。

2.14.



五类基本科目类型之间的关系图。净值经收入而增加，经费用而减少。箭头代表金额的移动。

资产负债科目

三个所谓的资产负债科目是〔资产〕、〔负债〕和〔所有者权益〕。资产负债科目用于跟踪你的资产或负债。

〔资产〕是一组你拥有的东西。你的资产可能包含汽车、现金、房产、股票，或其它任何具有交换价值的东西。交换价值是指理论上你可以卖出资产换取现金。

〔负债〕是一组你欠别人的东西。你的负债可能包括汽车贷款、学生贷款、房屋贷款、投资保证金帐户，或其它任何你必须在某一时刻偿还的东西。

〔所有者权益〕即“净值”，代表从资产中减去负债后所剩的东西。可以认为它是你完全拥有的、不含任何债务的一部分资产。

损益科目

两种〔损益科目〕用于增加或减少各科目的金额。所以，资产负债科目只是跟踪资产和负债的金额，而损益科目能够改变这些科目的金额。

〔收入〕是你通过你的时间、你提供的服务或对资金的利用而得到的报酬。例如，当你收到工资时，工资就是你为雇主提供劳动而得到的报酬。收入也包含佣金、小费、来自股票的股息收入和来自银行账户的利息收入。收入总是会增加资产的价值。

〔费用〕是指涉及用于购买他人提供的商品或服务的花销。费用可以是餐馆里的一顿晚饭、房屋租金、杂货、汽油或演出票。费用会减少你的净值。如果你立刻支付费用，资产会减少，而如果你使用信用卡支付费用，负债会增加。

GnuCash 科目

本节将阐明GnuCash定义的科目如何对应于五类基本科目类型。

首先，给出GnuCash中科目的定义。一个GnuCash〔科目〕是包含其它子科目或交易的实体。由于科目可以包含其它科目，你经常可以在GnuCash中看到科目树，逻辑上相关的科目在一个共同的父科目下分为一组。

GnuCash中一个科目必须具有（由你指定的）唯一名称，以及预定义GnuCash“科目类型”的其中之一。GnuCash总共有12种科目类型。这12种科目均可归类于5类基本科目类型，之所以GnuCash科目类型多于基本科目类型是为了帮助GnuCash针对某些科目进行特殊的跟踪和处理。有6种资产科目（现金、银行、股票、共同基金、应收账款和资产）、3种负债科目（信用卡、应付账款和负债）、1种所有者权益科目（所有者权益）、1种收入科目（收入）、1种费用科目（费用）。

下面详细列出这些GnuCash科目类型。

资产负债科目

第一种详细介绍的资产负债科目是资产，正如你在前文了解到的那样，资产涉及你拥有的东西。

为了帮助你组织资产科目并便于录入交易，GnuCash支持以下几种资产科目类型：

1. 〔现金〕。使用此科目跟踪你在手头、钱包里、存钱罐中、床垫下，或任何你选择放在手边的现金。这是最具流动性的、最方便交易的资产类型。

2. [银行]。此科目用于跟踪你在银行，信用合作社，储蓄、贷款或经纪公司等任何资产受保护的机构中的余额。这是第二最具流动性的科目类型，因为你可以方便地将之兑换为手头的现金。
3. [股票]。使用此科目类型跟踪你单独持有的股票和债券。股票科目账簿提供附加列，用于输入份额数量和投资价格。除非你找到买家，否则你可能无法方便地将这种类型的资产转换为现金，而且你无法保证卖出时得到的金额与买入时花费的金额相同。
4. [共同基金]。这一科目类型与股票科目类似，只不过是用于跟踪基金的。它的账簿也提供了相同的附加列用于输入份额和价格信息。基金代表多项投资的所有权份额，而且和股票一样不对现金价值提供任何保证。

GnuCash以相同的方式处理股票和共同基金这两种科目类型。

5. [应收账款]。此科目一般仅用于企业。你应当将别人欠你的重要借款录入应收账款科目中。一般视应收账款为一种资产，因为你能够指望对方归还这些资金。

不应以任何方式添加、更改或删除涉及应收账款科目的交易，除非使用

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

6. [资产]。GnuCash可以轻松的处理其它许多情况，不管这些情况有多么的变化多端。科目类型“资产”可以用于上文未列出的各种资产。

GnuCash以相同的方式处理现金、银行和资产科目类型。

科目是“信息仓库”，用以跟踪或记录与建立科目的目相关的活动。

对于企业，往往会把要跟踪和编制报表的活动细分得比这里考虑的更加细致。为了更深入地研究这一可能，请阅读本指南第 15 章 Other Assets 中的相关描述。

对于个人财务管理，可以使用与企业相同的科目结构，也可以不同。因为用于企业的科目结构可能对个人正在跟踪的财务活动和个人管理金融资产需要的各类报表有一定作用。更多信息可查阅本指南的第 15 章 Other Assets。

第二种资产负债科目是负债，正如前文所述，它指你欠的东西或你借入的钱，而且应当在将来偿还。这些都代表着借款人从你这里获得回报的权利。跟踪负债余额可以让你知道在某一时间点你承担了多少负债。

GnuCash 提供三种负债科目类型：

1. [信用卡] 用于跟踪信用卡收据并根据信用卡账单对账。信用卡代表一种应当向信用卡公司偿还的短期贷款，这种科目也可用于其它短期贷款，例如从银行获得的信用贷款。
2. [应付账款] 此科目一般用于企业，用于记录尚未支付的账单。

涉及应付账款的交易不应通过除使用

· process payment

以外的方式添加、修改或删除。

3. [负债] 用于所有其它的负债，一般是数额更大的长期贷款，例如房屋贷款或汽车贷款。此科目有助于跟踪你当前的负债金额或已偿还金额。

GnuCash以相同的方式处理信用卡和负债这两种科目类型。

负债的记账方法与资产相反：贷增加负债余额，借减少负债余额。

最后一种资产负债科目是所有者权益，也就是“净值”。它代表从资产中减去负债后的剩余，所以这是你的资产中不含任何负债的、完全由你拥有的部分。在GnuCash中此类科目用作期初余额的来源，因为这些余额代表你的初始净值。

通常科目表中只有一个GnuCash所有者权益科目，自然被叫做[所有者权益]。对于企业，你可以为每位合作伙伴创建其子科目。

在所有者权益科目中，贷增加科目余额，借减少科目余额。

关联资产负债科目的会计等式是“资产 = 负债 + 所有者权益”，也可改写为“负债 - 资产 = 所有者权益”。所以，通俗地讲，你拥有的东西减去你欠的东西等于净值。

损益科目

收入是你由于你的时间、你提供的服务或对钱的利用而获得的支付金额。在GnuCash中，使用[收入] 科目类型跟踪这些金额。

贷增加收入科目余额，借减少收入科目余额。正如“会计概念”一节所述，贷代表金额从科目中移出。所以在收入科目中，当你将钱从收入科目移到另一个科目时（贷），收入科目的余额会增加。例如，当你存入工资并记录了一笔从收入科目至银行账户的交易时，这两个科目的余额都增加了。

费用指你用于购买他人提供的商品或服务的钱。在GnuCash中，使用[费用] 类型科目跟踪费用。

借增加费用科目余额，贷减少费用科目余额。

在一段时间内，从总收入减去总费用得到净收入。净收入随后作为留存收益添加到资产负债表的[所有者权益] 科目。

下面展示了标准的〔收入〕和〔费用〕科目，在〔新建科目层级〕向导（〔功能〕→〔新建科目层级…〕）中选择〔通用科目〕后会创建。

2.15.

| 科目名称 | 类型 | 商品 | 描述 | 合计 |
|----------------------------------|----|------|----------|---------|
| 收入 | 收入 | 人民币元 | 收入 | CN¥0.00 |
| 薪金 | 收入 | 人民币元 | 薪金 | CN¥0.00 |
| 收到的礼物 | 收入 | 人民币元 | 收到的礼物 | CN¥0.00 |
| 其他收入 | 收入 | 人民币元 | 其他收入 | CN¥0.00 |
| 利息收入 | 收入 | 人民币元 | 利息收入 | CN¥0.00 |
| 其他利息 | 收入 | 人民币元 | 其他利息 | CN¥0.00 |
| 活期利息 | 收入 | 人民币元 | 活期利息 | CN¥0.00 |
| 储蓄存款利息收 | 收入 | 人民币元 | 储蓄存款利息收入 | CN¥0.00 |
| 奖金 | 收入 | 人民币元 | 奖金 | CN¥0.00 |
| CN¥ 总计: 净资产: CN¥0.00 利润: CN¥0.00 | | | | |

此图展示了标准的收入科目。

2.16.

| 科目名称 | 类型 | 商品 | 描述 | 合计 |
|----------------------------------|----|------|--------|---------|
| 费用 | 费用 | 人民币元 | 支出的费用 | CN¥0.00 |
| 在线服务 | 费用 | 人民币元 | 在线服务 | CN¥0.00 |
| 杂项 | 费用 | 人民币元 | 杂项 | CN¥0.00 |
| 娱乐 | 费用 | 人民币元 | 娱乐 | CN¥0.00 |
| 用餐 | 费用 | 人民币元 | 用餐 | CN¥0.00 |
| 银行服务收费 | 费用 | 人民币元 | 银行服务收费 | CN¥0.00 |
| 衣服 | 费用 | 人民币元 | 衣服 | CN¥0.00 |
| 医疗费用 | 费用 | 人民币元 | 医疗费用 | CN¥0.00 |
| 洗衣/干洗 | 费用 | 人民币元 | 洗衣/干洗 | CN¥0.00 |
| CN¥ 总计: 净资产: CN¥0.00 利润: CN¥0.00 | | | | |

此图展示了标准的费用科目。

其它科目类型

也有其它一些特殊的科目类型。

贸易。多货币交易需要录入与“贸易”科目关联的分录，以使交易在每种货币和整个账簿上均保持平衡。请阅读第 12 章 多货币详细了解。

货币市场和信用额度仅用于OFX导入器，以完整符合规范。

步步为营

现在根据从此章学习到的内容，体会构建通用个人财务科目表的过程。科目表只不过是一个新的GnuCash文件，你可以在其中组织科目结构用以跟踪财务状况。要构建科目表，第一步是將你想要跟踪的财务事项划分到各个基本会计科目类型中去。这相当简单，不妨看看下面这个例子。

简单示例

假定你在银行中有一个活期账户和一个储蓄账户，你自己受雇佣因此会收到工资。你持有信用卡，按月支付房租、话费和电费。当然，你也需要购买食品杂货。现在，我们不会关心你在银行里存了多少钱，你的信用卡欠了多少钱，等等。这里只是简单地构建科目表的框架。

你的〔资产〕包含银行储蓄账户和活期账户。你的〔负债〕是信用卡欠款。你的〔所有者权益〕是银行账户和信用卡的初始金额（我们目前还没有这些金额数据，但这些金额是存在的）。你以薪资的形式获得〔收入〕，以食品杂货、房租、电费、话费和薪资税费（所得税、社保、医保等）的形式产生〔费用〕。

基本顶级科目

现在你必须决定你想要怎样组织这些科目。一般来说，你想让你的资产分为一组，负债分为一组，所有者权益分为一组，收入分为一组，费用分为一组。这是最常见的构建科目表的方式，同时强烈建议你总是以之作为组织科目的起点。

一开始，从菜单中选择〔文件〕→〔新建〕创建空白的GnuCash文件。〔创建科目结构〕向导会启动。点击取消关闭向导，因为我们不想使用任何一种预定义的科目结构。我们将依照自己的想法自行构建基本初始科目结构。在空白的GnuCash窗口中选择菜单〔查看〕→〔打开新的科目标签页〕，随后将打开〔科目〕标签页。再选择〔功能〕→〔新建科目…〕。

现在可以着手构建基本初始科目结构了。

1. 科目名称：〔资产〕（科目类型：〔资产〕，父科目：〔新建顶级科目〕）

2.17.

此图显示了创建资产科目的对话框。

2. 科目名称：〔负债〕（科目类型：〔负债〕，父科目：〔新建顶级科目〕）
3. 科目名称：〔所有者权益〕（科目类型：〔所有者权益〕，父科目：〔新建顶级科目〕）
4. 科目名称：〔收入〕（科目类型：〔收入〕，父科目：〔新建顶级科目〕）
5. 科目名称：〔费用〕（科目类型：〔费用〕，父科目：〔新建顶级科目〕）

创建顶级科目后，GnuCash中的科目页应该如下图所示：

2.18.



此图展示了基本顶级科目。

创建子科目

现在可以向基本顶级科目树结构添加一些存储实际交易的子科目。注意各个税费科目置于名为税费的子科目之下。可以在子科目下再创建子科目。一般将一组相关的科目作为同一科目的子科目（如这个例子中的各个税费科目）。

除了从菜单中选择〔功能〕→〔新建科目…〕，也可以右击科目名称并选择〔新建科目…〕菜单项创建相应科目的子科目。随后会打开与“基本顶级科目”一节所示相似的对话框，其中将新建的科目会自动设为相应科目的子科目。

1. 科目名称：〔现金〕（科目类型：〔现金〕，父科目：资产）
2. 科目名称：〔活期〕（科目类型：〔银行〕，父科目：资产）
3. 科目名称：〔储蓄〕（科目类型：〔银行〕，父科目：资产）
4. 科目名称：〔信用卡〕（科目类型：〔信用卡〕，父科目：负债）
5. 科目名称：〔薪资〕（科目类型：〔收入〕，父科目：收入）
6. 科目名称：〔话费〕（科目类型：〔费用〕，父科目：费用）
7. 科目名称：〔电费〕（科目类型：〔费用〕，父科目：费用）
8. 科目名称：〔房租〕（科目类型：〔费用〕，父科目：费用）
9. 科目名称：〔杂货〕（科目类型：〔费用〕，父科目：费用）

10.科目名称：〔税费〕（科目类型：〔费用〕，父科目：费用）

11.科目名称：〔所得税〕（科目类型：〔费用〕，父科目：费用:税费）

12.科目名称：〔社保〕（科目类型：〔花费〕，父科目：费用:税费）

13.科目名称：〔医保〕（科目类型：〔花费〕，父科目：费用:税费）

14.科目名称：〔期初余额〕（科目类型：〔所有者权益〕，父科目：所有者权益）

在创建了这些子科目后，最终结果如下所示：

2.19.

| 科目名称 | 商品 | 合计 |
|---------|------|---------|
| ▼ 资产 | 人民币元 | CN¥0.00 |
| 现金 | 人民币元 | CN¥0.00 |
| 储蓄存款 | 人民币元 | CN¥0.00 |
| 活期存款 | 人民币元 | CN¥0.00 |
| ▼ 负债 | 人民币元 | CN¥0.00 |
| 信用卡 | 人民币元 | CN¥0.00 |
| ▼ 收入 | 人民币元 | CN¥0.00 |
| 薪酬 | 人民币元 | CN¥0.00 |
| ▼ 费用 | 人民币元 | CN¥0.00 |
| 电话费 | 人民币元 | CN¥0.00 |
| 房租 | 人民币元 | CN¥0.00 |
| ▼ 水电费 | 人民币元 | CN¥0.00 |
| 电费 | 人民币元 | CN¥0.00 |
| ▼ 税费 | 人民币元 | CN¥0.00 |
| 社会保险 | 人民币元 | CN¥0.00 |
| 所得税 | 人民币元 | CN¥0.00 |
| 医疗保险 | 人民币元 | CN¥0.00 |
| 杂货 | 人民币元 | CN¥0.00 |
| ▼ 所有者权益 | 人民币元 | CN¥0.00 |
| 期初余额 | 人民币元 | CN¥0.00 |

CN¥ 总计: 净资产: CN¥0.00 利润: CN¥0.00

此图展示了一个简单的科目表。

使用文件名 "gcashdata_3" 和 "gcashdata_3emptyAccts" 保存此科目表，我们将在后续各章中继续使用这些文件。

现在，你已经创建了一份简单的用于跟踪家庭财务状况的科目表。在准备好这一基本框架后，我们现在可以开始向科目中添加交易了。具体会在下一章详细介绍。

交易

本章会针对在GnuCash中理解和使用交易做出基本说明。科目是科目表的结构与框架，然而交易才是填充各个科目的数据。

基本概念

GnuCash这类复式记账系统中的交易是指至少两个科目间的转账。因此一条交易总是包含至少两部分：一个来源科目和一个目的科目。金额从来源科目转移到目的科目中。会计师将交易的这些组成部分称为账簿条目（Ledger Entries）。在GnuCash中称之为分录。

一条分录包含涉及的科目，以及从之转出或向之转入的金额，如有需要，分录也可以包含其它几项信息。GnuCash支持在一条交易中记录多条分录，各条分录可以向涉及的科目转入或从之转出任意的金额。

例如，你收到了工资并将之存入银行储蓄账户。发生的交易就是你的银行储蓄科目（属于资产科目）从收入科目中得到了钱。两个科目受到影响，此时所有者权益净值增加了。

在GnuCash中增删交易需要使用科目账簿。每个创建的科目都对应一份科目账簿。它很像用于跟踪支票簿的日志，所以可能你看着很眼熟。

科目账簿会在接下来的“科目账簿”一节详细介绍。

科目账簿

科目账簿是一类窗口，可以浏览或编辑已经存在的交易，或为某一科目添加新的交易。要从科目树打开某一科目的科目账簿，双击科目名称，或者右击科目名称并从菜单中选择〔打开科目〕，也可以使用工具栏上的〔打开〕按钮。GnuCash会显示科目账簿窗口。

科目账簿的功能

科目账簿和标题栏显示了科目名称。标题栏之下，菜单栏显示了能够在科目账簿中使用的菜单项，工具栏包含方便操作科目账簿的常用功能按钮。

在科目账簿窗口的左下方，GnuCash会随窗口中鼠标移动而显示有用的帮助消息。在右侧，可以看到当前科目余额和已结清分录的总计。

选择账簿风格

GnuCash提供了几种样式选项，便于浏览账簿。默认样式是〔基本账簿〕模式，仅将影响当前科目的分录显示在一起。这种样式看起来十分类似其它流行的个人财务软件。你可以从〔查看〕菜单选择不同的账簿风格。另有两个选项：

- [查看] → [自动分录账簿] 样式会自动展开当前的交易。当在账簿中选中新的交易时，交易会自动展开以展示其全部分录。
- [查看] → [交易日志] 样式会在账簿中显示所有交易的所有分录，此样式更像是会计日志。

这些样式均能够以单行或多行格式浏览数据。选择 [查看] → [双行]，可以看到交易行扩展成了两行。双行模式也会显示交易说明列。

下面的屏幕截图展示了基本账簿和交易日志在外观上的不同。

例如，假设你花1000元购买了三条牛仔裤，且已经将这次购买录入为分录交易，其中每条牛仔裤各录入一条分录。

下面的屏幕截图展示了不同浏览模式下的差异。

首先，从活期科目查看牛仔裤交易：

2.20. 资产：活期

txns-registersplit4.gnucash - 活期存款 - GnuCash

文件(F) 编辑(E) 查看(V) 交易(N) 操作(A) 企业(B) 报表(R) 工具(T) 窗口(W) 帮助(H)

保存

关闭

创建副本

删除

确认

取消

转到空行

展开分录

科目

活期存款

| 日期 | 编号 | 描述 | | 借方总计 | 贷方总计 | 余额 |
|------------|------|--------|------------|-------------|------------|-----------|
| 2024-02-01 | | 初始化账簿 | 所有者权益:期初余额 | n 10,000.00 | | 10,000.00 |
| 2024-02-26 | ▼ 编号 | 添置衣物 | | 借方总计 | 1,000.00 | 9,000.00 |
| | | 第一条牛仔裤 | 支出:衣物 | n 500.00 | | |
| | | 第二条牛仔裤 | 支出:衣物 | n 300.00 | | |
| | | 第三条牛仔裤 | 支出:衣物 | n 200.00 | | |
| | | | 资产:活期存款 | | n 1,000.00 | |
| | | | | | | |
| 2024-04-01 | | | | n | | |

现在: CN¥9,000.00 将来: CN¥9,000.00 已结清: CN¥0.00 已对账: CN¥0.00 预计最低: CN¥9,000.00 排序: 标准 ^

此图展示了一条分录交易，其中包含三项牛仔裤购买分录。

现在，打开费用：服装科目，以基本模式查看。

2.21. 费用：服装

txns-registersplit4.gnucash - 衣物 - GnuCash

文件(F) 编辑(E) 查看(V) 交易(N) 操作(A) 企业(B) 报表(R) 工具(T) 窗口(W) 帮助(H)

保存

关闭

创建副本

删除

确认

取消

转到空行

展开分录

科目

衣物

| 日期 | 编号 | 描述 | 转账 | 对 | 借方 | 贷方 | 余额 |
|------------|----|------|------------|---|--------|----|----------|
| 2024-02-26 | 编号 | 添置衣物 | -- 分录交易 -- | n | 300.00 | 贷方 | 300.00 |
| 2024-02-26 | | 添置衣物 | -- 分录交易 -- | n | 200.00 | | 500.00 |
| 2024-02-26 | | 添置衣物 | -- 分录交易 -- | n | 500.00 | | 1,000.00 |
| 2024-04-01 | | | | n | | | |

现在：CN¥1,000.00 将来：CN¥1,000.00 已结清：CN¥0.00 已对账：CN¥0.00 预计最低：CN¥1,000.00 排序：标准 ^

此图展示了基本账簿模式下的费用：服装科目。

这里出现了三项，但活期科目中只有一项分录。仔细观察可以发现每一行的金额不同，分别为200元、300元和500元。这说明，此模式下账簿的每一行对应原始交易中的一条分录。

更改为交易日志模式则仅显示原始的分录交易。

2.22. 费用：服装

txns-registersplit4.gnucash - 衣物 - GnuCash

文件(F) 编辑(E) 查看(V) 交易(N) 操作(A) 企业(B) 报表(R) 工具(T) 窗口(W) 帮助(H)

保存

关闭

创建副本

删除

确认

取消

转到空行

展开分录

科目

衣物

| 日期 | 编号 | 描述 | | 借方总计 | 贷方总计 | 余额 |
|------------|----|--------|-----------|----------|----------|----------|
| 2024-02-26 | 编号 | 添置衣物 | | 1,000.00 | 贷方总计 | 1,000.00 |
| | | 第一条牛仔裤 | 支出:衣物 n | 500.00 | | |
| | | 第二条牛仔裤 | 支出:衣物 n | 300.00 | | |
| | | 第三条牛仔裤 | 支出:衣物 n | 200.00 | | |
| | | | 资产:活期存款 n | | 1,000.00 | |
| | | | | | | |
| 2024-04-01 | | | | | | |
| | | | n | | | |

现在: CN¥1,000.00 将来: CN¥1,000.00 已结清: CN¥0.00 已对账: CN¥0.00 预计最低: CN¥1,000.00 排序: 标准 ^

此图展示了交易日志模式下的费用：服装科目。

调整列宽

和电子表格一样，左键点击并拖动表头的分隔线能够调整账簿中各列的宽度。列头右侧的分隔线控制此列的宽度。在GnuCash中，〔描述〕和〔说明〕列会生动调整大小，使得账簿自适应地填满整个窗口，所以向左侧拖动这些列的分隔线会缩小列宽，但会在释放时移回原处。向右侧拖动分隔线会加宽显示区域并在窗口底部显示滚动条，但这一效果只会在账簿打开时保持。关闭并重新打开账簿或退出GnuCash后，账簿仍会丝毫不差地填满空间。

很多用户在调整〔余额〕列时遇到了困难，特别是在增加其宽度的时候。只需将其右边界拖出窗口就可以增加此列的宽度。将鼠标从左侧移动到分隔线处，并在鼠标指针成为双向箭头<-|->时立刻按下鼠标。

在分录模式下，账簿表头中存在不同的列标题，根据当前激活的行类型，分别对应分录行和交易行。为了调整分录行的列宽，必须首先选中分录行；为了调整交易行的列宽，必须选中交易行。

除描述列之外的列宽设置会保存在用户账簿的状态文件中，并在账簿关闭并重新打开时恢复先前调整的列宽。

使用数据录入快捷键

GnuCash提供了几组快捷键，可以使录入交易数据更加快捷。

- 日期输入框会根据当地日期格式解析输入的短日期（例如 2/15 或 15/2）。短日期代表当前年份内，或由〔首选项〕的〔日期时间〕标签页中选择的包含12个月的滑动窗口内的月份和日期。下述按键会调整当前输入的日期：
 - 〈+〉或〈=〉 后一天，〈-〉或〈_〉 前一天。
 - 〈l〉或〈}〉 上个月，〈[〉或〈{〉 下个月。
 - 〈M〉或〈m〉 设为此月第一天。
 - 〈H〉或〈h〉 设为此月最后一天。
 - 〈Y〉或〈y〉 设为此年第一天。
 - 〈R〉或〈r〉 设为此年最后一天。
 - 〈T〉或〈t〉 设为今天。

不仅可以在账簿中使用这些快捷键设置日期，在其他输入日期的地方也可以使用这些快捷键。

输入日期时，也有另一种方式可以节约时间——输入部分日期。例如，如果只输入一个数字，GnuCash会解释为当前年份当前月份的日期。如果输入日期和月份，GnuCash会自动添加年份。默认情况下会从11个月前开始选取相应的日期。这说明，如果你输入1/5（1月5日），而当前日期是2015年12月，则日期会补全为16/1/5。此默认行为可以通过〔编辑〕→〔首选项〕→〔日期与时间〕+〔日期补全〕更改。可以更改为总是在当前日历年内选择日期，或更改滑动窗口的起始位置。

- 在交易行的〔编号〕列，按下〈+〉会在先前输入的数字基础上增大交易编号。按下〈-〉会减小编号。
- 当在〔描述〕列中输入时，GnuCash会寻找先前具有以此开头的描述的交易，将其作为自动补全的提示内容。在显示补全文本时按下〈Tab〉，GnuCash会复制先前的交易，并将光标聚焦选中〔借〕或〔贷〕的数值，以便重新输入新的数值。

此处复制的是整个交易，包括一些可能不显示的部分，如〔说明〕和〔价格〕。
除非你确实想要精确地复制交易，否则请确保已经修改了所有应与之不同的内容。

你可以使用这项功能一次添加多项交易，然后再回到〔描述〕列修改交易描述。

- 〔转账〕列支持科目名称的快速补全。可以先输入科目名，随后GnuCash会补全科目名称其余部分。任何时候输入分隔符会补全科目名称的当前层级，并将光标置于开始输入科目名称下一层级的位置。例如，输入资:活可能会补全为资产:活期科目。以这种方式依序输入名称会弹出列出了全部科目的列表，其中预测项会被选中。另外，也可以输入科目名称中的任何字符序列，会弹出科目列表，其中会筛选出包含当前输入字符序列的科目。不论哪种情况，都能使用〈上箭头〉或〈下箭头〉从弹出列表中选择科目。
- 可以在任何涉及数量的列中使用内置计算器。只需输入第一个值之后再接着输入〈+〉、〈-〉、〈*〉或〈/〉。然后输入第二个值。GnuCash会在按下〈Tab〉键时执行计算并将计算结果填入数量列中。
- 所有的菜单项都具有访问键，在按住〈Alt〉键时，访问键在菜单名称中使用下划线标出。按下〈Alt〉+〈下划线字符〉弹出菜单，然后按下相应的访问键选择对应的菜单项。例如，按下〈Alt〉+〈A〉弹出功能菜单，然后按下〈P〉选择分录交易功能。一些菜单项也分配了快捷键，能够立刻执行相应的功能（一般配合使用〈Ctrl〉键）。这些快捷键都在各自菜单项的右侧列出。
- 使用下面这些按键可以在账簿中快速移动：
 - 〈Tab〉移动到下一列，〈Shift〉+〈Tab〉移动到前一列。
 - 〈Home〉移动到第一列，〈End〉移动到最后一列。
 - 〈Enter〉或〈↓〉移动到下一个交易，〈↑〉移动到上一个交易。
 - 〈Page Up〉向上移动一屏，〈Page Down〉向下移动一屏。
 - 〈Shift〉+〈Page Up〉移动到第一个交易，〈Shift〉+〈Page Down〉移动到最后一个交易。

在〔对账〕窗口中可以使用这些键盘快捷键：

- 使用〈Tab〉移动到下一个列表框，〈Shift〉+〈Tab〉移动到上一个列表框。
- 使用空格键在已对账和未对账状态间切换。
- 使用〈↑〉和〈↓〉在当前列表框中各项间移动。

简单交易还是分录交易？

GnuCash中每一个交易都至少有两项分录，但一项交易也可以拥有多于两项的分录。只有两项分录的交易称为简单交易，因为它只涉及当前科目和另一个科目。涉及三个或更多科目的交易称为分录交易。

当账簿处于基本账簿模式时，你会看到影响当前科目的分录汇总。对于简单交易，〔转账〕列会显示同其转移金额的另一个科目。对于分录交易，〔转账〕列会显示〔-- 分录交易 --〕。选择某一分录交易后，可以点击工具栏上的分录按钮查看交易下的各分录。

对于分录交易，交易的第一行是交易行，包含〔日期〕、可选的〔编号〕，交易〔描述〕，影响当前科目的总额（此处的〔总计增加〕）和在此交易发生后新的科目〔余额〕。注意，当交易展开出多个分录行时，〔转账〕列头会隐藏，对应列中也不会显示科目名称。有关此交易的详细信息，需要查看组成该交易的各项分录。

科目账簿只会显示那些包含关联到此科目的分录的交易。如果某交易有超过一项分录关联到此科目，那么在基本账簿和自动分录视图下，该交易中多项与此科目关联的分录都会显示一次。在交易日志账视图下，这样的交易只会显示一次。

交易行之下是分录行，分录行之间使用灰色线条分隔。选中分录行时，表头会变为与分录相关的各项。

2.23. 资产：活期

txns-registersplit2.gnucash - 活期存款 - GnuCash

文件(F) 编辑(E) 查看(V) 交易(N) 操作(A) 企业(B) 报表(R) 工具(T) 窗口(W) 帮助(H)

保存

关闭

创建副本

删除

确认

取消

转到空行

展开分录

科目

活期存款

| | 行为 | 备注 | 科目 | 对 | 借方 | 贷方 | |
|------------|----|-------|------------|---|--------|----------|--------|
| 2024-03-14 | | 发放工资 | | | 670.00 | | 670.00 |
| | 行为 | 税后 | 资产:活期存款 | n | 670.00 | 贷方 | |
| | | 个人所得税 | 费用:税费:所得税 | n | 180.00 | | |
| | | 医疗保险 | 费用:税费:医疗保险 | n | 90.00 | | |
| | | 社会保险 | 费用:税费:社会保险 | n | 60.00 | | |
| | | 税前 | 收入:薪酬 | n | | 1,000.00 | |
| 2024-04-01 | | | | n | | | |

现在: CN¥670.00 将来: CN¥670.00 已结清: CN¥0.00 已对账: CN¥0.00 预计最低: CN¥670.00 排序: 标准 ^

此图展示了分录表头的变化。

每项分录都包含可选的〔动作〕，也就是分录类型，可以输入也可以从下拉列表中选择。分录也包含可选的用于描述此分录的〔备注〕。每一项分录影响一个〔科目〕，可以使用下拉列表在各科目中选择。〔R〕列指示分录是否已经对账。最后两列显示了转入或转出相应科目的金额。

正如“会计概念”一节介绍的那样，每项交易中借方总额（左列）必须等于贷方总额（右列）。在上面的例子中，借方总额等于贷方总额，所以此交易是平衡的。你可能已经留意到，交易行的借方金额与〔活期〕分录行的金额相同。为什么这一金额会出现两次呢？因为交易行只总结此交易对当前科目的影响。同一项交易，在不同的科目下显示的交易行也不尽相同，同一项交易在不同科目下显示的交易行代表对各自科目的影响。你可以选中另一分录行，再点击工具栏中的跳转按钮，来观察这一点。

在这个例子中，如果你跳转到收入：薪资科目，GnuCash会在收入：薪资科目账簿中显示同一项交易：

2.24. 收入：薪资



此图展示了跳转到收入：薪资科目后的界面。

注意，交易行现在汇总的是此交易对收入：薪资科目，而非对活期科目的影响，这是因为你现在浏览的是收入：薪资科目账簿。此交易下的分录行是完全相同的，但现在交易行反映的是对收入：薪资科目的贷记。

简单交易的例子

在之前章节创建的 "gcashdata_3"的科目表基础上，双击〔活期〕资产科目。我们现在向活期科目添加一项简单交易。当你第一次在GnuCash中创建科目时，往往会从期初余额开始录入交易数据。

在第一个交易行，输入日期（如2006年3月1日）、描述（如“期初余额”），点击〔转账〕弹出列表并选择所有者权益：期初余额，输入1000元的转入金额，并按下〈Enter〉键。科目账簿现在应当如下图所示：

2.25.

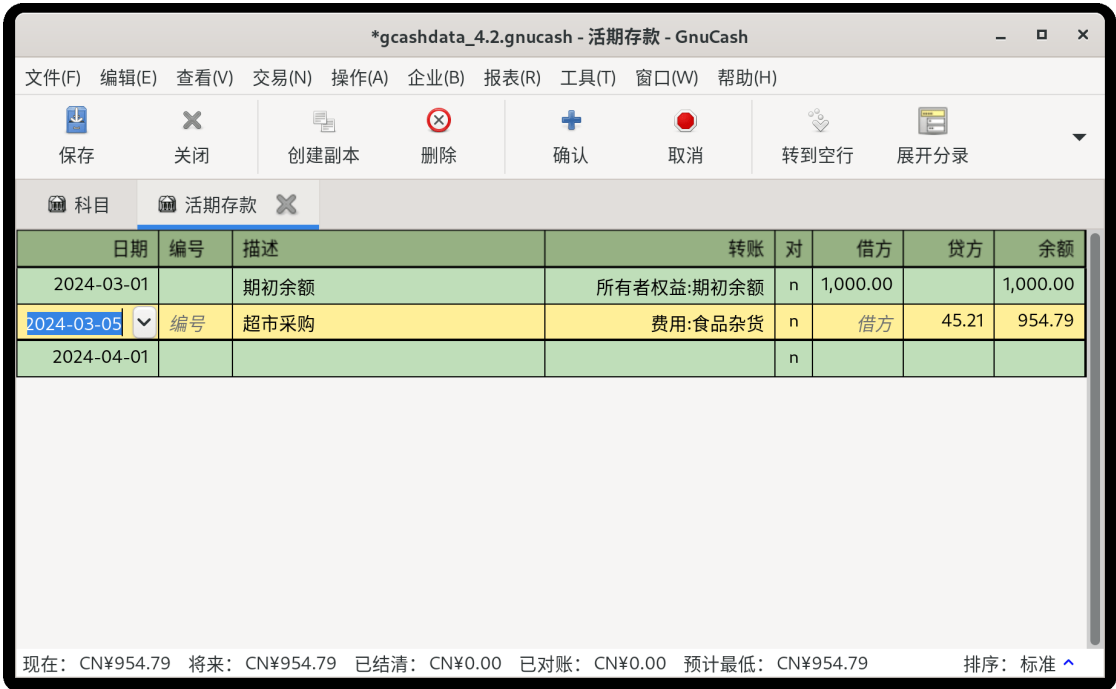


此图展示了插入期初余额交易后的资产:活期账簿。

设置科目的期初余额是简单双科目交易的一个例子，此交易影响的是资产:活期和所有者权益:期初余额科目。

再举一个简单双科目交易的例子。再添加一项交易，记录购买了45.21元的杂货。在资产:活期科目中，设置〔转账〕为费用:杂货。科目账簿现在应该显示为：

2.26.



此图显示了添加了购买杂货交易后的资产：活期账簿。

分录交易示例

有时需要在一项交易中添加三项或更多项分录，此情况往往出现在需要将交易中的来源科目或目的科目分拆进多个科目的时候。一个典型例子是在收到工资时。你带回家的钱往往少于支付给你的工资，其中的差异是由税费、养老保险和其它事项导致的。因此，收到工资这一笔交易除了资产：活期和收入：薪资这两个科目外，还会涉及其它科目。

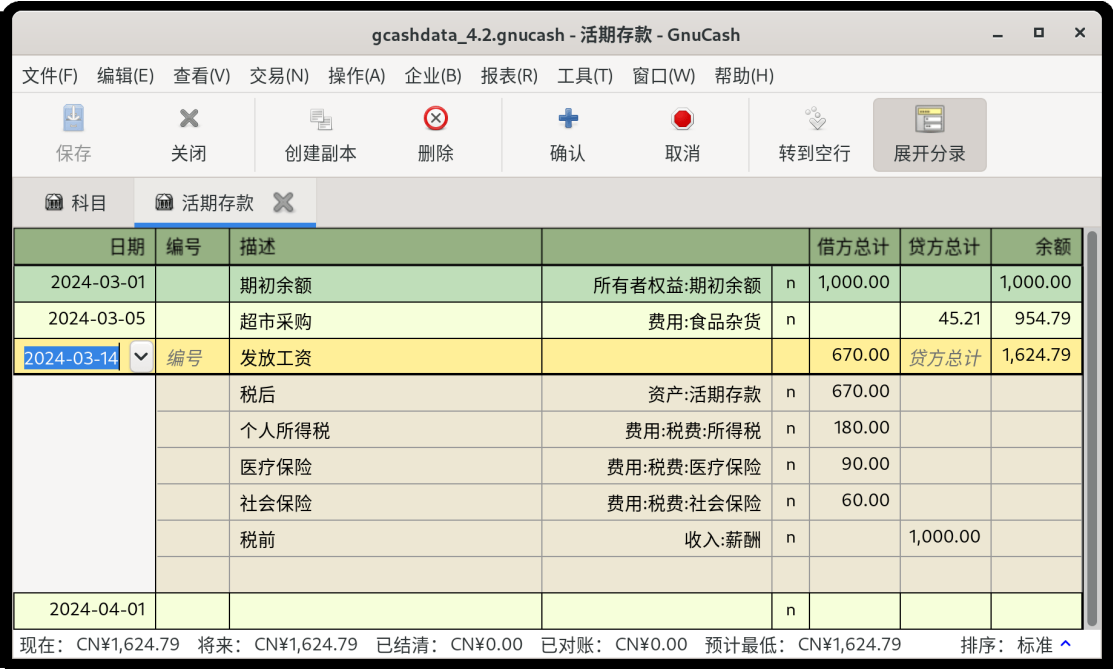
要从资产：活期科目账簿窗口添加工资分录交易，点击新交易行并点击〔分录〕。注意，如果你已经将账簿窗口视图设置为自动分录或交易日志，分录行会自动展开。在第一行输入对此交易的描述（例如“公司支付薪资”）。在其下的分录行中，逐一输入组成此交易的多项分录。要输入分录，先选择科目，再输入相应科目的变动金额。请记住，对于资产科目，在左列输入的金额会增加科目余额，在右列输入的金额会减少余额（详细了解请参考“交易”一节）。按下〈Tab〉键或点击下一个分录行，重复此过程。请注意，如果你使用键盘在不同交易间跳转，使用〈Tab〉可以在列之间跳转，使用〈Enter〉会确认此交易并在“不平衡的”科目中创建分录。

在GnuCash中创建交易时，能够以任何顺序录入分录。然而，在交易被确认之后（可能是输入光标离开此交易或者按下〈Enter〉键），所有借方分录会移动到所有贷方分录之前。

在此示例中，选择存入工资的科目（资产：活期）并输入要存入的金额（例如670元）。随后，输入各项税费的金额：费用：税费：所得税科目，180元；费用：税费：医疗保险科目，90元；以及费用：税费：社会保险，60元。最终，收入：薪资科目作为贷方，输入工资的总额。

最终的分录看起来应该像是图 2.27 “录入一笔分录交易”。记得在录入完毕后按下〈Enter〉键。不过要注意到在按下〈Enter〉之后，分录视图会“折叠”回简化的交易视图。分录仍然存在，你只需要点击〔分录〕来查看这些分录。参见“科目账簿的功能”一节了解详细信息。

2.27.



对账

一旦将交易录入进GnuCash，验证这些交易同金融机构的记录是否一致就十分关键了。这一验证过程被称为 对账，这是确保你的记录准确无误的关键。最常见的情况是，你需要同银行账单对照核对交易，当然也可以通过这一过程核对交易是否正确录入。

GnuCash会一直跟踪每个交易的对账状态，交易的对账状态在账簿中使用对账〔R〕列指示：y表示交易已经对账，n表示还没有对账，c表示已结清但没有对账。标记为已结清的分录表示得到了金融机构已接受指令的确认信息（比如，你拿到了使用信用卡付款的餐厅收据）。你可以点击〔R〕列在n和c间切换对账状态，也可以根据“对账窗口”一节置为y。

在科目窗口底部，有两个实时更新的余额（分别是〔已结清〕和〔已对账〕余额），以及〔总计〕余额。前者代表银行认为你的账户中有多少钱，而后者包含了未结清的交易。

举个例子，当你支付某些费用时，你应该向GnuCash录入交易。交易的对账〔R〕列初始为〔n〕（新添加）。新录入的交易会更新到〔总计〕余额中。随后，如果你收到了有关支付已被接受的确认信息，你可以点击交易的〔R〕列，将之更改为c（已结清）。在进行此操作后，〔已结清〕余额会更新，将此金额包括进去。在收到银行账单后，你可以打开对账窗口，对比银行账单和已在GnuCash中录入的交易，随后，你能够将〔R〕列更改为y（已对账）。

你不能直接在账簿窗口中点击〔R〕列来对账，必须使用对账窗口。一旦交易被标记为已对账，就不能再随意更改，除非在下一次对账中故意输入错误的〔期初余额〕。

特别重要的是，要充分理解对账是针对某一给定的日期进行的。当你根据某一特定日期的账单为科目对账时，你是在对 进行对账。因此，如果你添加或修改最后一次对账日期之前的交易，你的 余额会变回之前的数额。

对账窗口

对账窗口用于根据银行或其它机构发送给你的账单为某一GnuCash科目进行对账。对账是很有用的，不仅可以 根据银行的记录再次核对账簿中的交易记录，也可以得知哪些交易还未结清，比如未付款的账单。

要使用对账窗口，在科目树中选择一个科目并点击〔功能〕→〔对账〕打开对账窗口，如下所示，可以在此处输入对账信息。

2.28.



资产:活期存款 - 对账

对账信息

日期(D) 2024-04-01

期初余额 CN¥0.00

期末余额(E) 1,624.79

☐ 包含子科目(S)

记录利息收入(I)...

取消(C) 确认(O)

在初始对账窗口中，需要输入一些〔对账信息〕。

〔账单日期〕

对账所依据的银行账单的日期。

点击此输入框右侧的下拉箭头打开日历。

〔期初余额〕

此项不可编辑，显示前一次对账后的余额。此数字应当与银行账单的期初余额一致。

有时，GnuCash显示的期初余额与你的银行账单并不一致。在第一次对账，或者先前已经对账的交易被删除或取消对账的时候可能会出现这一情况。

首次对账时，〔期初余额〕会是〔0.00〕，所以此处显示的期初余额可能与该科目的“期初余额”交易不相同。在对科目对账时，“期初余额”交易也应当标记为已对账，这样做结果才会是相符的。

如果先前的交易因失误被取消对账，你可以放心与当前银行账单记录的交易一起重新对账，结果应该平衡的。

在因失误删除先前已经对账的交易时，事情有些棘手。如果你不确定删除了什么，也无法将之复原到账簿，那你可能需要创建一项虚拟交易，以完成对账。

不过，只有在个人记账时，后一种情况才不成问题；如果你是社团或公司的会计师，你应当跟审计师好好解释。

〔期末余额〕

此输入框应当按照银行账单所示金额填写期末余额。

GnuCash会自动将科目账簿下方显示的〔当前〕余额填入此输入框。

〔包含子科目〕

要把当前所选科目下的子科目的交易一同对账，请勾选此选项。注意，要使用此功能，子科目的商品应当与父科目相同。

输入利息付款

点击此按钮会打开新窗口，可以为要对账的科目输入利息交易。

〔利息付款〕窗口可能会在为银行、信用卡、共同基金、资产、应收账款、应付账款和负债科目启动对账时自动打开。如果你想为所有先前的科目禁用此行为，前往〔GnuCash首选项〕的〔账簿〕标签页，取消勾选〔自动利息转账〕选项。另外，如果只是为选择的科目禁用此行为，可在〔利息付款〕窗口中点击此科目无需自动利息转账按钮。

然后，点击确定按钮，可以看到对账窗口中列出了各项交易：

2.29.



对账窗口中的交易列表。

窗口中的两列分别叫做〔转入〕和〔转出〕，列出了此科目下所有未对账交易，接下来就是要为这些交易进行对账。其中的〔R〕列表示交易是否已对账。

现在，检查银行账单中的每一条记录，并在〔对账〕窗口中寻找其匹配项。

如果你找不到某项交易，你或许是忘记录入这项交易或不知道这项交易已经发生。你可以使用工具栏上的新建按钮或〔交易〕→〔新建〕菜单项打开账簿窗口录入遗漏的交易。新创建的交易会当在账簿中录入交易后按下〈Enter〉键时在〔对账〕窗口中出现。

当在〔对账〕窗口中发现了相应的交易时，对比此交易的金额与银行账单上的金额。如果两者不一致，你可能在GnuCash中把交易输入错了。你可以使用工具栏中的编辑按钮或〔交易〕→〔编辑〕菜单项打开账簿窗口修正错误。

如果金额相符，点击〔对账〕窗口中的相应项，所选交易的〔R〕列处会打上对勾。GnuCash会自动更新右下方〔合计〕面板中的金额。

可以使用〈上〉与〈下〉方向键在不同条目间切换，使用〈空格〉键将条目标记为已对账，使用〈Tab〉键在不同面板间切换。

接下来，对银行账单上的每一条记录重复这一操作，验证银行账单中各条记录的金额符合GnuCash中相应交易的金额，然后将GnuCash中的相应交易标记为已对账。

在任一面板中，点击其中一项交易，按下 `<Ctrl-A>`，再按下 `<空格>`，就可以勾选或取消勾选所有交易的对账状态。此操作将面板中所有交易的对账状态统一切换为勾选或取消勾选，可以重复此操作将面板中的交易切换为需要的状态。

〔对账〕窗口的底部有〔差额〕统计项，对账完成后应显示为〔0.00〕。如果此处显示了其它值，说明在GnuCash中要么遗漏了一些交易，要么一些交易的金额不正确。（也可能是银行出错了，不过基本上不可能。）

某些情况下，找出科目为什么账目不符很困难甚至不可能。如果不能修正账簿与银行账单之间的差异，可以使用工具栏上的结平按钮，自动用不平衡的金额创建交易来确保两者余额相符。

要使用此功能，尽量对账（将所有你能够辨识的交易标记为已对账）后，点击平衡按钮。GnuCash会使用特殊科目：孤立的-XXX科目（其中“XXX”代表你的货币）根据仍存在的差额创建余额平衡交易项。此时〔对账〕窗口会关闭。重新打开对账窗口后，可以检查新创建的余额平衡交易并结束对账过程。

在将所有与银行账单上的条目相对应的交易标记为已对账且差额为0.00时，点击工具栏上的完成按钮，或从菜单中选择〔对账〕→〔完成〕。〔对账〕窗口会关闭。在账簿窗口中，已对账交易的〔R〕列会变为y。

不过目前，我们尚未掌握所有信息，因此点击推迟按钮，这样就可以继续进行后面的操作了。值得注意的是〔R〕表明我们已经结清(c)了两项交易。我们还没有为这两项交易对账，但已经核对了它们，所以它们标记为已结清。你可以看到账簿窗口下方的状态栏已经汇总出了已对账金额和已结清金额（〔已结清：¥954.79 已对账：¥0.00〕）。

2.30.

*gcashdata_4.2.gnucash - 活期存款 - GnuCash

文件(F) 编辑(E) 查看(V) 交易(N) 操作(A) 企业(B) 报表(R) 工具(T) 窗口(W) 帮助(H)

保存

关闭

创建副本

删除

确认

取消

转到空行

展开分录

科目 活期存款

| 日期 | 编号 | 描述 | 转账 | 对 | 借方 | 贷方 | 余额 |
|------------|----|------|------------|---|----------|-------|----------|
| 2024-03-01 | 编号 | 期初余额 | 所有者权益:期初余额 | c | 1,000.00 | 贷方 | 1,000.00 |
| 2024-03-05 | | 超市采购 | 费用:食品杂货 | c | | 45.21 | 954.79 |
| 2024-03-14 | | 发放工资 | -- 复合分录 -- | n | 670.00 | | 1,624.79 |
| 2024-04-01 | | | | n | | | |

现在: CN¥1,624.79 将来: CN¥1,624.79 已结清: CN¥954.79 已对账: CN¥0.00 预计最低: CN¥1,624.79 排序: 标准

计划交易

计划交易功能可以帮助创建重复的交易，像是报刊订阅费、保险或税费。使用计划交易，只需录入一次关心的交易，设置开始日期、频率和些许描述等参数，然后GnuCash就会告知到底什么时候准备创建计划好的交易，并到那时候创建交易。

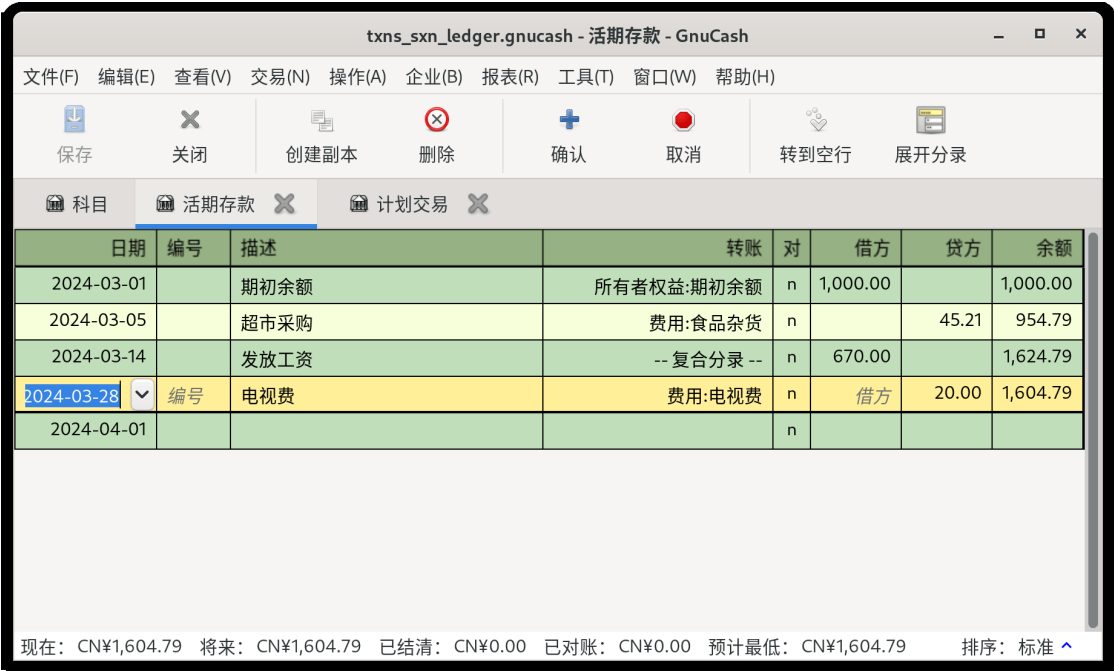
在这一部分的指南中，我们拿每月28日交纳20元的互联网接入费为例子。

在GnuCash中，有两种创建计划交易的方式，从账簿创建计划交易或是使用计划交易编辑器。

从账簿创建计划交易

在账簿中录入一系列计划交易中的第一次交易。在〔转账〕列中，输入费用:网费，如下图所示。

2.31. 1



由于我们还没有创建费用:网费科目，GnuCash会提示创建此科目。
然后，右击刚才创建的交易，选择〔计划…〕。

2.32. 2



会弹出如下窗口：

2.33.

3

创建计划交易

名称

电视费

频率

每天

开始日期

2024-03-29

结束

永不结束

结束日期

2024-04-01

执行次数

1

查看

四个月

| | 一 | 二 | 三 | 四 | 五 | 六 | 日 |
|----|----|----|----|----|----|----|----|
| | | | | | 1 | 2 | 3 |
| | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| 3月 | 11 | 12 | 13 | 14 | 15 | 16 | 17 |
| | 18 | 19 | 20 | 21 | 22 | 23 | 24 |
| | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
| | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| | 8 | 9 | 10 | 11 | 12 | 13 | 14 |
| 4月 | 15 | 16 | 17 | 18 | 19 | 20 | 21 |
| | 22 | 23 | 24 | 25 | 26 | 27 | 28 |
| | 29 | 30 | 1 | 2 | 3 | 4 | 5 |
| | 6 | 7 | 8 | 9 | 10 | 11 | 12 |
| 5月 | 13 | 14 | 15 | 16 | 17 | 18 | 19 |
| | 20 | 21 | 22 | 23 | 24 | 25 | 26 |
| | 27 | 28 | 29 | 30 | 31 | 1 | 2 |
| | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
| 6月 | 10 | 11 | 12 | 13 | 14 | 15 | 16 |
| | 17 | 18 | 19 | 20 | 21 | 22 | 23 |
| | 24 | 25 | 26 | 27 | 28 | 29 | 30 |

高级...

取消(C)

确认(O)

现在来填写这些参数，我们知道网费在每月28日缴纳，下一次交费就是在下个月了（我们已经手动录入了本月的交易）：

60

2.34.

创建计划交易

名称 每月缴纳电视费

频率 每月

开始日期 2024-04-28

结束

☒ 永不结束

☐ 结束日期 2024-04-01

☐ 执行次数 1

查看 四个月

| | 一 | 二 | 三 | 四 | 五 | 六 | 日 |
|----|----|----|----|----|----|----|----|
| 4月 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| | 8 | 9 | 10 | 11 | 12 | 13 | 14 |
| | 15 | 16 | 17 | 18 | 19 | 20 | 21 |
| | 22 | 23 | 24 | 25 | 26 | 27 | 28 |
| | 29 | 30 | 1 | 2 | 3 | 4 | 5 |
| 5月 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |
| | 13 | 14 | 15 | 16 | 17 | 18 | 19 |
| | 20 | 21 | 22 | 23 | 24 | 25 | 26 |
| | 27 | 28 | 29 | 30 | 31 | 1 | 2 |
| 6月 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
| | 10 | 11 | 12 | 13 | 14 | 15 | 16 |
| | 17 | 18 | 19 | 20 | 21 | 22 | 23 |
| | 24 | 25 | 26 | 27 | 28 | 29 | 30 |
| 7月 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| | 8 | 9 | 10 | 11 | 12 | 13 | 14 |
| | 15 | 16 | 17 | 18 | 19 | 20 | 21 |
| | 22 | 23 | 24 | 25 | 26 | 27 | 28 |
| | 29 | 30 | 31 | | | | |

高级... 取消(C) 确认(O)

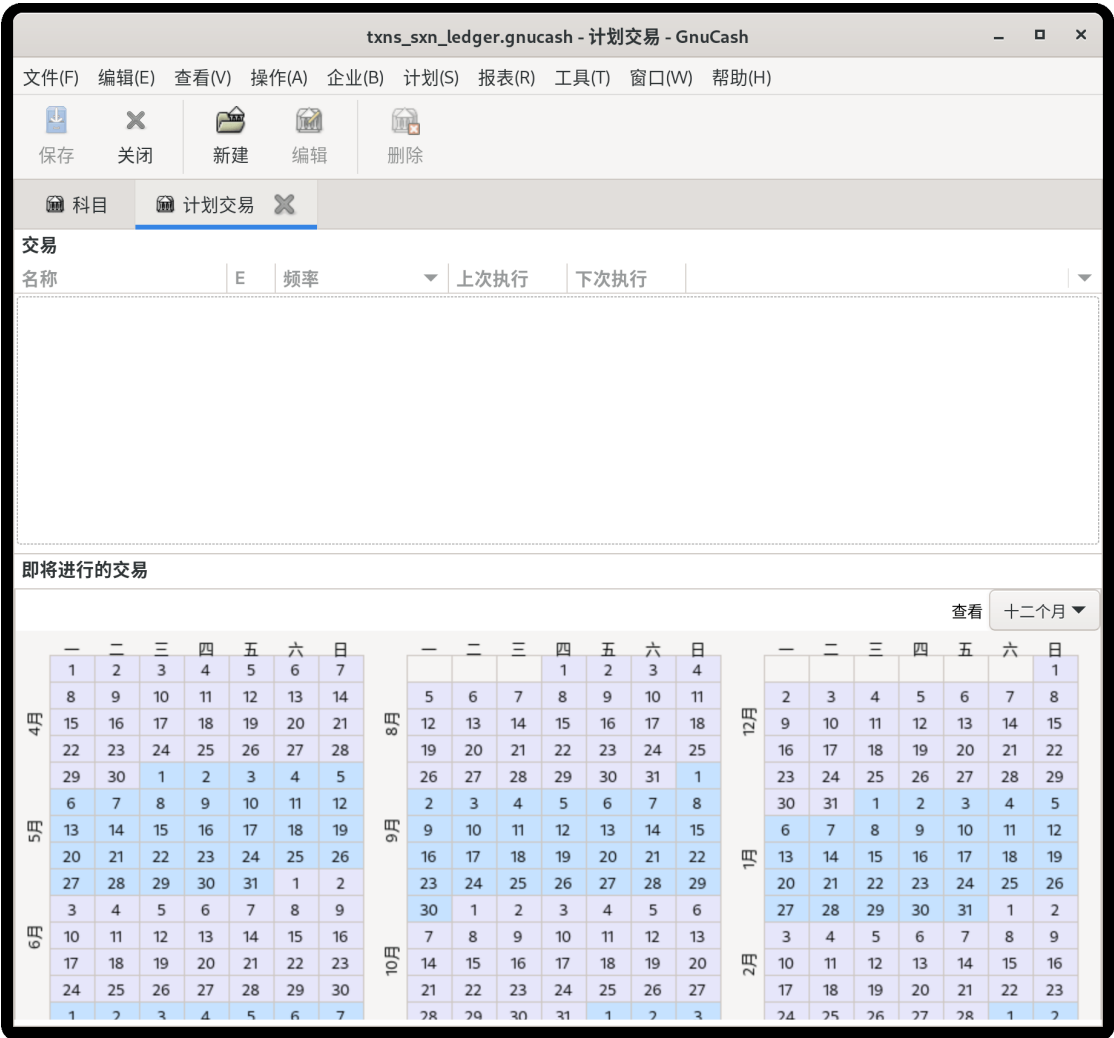
点击确认按钮，计划交易就创建好了。GnuCash现已记下了此计划交易，到下个月28日会弹出窗口询问是否要创建新的交易（可见后文的屏幕截图）。

从编辑器创建计划交易

录入计划交易的另一种方式是通过〔计划交易编辑器〕，如果一次要创建几项计划交易，那样做会快些。

选择科目窗口菜单栏中的〔功能〕→〔计划交易〕→〔计划交易编辑器〕。〔计划交易〕标签页会在当前GnuCash窗口中打开，如下图所示：

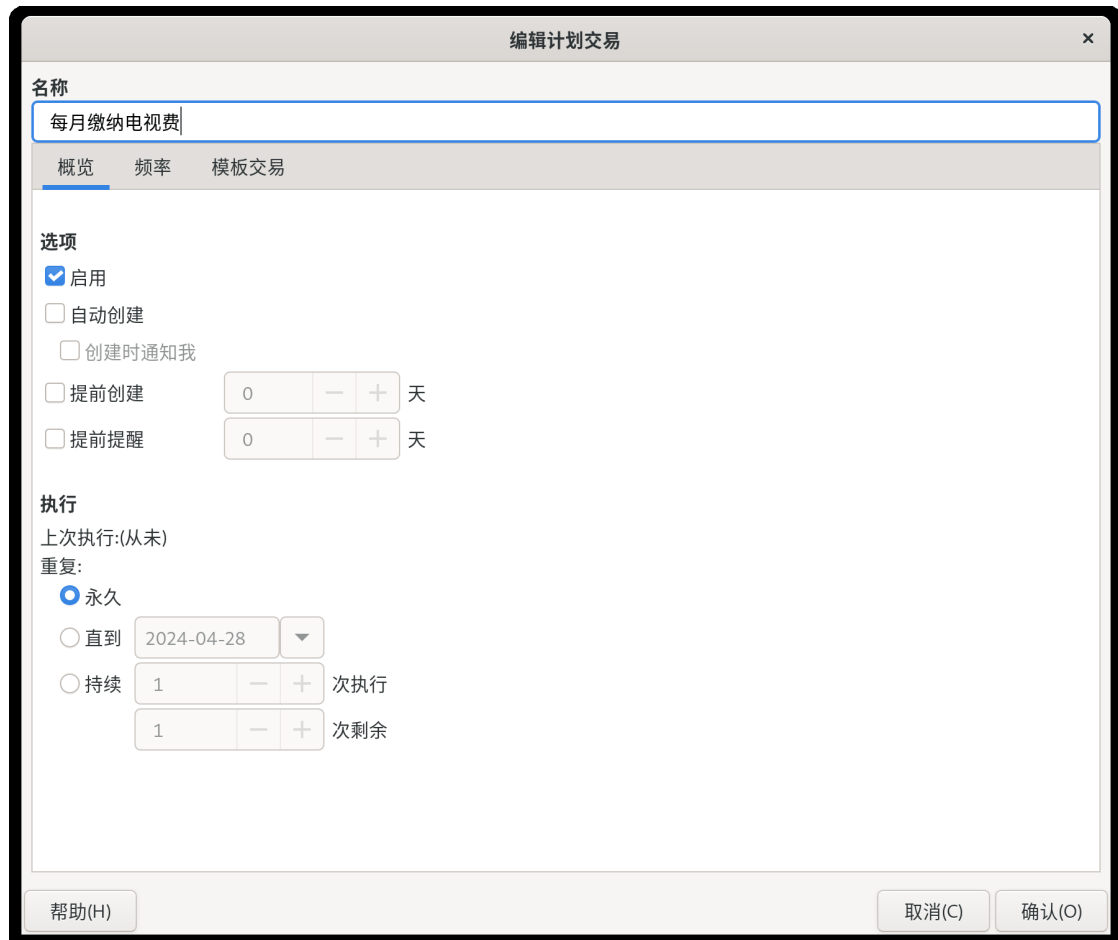
2.35.



空白的〔计划交易〕标签页。

此标签页包含一份列表（现在是空的），其中会列出全部已创建的计划交易。先来创建新的计划交易，点击工具栏中的新建按钮，会弹出如下所示的新窗口：

2.36. [编辑计划交易] [概览]



编辑计划交易

名称
每月缴纳电视费

概览 频率 模板交易

选项

☒ 启用

☐ 自动创建

☐ 创建时通知我

☐ 提前创建 0 - + 天

☐ 提前提醒 0 - + 天

执行

上次执行:(从未)

重复:

☒ 永久

☐ 直到 2024-04-28 ▼

☐ 持续 1 - + 次执行

1 - + 次剩余

帮助(H) 取消(C) 确认(O)

[编辑计划交易] 窗口，[概览] 标签页。

首先，在窗口上部为新建的计划交易输入一个名字。

这个名字仅用于在 [编辑计划交易] 窗口中区分不同的计划交易，从不会出现在账簿中。

[概览] 标签页的 [选项] 面板提供了四个选项：

[启用]

设置计划交易的状态。

[自动创建]

若勾选，GnuCash会不经提前询问而插入此交易（详见下文）。如果需要，勾选 [在创建时通知我] 选项，来在创建交易时获得提示。

[提前创建]

设置交易要提前多少天创建。

〔提醒〕

设置要提前多少天提醒。例如，这个选项可以用于当你需要使用支票支付费用的时候，提前一周提醒有助于你在截止日期前送出支票。

〔执行〕面板中的选项可以告诉GnuCash计划交易不会永远持续下去。例如，如果你在偿还贷款，你可以输入还款结束日期或贷款剩余期数。

现在选择〔编辑计划交易〕窗口中的〔频率〕选项卡。此处可以设置与时间相关的选项。

2.37. 〔编辑计划交易〕 〔频率〕

编辑计划交易

名称
每月缴纳电视费

概览 频率 模板交易

频率 每月 开始日期 2024-04-28

每 1 月

在 28 号 避开周末 不变

查看 十二个月

| 一 | 二 | 三 | 四 | 五 | 六 | 日 |
|----|----|----|----|----|----|----|
| 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| 8 | 9 | 10 | 11 | 12 | 13 | 14 |
| 15 | 16 | 17 | 18 | 19 | 20 | 21 |
| 22 | 23 | 24 | 25 | 26 | 27 | 28 |
| 29 | 30 | 1 | 2 | 3 | 4 | 5 |

| 一 | 二 | 三 | 四 | 五 | 六 | 日 |
|----|----|----|----|----|----|----|
| 8 | 9 | 10 | 11 | 12 | 13 | 14 |
| 15 | 16 | 17 | 18 | 19 | 20 | 21 |
| 22 | 23 | 24 | 25 | 26 | 27 | 28 |
| 29 | 30 | 31 | 1 | 2 | 3 | 4 |
| 5 | 6 | 7 | 8 | 9 | 10 | 11 |

| 一 | 二 | 三 | 四 | 五 | 六 | 日 |
|----|----|----|----|----|----|----|
| 7 | 8 | 9 | 10 | 11 | 12 | 13 |
| 14 | 15 | 16 | 17 | 18 | 19 | 20 |
| 21 | 22 | 23 | 24 | 25 | 26 | 27 |
| 28 | 29 | 30 | 31 | 1 | 2 | 3 |
| 4 | 5 | 6 | 7 | 8 | 9 | 10 |

| 一 | 二 | 三 | 四 | 五 | 六 | 日 |
|----|----|----|----|----|----|----|
| 6 | 7 | 8 | 9 | 10 | 11 | 12 |
| 13 | 14 | 15 | 16 | 17 | 18 | 19 |
| 20 | 21 | 22 | 23 | 24 | 25 | 26 |
| 27 | 28 | 29 | 30 | 31 | 1 | 2 |
| 3 | 4 | 5 | 6 | 7 | 8 | 9 |

帮助(H) 取消(C) 确认(O)

〔编辑计划交易〕窗口，〔频率〕标签页。

〔频率〕

设置交易的基本频率，提供的选项有〔单次〕、〔每天〕、〔每周〕、〔每半月〕和〔每月〕。在此例中，我们设为〔每月〕。

也可以更改〔每…〕选项，将交易的发生周期设为〔频率〕中所提供选项以外的值，详见下文。

〔起始日期〕

设置交易开始时间。在此例中设为下个月。

〔每…〕

此选项可以倍乘〔频率〕选项设置的交易周期。例如，要每3周执行交易，将〔频率〕设为〔每周〕再将〔每…周〕设为〔3〕。

〔在〕和〔排除周末〕

设置交易计划在月中的哪一天，并控制当那一天是周末时GnuCash怎样处理。

我们知道网费在每月28号交，所以〔起始日期〕是11月1号（假设下个月是11月），〔频率〕是〔每月〕。在此例中，网费交易会添加，所以无需提前创建，也不用给出结束日期。

当此标签页中的选项填写完毕后，GnuCash会在下方的日历中标出将来的交易会在哪些日期执行。

最后选择〔交易模板@〕标签页，像在账簿中那样在下面录入交易，只是不填日期。别忘了点击确认图标以确认录入交易。

现在，你看到的窗口应该像这样：

2.38. [编辑计划交易] [交易模板]

编辑计划交易

名称

每月缴纳电视费

概览

频率

模板交易

编辑(E) 交易(T) 查看(V) 操作(A)

创建副本

删除

确认

取消

转到空行

| 日期 | 编号 | 描述 | | | 借方总计 | 贷方总计 |
|-----|----|-----|---------|---|------|------|
| | | 说明 | | L | | |
| 按计划 | | 电视费 | | | | |
| | | | | | | |
| | | | 资产:活期存款 | n | | 20 |
| | | | 费用:电视费 | n | 20 | |
| 按计划 | | 描述 | | | 借方总计 | 贷方总计 |
| | | 说明 | | | | |
| | | | | n | | |

排序: 标准 ^

输入交易参考号, 由所有分录行共享。

帮助(H)

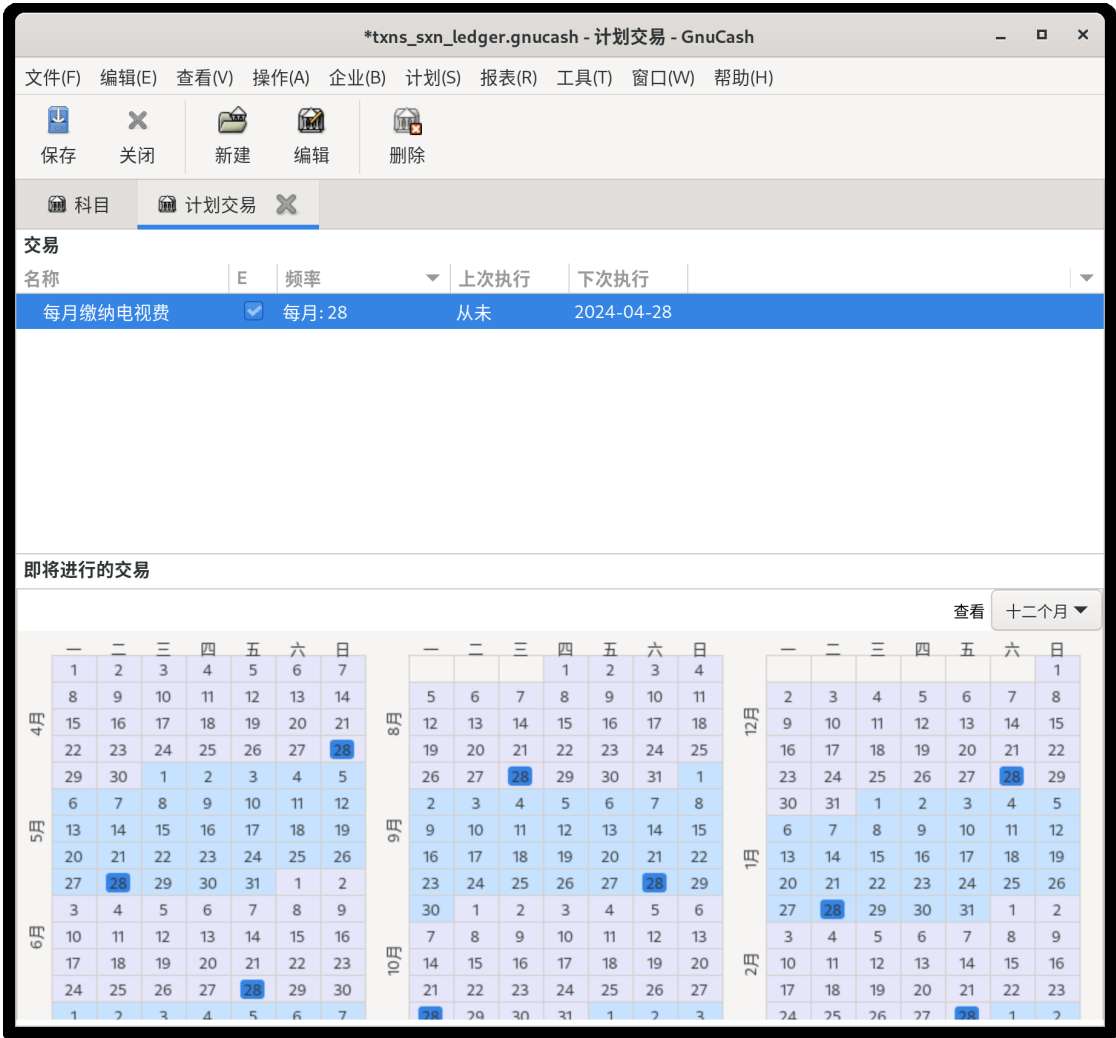
取消(C)

确认(O)

[编辑计划交易] 窗口, [交易模板] 标签页。

现在点击确定, 回到 [计划交易] 标签页, 现在 [交易] 列表中会显示刚才添加的计划交易。

2.39.



点击日历区域中的某天，一个跟随鼠标的小窗口会显示鼠标指向的这天有哪些计划的交易。再次点击鼠标来隐藏小窗口。

现在可以关闭〔计划交易〕标签页，并〔保存〕账簿。

下文是解释性的描述，不代表当下就会录入GnuCash数据库。在每一次计划交易运行时，下面的对话框就会出现。

从现在开始，当GnuCash启动且计划好的交易需要录入时，会看到〔自从上次运行…〕窗口，其中汇总了计划好的交易操作（〔提醒〕、〔将要创建〕等等，GnuCash手册对每一选项进行了更详细的介绍）：

2.40.



若需要，可以点击〔状态〕列更改计划交易状态。然后点击确定关闭〔自从上次运行…〕窗口并应用接下来的操作。点击〔取消〕跳过接下来录入账簿的操作。

如果〔状态〕列设为〔将要创建〕，浏览相应科目账簿，会发现交易已经被创建了。

若勾选〔自从上次运行…〕窗口右下角〔检查创建的交易〕选项，点击确定后，GnuCash会自动打开每个被录入计划交易的科目。

如果你在这一步录入了4月28号的交易，请恢复到最后保存的GnuCash文件，选择〔文件〕→〔打开〕并选择之前保存的 "gcashdata_3" GnuCash 文件。

步步为营

本章前面的小节讨论了处理交易的概念和方法。这一部分会以前文构建的科目表为基础进行拓展，包括设置期初余额、添加交易和计划交易。

打开GnuCash文件

先是打开之前存储的数据文件 "gcashdata_3emptyAccts"，并直接将之存储为 "gcashdata_4"。主窗口应当如下图所示：

2.41.



此图展示了本小节操作的最初状态。

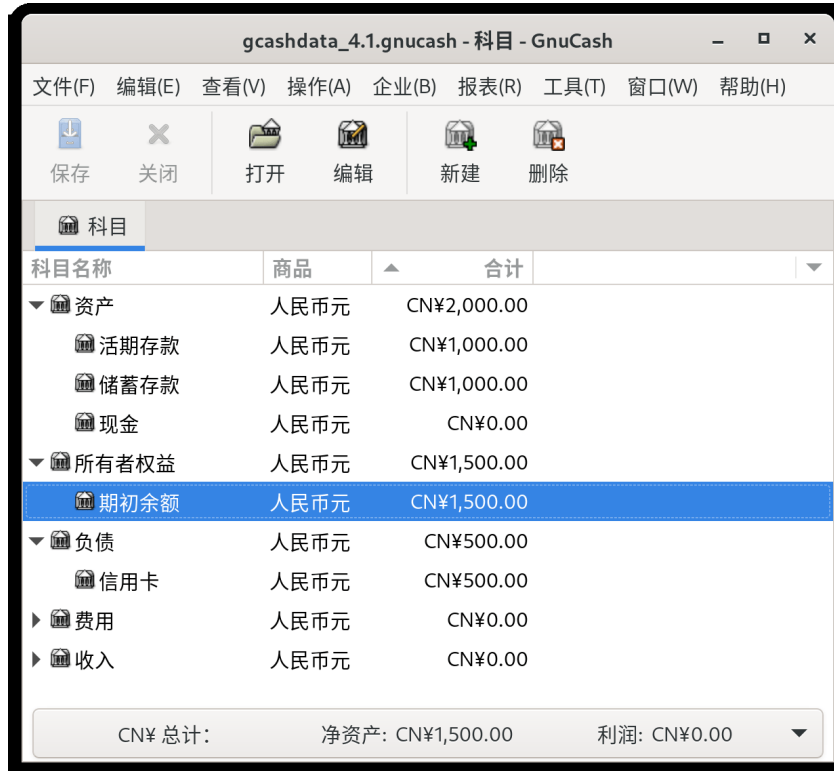
期初余额

正如前文“简单交易的例子”一节以资产:活期为例所述，科目的期初余额交易一般设所有者权益:期初余额为转账科目。要为科目表添加交易，就从为科目设置期初余额开始。假定储蓄科目中有1000元余额，信用卡有500元的欠款。

- 1. 打开资产:储蓄科目账簿。从菜单中选择〔查看〕确定处于基本账簿模式下。后文会以其它模式查看交易，但现在先使用默认的基本模式录入交易。
- 2. 在资产:储蓄科目账簿窗口中，录入一项基本双科目交易，将期初余额设为1000元，由所有者权益:期初余额转账。记住，基本交易是在单个来源科目和单个目的科目之间进行转账。然后确认交易（按〈Enter〉键或点击确认图标）。
- 3. 在资产:活期科目账簿窗口中，录入一项基本双科目交易，将期初余额设为1000元，由所有者权益:期初余额转账。
- 4. 在负债:信用卡科目账簿窗口中，录入基本双科目交易，将期初余额设为500元，由所有者权益:期初余额转账。此操作将500元录入为信用卡科目负债的增加（或者说是期初余额科目金额减少），因为这是你借入的一笔钱。然后确认交易（按〈Enter〉键或点击确认图标）。

你现在应该已经设置好了3个科目的期初余额，分别是资产:活期、资产:储蓄和负债:信用卡。

2.42.



| 科目名称 | 商品 | 合计 |
|---------|------|-------------|
| ▼ 资产 | 人民币元 | CN¥2,000.00 |
| 活期存款 | 人民币元 | CN¥1,000.00 |
| 储蓄存款 | 人民币元 | CN¥1,000.00 |
| 现金 | 人民币元 | CN¥0.00 |
| ▼ 所有者权益 | 人民币元 | CN¥1,500.00 |
| 期初余额 | 人民币元 | CN¥1,500.00 |
| ▼ 负债 | 人民币元 | CN¥500.00 |
| 信用卡 | 人民币元 | CN¥500.00 |
| ▶ 费用 | 人民币元 | CN¥0.00 |
| ▶ 收入 | 人民币元 | CN¥0.00 |

CN¥ 总计: 净资产: CN¥1,500.00 利润: CN¥0.00

此图展示了设置期初余额后的结果。

更多交易示例

接下来添加更多交易用以模拟一个月内的花销。在这个月里，电费花了78元，话费花了45元，房租花了350元，这些都是用银行活期账户支付的。我们还花了45.21元购置杂货，收到了670元的薪水，也支付了这个月的网费。最后，从储蓄账户向活期账户转账100元。

1. 打开费用:电费科目账簿，录入一项简单双科目交易，内容为本月末（如：2006年3月28日）支付78元电费。输入描述（如：国家电网）和交易编号（如：102）。〔转账〕科目应为资产:活期。
2. 打开资产:活期科目，录入一项基本双科目交易，在本月末（如：2006年3月28日）支付45元电话费。输入描述（如：中国移动）和交易编号（如：103）。〔转账〕科目应为费用:话费。注意，你既可以从借方（花费科目）录入这项支出交易，也可以从贷方（资产科目）录入。
3. 打开费用:房租科目，录入一项基本双科目交易，在本月末支付350元租金（如：2006年3月28日）。输入描述（如：四月房租）和交易编号（如：104）。〔转账〕科目应为资产:活期。
4. 使用工具栏上的创建副本按钮复制此交易。先点击当前的房租交易，再点击〔创建副本〕图标。输入的交易日期为从今往后推一个月（如：2006年4月28号）。注意GnuCash使用蓝色的线条分隔将来的交易与当前的交易。如此可以在交易实际发生之前就录入交易。

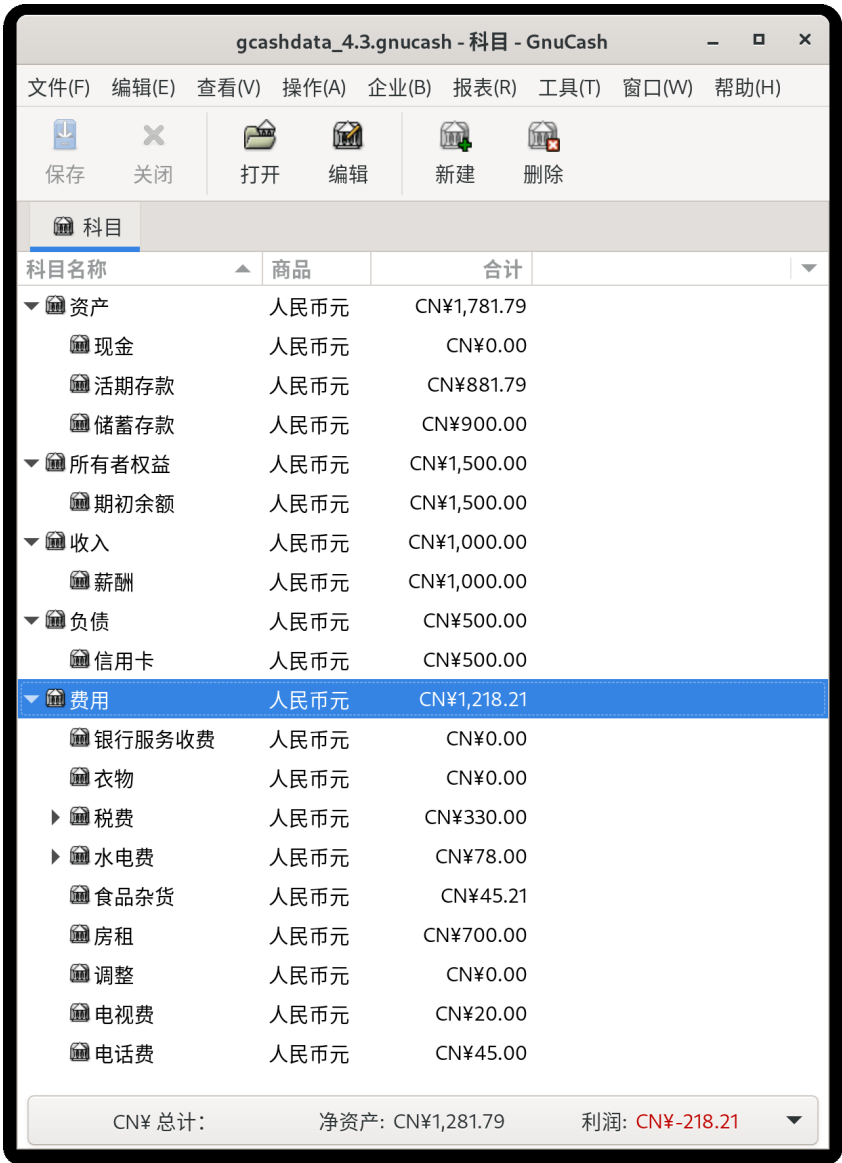
你也可以设置计划交易记录房租支出，因为在可预见的将来，房租金额大概是不变的。

· 先点击当前（4月28日）的房租交易，再点击〔计划〕图标。

· 更改为〔每月〕，如果需要也可以更改描述，点击确定。

5. 要把钱从储蓄账户转到活期账户，打开资产：储蓄科目账簿，添加一项新的交易，将其〔转账〕设为资产：活期，金额为100元（日期为2006年3月6日）。
6. 添加另一项交易，描述发生在3月5日价值45.21元的杂货购买，这是简单双科目交易的另一个例子。在资产：活期科目中，将〔转账〕设为费用：杂货。
7. 要从资产：活期科目账簿窗口添加薪资交易，点击新交易行，再点击〔分录〕。首先在第一行输入交易描述（如：“公司发放工资”）和日期（3月14日）。在下方的分录行中，输入向资产：活期存入的金额（如：670元）。随后录入多项税费缴纳分录：费用：税费：所得税（如：180元）、费用：税费：医保（如：90元）和费用：税费：社保（如：60元）。最后，录入自收入：薪资科目的取出转账，金额为薪资的原总额（如：1000元）。
8. 你也需要记录本月28号支出了20元网费。

2.43.



此图展示了按“交易”一节操作后的期末余额。

保存文件

在接下来介绍报表功能之前，先保存GnuCash数据文件（"gcashdata_4"）。

报表

不过，在电脑屏幕上展示这些枯燥的数据并不能令会计师满意，因此GnuCash提供了大量可高度自定义的报表。

有关报表的更多信息可见第 10 章 Reports。

先来看看〔现金流量〕报表和〔交易汇总〕报表。

1. 首先来看三月份的〔现金流量〕报表。

由〔报表〕→〔收入与花费〕→〔现金流量表〕可以打开现金流量表。

2.44.

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

Selected Accounts

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

| | |
|--|---------------------|
| Money into selected accounts comes from | |
| Income:Salary | USD 1,000.00 |
| Money In | USD 1,000.00 |
| Money out of selected accounts goes to | |
| Expenses:Electricity | USD 78.00 |
| Expenses:Groceries | USD 45.21 |
| Expenses:Internet | USD 20.00 |
| Expenses:Phone | USD 45.00 |
| Expenses:Rent | USD 350.00 |
| Expenses:Taxes:Federal | USD 180.00 |
| Expenses:Taxes:Medicare | USD 90.00 |
| Expenses:Taxes:Social Security | USD 60.00 |
| Money Out | USD 868.21 |
| Difference | USD 131.79 |

此图展示了按“交易”一节操作后得到的现金流量表。

要自定义报表，在报表中右击并选择〔报表选项〕，然后设置起止日期并指定想在报表中展示的科目。

2. 现在来看活期科目的交易汇总表。

从〔报表〕→〔交易汇总表〕打开交易汇总表。

2.45.

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

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

此图展示了三月份期间活期科目的交易汇总表。

3. 现在更改交易汇总表，令其仅展示各个费用科目。

2.46.

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

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

此图展示了三月份期间费用科目的交易汇总表。

第 3 章 Importing Data into GnuCash

This chapter will detail procedures for importing data into GnuCash.

Importing Transactions from Files

Outdated

This is an older version of “Common Transaction Operations” in the manual [<https://www.gnucash.org/docs/v5/C/gnucash-manual/trans-import.html>]. Better continue there until this update anomaly is resolved.

Imported transactions will generally be to a specific account in your account tree. In the following this will be referred to as the `import` or `base` account. It may or may not be specified in the data being imported, depending on the import format. It is usually the first split of a transaction being imported.

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

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

File Import Formats

Gnucash allows transactions to be imported in the following formats:

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

These import methods can be accessed from `[File] → [Import ...]` .

Import Matcher

Several of the Import Assistants use an Import Matcher to implement a Bayesian approach to assign destination accounts, if such accounts are not specified in the

imported data, to each imported transaction based on the previous import history of the import account. It also attempts to match the transactions being imported to any existing transactions based on the date and the description fields .

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

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

Assign a Destination Account to a Single Transaction

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

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

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

Assign a Destination Account to Multiple Transactions

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

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

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

To select rows either:

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

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

Completing the Import

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

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

Import QIF

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

To import QIF files:

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

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

This image shows the start of the QIF Import assistant.

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

Review the GnuCash accounts to be created.

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

making changes, and proceed through a similar category matching process. QIF income and expense categories import as GnuCash income and expense accounts. Make changes to these account names if necessary, and click Next to continue.

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

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

Tell GnuCash to import the data.

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

Import OFX/QFX

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

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

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

Import CSV

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

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

Load and Save Settings

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

Account

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

File Format

This section allows you to define whether the file has:

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

Miscellaneous

The miscellaneous settings allow you to set:

- Encoding This is usually the UTF-8 variant for your locale;

- **Date Format** This does not default to the Locale setting so check it matches the data you are importing;
- **Currency Format**;
- **Leading Lines to Skip** Can be set to skip a number of non-transaction lines preceding the actual data. This can include a field name header or an introduction section in the import file;
- **Trailing Lines to Skip** Can be set to skip a number of non-transaction lines following the actual data. This can include a summary section;
- **Skip alternate lines** When set only every other line will be imported, starting from the first line not skipped with the "Leading Lines to Skip" option;

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

Import Panel

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

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

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

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

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

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

Import MT940

Use this option to import MT940 data.

Import MT942

Use this option to import MT942 data.

Import DTAUS

Use this option to import DTAUS data.

第 4 章 Configuring GnuCash

This chapter will detail preference settings for GnuCash.

Overview

Configuring is fun.

Main Configuration Screens

GnuCash stores settings in two main locations.

Preferences: Accounting Period Tab

Stuff about the Accounting Period tab

Preferences: Accounts Tab

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

Preferences: Online Banking

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

部分 II. 常规使用

目录

| | |
|---------------------|-----|
| 5. 活期账户与现金 | 87 |
| 创建科目 | 87 |
| 录入存款 | 87 |
| 录入取款 | 88 |
| 使用ATM机取出现金 | 88 |
| 科目对账 | 88 |
| 步步为营 | 89 |
| 期初余额 | 89 |
| 添加交易 | 90 |
| 期初余额 | 92 |
| 保存文件 | 95 |
| 报表 | 96 |
| 6. 费用 | 100 |
| 概念 | 100 |
| 创建科目 | 100 |
| 建立简单的费用科目 | 100 |
| 复杂的费用科目结构 | 100 |
| 录入费用交易 | 100 |
| 有关费用科目的其它注意事项 | 101 |
| 7. 信用卡 | 102 |
| 概念 | 102 |
| 创建科目 | 102 |
| 简易科目设置 | 102 |
| 完整科目设置 | 103 |
| 记录付款 | 103 |
| 记录还款 | 104 |
| 步步为营 | 106 |
| 打开GnuCash文件 | 106 |
| 购买 | 107 |
| 退款 | 108 |
| 缴纳利息 | 109 |
| 对账 | 110 |
| 还款 | 111 |
| 保存文件 | 113 |
| 报表 | 113 |
| 8. 贷款 | 117 |
| 基本概念 | 117 |
| 术语 | 117 |
| 创建科目 | 118 |
| 计算器 | 118 |

| | |
|-------------------------|-----|
| 示例：每月还款 | 120 |
| 示例：贷款总期数 | 120 |
| 进阶：计算详细过程 | 120 |
| 怎样：记录房屋按揭贷款 | 122 |
| 怎样：记录汽车贷款 | 125 |
| 怎样：记录给朋友的私人借款 | 126 |
| 借款约定 | 126 |
| 借款科目 | 127 |
| 借出钱款 | 127 |
| 收到第一笔还款 | 128 |
| 收到第二笔还款 | 129 |
| 怎样：为贷款账单对账 | 131 |
| 怎样：出售房屋或汽车 | 132 |
| 简单交易 | 132 |
| 更复杂的交易 | 133 |
| 9. 投资 | 135 |
| 基本概念 | 135 |
| 术语 | 135 |
| 投资类型 | 136 |
| 创建科目 | 137 |
| 使用预置的投资科目 | 137 |
| 手动创建投资科目 | 138 |
| 自建科目示例 | 138 |
| 计息科目 | 140 |
| 创建科目 | 140 |
| 示例 | 140 |
| 创建投资组合 | 143 |
| 创建股票和公募基金科目 | 144 |
| 股票科目示例 | 144 |
| 购入份额 | 148 |
| 录入已持有的份额 | 148 |
| 新购份额 | 149 |
| 设置份额价格 | 151 |
| 价格数据库初始设置 | 151 |
| 手动设置股票价格 | 152 |
| 配置自动获取报价 | 153 |
| 显示份额价值 | 153 |
| 报表中的“价格来源” | 154 |
| 卖出份额 | 156 |
| 卖出份额时手动计算资本损益 | 157 |
| 使用批次卖出份额并自动计算资本损益 | 164 |
| 分红 | 176 |
| 现金分红 | 176 |

| | |
|---|-----|
| 红利再投 | 177 |
| 资本退回 | 178 |
| 拆股与合股 | 178 |
| 简单的拆股 | 179 |
| 较为复杂的合股 | 181 |
| 10. Reports | 184 |
| Overview | 184 |
| Saved Reports Configurations Concepts | 184 |
| Standard Reports Overview | 185 |
| Transaction Report | 185 |
| Assets & Liabilities Group | 186 |
| Budget Group | 191 |
| Business Group | 192 |
| Income and Expense Group | 193 |
| Sample & Custom Group | 198 |
| Other Reports | 199 |
| Using the Transaction Report | 200 |
| 11. 资本利得 | 203 |
| 基本概念 | 203 |
| 估值 | 203 |
| 建立科目 | 204 |
| 示例 | 204 |
| 未实现损益 | 205 |
| 出售 | 206 |
| 缴税 | 208 |
| 12. 多货币 | 210 |
| 基本概念 | 210 |
| 手动记录外币交易 | 210 |
| 建立科目 | 210 |
| 记录并更新货币汇率 | 213 |
| 记录通过外币的购买 | 222 |
| 跟踪外币投资 | 228 |
| 对账外币账单 | 229 |
| 使用贸易科目自动记录外币交易 | 229 |
| 贸易科目的概念 | 230 |
| 使用贸易科目记录外币交易 | 230 |

第 5 章 活期账户与现金

这一章会给出使用GnuCash管理活期账户的具体信息。管理活期账簿是跟踪个人财务状况的第一步。使用GnuCash管理起来比传统的纸制账簿更加方便。

在正式开始之前，先来回想一下“交易”一节介绍的内容。借助GnuCash的自动补全功能，常见交易的数据录入工作实际很简单。GnuCash实时更新每个科目的余额，因此科目对账也相当方便。复式记账法要求为每项交易指定转出科目，有助于你理清各项开销，也方便查明在各个方面都花了多少钱。

在熟练使用GnuCash管理活期账户和其它银行账户之后，你可能想要再跟踪其它金融账户。第6章至第9章详细讨论了管理其它账户的方法。

创建科目

管理活期账户的第一步就是创建必要的科目。你既可以使用默认的GnuCash科目，也可以自行创建。有关创建新科目的具体步骤，请参考“基本顶级科目”一节。有关由其它软件导入科目结构的方法，请参考GnuCash帮助手册。

现在先来创建接下来要用到的银行账户科目。你需要为每个想要跟踪的实际银行账户创建一个GnuCash银行类型科目。不论是自行创建科目还是使用GnuCash的默认科目，确保每个银行科目都拥有一项期初余额交易。最简单的方法就是直接使用最近一张银行账单的余额作为期初余额。你可以在〔建立科目结构〕向导的科目信息窗口中输入期初余额，也可以直接在科目中手动录入交易。手动录入的交易应设为从〔期初余额〕科目（科目类型为所有者权益）向银行科目的转账。

你要跟踪的银行科目一般有：

- 活期存款 - 任何提供支付能力的机构账户。
- 储蓄存款 - 付息的机构账户，通常用于期限比活期账户更长的存款。

影响这些银行科目的常见交易有付款和存款。付款指将钱从银行科目中转出，通常转至费用科目。存款指将钱转入银行科目，通常转自收入科目。你需要创建收入和费用科目跟踪钱从哪里来，又到了哪里去。请记住，平衡的交易中来源科目与目的科目的转账总额相等。所以，你如果向活期科目存款，也必须输入款项来自于哪个科目。你如果使用活期科目支付账单，也必须输入款项转去了哪个科目。

录入存款

存款交易大多录入为由收入科目向银行科目的转账。收入来源各式各样，最好为不同的收入来源创建各自的收入类型科目。例如，你的收入可能主要来自于工资，但你可能也会收到储蓄存款的利息。在这种情况下，你应该为工资建立一个收入科目，并为收到的利息建立另一个收入科目。

确保核对纳税申报单，并在建立应税收入科目时在〔收入纳税信息〕对话框（〔编辑〕→〔报税选项〕）中指定适合的纳税类型。一些收入类型，如礼品收入，可能算作应税的，所以请查阅适合的税务法规确定某项收入是否应税。跟踪资本利得收入的方法请参考第 11 章 资本利得。

在着手录入工资存款时，请明确记录的细致程度。最基本的是录入你的净工资，就像你会在纸制账簿上记录的那样。这是最简单的记录方式，不过你如果录入包含扣除额的毛工资，则可以使用GnuCash分析出更多信息。录入扣除额是要花点工夫，但你在录入整年的预扣税信息后，可以随时在GnuCash中生成实用的税务状况报告。这些报表能够帮助你确定是否已足额纳税，也可以帮助预估税单。如果你不确定记录的细致程度，则可以先从录入净工资开始。你如果想记录更详细的信息，随时可以回头编辑这些交易。

录入取款

不论你出于什么原因要从银行账户中取款时，你就在将钱从银行科目转至其它地方。在GnuCash中，这里所说的“其它地方”也是对应一个科目。你用于记录花销的科目建立得越细致，你就能将钱款的去向分析得越详尽。

取款有多种形式。ATM机取款是最常见的交易形式之一。填写支票是一种取款用以支付账单、采购商品或捐款的形式。根据银行账户的不同，你可能还要支付服务费，银行会从账户中扣除这笔钱。将款项转出至其它账户也是一种取款。下面将逐一介绍这几种取款的形式，并说明在GnuCash中如何记录各式取款。

使用ATM机取出现金

现金取款处理为从银行科目到现金科目的转账。GnuCash特地提供现金类型科目用于跟踪现金交易，所以你应当建立现金科目记录ATM机和其它现金取款。

在现金科目中，可以按不同的细致程度录入交易。最基本的细致程度是只是简单地录入从活期科目向现金科目的转账。这样做，你可以得知某天你从活期账户中取出了多少钱，但你无从得知这些现金花到哪里去了。再费点心思，你可以使用现金科目记录现金支出，这样一来，你就可以看到现金花到哪里去了。你应将这些支出录入为从现金科目到费用科目的转账。

一些人会记下每一笔现金支出，但这样会花很多精力。一种更简单的方式是只记录手头有相应收据的支出，之后再调整科目余额，使之与钱包现金总额相符。

最好至少为取款建立现金科目。你如果以后决定跟踪现金的去处，就可以为花出去的钱录入交易。你可以自由决定想要的记录细致程度。

科目对账

借助GnuCash，根据银行月账单为银行科目对账变得轻松许多。“对账”一节给出了根据银行月账单进行交易对账的步骤。这是每个月都要做的主要对账任务。

不过该怎么处理其它你创建的科目呢？这些科目也应当对账吗？如果你得到了某一科目对应的账单，那你就该考虑为这个科目对账了。常见的例子有，活期账户账单、储蓄账户账单和信用

卡账单。信用卡账单和信用卡交易会在第 7 章 信用卡介绍，所以如果你想在GnuCash中跟踪信用卡，不妨看看那一章给出的操作指导。

收入和费用科目通常不对账，因为没有账单可用作对账依据。出于同样的原因，你也不必为现金科目对账。不过，你可能要不时调整现金科目的余额，以使你手头实际的现金总额与现金科目的余额相符。下面一节会介绍调整余额的方法。

步步为营

在“交易”一节，你已经向活期科目录入了一些交易。在本章中，我们会添加更多的交易，并为其对账。

期初余额

现在，先打开在上一章保存的GnuCash数据文件（"gcashdata_4"）。科目表应该如下图所示：

5.1.

*gcashdata_4.7.gnucash - 科目 - GnuCash

文件(F) 编辑(E) 查看(V) 操作(A) 企业(B) 报表(R) 工具(T) 窗口(W) 帮助(H)

保存

关闭

打开

编辑

新建

删除

科目

| 科目名称 | 类型 | 商品 | 合计 |
|-------|-------|------|-------------|
| 资产 | 资产 | 人民币元 | CN¥1,781.79 |
| 流动资产 | 资产 | 人民币元 | CN¥1,781.79 |
| 储蓄存款 | 银行 | 人民币元 | CN¥900.00 |
| 活期存款 | 银行 | 人民币元 | CN¥881.79 |
| 所有者权益 | 所有者权益 | 人民币元 | CN¥1,500.00 |
| 费用 | 费用 | 人民币元 | CN¥1,218.21 |
| 房租 | 费用 | 人民币元 | CN¥700.00 |
| 税费 | 费用 | 人民币元 | CN¥330.00 |
| 水电费 | 费用 | 人民币元 | CN¥78.00 |
| 食品杂货 | 费用 | 人民币元 | CN¥45.21 |
| 电话费 | 费用 | 人民币元 | CN¥45.00 |
| 电视费 | 费用 | 人民币元 | CN¥20.00 |
| 收入 | 收入 | 人民币元 | CN¥1,000.00 |
| 负债 | 负债 | 人民币元 | CN¥500.00 |

CN¥ 总计:

净资产: CN¥1,101.79

利润: CN¥-398.21

此图展示了科目表。

添加交易

1. 现在从主窗口打开活期科目。在上一章中，你录入了工资存入活期银行账户的交易。现在录入另一种类型的存款交易——由储蓄账户到活期账户的转账。在空白行中，录入一项由储蓄存款转账500元到活期存款的交易。（在这里，最后一项交易的日期是3月28日，但这项转账交易发生在3月24日。）交易的转账科目应设为储蓄存款，因为你是活期存款科目中。现在活期存款科目应当如下所示：

5.2. 活期

*gcashdata_4.7.gnucash - 活期存款 - GnuCash

文件(F) 编辑(E) 查看(V) 交易(N) 操作(A) 企业(B) 报表(R) 工具(T) 窗口(W) 帮助(H)

保存

关闭

创建副本

删除

确认

取消

转到空行

展开分录

科目

活期存款

| 日期 | 编号 | 描述 | 转账 | 对 | 借方 | 贷方 | 余额 |
|------------|------|----------|--------------|---|----------|--------|----------|
| 2024-03-01 | | | 所有者权益:期初余额 | n | 1,000.00 | | 1,000.00 |
| 2024-03-05 | 101 | 超市采购 | 费用:食品杂货 | n | | 45.21 | 954.79 |
| 2024-03-06 | | 定期存款到期 | 资产:流动资产:储蓄存款 | n | 100.00 | | 1,054.79 |
| 2024-03-14 | | 发放工资 | -- 复合分录 -- | n | 670.00 | | 1,724.79 |
| 2024-03-24 | ▼ 编号 | 定期存款提前支取 | 资产:流动资产:储蓄存款 | n | 500.00 | 贷方 | 2,224.79 |
| 2024-03-28 | 102 | 国家电网 | 费用:水电费:电费 | n | | 78.00 | 2,146.79 |
| 2024-03-28 | 103 | 中国移动 | 费用:电话费 | n | | 45.00 | 2,101.79 |
| 2024-03-28 | 104 | 四月份房租 | 费用:房租 | n | | 350.00 | 1,751.79 |
| 2024-03-28 | 105 | 广电网络 | 费用:电视费 | n | | 20.00 | 1,731.79 |
| 2024-04-28 | | 五月份房租 | 费用:房租 | n | | 350.00 | 1,381.79 |
| 2024-04-01 | | | | n | | | |

现在: CN¥1,731.79 将来: CN¥1,381.79 已结清: CN¥0.00 已对账: CN¥0.00 预计最低: CN¥1,381.79 排序: 标准 ^

此图展示了活期科目账簿。

2. 现在来在科目中记录一些支付活动。首先，向家门口杂货店总计支付75元（3月5日）。转账科目为食品杂货，因为这笔钱都用来买杂货了。接下来，向壹贰叁五金店支付100元（3月6日），并将此金额分录为两项费用：家居用品的50元和工具的50元。你需要为这两项支出各创建一个费用类型科目，然后再为其录入分录。现在活期存款科目应当如下所示：

5.3. 活期

*gcashdata_4.7.gnucash - 活期存款 - GnuCash

文件(F) 编辑(E) 查看(V) 交易(N) 操作(A) 企业(B) 报表(R) 工具(T) 窗口(W) 帮助(H)

保存

关闭

创建副本

删除

确认

取消

转到空行

展开分录

科目

活期存款

| 日期 | 编号 | 描述 | | 借方总计 | 贷方总计 | 余额 |
|------------|-----|--------|--------------|------------|--------|----------|
| 2024-03-01 | | | 所有者权益:期初余额 | n 1,000.00 | | 1,000.00 |
| 2024-03-05 | 101 | 超市采购 | 费用:食品杂货 | n | 45.21 | 954.79 |
| 2024-03-05 | 105 | 家滋味便利店 | 费用:食品杂货 | n | 75.00 | 879.79 |
| 2024-03-06 | | 定期存款到期 | 资产:流动资产:储蓄存款 | n 100.00 | | 979.79 |
| 2024-03-06 | 106 | 甲乙丙五金店 | | 借方总计 | 100.00 | 879.79 |
| | | | 费用:家居用品 | n 50.00 | | |
| | | | 费用:五金工具 | n 50.00 | | |
| | | | 资产:流动资产:活期存款 | | 100.00 | |
| 2024-03-14 | | 发放工资 | -- 复合分录 -- | n 670.00 | | 1,549.79 |
| 2024-03-24 | | 定期存款到期 | 资产:流动资产:储蓄存款 | n 500.00 | | 2,049.79 |

现在: CN¥1,556.79 将来: CN¥1,206.79 已结清: CN¥0.00 已对账: CN¥0.00 预计最低: CN¥1,206.79 排序: 标准

此图展示了填入更多支付记录后的活期科目账簿。

3. 假设你现在需要取些钱出来。你现在还没有在科目表中创建现金科目，所以你需要创建一个。在资产顶级科目下创建一个子科目，名为〔现金〕，类型为〔资产〕。然后，在活期科目账簿中，录入一项ATM机取款交易，记为3月25日由活期到现金转账100元。

5.4. ATM 活期

*gcashdata_4.7.gnucash - 活期存款 - GnuCash

文件(F) 编辑(E) 查看(V) 交易(N) 操作(A) 企业(B) 报表(R) 工具(T) 窗口(W) 帮助(H)

保存

关闭

创建副本

删除

确认

取消

转到空行

展开分录

科目

活期存款

| 日期 | 编号 | 描述 | 转账 | 对 | 借方 | 贷方 | 余额 |
|------------|-----|----------|--------------|---|-----------|--------|----------|
| 2024-03-01 | | | 所有者权益:期初余额 | n | 1,000.00 | | 1,000.00 |
| 2024-03-05 | 101 | 超市采购 | 费用:食品杂货 | n | | 45.21 | 954.79 |
| 2024-03-05 | 105 | 家滋味便利店 | 费用:食品杂货 | n | | 75.00 | 879.79 |
| 2024-03-06 | | 定期存款到期 | 资产:流动资产:储蓄存款 | n | 100.00 | | 979.79 |
| 2024-03-06 | 106 | 甲乙丙五金店 | -- 复合分录 -- | n | | 100.00 | 879.79 |
| 2024-03-14 | | 发放工资 | -- 复合分录 -- | n | 670.00 | | 1,549.79 |
| 2024-03-24 | | 定期存款提前支取 | 资产:流动资产:储蓄存款 | n | 500.00 | | 2,049.79 |
| 2024-03-25 | | ATM取现 | 资产:流动资产:现金 | n | 借方 100.00 | | 1,949.79 |
| 2024-03-28 | 102 | 国家电网 | 费用:水电费:电费 | n | | 78.00 | 1,871.79 |
| 2024-03-28 | 103 | 中国移动 | 费用:电话费 | n | | 45.00 | 1,826.79 |
| 2024-03-28 | 104 | 西昌公寓 | 费用:房租 | n | | 350.00 | 1,476.79 |

现在: CN¥1,456.79 将来: CN¥1,106.79 已结清: CN¥0.00 已对账: CN¥0.00 预计最低: CN¥1,106.79 排序: 标准

此图展示了录入ATM机取款交易后的活期科目账簿。

期初余额

现在已经准备好为活期科目对账了，依据的是下面这份示例银行账单：

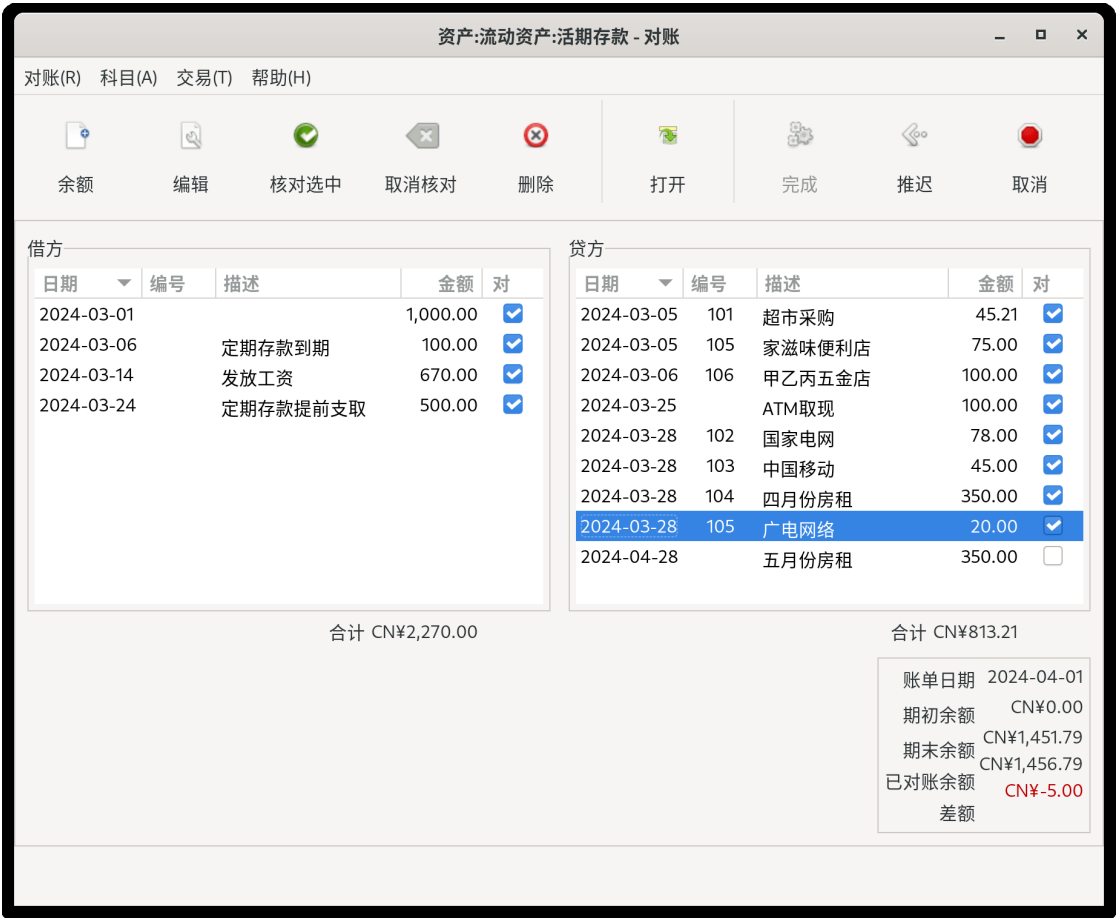
5.5.

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

此图展示了一份示例银行账单。

1. 从菜单中选择〔功能〕→〔对账〕，期末余额填写1451.79元。点击 确认 开始为此科目对账。勾选出现在银行账单中的条目。在所有条目均被勾选后，对账窗口将如下图所示：

5.6.



此图展示了对账窗口，其中有5元的差额。

注意到对账余额与期末余额相差5元。查看银行账单，会发现还有一项5元的服务费没有添加到活期科目中。

2. 所以，点击活期账簿，向活期科目添加5元的服务费。在活期账簿的空白行中，录入一项从活期转账5元到服务费科目的交易。（你需要创建类型为〔费用〕的服务费科目。）日期填写为此交易在银行账单上对应的日期。现在活期存款科目应当如下所示：

5.7. 活期

gcashdata_4.7.gnucash - 活期存款 - GnuCash

文件(F) 编辑(E) 查看(V) 交易(N) 操作(A) 企业(B) 报表(R) 工具(T) 窗口(W) 帮助(H)

保存

关闭

创建副本

删除

确认

取消

转到空行

展开分录

科目

活期存款

| 日期 | 编号 | 描述 | 转账 | 对 | 借方 | 贷方 | 余额 |
|------------|-----|----------|--------------|---|--------|--------|----------|
| 2024-03-06 | 106 | 甲乙丙五金店 | -- 复合分录 -- | n | | 100.00 | 879.79 |
| 2024-03-14 | | 发放工资 | -- 复合分录 -- | n | 670.00 | | 1,549.79 |
| 2024-03-24 | | 定期存款提前支取 | 资产:流动资产:储蓄存款 | n | 500.00 | | 2,049.79 |
| 2024-03-25 | | ATM取现 | 资产:流动资产:现金 | n | | 100.00 | 1,949.79 |
| 2024-03-28 | 102 | 国家电网 | 费用:水电费:电费 | n | | 78.00 | 1,871.79 |
| 2024-03-28 | 103 | 中国移动 | 费用:电话费 | n | | 45.00 | 1,826.79 |
| 2024-03-28 | 104 | 四月份房租 | 费用:房租 | n | | 350.00 | 1,476.79 |
| 2024-03-28 | 105 | 广电网络 | 费用:电视费 | n | | 20.00 | 1,456.79 |
| 2024-03-31 | | 服务费 | 费用:银行服务收费 | n | 借方 | 5.00 | 1,451.79 |
| 2024-04-28 | | 五月份房租 | 费用:房租 | n | | 350.00 | 1,101.79 |

现在: CN¥1,451.79 将来: CN¥1,101.79 已结清: CN¥0.00 已对账: CN¥0.00 预计最低: CN¥1,101.79 排序: 标准 ^

此图展示了补充服务费后的活期科目。

3. 回到对账窗口，你应该会看到服务费现已在〔转出〕下列出。点击此项标记为已对账，可见下方的差额现在变成了〔0.00〕。点击工具栏上的完成按钮完成对账。在活期科目账簿中表示对账状态的〔对〕列下，刚才对账过的每笔交易现在都应该显示为〔y〕。同时，底部状态栏现在会显示〔已对账：CNY 1451.79〕。

5.8. 活期

*gcashdata_4.7.gnucash - 活期存款 - GnuCash

文件(F) 编辑(E) 查看(V) 交易(N) 操作(A) 企业(B) 报表(R) 工具(T) 窗口(W) 帮助(H)

保存 关闭 创建副本 删除 确认 取消 转到空行 展开分录

科目 活期存款

| 日期 | 编号 | 描述 | 转账 | 对 | 借方 | 贷方 | 余额 |
|------------|-----|----------|--------------|---|--------|--------|----------|
| 2024-03-14 | | 发放工资 | -- 复合分录 -- | y | 670.00 | | 1,549.79 |
| 2024-03-24 | | 定期存款提前支取 | 资产:流动资产:储蓄存款 | y | 500.00 | | 2,049.79 |
| 2024-03-25 | | ATM取现 | 资产:流动资产:现金 | y | | 100.00 | 1,949.79 |
| 2024-03-28 | 102 | 国家电网 | 费用:水电费:电费 | y | | 78.00 | 1,871.79 |
| 2024-03-28 | 103 | 中国移动 | 费用:电话费 | y | | 45.00 | 1,826.79 |
| 2024-03-28 | 104 | 四月份房租 | 费用:房租 | y | | 350.00 | 1,476.79 |
| 2024-03-28 | 105 | 广电网络 | 费用:电视费 | y | | 20.00 | 1,456.79 |
| 2024-03-31 | | 服务费 | 费用:银行服务收费 | y | | 5.00 | 1,451.79 |
| 2024-04-28 | | 五月份房租 | 费用:房租 | n | | 350.00 | 1,101.79 |
| 2024-04-01 | ▼ | 编号 描述 | 转账 | n | 借方 | 贷方 | 余额 |

现在: CN¥1,451.79 将来: CN¥1,101.79 已结清: CN¥1,451.79 已对账: CN¥1,451.79 预计最低: CN¥1,101.79 排序: 标准 ^

此图展示了对账后的活期科目账簿。

保存文件

回到主窗口，使用新文件名 "gcashdata_5"保存你的文件。你的科目表规模在逐步扩大，现在如下图所示：

5.9.

*gcashdata_4.7.gnucash - 科目 - GnuCash

文件(F) 编辑(E) 查看(V) 操作(A) 企业(B) 报表(R) 工具(T) 窗口(W) 帮助(H)

保存

关闭

打开

编辑

新建

删除

科目

| 科目名称 | 类型 | 商品 | 合计 |
|--------|-------|------|-------------|
| 资产 | 资产 | 人民币元 | CN¥1,601.79 |
| 流动资产 | 资产 | 人民币元 | CN¥1,601.79 |
| 活期存款 | 银行 | 人民币元 | CN¥1,101.79 |
| 储蓄存款 | 银行 | 人民币元 | CN¥400.00 |
| 现金 | 现金 | 人民币元 | CN¥100.00 |
| 所有者权益 | 所有者权益 | 人民币元 | CN¥1,500.00 |
| 费用 | 费用 | 人民币元 | CN¥1,398.21 |
| 房租 | 费用 | 人民币元 | CN¥700.00 |
| 税费 | 费用 | 人民币元 | CN¥330.00 |
| 食品杂货 | 费用 | 人民币元 | CN¥120.21 |
| 水电费 | 费用 | 人民币元 | CN¥78.00 |
| 五金工具 | 费用 | 人民币元 | CN¥50.00 |
| 家居用品 | 费用 | 人民币元 | CN¥50.00 |
| 电话费 | 费用 | 人民币元 | CN¥45.00 |
| 电视费 | 费用 | 人民币元 | CN¥20.00 |
| 银行服务收费 | 费用 | 人民币元 | CN¥5.00 |
| 收入 | 收入 | 人民币元 | CN¥1,000.00 |
| 负债 | 负债 | 人民币元 | CN¥500.00 |

CN¥ 总计:

净资产: CN¥1,101.79

利润: CN¥-398.21

此图展示了科目表。

报表

就像在前文章节做的那样，现在来看看〔现金流量表〕和〔交易汇总表〕这两份报表。

1. 首先查看3月的〔现金流量表〕。

选择〔报表〕→〔收入与费用〕→〔现金流量表〕。

5.10. 3

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

Selected Accounts

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

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

此图展示了按第 5 章 活期账户与现金所述操作后的现金流量表。

2. 接下来查看各个资产科目对应的交易汇总表。

选择 [报表] → [交易汇总表] 。

5.11. 3

Transaction Report

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

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

此图展示了3月份期间资产科目的交易汇总表。

- 现在，将交易汇总表修改为只展示各个费用科目的情况。

5.12. Transaction Report For The Expenses During March

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

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

此图展示了3月份期间各费用科目的交易汇总表。

你可能已经注意到，目前还有一个在科目表中列出的科目没有使用，那就是信用卡科目。到这里，你知道了怎样在GnuCash中跟踪你的银行科目和现金科目。现在，你可能也想要开始跟踪信用卡的使用情况。GnuCash为信用卡提供了一种特殊的科目类型，相关内容会在后面的章节详细讨论。

第 6 章 费用

如果说管理活期账户是跟踪财务状况的第一步，那借助费用科目了解钱款去处就是紧随其后的第二步。这一章会帮助你理解在GnuCash中怎样借助费用科目跟踪各类交易。

概念

费用类型科目可帮助你跟踪在某项支出上花了多少钱。许多人第一次跟踪花销是在Quicken(tm)上，其中交易可以被指定到一个或多个分类。在GnuCash中，这些分类对应于各个相互独立的费用类型科目。这样的组织形式可以让GnuCash将复式记账规则同样用于管理费用。根据需要，费用科目的设立可详可略。一些用户只需几个科目就足以跟踪个人财务。还有一些用户借助GnuCash的费用科目事无巨细地管理他们的花销情况。你可以自由决定记录的细致程度。请记住，在GnuCash中，你可以更改交易所属的科目，因此如果你以后有不同的需求，将交易在不同科目间移动也是可以的。

创建科目

建立简单的费用科目

对于许多用户而言，创建费用科目最轻松的方法还是在〔建立科目结构〕中勾选〔常用科目〕。这样会创建各种最常用的费用科目。了解更多信息，请查看GnuCash帮助手册第3章中的“新建科目层次结构”一节。

复杂的费用科目结构

你如果有不同的费用记账需求，可参考“交易”一节或GnuCash帮助手册第5.4章，了解如何创建科目。

新建费用科目的理由一般有：要跟踪特定企业用途费用（如具体类别的原料费用），要跟踪具有特定税务用途的费用（如必须向他人报告的税务费用）或者只是跟踪在你看来有意义的支出（如向特定慈善组织支付的善款）。

录入费用交易

虽然可以直接将交易录入费用科目，但大家一般不这样做。对于大多数人，将一项交易录入其涉及的其它科目，这项交易也就添加到费用科目下了。换言之，你如果有一个用于记录慈善捐款的科目（如费用：慈善），则一般会在活期科目账簿中添加一项同慈善科目的交易，那也就将这项交易添加到了费用科目。

打开某个费用科目，你会看到与其它GnuCash科目相似的账簿。不过，使用非正式术语时，交易金额的列头稍有不同。左（借）列显示为支出总计，而右（贷）列显示为退款总计。

有关费用科目的其它注意事项

由于费用科目是完全根据你的需要建立的，所以没有账单用来对账。因此从技术上讲，费用科目没有对账的概念。当然，你也可以在费用科目上使用对账功能，这样会锁定交易阻止后续编辑。

还要考虑的一点是，通常这些科目中很少有减少其余额的贷记交易，因此，随着你持续使用 GnuCash，科目余额会持续增长。这样当然没什么问题，但是一些用户可能希望定期将余额归零。可以录入归零交易，将余额转至所有者权益科目。GnuCash提供〔账簿结转〕功能，可以将费用科目归零。请注意，不是必须要这么做，如果你需要获得某一费用科目的信息，你可以使用各种报表提取相关数据，而不必将科目归零。

第 7 章 信用卡

本章将介绍怎样使用GnuCash管理信用卡。

概念

可能你每个月通过支票或网上支付向信用卡公司偿还账单，所以你可能以为你的信用卡账单是一种费用，但它实际上并不属于费用。为什么？信用卡其实是一种短期贷款，你用这个贷款账户购物消费，而且最后要偿还这些钱，常常还要支付利息（你的金融服务费）。你使用信用卡产生的购买才属于费用。

你有两种方法记录信用卡交易，选择符合你所需细致程度的那一种方法。最简单的方法是只记录每个月向信用卡公司的还款。你每个月录入一项由银行科目向信用卡费用科目的转账。这样会显示你每个月向信用卡公司的还款，但不会显示任何有关信用卡欠款或信用卡购买记录的信息。

还有一种更复杂的记录信用卡交易的方法，那就是将每次购买或付款记为独立的交易。你需要在信用卡账簿中录入一整个月的单据。在收到信用卡账单时，你能够根据账单为信用卡科目对账。这种方法会向你展现整个月期间的欠款信息，并在对账时指出不一致之处，但是需要你录入更多数据条目。

创建科目

要开始在GnuCash中管理信用卡，你应当建立一个〔负债〕顶级科目，并在此父科目之下为你使用的每张信用卡创建信用卡类型的科目。如果你只是跟踪向信用卡公司的还款记录，那么你只需要银行科目和信用卡科目就足够用来录入交易了。

信用卡产生的服务费属于费用，所以你需要在名为〔费用〕的顶级科目下建立这些科目。你如果决定更详细地记录消费情况，还需要为你产生的每种消费创建相应的费用科目。你由于以后也要根据信用卡账单检查科目余额并为科目对账，则也应该在每个信用卡科目中录入期初余额。录入期初余额最简单的方法是直接将上一份信用卡账单给出的欠款额作为期初余额。

简易科目设置

你如果不想跟踪信用卡的每笔支出，可以建立如下所示的简单科目层次结构：

- 资产
 - 银行
- 负债
 - 信用卡
- 费用

-信用卡

在这个例子中，你要将每个月信用卡的还款总额录入为负债：信用卡与费用：信用卡间的交易。当还信用卡的时候，你要录入一项资产：银行和负债：信用卡间的交易。

简易信用卡记账方法一个显而易见的缺点就是你看不到钱花到哪里去了。所有信用卡消费都合并到了信用卡费用科目下。不过，这样做起来十分简单且易于维护。

完整科目设置

你如果想要更完整地跟踪花销情况，则应建立多个费用科目，分别命名为你产生的各种花销类型。信用卡的每笔付款应独立地录入为信用卡负债科目与具体费用科目间的交易。下面是相应的科目层次结构示例：

- 资产
 - 银行
- 负债
 - 信用卡
- 费用
 - 食品
 - 汽车
 - 服装
 - 娱乐
 - 利息
 - 服务费

需要明确的是，设置的具体费用科目应当符合你的消费习惯。这种科目设置与简易科目设置相比，唯一的不同是费用细分为几组。也要注意到其中包含一个“利息”费用科目，它是在信用卡为每月的未还款额收取利息时使用的。“服务费”费用科目用于记录信用卡收取的服务费，如可能会收取的年费。借助这样的科目设置，你将能够按照费用科目的分组看到每个月你的钱花到哪里去了。

本章其余部分将假定你在使用完整科目设置。

记录付款

记录信用卡的每笔付款能够为你给出更完整的消费习惯画像。信用卡付款记录为信用卡负债科目与适当的费用科目间的交易。

当你使用信用卡为商品或服务付款时，你就是在告诉信用卡公司替你向对方付费。这项交易会增加你在信用卡公司的欠款，即信用卡科目余额增加。这些交易的另一端大多是费用科目。例如，如果你使用信用卡在商店花50元买了一件衣服，则相当于将钱由信用卡科目转账到费用：服装。

将这些交易录入GnuCash有助于你随时掌握目前欠信用卡公司多少钱，也有助于你更清楚地了解各科目的整体情况。这样做还可以帮助你跟踪账户状态，确保避免盗刷。

向信用卡科目添加交易很像在其它科目中添加交易。你可以手动录入交易，或通过兼容的格式从信用卡公司导入交易。

有关如何录入交易，请查看GnuCash帮助手册第6章和“交易”一节。

记录还款

偿还信用卡账单往往录入为由银行科目（资产）到信用卡科目（负债）的转账。偿还月账单时，你就是在从银行科目取款，以冲抵信用卡欠款。此交易同时减少银行科目余额和信用卡负债额。

当你退货时，你的信用卡会收到退款。这是另一种形式的还款，它会减少你的信用卡欠款。你如果将原先的购买交易录入为由信用卡科目到费用科目的转账，现在则简单地将那项交易反转过来：将钱从费用科目转回信用卡科目。此交易同时减少费用科目余额和信用卡科目余额。例如，如果你原先录入了一笔使用信用卡付款的服装购买，交易是从信用卡科目到服装费用科目的转账。你如果之后退掉衣服，收到退款，则简单地将钱从服装费用科目转回信用卡科目。

一个常见的错误是将退款录入为收入。退款不是收入，而是一种“负费用”。这就是为什么必须在收到退款时将钱从费用科目转回信用卡科目。

为了说清楚这个规则，来看看一个例子。你使用VISA卡花74.99元买了一些牛仔裤，但之后某天你发现它们太大，想退回去。店家爽快地同意了，并将钱款退到了你的信用卡里。

1. 先打开之前保存的数据文件（"gcashdata_5"），将之另存为"gcashdata_6"。
2. 打开负债:Visa科目账簿，录入一项简单的双科目转账交易，表示买牛仔裤花了74.99元。〔转账〕科目应为费用:服装，使用Visa〔付款〕74.99元。

由于之前没有创建费用:服装科目，GnuCash会提示我们创建科目。要记得将其创建为〔费用〕科目。

3. 按下列方法之一录入退款交易：

- 录入与当初购买时相同的交易，但在信用卡科目账簿的〔还款〕处，而非〔付款〕处输入金额。
- 选择想要退款的购买交易（这个例子中也就是购买牛仔裤的交易），并选择〔交易〕→〔冲销交易〕。按需修改日期。

在对冲交易后，你的信用卡科目应该如下所示：

7.1.

*gcashdata_6.1.gnucash - 信用卡 - GnuCash

文件(F) 编辑(E) 查看(V) 交易(N) 操作(A) 企业(B) 报表(R) 工具(T) 窗口(W) 帮助(H)

保存

关闭

创建副本

删除

确认

取消

转到空行

展开分录

科目

信用卡

| 日期 | 编号 | 描述 | 转账 | 对 | 借方 | 贷方 | 余额 |
|------------|------|-----------|------------|---|-------|--------|--------|
| 2024-03-01 | | | 所有者权益:期初余额 | n | | 500.00 | 500.00 |
| 2024-04-03 | | 牛仔博士 | 费用:衣物 | n | | 74.99 | 574.99 |
| 2024-04-05 | ▼ 编号 | 牛仔博士 - 退款 | 费用:衣物 | n | 74.99 | 贷方 | 500.00 |
| 2024-05-07 | | | | n | | | |

现在: CN¥500.00 将来: CN¥500.00 已结清: CN¥0.00 已对账: CN¥0.00 预计最低: CN¥500.00 排序: 标准 ^

此图展示了对冲购买交易后的负债:信用卡科目账簿。

同时费用:服装科目账簿应该如下所示:

7.2.



此图展示了对冲购买交易后的费用:服装科目账簿。

4. 保存GnuCash数据文件。

步步为营

现在我们已经介绍了在跟踪信用卡账目所需创建的交易背后的基本思想。现在来研究一个例子。在这个例子中，我们将使用信用卡买东西，退还两样购买的商品，缴纳由未还款额产生的利息，为信用卡科目对账，最后偿还部分信用卡欠款。

打开GnuCash文件

首先打开先前保存的数据文件 "gcashdata_5"，并将其直接保存为 "gcashdata_6"主窗口应该如下所示：

7.3.

gcashdata_6.0.gnucash - 科目 - GnuCash

文件(F) 编辑(E) 查看(V) 操作(A) 企业(B) 报表(R) 工具(T) 窗口(W) 帮助(H)

保存 关闭 打开 编辑 新建 删除

科目 信用卡

| 科目名称 | 类型 | 商品 | 合计 |
|--------|-------|------|-------------|
| 资产 | 资产 | 人民币元 | CN¥1,601.79 |
| 现金 | 现金 | 人民币元 | CN¥100.00 |
| 活期存款 | 银行 | 人民币元 | CN¥1,101.79 |
| 储蓄存款 | 银行 | 人民币元 | CN¥400.00 |
| 所有者权益 | 所有者权益 | 人民币元 | CN¥1,500.00 |
| 收入 | 收入 | 人民币元 | CN¥1,000.00 |
| 负债 | 负债 | 人民币元 | CN¥500.00 |
| 信用卡 | 信用卡 | 人民币元 | CN¥500.00 |
| 费用 | 费用 | 人民币元 | CN¥1,398.21 |
| 银行服务收费 | 费用 | 人民币元 | CN¥5.00 |
| 衣物 | 费用 | 人民币元 | CN¥0.00 |
| 五金工具 | 费用 | 人民币元 | CN¥50.00 |
| 税费 | 费用 | 人民币元 | CN¥330.00 |
| 水电费 | 费用 | 人民币元 | CN¥78.00 |
| 食品杂货 | 费用 | 人民币元 | CN¥120.21 |
| 家居用品 | 费用 | 人民币元 | CN¥50.00 |
| 房租 | 费用 | 人民币元 | CN¥700.00 |
| 调整 | 费用 | 人民币元 | CN¥0.00 |
| 电视费 | 费用 | 人民币元 | CN¥20.00 |
| 电话费 | 费用 | 人民币元 | CN¥45.00 |

CN¥ 总计: 净资产: CN¥1,101.79 利润: CN¥-398.21

在这个步步为营例子中跟踪信用卡账目前的初始科目结构。

购买

现在使用Visa卡买一些东西。先在苍蝇茶馆买了25元的食品，在不着调时装店购买100元的衣服，向加价不加量加油站支付25元的汽油钱，在不靠谱杂货店购买125元的食品杂货和家居用品（分录为85元的杂货和40元的家居），最后又在廉价市场买了60元的家居用品。

我们也会重复前面几章的操作，即在4月3日花74.99元购买一条牛仔褲，并在两天后退货。

信用卡负债账簿窗口应该如下所示：

7.4.

*gcashdata_6.1.gnucash - 信用卡 - GnuCash

文件(F) 编辑(E) 查看(V) 交易(N) 操作(A) 企业(B) 报表(R) 工具(T) 窗口(W) 帮助(H)

保存

关闭

创建副本

删除

确认

取消

转到空行

展开分录

科目 信用卡

| 日期 | 编号 | 描述 | 转账 | 对 | 借方 | 贷方 | 余额 |
|------------|----|-----------|------------|---|-------|--------|--------|
| 2024-03-01 | | | 所有者权益:期初余额 | n | | 500.00 | 500.00 |
| 2024-04-03 | | 牛仔博士 | 费用:衣物 | n | | 74.99 | 574.99 |
| 2024-04-05 | | 牛仔博士 - 退款 | 费用:衣物 | n | 74.99 | | 500.00 |
| 2024-04-10 | | 苍蝇小馆 | 费用:食品 | n | | 25.00 | 525.00 |
| 2024-04-11 | | 不着调服装店 | 费用:衣物 | n | | 100.00 | 625.00 |
| 2024-04-12 | | 金牌加油站 | 费用:汽车:能源 | n | | 25.00 | 650.00 |
| 2024-04-13 | | 万有杂货店 | -- 复合分录 -- | n | | 125.00 | 775.00 |
| 2024-04-13 | | 两元店 | 费用:家居用品 | n | | 60.00 | 835.00 |
| 2024-05-07 | | | 转账 | n | 借方 | 贷方 | 余额 |

现在: CN¥835.00 将来: CN¥835.00 已结清: CN¥0.00 已对账: CN¥0.00 预计最低: CN¥835.00 排序: 标准

初始的信用卡购买。

退款

现在假定在4月15日，你退还了4月11日从不着调时装店购买的衣服，他们将退款退回了你的信用卡。录入一项总额100元的信用卡退款交易。别忘了所用的转账科目与原先购买时所用科目相同，并在〔还款〕列下输入金额。GnuCash会自动补全名称和转账科目，但仍会将100元自动填入〔付款〕列中。你需要重新在〔还款〕列中输入金额。交易显示如下：

7.5.

*gcashdata_6.1.gnucash - 信用卡 - GnuCash

文件(F) 编辑(E) 查看(V) 交易(N) 操作(A) 企业(B) 报表(R) 工具(T) 窗口(W) 帮助(H)

保存

关闭

创建副本

删除

确认

取消

转到空行

展开分录

科目 信用卡

| 日期 | 编号 | 描述 | 转账 | 对 | 借方 | 贷方 | 余额 |
|------------|------|-------------|------------|---|--------|--------|--------|
| 2024-03-01 | | | 所有者权益:期初余额 | n | | 500.00 | 500.00 |
| 2024-04-03 | | 牛仔博士 | 费用:衣物 | n | | 74.99 | 574.99 |
| 2024-04-05 | | 牛仔博士 - 退款 | 费用:衣物 | n | 74.99 | | 500.00 |
| 2024-04-10 | | 苍蝇小馆 | 费用:食品 | n | | 25.00 | 525.00 |
| 2024-04-11 | | 不着调服装店 | 费用:衣物 | n | | 100.00 | 625.00 |
| 2024-04-12 | | 金牌加油站 | 费用:汽车:能源 | n | | 25.00 | 650.00 |
| 2024-04-13 | | 万有杂货店 | -- 复合分录 -- | n | | 125.00 | 775.00 |
| 2024-04-13 | | 两元店 | 费用:家居用品 | n | | 60.00 | 835.00 |
| 2024-04-15 | ▼ 编号 | 不着调服装店 - 退款 | 费用:衣物 | n | 100.00 | 贷方 | 735.00 |
| 2024-05-07 | | | | n | | | |

现在: CN¥735.00 将来: CN¥735.00 已结清: CN¥0.00 已对账: CN¥0.00 预计最低: CN¥735.00 排序: 标准 ^

将衣服退还给不靠谱时装店，退款退回信用卡。

缴纳利息

在几个月的消费后，你收到了寄来的信用卡账单或在网上看到了账单。很不幸，由于你上个月的部分欠款拖到了现在，你在4月的最后一天被收取了20元的利息。这笔利息要作为费用录入到信用卡科目。

7.6.

*gcashdata_6.1.gnucash - 信用卡 - GnuCash

文件(F) 编辑(E) 查看(V) 交易(N) 操作(A) 企业(B) 报表(R) 工具(T) 窗口(W) 帮助(H)

保存

关闭

创建副本

删除

确认

取消

转到空行

展开分录

科目

信用卡

| 日期 | 编号 | 描述 | 转账 | 对 | 借方 | 贷方 | 余额 |
|------------|----|-------------|------------|---|--------|--------|--------|
| 2024-03-31 | | | 所有者权益:期初余额 | n | | 500.00 | 500.00 |
| 2024-04-03 | | 牛仔博士 | 费用:衣物 | n | | 74.99 | 574.99 |
| 2024-04-05 | | 牛仔博士 - 退款 | 费用:衣物 | n | 74.99 | | 500.00 |
| 2024-04-10 | | 苍蝇小馆 | 费用:食品 | n | | 25.00 | 525.00 |
| 2024-04-11 | | 不着调服装店 | 费用:衣物 | n | | 100.00 | 625.00 |
| 2024-04-12 | | 金牌加油站 | 费用:汽车:能源 | n | | 25.00 | 650.00 |
| 2024-04-13 | | 万有杂货店 | -- 复合分录 -- | n | | 125.00 | 775.00 |
| 2024-04-13 | | 两元店 | 费用:家居用品 | n | | 60.00 | 835.00 |
| 2024-04-15 | | 不着调服装店 - 退款 | 费用:衣物 | n | 100.00 | | 735.00 |
| 2024-04-30 | ▼ | 利息 | 费用:利息 | n | 借方 | 20.00 | 755.00 |
| 2024-05-07 | | | | n | | | |

现在: CN¥755.00 将来: CN¥755.00 已结清: CN¥0.00 已对账: CN¥0.00 预计最低: CN¥755.00 排序: 标准 ^

信用卡收取利息。

对账

在收到信用卡账单后，你应该根据账单为信用卡科目对账。使用GnuCash内置的对账功能就可以。选中信用卡科目并点击 [功能] → [对账...] 。对账的具体操作步骤在“对账”一节中有详细的描述，但在这里我们会再回顾一遍相关步骤。例如，假定信用卡账单截止到5月1日，最后的欠额是455元。将日期和金额输入到初始对账窗口中，如下图所示。

7.7.

负债:信用卡 - 对账

对账信息

日期(D)

2024-04-30

期初余额

CN¥0.00

期末余额(E)

455.00

☐ 包含子科目(S)

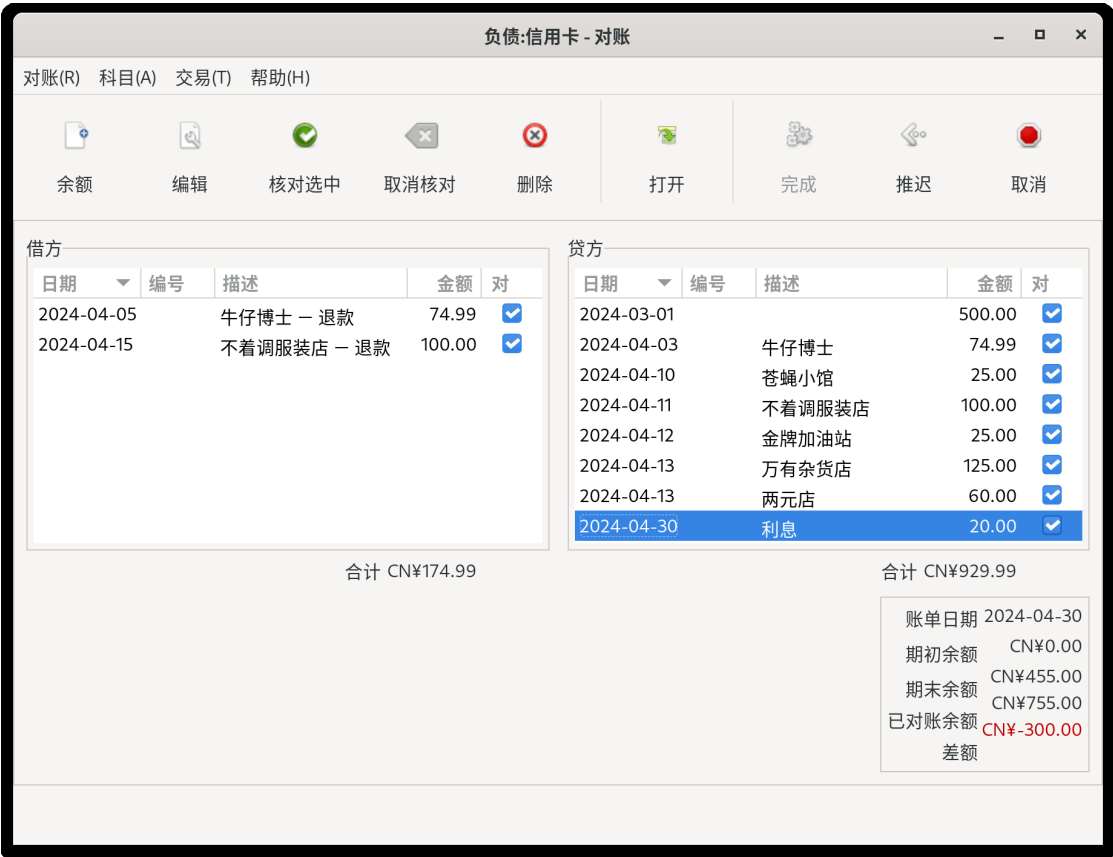
记录利息支出(I)...

取消(C)

确认(O)

在对账过程中，你需要勾选每一项确认同时出现在GnuCash科目和信用卡账单中的交易。在这个例子中，GnuCash科目与信用卡账单间存在300元的差额，如下图所示。

7.8.



提示300元差额的科目对账主窗口。

经过一番调查，你发现你忘了记录3月5日向信用卡公司300元的还款，你要录入这项由银行科目到信用卡科目的还款交易。现在信用卡账单和GnuCash科目账目相符，余额为455元。

还款

完成了信用卡科目的对账工作后，你需要向信用卡公司还款。在这个例子中，你欠了455元，但这个月还是只偿还部分欠款，共300元。相应地，录入一项300元的由银行科目到信用卡科目的交易，这样，信用卡科目余额应减少到155元。你的信用卡科目账簿现在应该如下所示：

7.9.

*gcashdata_6.1.gnucash - 信用卡 - GnuCash

文件(F) 编辑(E) 查看(V) 交易(N) 操作(A) 企业(B) 报表(R) 工具(T) 窗口(W) 帮助(H)

保存

关闭

创建副本

删除

确认

取消

转到空行

展开分录

科目 信用卡

| 日期 | 编号 | 描述 | 转账 | 对 | 借方 | 贷方 | 余额 |
|------------|----|-------------|------------|---|--------|--------|--------|
| 2024-04-05 | | 牛仔博士 - 退款 | 费用:衣物 | y | 74.99 | | 200.00 |
| 2024-04-10 | | 苍蝇小馆 | 费用:食品 | y | | 25.00 | 225.00 |
| 2024-04-11 | | 不着调服装店 | 费用:衣物 | y | | 100.00 | 325.00 |
| 2024-04-12 | | 金牌加油站 | 费用:汽车:能源 | y | | 25.00 | 350.00 |
| 2024-04-13 | | 万有杂货店 | -- 复合分录 -- | y | | 125.00 | 475.00 |
| 2024-04-13 | | 两元店 | 费用:家居用品 | y | | 60.00 | 535.00 |
| 2024-04-15 | | 不着调服装店 - 退款 | 费用:衣物 | y | 100.00 | | 435.00 |
| 2024-04-30 | | 利息 | 费用:利息 | y | | 20.00 | 455.00 |
| 2024-05-05 | | 部分还款 | 资产:活期存款 | n | 300.00 | 贷方 | 155.00 |
| 2024-05-07 | | | | n | | | |

现在: CN¥155.00 将来: CN¥155.00 已结清: CN¥455.00 已对账: CN¥455.00 预计最低: CN¥155.00 排序: 标准 ^

对账和还款后的科目账簿。

回到主窗口保存文件（"gcashdata_6"）。科目表在稳步扩充，现在应当如下所示：

7.10.

| 科目名称 | 类型 | 商品 | 合计 |
|---------|-------|------|-------------|
| ▶ 资产 | 资产 | 人民币元 | CN¥1,001.79 |
| ▼ 负债 | 负债 | 人民币元 | CN¥155.00 |
| 信用卡 | 信用卡 | 人民币元 | CN¥155.00 |
| ▶ 收入 | 收入 | 人民币元 | CN¥1,000.00 |
| ▼ 费用 | 费用 | 人民币元 | CN¥1,653.21 |
| 电话费 | 费用 | 人民币元 | CN¥45.00 |
| 电视费 | 费用 | 人民币元 | CN¥20.00 |
| 调整 | 费用 | 人民币元 | CN¥0.00 |
| 房租 | 费用 | 人民币元 | CN¥700.00 |
| 家居用品 | 费用 | 人民币元 | CN¥150.00 |
| 利息 | 费用 | 人民币元 | CN¥20.00 |
| ▶ 汽车 | 费用 | 人民币元 | CN¥25.00 |
| 食品 | 费用 | 人民币元 | CN¥25.00 |
| ▶ 水电费 | 费用 | 人民币元 | CN¥78.00 |
| ▶ 税费 | 费用 | 人民币元 | CN¥330.00 |
| 五金工具 | 费用 | 人民币元 | CN¥50.00 |
| 衣物 | 费用 | 人民币元 | CN¥0.00 |
| 银行服务收费 | 费用 | 人民币元 | CN¥5.00 |
| 杂货 | 费用 | 人民币元 | CN¥205.21 |
| ▶ 所有者权益 | 所有者权益 | 人民币元 | CN¥1,500.00 |

CN¥ 总计: 净资产: CN¥846.79 利润: CN¥-653.21

保存文件

最后，保存GnuCash数据文件（"gcashdata_6"）。

报表

和前几章一样，我们来看看〔现金流量表〕和〔交易汇总表〕这两份报表。

1. 首先来看3月期间负债科目〔Visa〕的〔现金流量表〕。

选择〔报表〕→〔收入与费用〕→〔现金流量表〕打开现金流量表。

7.11. 3

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

Selected Accounts

● Liabilities:Visa

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

此图展示了第 7 章 信用卡后得到的现金流量表。

2. 现在来看Visa科目相应的交易汇总表。

选择〔报表〕→〔交易汇总表〕打开交易汇总表。

7.12.3 4 Visa

Transaction Report

From 01/03/06 To 30/04/06

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

此图展示了3、4月间Visa科目的交易汇总表。

3. 现在更改交易汇总表，使之只展示各费用科目的交易汇总。

7.13. 4

Transaction Report

From 01/04/06 To 30/04/06

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

此图展示了4月期间各个费用科目的交易汇总表。

第 8 章 贷款

本章将介绍怎样使用GnuCash管理你的贷款。

基本概念

贷款定义为某人为借用其它人的钱而支付一定费用的金融交易。生活中常见的贷款形式有很多，如信用卡、汽车贷款、房屋贷款或商业贷款。

术语

在具体讨论如何使用GnuCash跟踪贷款之前，先列出相关术语是很有帮助的。下面列出的一些术语对应贷款领域中的基本概念。建议尽量熟悉这些术语，起码在后文遇不熟悉的词汇时记得回头查阅这份列表。

分期

确保最终偿清贷款的还款计划，一般设计为使每月的还款金额相等。还款金额通常划分为本金和利息两部分，随分期还款的推进，其中偿还本金的金额增加，支付利息的金额减少。

借款人

经贷款得到钱的个人或企业。

违约

借款人无法根据同放款人达成的协议偿还贷款的時刻。

延期

临时推迟偿还贷款。

拖欠

未能如期还款。

放款

支付给借款人的贷款金额。一些贷款包含多次放款，意味着借款人不会一次性收到贷款的全部金额。

利息

借款人因使用贷款资金而向放款人缴纳的费用。一般以借入本金的年百分比表示，也就是常说的年利率。

放款人

将钱借出给借款人的企业或个人。

贷款手续费

借款人收到贷款时从本金中扣除的贷款处理手续费。

本金

贷款的初始金额，或贷款中目前仍未偿还的金额。按月偿还贷款时，一部分还款金额用于偿还利息，一部分还款金额用于偿还本金。

借据

借款人与放款人就贷款事宜达成的法律协定。

创建科目

借款人在获得贷款时，往往有意要购买一些值钱的东西。事实上，多数贷款往往要求借款人购买事先约定的资产，如房屋。这项资产也是对借款人贷款违约的保障。当然，也有一些贷款不一定关联到高价值资产，如教育贷款。

此处给出的科目结构假定贷款用于购买将被用作抵押物的资产。

贷款是一种负债，逐渐增长的贷款利息是一项持续存在的花费，可能要支付的管理费是另一项费用。使用贷来的钱购买的东西是资产。现在根据上述界定，可以给出基本的贷款科目结构如下：

基本贷款科目结构

- 资产
 - 流动资产
 - 储蓄存款
 - 固定资产
 - 已购入资产
- 负债
 - 贷款
 - 房屋按揭贷款
- 费用
 - 利息
 - 房屋按揭贷款利息
 - 房屋按揭贷款管理费

GnuCash内置多种预定义的贷款科目层次结构，包括汽车贷款和住宅按揭贷款。要使用这些预定义的科目结构，点击〔功能〕→〔建立科目结构〕并选择你想使用的贷款类型。

计算器

确定分期贷款还款计划、每期还款金额、总还款额或利率可能并不容易。GnuCash内置的〔贷款还款计算器〕可以帮助你方便地进行这些计算。通过〔工具〕→〔贷款还款计算器〕打开。

8.1.

GnuCash〔贷款还款计算器〕。

〔贷款还款计算器〕能够计算的数值有〔还款期数〕、〔利率〕、〔现值〕，〔每期还款金额〕和〔终值〕，只要给出其它四项数值就可以计算任一数值。另外还要指定复利方案和还款方案。

还款期数

要支付多少次还款。

利率

贷款的名义利率(NAR)，也就是常说的年利率。

现值

贷款当前的价值，也就是当前欠款金额。

每期还款

每期要偿还的金额。

终值

贷款将来的价值，也就是全部还款期结束后剩余的欠款金额。

复利

有两种复利计算方法，离散复利和连续复利。选择离散复利时，要从下拉框中指定从每年到每天不等的复利频率。

还款

通过下拉框选择还款频率，每年到每天不等。也可以选择在期初还是在期末时还款。如果选择期初还款，还款额会同当前欠款金额一起计算利息。

示例：每月还款

一份总计100,000元、时长30年、固定利率4%、复利计息的贷款，其月还款额是多少呢？

这一分期方案如下图所示。要执行计算，将〔还款期数〕设为**360**（12个月×30年），〔利率〕为**4**，〔现值〕为**100000**，〔每期还款金额〕留空，〔终值〕设为**0**（你不想在还款结束后仍欠款）。〔复利〕是〔每月〕，〔还款〕是〔每月〕，另假设〔期末〕还款、〔离散复利〕。现在点击计算按钮，可见结果为-477.42。

答案：每月需还款477.42元。

示例：贷款总期数

一份20,000元、10%固定利率、按月复利的贷款，每月还款500元，要多久才能偿清？

要执行计算，〔还款期数〕留空，设置〔利率〕为**10**、〔现值〕为**20000**、〔每期还款金额〕为**-500**、〔终值〕为**0**（你不想在还款结束后仍欠款）。〔复利〕为〔每月〕、〔还款〕为〔每月〕，假设〔期末〕还款、〔离散复利〕。现在点击计算按钮，可见〔还款期数〕处显示49。

答案：用4年零1个月内偿清贷款（49个月）。

进阶：计算详细过程

在讨论〔贷款还款计算器〕所使用的数学公式之前，先定义一些变量。

```
n    == 还款期数
%i   == 收取的名义利率
PV   == 现值
PMT  == 每期还款金额
FV   == 终值
CF   == 每年计息频率
PF   == 每年还款频率
```

每年计息频率(CF)和每年还款频率(PF)一般为：

```
1 == 每年
2 == 每半年
3 == 每三分之一年
4 == 每季度
6 == 每两个月
```

12 == 每月
 24 == 每半月
 26 == 每两周
 52 == 每周
 360 == 每天
 365 == 每天

名义利率与有效利率间的转换

要求解 n 、 PV 、 PMT 或 FV ，先要将名义利率(i)转换为还款期时长的有效利率($ieff$)。这一利率($ieff$)随后用于计算需要求解的变量。需要求解 i 时，计算结果是有效利率($ieff$)。因此，需要将 i 转换为 $ieff$ ，也要将 $ieff$ 转换为 i 。

将 i 转换为 $ieff$ ，使用公式：

离散复利： $ieff = (1 + i/CF)^{(CF/PF)} - 1$

连续复利： $ieff = e^{(i/PF)} - 1 = \exp(i/PF) - 1$

将 $ieff$ 转换为 i ，使用公式：

离散复利： $i = CF * [(1+ieff)^{(PF/CF)} - 1]$

连续复利： $i = \ln[(1+ieff)^{PF}]$

在下文处理金融业务的公式中，所有利率均指有效利率“ $ieff$ ”。出于简洁考虑，此符号缩写为“ i ”。

基本金融公式

一个方程可将所有这五个变量联系起来，这也就是常说的基本金融方程：

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

其中： $X = 0$ 表示期末还款，and

$X = 1$ 表示期初还款

由此方程可推导出计算各独立变量的函数表达式。有关方程推导的详细解释，参见 GnuCash源代码文件 `"libgnucash/app-utils/calculation/fin.c"` [<https://github.com/Gnucash/gnucash/tree/stable/libgnucash/app-utils/calculation/fin.c>] 中的注释。首先定义 A 、 B 、 C 三个变量，以使后面给出的公式更易读。

$$A = (1 + i)^n - 1$$

$$B = (1 + iX) / i$$

$$C = PMT * B$$

$$n = \ln[(C - FV)/(C + PV)] / \ln((1 + i))$$

$$PV = -[FV + A * C] / (A + 1)$$

$$PMT = -[FV + PV * (A + 1)] / [A * B]$$

$$FV = -[PV + A * (PV + C)]$$

求解利率分为两种情况。

简单情况为 $PMT == 0$ 时，相应解为：

$$i = [FV/PV]^{(1/n)} - 1$$

$PMT \neq 0$ 的情况相当复杂，这里不会列出。。当在 $PMT \neq 0$ 的情况下求解利率时，无法给出准确的可解函数，相关计算需涉及迭代过程。详细说明请参见 "libgnucash/app-utils/calculation/fin.c" 源文件。

示例：每月还款

现在重新计算“示例：每月还款”一节，这次我们使用数学公式而非〔贷款还款计算器〕。对于一笔为期30年、固定利率4%、复利计息、总计100,000元的贷款，你每个月要还多少钱呢？

首先，定义变量 $n = (30 * 12) = 360$, $PV = 100000$, $PMT = \text{未知}$, $FV = 0$, $i = 4\% = 4/100 = 0.04$, $CF = PF = 12$, $X = 0$ (所有还款期结束)

第二步，将名义利率(i)转换为有效利率($ieff$)。由于利率是按月复利，而且是离散的，使用： $ieff = (1 + i/CF)^{(CF/PF)} - 1$ ，其中 $ieff = (1 + 0.04/12)^{(12/12)} - 1$ ，因此 $ieff = 1/300 = 0.0033333$ 。

现在我们可以计算A和B了。 $A = (1 + i)^n - 1 = (1 + 1/300)^{360} - 1 = 2.313498$ 。 $B = (1 + iX)/i = (1 + (1/300)*0)/(1/300) = 300$ 。

算出A和B，可以计算 PMT。 $PMT = -[FV + PV * (A + 1)] / [A * B] = -[0 + 100000 * (2.313498 + 1)] / [2.313498 * 300] = -331349.8 / 694.0494 = -477.415296 = -477.42$ 。

答案：你每月须还款477.42元。

怎样：记录房屋按揭贷款

房屋按揭贷款可以使用“创建科目”一节中给出的科目结构。

举个例子，假定你在储蓄科目中有60,000元存款，你买了150,000元的房屋。按揭贷款收取6%的年利率和3%的管理费（如，过户费）。你决定先拿出50,000元的首付款，因此还需要借103,000元。

借钱之前的科目：

8.2.

| 科目名称 | 类型 | 合计 |
|-----------|-------|--------------|
| ▼ 所有者权益 | 所有者权益 | CN¥60,000.00 |
| 期初余额 | 所有者权益 | CN¥60,000.00 |
| ▼ 资产 | 资产 | CN¥60,000.00 |
| ▼ 流动资产 | 资产 | CN¥60,000.00 |
| 储蓄存款 | 资产 | CN¥60,000.00 |
| ▼ 固定资产 | 资产 | CN¥0.00 |
| 住房 | 资产 | CN¥0.00 |
| ▼ 费用 | 费用 | CN¥0.00 |
| 住房按揭贷款管理费 | 费用 | CN¥0.00 |
| ▼ 利息 | 费用 | CN¥0.00 |
| 住房按揭贷款利息 | 费用 | CN¥0.00 |
| 收入 | 收入 | CN¥0.00 |
| ▼ 负债 | 负债 | CN¥0.00 |
| ▼ 贷款 | 负债 | CN¥0.00 |
| 住房按揭贷款 | 负债 | CN¥0.00 |

CN¥ 总计: 净资产: CN¥60,000.00 利润: CN¥0.00

房屋购买记录为资产:固定资产:房屋科目下的分录交易, 其中包含来自银行的50,000元(即你的首付款), 以及来自按揭贷款的100,000元。你可以将3,000元的过户费记到同一分录中, 我们会将房屋贷款增加到103,000元以将过户费纳入其中。

8.1.

| 科目 | 增加 | 减少 |
|--------------|-----------|-----------|
| 资产:固定资产:房屋 | ¥ 150,000 | |
| 资产:流动资产:银行 | | ¥ 50,000 |
| 负债:贷款:房屋按揭贷款 | | ¥ 103,000 |
| 费用:房屋按揭贷款管理费 | ¥ 3,000 | |

资产:固定资产:房屋中的分录交易如下:

8.4.

gnc-homeloan.1.gnucash - 科目 - GnuCash

文件(F) 编辑(E) 查看(V) 操作(A) 企业(B) 报表(R) 工具(T) 窗口(W) 帮助(H)

保存 关闭 打开 编辑 新建 删除

科目

| 科目名称 | 类型 | 合计 |
|-----------|-------|---------------|
| ▼ 资产 | 资产 | CN¥160,000.00 |
| ▼ 固定资产 | 资产 | CN¥150,000.00 |
| 住房 | 资产 | CN¥150,000.00 |
| ▼ 流动资产 | 资产 | CN¥10,000.00 |
| 储蓄存款 | 资产 | CN¥10,000.00 |
| ▼ 负债 | 负债 | CN¥103,000.00 |
| ▼ 贷款 | 负债 | CN¥103,000.00 |
| 住房按揭贷款 | 负债 | CN¥103,000.00 |
| ▼ 所有者权益 | 所有者权益 | CN¥60,000.00 |
| 期初余额 | 所有者权益 | CN¥60,000.00 |
| ▼ 费用 | 费用 | CN¥3,000.00 |
| 住房按揭贷款管理费 | 费用 | CN¥3,000.00 |
| ▼ 利息 | 费用 | CN¥0.00 |
| 住房按揭贷款利息 | 费用 | CN¥0.00 |
| 收入 | 收入 | CN¥0.00 |

CN¥ 总计: 净资产: CN¥57,000.00 利润: CN¥-3,000.00

怎样：记录汽车贷款

汽车贷款，即通常所说的车贷，处理起来多多少少与房屋贷款相通。唯一的区别在于科目的不同和利率的差异。基本的汽车贷款科目结构如下：

- 资产
 - 流动资产
 - 储蓄存款
 - 固定资产
 - 汽车
- 负债
 - 贷款
 - 汽车贷款
- 费用
 - 利息
 - 汽车贷款利息

-汽车贷款管理费

更多信息可参阅“怎样：记录房屋按揭贷款”一节。

怎样：记录给朋友的私人借款

你不总是从银行借钱，有时你找家人借钱，或是借钱给朋友。这一部分会介绍一种处理借给朋友借款的方法。

我们以如下普遍适用的科目结构为基础介绍具体做法：

- 资产
 - 银行
 - 银行账户
 - 欠你的钱
 - 个人
- 收入
 - 利息收入
 - 个人

这个例子展示怎样跟踪借给朋友张三总计2,000元的个人借款。

借款约定

张三想从你这里借2,000元，并计划在接下来的18个月里每个月偿还一部分。因为他是你的朋友（不过也不太熟），所以你们一致认可5%的年利率。

现在将已知具体信息汇总在下面。张三的借款具体信息：

- 本金金额——¥2,000
- 借款期限——18个月，一年12次还款。
- 年利率——5%
- 每月还款金额——??

所以你要怎样计算每月还款金额呢？

你可以用很多工具进行计算，纸笔、Linux计算器、Apache OpenOffice/LibreOffice的电子表格组件，等等。但还是使用GnuCash的〔贷款还款计算器〕最为简单。可算得每月还款金额为¥115.56。

不过，你还得知道每月还款金额中的多少偿还的是利息、多少偿还的是本金，这样你才能将收到的还款恰当地记录到账簿中。为此，你需要更强大的工具，像是Apache OpenOffice/LibreOffice中的电子表格组件，特别是其中的PMT函数。

8.5.

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

有关向张三借款的详细信息。

借款科目

先从下述科目开始，其中各科目的货币相同，均为人民币。

8.1.

- 资产:银行:人民币
- 资产:他人欠款:张三
- 收入:利息收入:张三
- 所有者权益:期初余额:人民币

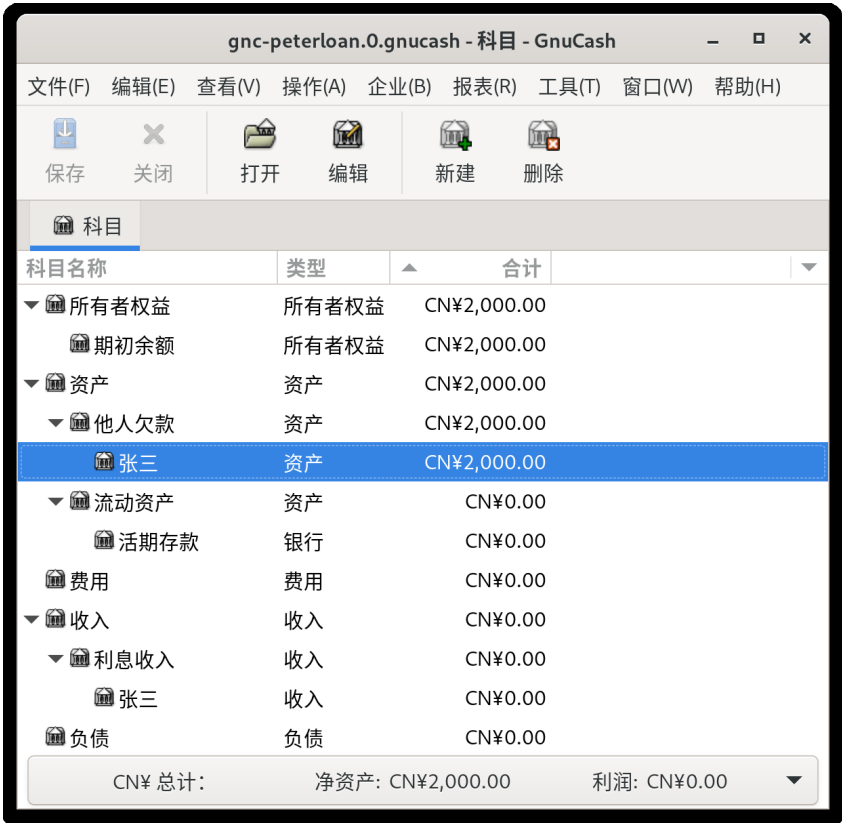
借出钱款

当你把钱借给朋友的时候，你实际上就是将钱从资产科目（如银行、活期存款或其它类似的科目）转账到另一个资产科目他人欠款。要记录这笔出借的资金，在资产:他人欠款中录入如下交易。

8.2.

| 科目 | 增加 | 减少 |
|------------|---------|---------|
| 资产:他人欠款:张三 | ¥ 2,000 | |
| 资产:银行:人民币 | | ¥ 2,000 |

8.6.



收到第一笔还款

收到第一笔还款（¥115.56）后，你需要确定其中多少属于借款本金，多少属于借款利息。

- 本期未偿还借款金额 = ¥2000
- 每月还款 = ¥115.56
- 还款金额分解
 - 利息： $5\% \div 12 \times ¥2,000 = ¥8.33$
 - 本金：¥115.56 - ¥8.33 = ¥107.23

此过程可录入为如下GnuCash交易。

8.7.

gnc-peterloan.1.gnucash - 张三 - GnuCash

文件(F) 编辑(E) 查看(V) 交易(N) 操作(A) 企业(B) 报表(R) 工具(T) 窗口(W) 帮助(H)

保存

关闭

创建副本

删除

确认

取消

转到空行

展开分录

科目

张三

| 日期 | 编号 | 描述 | | 借方总计 | 贷方总计 | 余额 |
|------------|----|---------|----------------|----------|----------|----------|
| 2024-06-10 | | 借钱给张三 | | 2,000.00 | | 2,000.00 |
| | | | 资产:他人欠款:张三 n | 2,000.00 | | |
| | | | 资产:流动资产:活期存款 n | | 2,000.00 | |
| 2024-07-10 | 编号 | 张三第一次还钱 | | 借方总计 | 107.23 | 1,892.77 |
| | | | 资产:流动资产:活期存款 n | 115.56 | | |
| | | | 资产:他人欠款:张三 n | | 107.23 | |
| | | | 收入:利息收入:张三 n | | 8.33 | |
| | | | | | | |
| 2024-09-01 | | | | | | |
| | | | n | | | |

现在: CN¥1,892.77 将来: CN¥1,892.77 已结清: CN¥0.00 已对账: CN¥0.00 预计最低: CN¥1,892.77 排序: 标准 ^

第一笔还款的账簿细节。

张三的借款余额现在是 $¥2,000 - ¥107.23 = ¥1,892.77$ 。

收到第二笔还款

收到第二笔还款（¥115.56）后，你同样需要确定其中多少是借款本金，多少是借款利息。

- 本期未偿还借款金额 = ¥1,892.77
- 每月还款 = ¥115.56
- 还款金额分解
- $5\% \div 12 \times ¥1,892.77 = ¥7.89$ 利息
- $¥115.56 - ¥7.89 = ¥107.67$ 本金

此过程可记为如下GnuCash交易。

8.8.

gnc-peterloan.2.gnucash - 张三 - GnuCash

文件(F) 编辑(E) 查看(V) 交易(N) 操作(A) 企业(B) 报表(R) 工具(T) 窗口(W) 帮助(H)

保存

关闭

创建副本

删除

确认

取消

转到空行

展开分录

科目

张三

| 日期 | 编号 | 描述 | | 借方总计 | 贷方总计 | 余额 |
|------------|------|---------|----------------|----------|--------|----------|
| 2024-06-10 | | 借钱给张三 | 资产:流动资产:活期存款 n | 2,000.00 | | 2,000.00 |
| 2024-07-10 | | 张三第一次还钱 | -- 分录交易 -- n | | 107.23 | 1,892.77 |
| 2024-08-10 | ▼ 编号 | 张三第二次还钱 | | 借方总计 | 107.67 | 1,785.10 |
| | | | 资产:流动资产:活期存款 n | 115.56 | | |
| | | | 资产:他人欠款:张三 n | | 107.67 | |
| | | | 收入:利息收入:张三 n | | 7.89 | |
| 2024-09-01 | | | n | | | |

现在: CN¥1,785.10 将来: CN¥1,785.10 已结清: CN¥0.00 已对账: CN¥0.00 预计最低: CN¥1,785.10 排序: 标准 ^

第二笔还款的账簿细节。

张三的借款余额现在是 $¥1,892.77 - ¥107.67 = ¥1,785.10$ 。

至此，科目表如下所示：

8.9.

gnc-peterloan.2.gnucash - 科目 - GnuCash

文件(F) 编辑(E) 查看(V) 操作(A) 企业(B) 报表(R) 工具(T) 窗口(W) 帮助(H)

保存

关闭

打开

编辑

新建

删除

科目

| 科目名称 | 类型 | 合计 |
|-------|-------|-------------|
| 资产 | 资产 | CN¥2,016.22 |
| 他人欠款 | 资产 | CN¥1,785.10 |
| 张三 | 资产 | CN¥1,785.10 |
| 流动资产 | 资产 | CN¥231.12 |
| 活期存款 | 银行 | CN¥231.12 |
| 所有者权益 | 所有者权益 | CN¥2,000.00 |
| 期初余额 | 所有者权益 | CN¥2,000.00 |
| 收入 | 收入 | CN¥16.22 |
| 利息收入 | 收入 | CN¥16.22 |
| 张三 | 收入 | CN¥16.22 |
| 费用 | 费用 | CN¥0.00 |
| 负债 | 负债 | CN¥0.00 |

CN¥ 总计:

净资产: CN¥2,016.22

利润: CN¥16.22

显而易见，每个月的利息在变化，每个月的本金也不同。所以，你每收到一笔还款就要计算各交易分录的金额。

每笔还款中的利息金额会越来越少（因为利息一直是根据持续减少的借款本金计算出来的），直到最后一笔还款后本金几乎为零。请回顾 有关向张三借款的详细信息。详细了解。

怎样：为贷款账单对账

根据贷款账单对账与根据银行账单或信用卡账单对账没有什么不同。

在此期间，你应该已经悉数记录了各种与贷款相关的交易，其中每一项交易都涉及〔负债：贷款：贷款类型〕科目。例如，偿还一部分贷款会减少银行科目和贷款科目的余额，同时会增加贷款利息，还可能会增加贷款管理费。

收到贷款账单后，打开相应贷款科目，启动对账向导，根据账单信息勾选所有先前记录的交易。完成后，对账差额应当为0。否则，你需要仔细核对科目账簿并同贷款账单对比找出差异。当差额为0时，贷款科目对账完成，此时你就可以结束对账向导了。

有关对账的更多信息，请参考“对账”一节。

怎样：出售房屋或汽车

当你要出售房屋时，你可以使用不同的记账方法在GnuCash中记录。这里，我们会详细研究其中两种记账方法。在其中一种方法中，你只需记录购买金额和当下的出售金额。在另一种方法中，你需要跟踪房地产市场的起落，并定期记录随时间变动的未实现损益。

简单交易

这种方法只需要恰当记录出售金额。

我们要记录两笔房屋出售，一笔盈利，另一笔亏损。如果你想出售汽车，只要把房屋科目换成汽车科目就可以了。

- 资产
- 固定资产
- 房屋
- 流动资产
- 储蓄存款
- 收入
- 长期资本利得
- 房屋

· 你之前花300,000元购买了房屋，现在又成功以600,000元出售。你要怎样记录这项交易呢？

要记录这项交易，你需要将储蓄存款科目增加600,000元，并将某些其它科目减少600,000元。房屋科目只有300,000元，正好为当初的购买金额，所以你将这些钱转账至储蓄存款科目。这样下来你还有300,000元的缺口。这笔钱你要从收入：长期资本利得：房屋科目中转账过来。录入储蓄存款科目（资产：流动资产：储蓄存款）的分录交易应当如下所示。

8.3.

| 科目 | 增加 | 减少 |
|--------------|-----------|-----------|
| 资产:流动资产:储蓄存款 | ¥ 600,000 | |
| 资产:固定资产:房屋 | | ¥ 300,000 |
| 收入:长期资本利得:房屋 | | ¥ 300,000 |

· 你之前花300,000元购买了房屋，但由于附近新建了一座机场，现在只能以230,000元出售。你要怎样记录这项交易呢？

要记录这项交易，你需要将储蓄存款科目增加230,000元，同时将某些其它科目减少230,000元。房屋科目包含300,000元，多于出售金额。所以应将230,000元转至储蓄科目。其后，房屋科目中仍剩70,000元需要转出。你应将这些钱转至收入：长期资本利得：房屋科目，表示亏损。录入房屋科目（资产：固定资产：房屋）的分录交易如下所示。

8.4.

| 科目 | 增加 | 减少 |
|--------------|-----------|-----------|
| 资产:固定资产:房屋 | | ¥ 300,000 |
| 资产:流动资产:储蓄存款 | ¥ 230,000 | |
| 收入:长期资本利得:房屋 | ¥ 70,000 | |

更复杂的交易

在这个例子中，我们将浅显地接触些更复杂的会计准则。相关详细内容请参考第 11 章 资本利得。

这里只涉及已经准确估计房屋现值的情况。对于其它情况（高估和低估），请参考第 11 章 资本利得。

- 资产
 - 固定资产
 - 房屋
 - 成本
 - 未实现损益
 - 流动资产
 - 储蓄存款
- 收入
 - 已实现损益
 - 房屋
 - 未实现损益
 - 房屋

你先前花300,000元购买了房屋，同时这些年一直密切关注市场动态并将估计的现值更新到账簿中。你现在打算出售房屋，查明其当前市场价值为600,000元。

600,000元（估计市场价值）与300,000元（购买金额）之间的差额就是当前的未实现损益金额。因此，你在资产:固定资产:房屋:未实现损益科目中有300,000元。

你该怎样记录这笔出售交易呢？

要记录出售交易，你需要将储蓄存款科目增加600,000元，并将某些其它科目减少600,000元。你必须首先将收入科目中的未实现损益变换为已实现损益。你最后需要将全部金额从资产:固定资产:房屋的子科目中转出。

录入收入:已实现损益:房屋科目的交易应该如下所示。

8.5.

| 科目 | 增加 | 减少 |
|----|----|----|
|----|----|----|

贷款

| | | |
|-------------|-----------|-----------|
| 收入:已实现损益:房屋 | ¥ 300,000 | |
| 收入:未实现损益:房屋 | | ¥ 300,000 |

录入资产:流动资产:储蓄存款科目的交易应该如下所示。

8.6.

| 科目 | 增加 | 减少 |
|------------------|-----------|-----------|
| 资产:流动资产:储蓄存款 | ¥ 600,000 | |
| 资产:固定资产:房屋:成本 | | ¥ 300,000 |
| 资产:固定资产:房屋:已实现损益 | | ¥ 300,000 |

在录入这些交易后，可以看到资产:固定资产:房屋的余额为0，储蓄存款科目增加了600,000元，以及收入:已实现损益增加到了300,000元。

第 9 章 投资

本章介绍怎样使用GnuCash管理投资。大部分用户都有投资计划，不论是银行存单、就职企业的激励计划还是通过经纪人买卖的股票和债券。GnuCash会提供给你有助于管理这些投资的工具，例如价格数据库就可以帮助你记录所持股票的价格起落。

基本概念

投资是你希望用于生产收入或希望未来以更高价格卖出而买入的东西。根据这个简单的定义，许多东西均可被视作投资：你居住的房屋、贵重的画作、上市公司的股票、储蓄存款账户或银行存单。本章会讨论怎样使用GnuCash跟踪这些纷杂多样的投资类型。

术语

具体讨论管理投资的相关事宜前，有必要理清投资领域的术语。下面列出的术语是投资中的基本概念。建议尽量熟悉这些术语，起码记得在后面的章节中遇到不了解的术语时回过头看看这份列表。

资本利得

投资品买卖价格差。如果卖出价比买入价低，就称为资本损失。资本利得也常被称作已实现损益。

佣金

买卖证券时支付给经纪人的费用。

普通股

代表拥有某家企业部分所有权的证券。这就是你在公开市场上“买入公司股票”时所得到的。有时也称为股本。

复利

将利息再次投资，以便将来得到它自身产生的利息（利息的利息）。

分红

分红是企业分发给股东的现金。分红金额往往大致根据企业利润额确定。注意不是所有普通股都会分红。

所有权

所有权指使投资者部分（或完全）拥有某物的投资，包括企业的普通股或房地产。

利息

借款人为使用放款人的钱而向放款人支付的费用，通常表示为每年相对于本金的百分比。例如，提供1%利息的储蓄账户（这里你自己是放款人，银行是借款人）每年会为你存入的每100元钱付给你1元的利息。

流动性

凭投资套取现金的难易程度。储蓄账户中的钱流动性极佳，而投入房产的钱流动性较差，因为出售房产要花时间。

本金

投资或借款的初始金额。

已实现损益 与 未实现损益

当资产价格变化时，你就取得了未实现损益。当实际卖出资产时，你就取得了已实现损益。另见资本利得。

回报

计入资本利得或资本损失的总收入。另见收益率。

风险

投资回报不同于预期的可能性。投资往往按照风险从低（储蓄存款、政府债券）到高（普通股、垃圾债券）归类。经验法则是，风险越高，可能的回报就越高。

股东

股东指持有企业普通股的人。

股票拆分（拆股）

当一家公司主动为目前已发行的股票额外配发相应倍数的份数时，这家公司就是在进行股票分割。例如，“2配1”的股票分割意味着如果你拥有一支股票的100份，你将会无偿得到额外的100份股票。每份股票的价格往往会发生调整，最终股票的总价值不会发生改变，所以在这个例子中每股的价格会减半。

估值

在合理的时间范围内确定投资品出售的市场价值或价格。

收益率

对从投资中赚取收益的度量，也就是你从投资中挣到了多少钱。一般将收益率记为相对本金金额的百分比。资本利得或损失不计入收益率（见回报）。例如，价格100元，每年分红2元的股票收益率为2%。

投资类型

下面列举了一些常见的投资大类，并为每类附上实际例子。

· 计息账户或凭证

这类投资往往能随时换取现金，一般会根据已存入的金额按月支付利息，如银行储蓄账户（及某些计息的活期账户）和经纪人提供的现金账户。这是一类风险极低的投资，这类账户一般受保险保护，可以防止一定限额内的损失。

有时，计息投资会锁定一段时间，即要求钱款需投资一段指定的时间，并获得约定的回报率。通常，投资的时间越长，利率就越高。如果在投资结束日期前取出钱款，则通常要支付提

前取款的罚金。这是风险相对较低的投资，如银行存单或某些政府债券。其它类型的债券可能因由发行方“信用评级”带来的更高风险而具有更高的收益率。

- 股票与公募基金

这是一种对企业的投资，在这一过程中你实际上成为了企业的部分所有者。公开交易股票通常没有锁定期，然而根据持有股票的时间长短，资本利得税率或手续费可能不同。因此，股票往往具有很好的流动性，即用股票换取现金相对方便。同时，这种类投资风险更高，因为股票未来的价格得不到保证。

公募基金是一种集合投资机制，你可以通过购入公募基金一次同时买入许多股票。例如，“中证500指数基金”是一种买入列入中证500指数中的全部500支股票的基金。当购买该基金的份额时，你其实就是对基金中包含的这500支股票中的每支都买入一小部分。不论是在税务上还是在会计上，公募基金的处理方法与单支股票几乎相同。

- 固定资产

价值随时间增长的资产是另一种形式的投资对象。常见的如房屋、小块土地或值钱的画作。这类投资直到出售前都是很难评估价值的。根据投资对象的不同，出售这类投资品的税收政策也不尽相同。例如，如果你出售的房屋是你的首要住所，那这笔出售会有税收减免，但是出售贵重的画作或许就不能享受类似的税收优惠。

固定资产投资将在第 11 章 资本利得和第 16 章 Depreciation 讨论。除了记录购买和出售交易，固定资产投资一般没有太多会计工作要做。

创建科目

要在GnuCash创建投资科目，可以使用预定义的投资科目结构，也可以自行创建。要跟踪投资，至少需要为每类持有的投资建立一个资产科目。然而，如上文所述，创建结构化的科目层级，并将相关的投资分为一组通常是更有条理的。例如，你可能想要将所有公开交易股票编为一组，放置在以用来买卖股票的经纪公司命名的父科目之下。

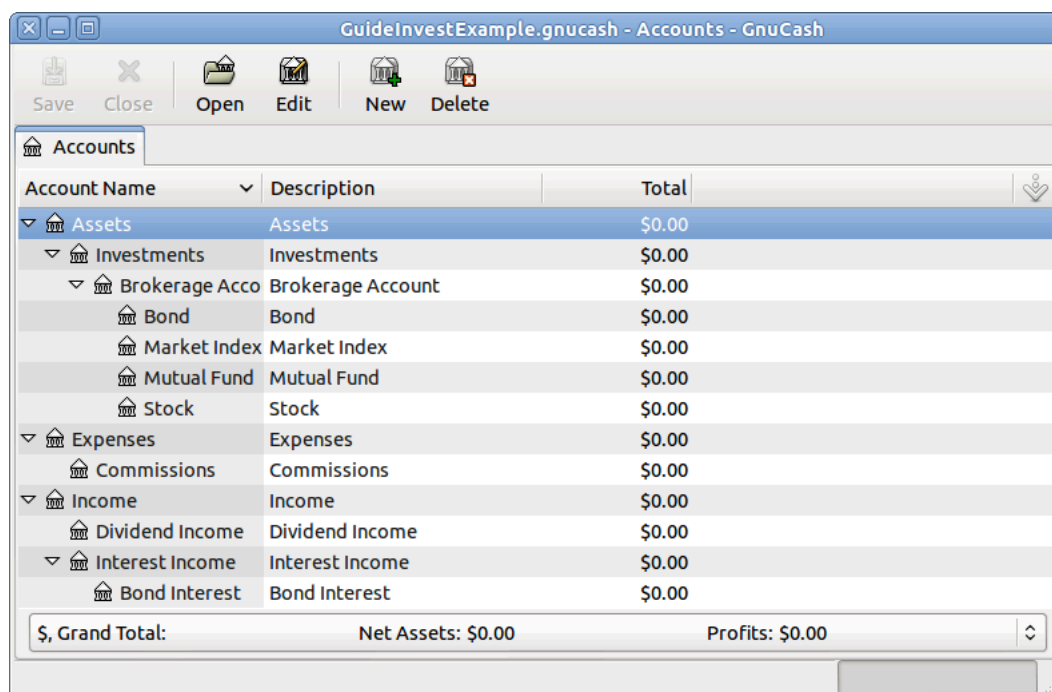
请注意，不论怎样组织科目层次结构，总能将来再随意移动这些科目，同时已经录入的信息不会丢失。所以，一开始建立科目层次无需一步到位。

使用预置的投资科目

〔建立科目结构〕向导中的〔投资科目〕选项会自动为你创建基本的投资科目层次结构。要使用预置的投资科目层次结构，首先打开GnuCash文件，切换到〔科目〕标签页，选择〔功能〕→〔建立科目结构〕。此菜单项会启动〔建立科目结构〕向导，引导你选择要新添加到层次结构中的科目。选择〔投资科目〕选项（也可一并选择你感兴趣的其它科目结构）。假设这里只选择了投资科目，则此向导创建的科目结构如下所示。

你也可以通过创建新的GnuCash文件运行〔建立科目结构〕向导。

9.1.



创建新文件且只选择默认投资科目后〔科目〕标签页的屏幕截图。

或许起码你会按照前几章所述向资产科目添加银行科目，并添加所有者权益：期初余额科目。不要忘记以贴切的文件名保存这份新的GnuCash文件。

手动创建投资科目

当然也可以自行建立投资科目层次结构。处理投资事宜常常需要创建几个相关联的科目：用于跟踪投资资金自身的资产科目、用于跟踪分红明细的收入科目，以及用于记录投资手续费的费用科目。

在典型的科目结构中，证券科目是资产科目的子科目，代表在某一经纪公司处开立的一个账户。经纪账户以本位货币计价，其下包括与每种在该经纪公司处交易的证券相对应的子科目。

相关的买入、卖出、收入和费用科目的货币也应当与经纪科目一致。

每个证券子科目应当设为从（用户定义的）主证券列表中选择的一种证券单位，同时使用与经纪科目一致的货币。

证券价格记录在GnuCash的〔价格数据库〕中（参见“设置份额价格”一节）。其中包含每一种证券单位（而非证券科目）的价格。单种证券的所有价格都由同一种货币记录。如果某种证券使用多种货币交易，则应当为每种货币建立各自独立的证券单位和证券科目。

自建科目示例

下面是一个稍显复杂的示例，展示了怎样设置用于跟踪投资的GnuCash的科目结构。其优势在于将每种不同的投资对象分组在处理投资交易的经纪商之下。这样更便于比对经纪商给出的账单与GnuCash中已有的科目交易记录，并定位GnuCash与账单间的差异。

资产

投资

经纪商科目

海通证券

股票

ACME公司

货币市场基金

海通证券市政基金

现金

天天基金

货币市场基金

流动资产基金

政府债券

Treas Bond xxx

Treas Note yyy

公募基金

基金A

基金B

现金

收入

投资

经纪商科目

资本利得

海通证券

天天基金

分红

海通证券

应税

免税

天天基金

应税

免税

利息收入

海通证券

应税

免税

天天基金

应税

免税

费用

投资费用

手续费

海通证券
天天基金
管理费用
海通证券
天天基金

确实没什么组织投资科目结构的标准做法。多尝试不同的组织结构，直到找到了某种在你看来将投资科目分组得既有逻辑又很明了的方式。

计息科目

利率固定或可变的投资是最简单也是最常见的投资形式。计息投资包括你的银行账户、存单或其它任何由本金产生利息的投资形式。这一节会介绍怎样在GnuCash中管理这类投资。

创建科目

当参与计息投资时，你需要创建一个资产科目用于记录这笔投资的买入、一个收入科目用于记录赚到的利息，还有一个费用科目记录银行收费。下面为一份科目结构的示例，其中，你在银行有一个计息的储蓄账户和一张存单。

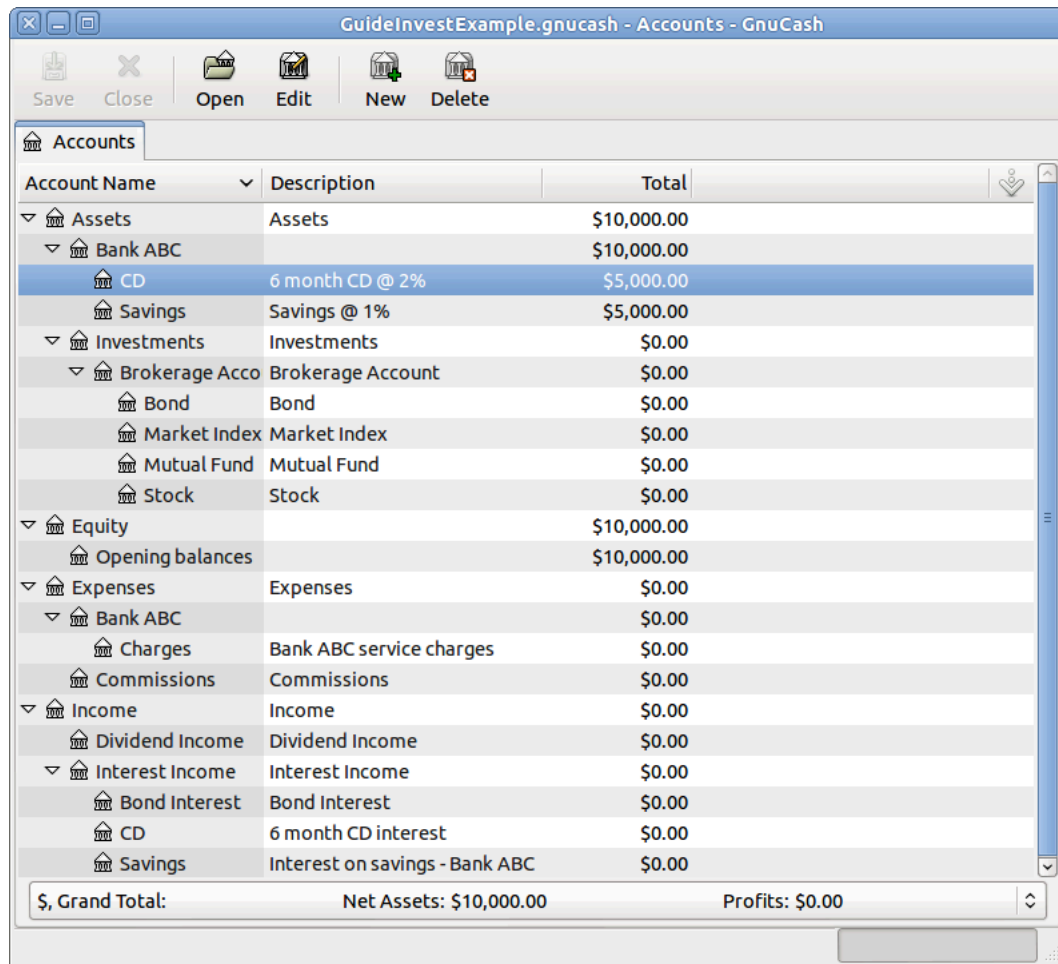
资产
 农业银行
 存单
 储蓄存款
费用
 农业银行
 服务费
收入
 利息收入
 存单
 储蓄存款

一般来讲，这份科目结构不过是一个示例，你应当以最符合实际情况的形式创建科目。

示例

现在将实际的金额填入这些科目。首先，假定一开始你的银行账户中有10000元，以1%的利率支付利息，同时你买了一份5000元，时长6个月，2%收益率的存单。显然，把你的钱存到存单中比存在储蓄账户中要好得多。最初购入存单后，你的科目应如下所示：

9.2.



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

这是在创建并投资存单后的科目表。

现在，在接下来6个月的时间里，你每月都会收到银行账单，其中记录了你的银行账户的活动。在这个虚构的例子中，我们没有动过这家银行的钱，因此唯一的活动就是利息收入和银行服务收费。银行服务费每月2元。6个月后，存单和储蓄科目账簿窗口应分别如下所示：

9.3. 6

| Date | Num | Description | Transfer | R | Increase | Decrease | Balance |
|----------|-----|------------------------|---------------------------|----------|-----------------|-----------------|----------------|
| 01/01/05 | | Purchase CD | Assets:Bank ABC | n | 5,000.00 | | 5,000.00 |
| 01/31/05 | | Interest | Income:Interest Income:CD | n | 8.33 | | 5,008.33 |
| 02/28/05 | | Interest | Income:Interest Income:CD | n | 8.35 | | 5,016.68 |
| 03/31/05 | | Interest | Income:Interest Income:CD | n | 8.36 | | 5,025.04 |
| 04/30/05 | | Interest | Income:Interest Income:CD | n | 8.38 | | 5,033.42 |
| 05/31/05 | | Interest | Income:Interest Income:CD | n | 8.39 | | 5,041.81 |
| 06/30/05 | | Interest | Income:Interest Income:CD | n | 8.40 | | 5,050.21 |
| 06/30/05 | | <i>Num Description</i> | <i>Transfer</i> | <i>n</i> | <i>Increase</i> | <i>Decrease</i> | <i>Balance</i> |

Present: \$5,050.21 Future: \$5,050.21 Cleared: \$0.00 Reconciled: \$0.00 Projected Minimum: \$5,050.21
Thursday 30 June 2005

6个月后的存单科目账簿。

9.4. 6

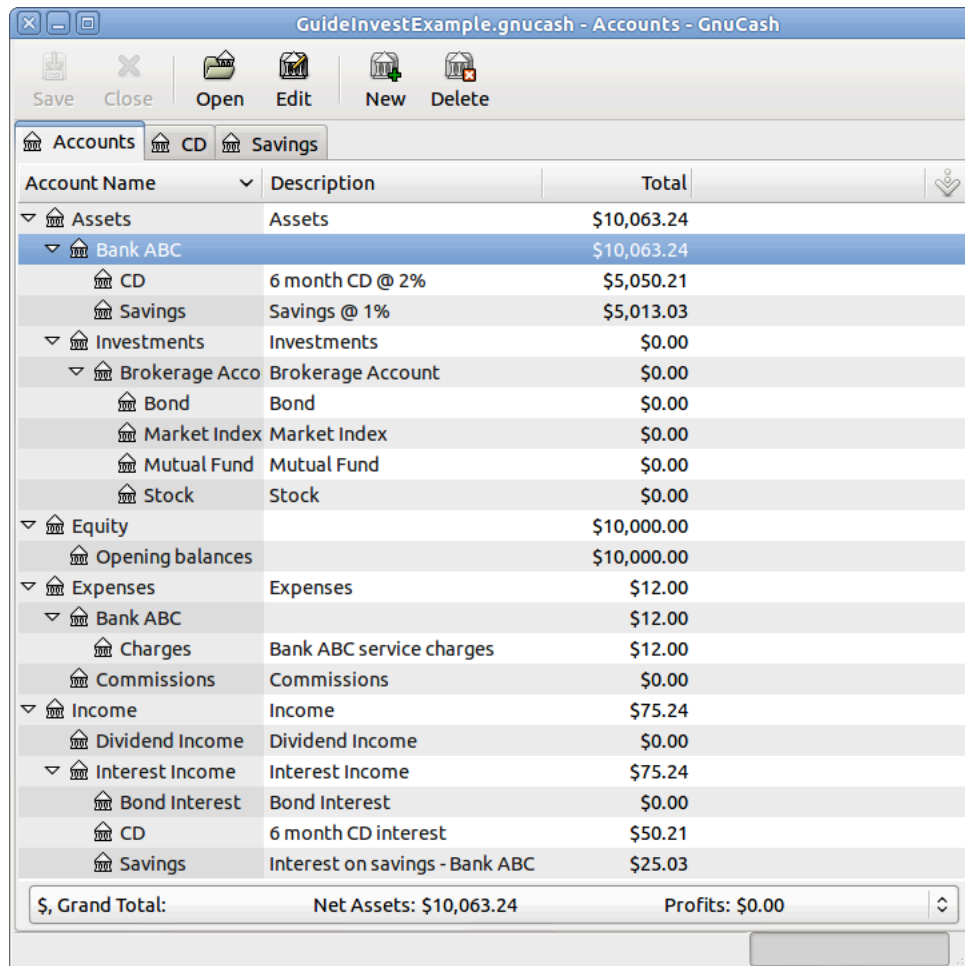
| Date | Num | Description | Transfer | R | Increase | Decrease | Balance |
|----------|-----|-------------|--------------------------------|----------|-----------------|-----------------|----------------|
| 02/28/05 | | Charges | Expenses:Bank ABC:Charges | n | | 2.00 | 5,004.34 |
| 03/31/05 | | Interest | Income:Interest Income:Savings | n | 4.17 | | 5,008.51 |
| 03/31/05 | | Charges | Expenses:Bank ABC:Charges | n | | 2.00 | 5,006.51 |
| 04/30/05 | | Interest | Income:Interest Income:Savings | n | 4.17 | | 5,010.68 |
| 04/30/05 | | Charges | Expenses:Bank ABC:Charges | n | | 2.00 | 5,008.68 |
| 05/31/05 | | Interest | Income:Interest Income:Savings | n | 4.17 | | 5,012.85 |
| 05/31/05 | | Charges | Expenses:Bank ABC:Charges | n | | 2.00 | 5,010.85 |
| 06/30/05 | | Interest | Income:Interest Income:Savings | n | 4.18 | | 5,015.03 |
| 06/30/05 | | Charges | Expenses:Bank ABC:Charges | n | | 2.00 | 5,013.03 |
| 06/30/05 | | <i>Num</i> | <i>Transfer</i> | <i>n</i> | <i>Increase</i> | <i>Decrease</i> | <i>Balance</i> |

Present: \$5,013.03 Future: \$5,013.03 Cleared: \$0.00 Reconciled: \$0.00 Projected Minimum: \$5,013.03
Enter a description of the transaction

6个月后的储蓄存款科目账簿。

这是GnuCash主科目窗口：

9.5. 6



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

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

6个月后的科目表。

上面展示的GnuCash主科目窗口充分汇总了6个月时间中这些投资项目发生的事情。由于存单的收益率是储蓄账户的两倍，存单的收益金额是50.21元，而储蓄账户的是13.03元，也就是几乎差了4倍。怎么回事？这是因为恼人的2元银行服务费把储蓄科目的金额拉下来了（6个月总计12元）。

6个月到了，存单到期了，也就是你不必交罚息就可以卖出存单。此时只需将5050.21元从存单科目转至储蓄存款科目即可。

创建投资组合

现在，你已经按前文所述建立了科目结构，这一部分会展示怎样根据投资组合创建科目并向其填入金额。建立好投资组合之后，你可能有一些在开始使用GnuCash之前就已经购入的股票。对于这部分股票，请根据录入已持有的份额一节的指引操作。如果你不久前才购入了这些股票，则按新购份额一节操作。

创建股票和公募基金科目

这一部分会展示怎样向GnuCash中添加股票和公募基金科目。这里，假定你正在使用前文介绍的基本科目结构，但其原理可以应用于任何科目层次结构。

你现在应该有一个顶层的〔资产〕科目，向下几层还有一个名为〔股票〕的科目。点击科目名称旁指向右方的三角记号，将科目树一直展开到要创建科目的层级。你接下来需要在〔股票〕科目下为你持有的每支股票创建子科目，其类型为股票。每支股票都对应独立的科目。虽说科目名称可以是在你看来意义明确的任何字词，但这些股票科目通常使用相应的股票名称命名。下面是这种结构的样板（仅展示资产子科目）。

```
资产
├── 投资
│   ├── 经纪账户
│   ├── 债券
│   ├── 公募基金
│   ├── 股指
│   ├── 股票
│       ├── 浪潮信息
│       └── 中国黄金
```

如果你还想跟踪每支股票或基金的收入（分红、派息、资本利得），你还需要为每支你持有的会支付分红或利息的股票创建收入：分红：股票名称、收入：长期资本利得：股票名称、收入：短期资本利得：股票名称和收入：利息：股票名称科目。

股票科目示例

在这个例子中，假定目前持有100股浪潮信息的股票。首先创建股票科目“浪潮信息”，选中〔股票〕科目并点击菜单〔功能〕+〔新建科目…〕。会弹出新建科目对话框，按如下步骤依次操作，创建新的股票科目。

9.6.

The screenshot shows a software window titled "New Account - Assets:Investments:Brokerage Account:Stock:AMZN". It has two tabs: "General" and "Opening Balance". The "General" tab is selected and contains the following sections:

- Identification:**
 - Account name:
 - Account code:
 - Description:
 - Security/currency: - Smallest fraction:
 - Account Color:
 - Notes:
- ☐ Tax related
- ☐ Hidden
- ☐ Placeholder
- Account Type:** A list box with "Stock" selected, and "Mutual Fund", "A/Receivable" visible.
- Parent Account:** A list box with "Stock" selected, and "Equity", "Expenses" visible.

At the bottom are three buttons: "Help", "Cancel", and "OK".

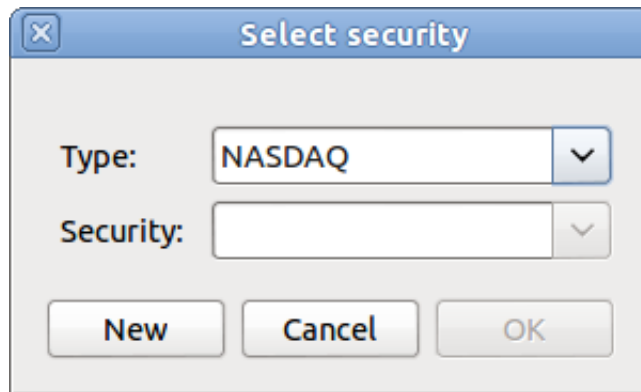
1. 科目名称 - 通常使用股票名称，如“浪潮信息”。
2. 科目代码 - 选填项，使用股票或基金的公开代码，或是你自己设定的编号。
3. 描述 - 选填项，详细描述商品或股票。注意这一项默认在〔科目〕标签页的树状列表中显示。
4. 科目类型 - 在左下方列表中为将要创建的科目选择类型。
5. 父科目 - 在右侧列表中为新建的科目选择父科目。可按需展开科目列表。
6. 创建新证券 - 要设为新的股票，你必须将股票创建为一种新的商品。

确认科目类型选为股票或公募基金，这样选择…按钮弹出的才是证券列表而非货币列表。

- 选择证券/货币 - 点击〔证券/货币〕行一侧的选择…按钮，弹出〔选择证券〕对话框。必须将证券由默认选项（最初设置的本位币）改为特定的股票。
- 类型 - 选择证券或商品交易的交易所（在这个例子中是沪市）。

点击〔新建〕按钮打开〔新建证券〕窗口。

9.7. [选择证券]



- 创建证券 - 点击新建…按钮，并在弹出的〔新建证券〕窗口中填入相应股票的信息。
- [全名]：这里是####。
- [简称]：这里是####。简称会提供给报价数据源（往下几行）。

不同的报价数据源可能会对同一支股票使用不同的简称。

- [类型]：应当已经设为〔沪市〕，因为这是在证券选择器中选中的选项。不过,你可以在这里修改，也可以添加更多类型。更多信息可参阅GnuCash用户手册8.7节“证券编辑器”。
- [ISIN、CUSIP或其它编码]：这里可以输入其它一些编号或文本（本例中留空）。
- [最小交易单位]：应当调整为这种证券能够交易的最小单位，通常是〔1/100〕或〔1/10000〕。
- “获取在线报价”复选框、报价数据源和时区：应选择以指定在线更新报价的数据源。参见“配置自动获取报价”一节。

如果获取在线报价未选中也无法点击，则说明Finance::Quote软件包尚未安装。详细信息请参阅用户手册中设置报价检索 [https://www.gnucash.org/docs/v5/C/gnucash-manual/finance-quote.html]一节。

设置完成后，此窗口应如下所示：

9.8. “ ”

New security

Security Information

Full name: Amazon.com

Symbol/abbreviation: AMZN

Type: NASDAQ

ISIN, CUSIP or other code:

Fraction traded: 1 / 10000

Quote Source Information

☒ Get Online Quotes

Type of quote source:

☒ Single: Yahoo USA

☐ Multiple: Asia (Yahoo, ...)

☐ Unknown: adig

Timezone: Use local time

Help Cancel OK

· 保存证券 - 点击确定按钮保存新创建的证券，〔新建证券〕窗口会关闭，并返回〔选择证券〕窗口。

7. 选择证券 - 你现在应该看到可以在〔证券/货币〕下拉列表中选择刚才新创建的证券了。选择此证券（可能默认已经选中了）并点击确定。

8. 最小交易单位 - 指定证券或商品交易时允许的最小单位。

9. 说明 - 输入与证券或商品相关的任何说明。

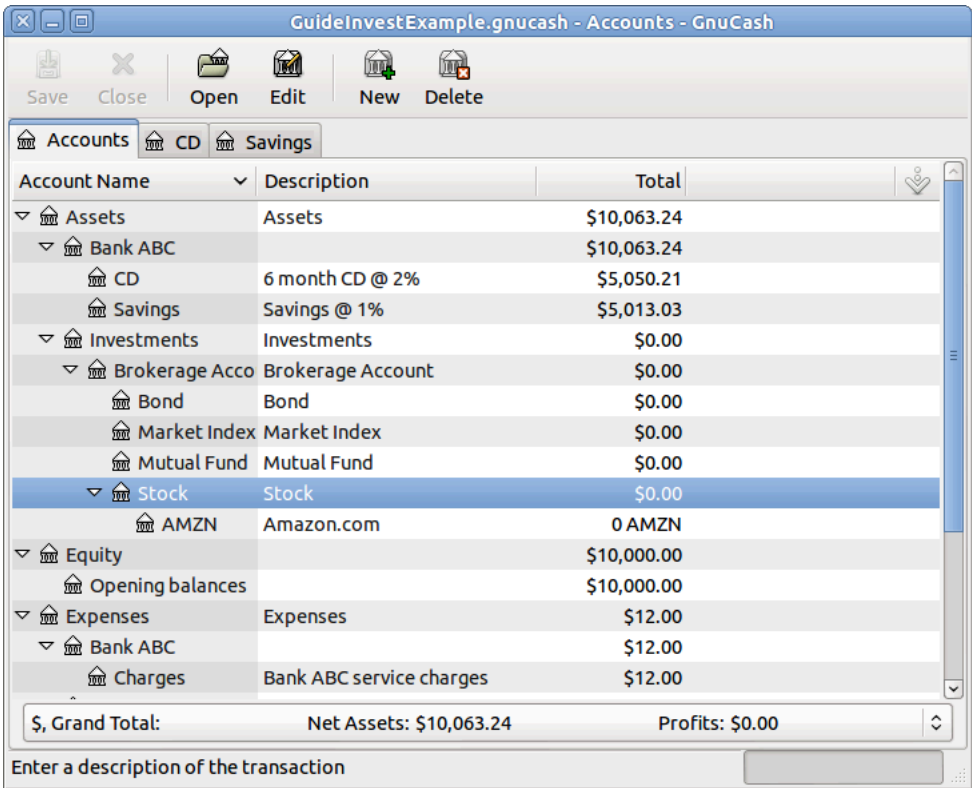
10. 应税 - 如果此科目的交易涉及所得税，则前往〔编辑〕→〔税务报表选项〕以勾选此框。

11. 占位符 - 如果此科目为占位符，即不包含任何交易，则勾选此项。

12. 完成 - 现在回到了〔新建科目〕对话框，其中〔证券〕一行已设为“浪潮信息”。点击确定按钮保存新创建的股票科目。

你已经创建了“浪潮信息”股票科目，你的科目表应当如下图所示。注意其中存在几个额外的科目、一个银行科目和一个所有者权益科目。

9.9.



在创建第一个股票科目浪潮信息后。

打开浪潮信息股票科目账簿，弹出的是 视图。此视图提供了对该商品相关交易的概览，包括买入卖出的交易单位数（对股票或公募基金而言是份额）、每单位的净值和交易总金额。显然我们还没有买入或卖出任何份额，所以账簿现在不含任何交易。

购入份额

录入已持有的份额

这部分示例使用〔交易日志〕视图。

要录入你先前买入的100份股票，在第一行交易处，录入买入日期（如，2005年1月1日）和〔描述〕（如，期初余额）。在第一行分录处，于〔份额〕列下输入**100**，删除〔价格〕（价格会在你按〈Tab〉键跳出分录行后自动计算），并在〔买入〕列下输入**2000**。

让GnuCash根据其它两列自动计算〔份额〕或〔购买〕也是可以的。但是为了避免舍入误差，最好还是让GnuCash自动计算〔价格〕。

按下 <Tab> 跳转到第二行分录，录入自科目所有者权益:期初余额的转账。为了便于理解，本示例假定此交易不收取手续费。现在你的浪潮信息科目账簿应该如下所示：

9.10. 浪潮信息

| Date | Action | Memo | Account | R | Shares | Price | Buy | Sell |
|------------|--------|-----------------|-------------------------------------|---|--------|-------|-------|----------|
| 01/01/2005 | | Opening Balance | | | 100 | | 100 | 100 |
| | Buy | | Assets:Brokerage Account:Stock:AMZN | n | 100 | 20 | 2,000 | Sell |
| | | | Equity:Opening Balances | n | | | | 2,000.00 |
| 01/01/2005 | | | | | 0.00 | | 1 | |
| | | | | n | | 1 | | |

Shares: 100 AMZN Current Value: \$3,600.00

此图展示了购入第一笔股票后的浪潮信息科目账簿。

注意〔余额〕（最后一列）以商品（浪潮信息的份额）为单位，而非货币单位。因此，余额显示为100（浪潮信息的份额）而非2000元。不过，本应如此。

新购份额

录入新买入的股票与录入先前已经买入过的股票之间的主要差异在于，不要从所有者权益:期初余额转入用于买入股票的钱，而是从资产:农业银行或资产:经纪账户科目转入。

处理佣金和手续费

为了行文简洁，本文档会将你支付给经纪商以买入卖出证券的钱称为佣金。可能还需支付政府税费。除非另行说明，这类费用的处理方式与佣金类似。

GnuCash中，录入佣金有两种方式，可以选择其中的任意一种，为了行文简洁，下文会将这两种方式分别称为净价格法和毛价格法。

毛价格法

录入未经佣金调整的价格，并将佣金费用录入为交易下的独立分录。这种方式可以跟踪佣金支出，但不兼容“使用批次卖出份额并自动计算资本损益”一节功能。处理程序不知道怎样从收益中扣除佣金，所以你需要手动计算资本损益（参见“卖出份额时手动计算资本损益”一节）。

净价格法

在买入卖出证券时录入净价格，即经佣金调整过的价格。同时不应将佣金录入为一项来自某一佣金科目的费用，因为如此一来佣金实际上会被记录两次。这种方式与“使用批次卖出份

额并自动计算资本损益”一节功能兼容。这会致使将稍有误导性的价格（即你支付的等效价格）添加到价格数据库中，但通常没什么问题。

如果你仍然不确定哪种方式对你而言更加合适，则请寻求专业人士的建议。

示例：使用毛价格法买入份额

在这个示例中，你会以50元每股的价格买入100股科大讯飞的股票，支付100元佣金。第一步是创建科大讯飞的股票科目。这里沿用费用：佣金科目。

现在录入交易，在第一行交易中输入买入〔日期〕（如，2005年1月3日）和〔描述〕（如，科大讯飞首次买入）。在第一行分录中，于〔份额〕列输入**50**，清空〔价格〕（将之留空以便稍后计算），并在〔买入〕列下输入**5000**。无需填写〔价格〕列，因为它会在你按〈Tab〉键跳至下一行分录时自动计算。分录交易的下一行是费用：佣金，在〔买入〕列下填入**100**。第三行分录是从资产：农业银行科目转账5100元以使交易平衡。你的科大讯飞科目现在应该如下所示：

9.11. 科大讯飞

| Date | Action | Memo | Account | R | Shares | Price | Buy | Sell | Balance |
|------------|--------|----------------------------|------------------------------------|---|--------|-------|--------|----------|---------|
| 01/03/2005 | | Initial IBM Purchase | | | 50 | | 50 | | 50 |
| | Buy | 50 shares @ \$100 = \$5000 | Assets:Brokerage Account:Stock:IBM | n | 50 | 100 | 5,000 | | |
| | | | Expenses:Commission:IBM | n | | | 100.00 | | |
| | | | Assets:Bank ABC | n | | | | 5,100.00 | |
| | | | | | | 1 | | | |
| 16/03/2017 | | | | | 0.00 | | | | |

Shares: 50 IBM Current Value: \$5,000.00
50 shares @ \$100 = \$5000

此图展示了在第一次买入股票并支付佣金后科大讯飞科目的交易账簿。

示例：使用净价格法买入份额

在交易日志视图中重复前文示例，但使用净价格法而非毛价格法。

购买5000元的科大讯飞股票，共计50股，每股100元，并支付100元佣金。

现在录入交易，在第一行交易中输入买入时的〔日期〕（如，2005年1月3日）和〔描述〕（如，科大讯飞首次买入）。在第一行分录中，若需要可在〔备注〕列下输入更多详细信息，然后在〔份额〕列下输入**50**，删除〔价格〕列下的所有内容（这样当按下〈Tab〉键跳出分录时会通过将〔买入〕除以〔份额〕计算出价格数值），在〔买入〕列下输入**5100**（ $50 \times 100 \text{元} + 100 \text{元}$ ）。可以使用GnuCash计算〔买入〕的值，在〔买入〕列下输入算式 **$5000+100$ 或 $(50*100)+100$** ，然后按下〈Tab〉键跳出〔买入〕列时会自动计算。继续按〈Tab〉键录入下一行分录。

不要再为佣金录入分录，因为佣金已经计入〔买入〕金额中了。第二行分录是从资产:农业银行科目转账5100元使交易平衡。正确录入分录后，按下〈Enter〉键保存交易。你的科大讯飞账簿现应如下所示：

9.12.

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

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

此图展示了在第一次买入股票并支付佣金后科大讯飞科目的交易账簿。

设置份额价格

股票等商品的价格必须明确设置。股票科目跟踪你持有的股票的数量，而股票的价值存储在价格数据库中。〔价格数据库〕中记录的价格可以手动更新，也可以自动更新。

价格数据库初始设置

要使用〔价格数据库〕跟踪股票价格变化，必须首先创建股票。为此，打开〔价格数据库〕（〔工具〕→〔价格数据库〕）并点击添加按钮。在弹出的窗口中输入你想向〔价格数据库〕添加的商品。这里可以手动输入商品价格。新建商品窗口中有六项内容：

〔命名空间〕

买卖证券或商品的交易所（本例中是上海证券交易所，即沪市）。

〔证券〕

商品名称，必须从下拉列表中选择。

〔货币〕

价格使用的货币单位。

〔日期〕

此价格生效的日期。

〔类型〕

为以下之一：〔买入价〕（市场内买入份额的价格）、〔卖出价〕（市场内卖出份额的价格）、〔最新价〕（最新的交易价格）、〔净值〕（基金每份额的价格）或〔未知〕。股票

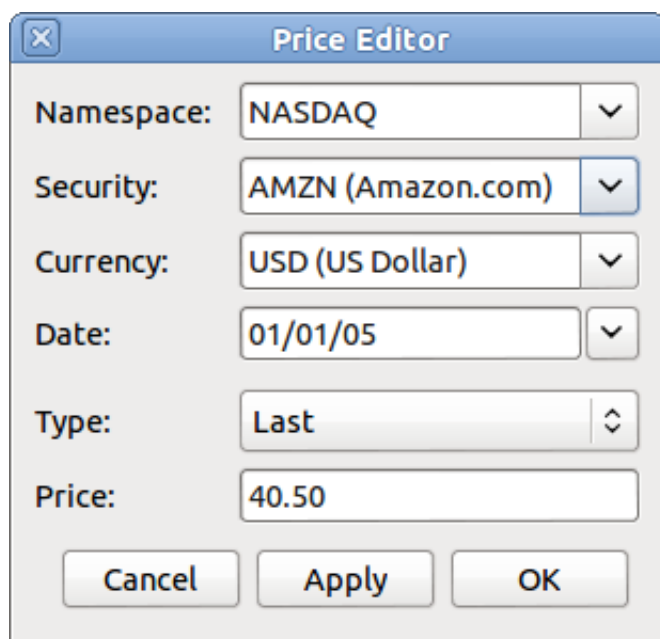
和外汇通常以买入价、卖出价或最新价中的一种形式给出报价，而基金往往以净值的形式给出报价。对于其它商品，选择〔未知〕即可。此选项只是随报价记录相关信息，本身不会被GnuCash使用。

〔价格〕

单位商品的价格。

下面的例子是向价格数据库添加浪潮信息股票报价，每股初始价格为40.50元。

9.13.



向价格数据库添加浪潮信息股票报价，每股初始价格为40.50元。

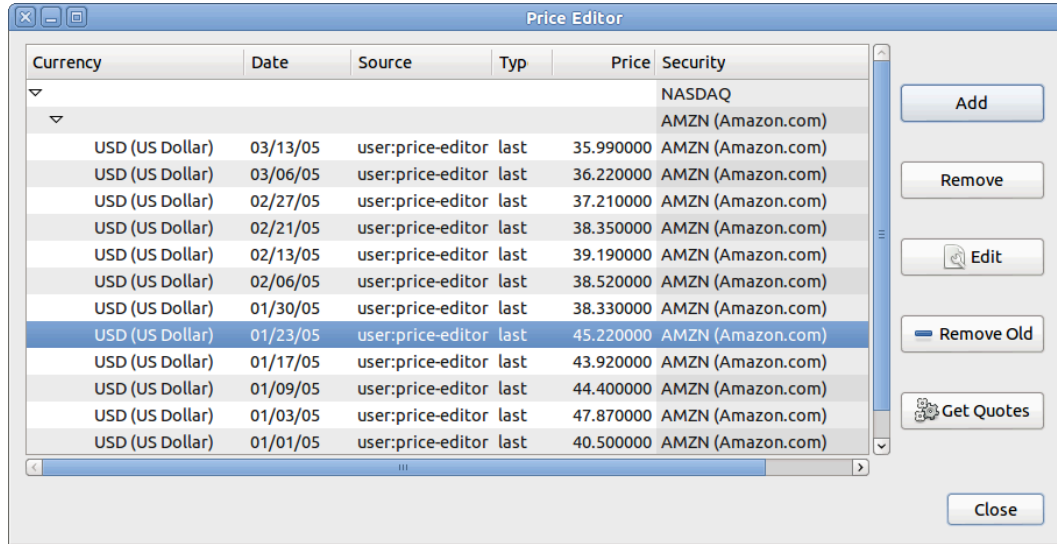
录入完成后点击确定。首次向〔价格数据库〕添加该商品后，就无须再次添加，即使在其它科目中使用同一商品后也不必重复此操作。

如果激活了获取在线报价功能（详见“配置自动获取报价”一节），不必手动录入报价就可以初始化商品价格。一开始在〔证券编辑器〕中添加证券时，勾选〔获取在线报价〕并保存证券。然后，在〔价格数据库〕中点击获取报价，则新添加的证券会随获取到的价格一起添加到价格列表中。

手动设置股票价格

商品（股票）的价格变化后，进入〔价格数据库〕，选择此商品，点击编辑并录入新价格，就可以完成对商品价格的调整。

9.14.



| Currency | Date | Source | Type | Price | Security |
|-----------------|----------|-------------------|------|-----------|-------------------|
| | | | | | NASDAQ |
| | | | | | AMZN (Amazon.com) |
| USD (US Dollar) | 03/13/05 | user:price-editor | last | 35.990000 | AMZN (Amazon.com) |
| USD (US Dollar) | 03/06/05 | user:price-editor | last | 36.220000 | AMZN (Amazon.com) |
| USD (US Dollar) | 02/27/05 | user:price-editor | last | 37.210000 | AMZN (Amazon.com) |
| USD (US Dollar) | 02/21/05 | user:price-editor | last | 38.350000 | AMZN (Amazon.com) |
| USD (US Dollar) | 02/13/05 | user:price-editor | last | 39.190000 | AMZN (Amazon.com) |
| USD (US Dollar) | 02/06/05 | user:price-editor | last | 38.520000 | AMZN (Amazon.com) |
| USD (US Dollar) | 01/30/05 | user:price-editor | last | 38.330000 | AMZN (Amazon.com) |
| USD (US Dollar) | 01/23/05 | user:price-editor | last | 45.220000 | AMZN (Amazon.com) |
| USD (US Dollar) | 01/17/05 | user:price-editor | last | 43.920000 | AMZN (Amazon.com) |
| USD (US Dollar) | 01/09/05 | user:price-editor | last | 44.400000 | AMZN (Amazon.com) |
| USD (US Dollar) | 01/03/05 | user:price-editor | last | 47.870000 | AMZN (Amazon.com) |
| USD (US Dollar) | 01/01/05 | user:price-editor | last | 40.500000 | AMZN (Amazon.com) |

价格数据库主窗口，显示了包含全部已知商品的列表。

配置自动获取报价

如果添加了多种商品，你就得一直更新它们的价格，那估计很快就会觉得烦了。GnuCash 可以从互联网上自动下载各商品的最新价格。这是通过 Perl 模块 Finance::Quote 实现的，必须安装该模块才能激活此功能。

参见手册中配置自动获取报价 [<https://www.gnucash.org/docs/v5/C/gnucash-manual/finance-quote.html>]一章。

配置证券的在线报价

在安装了 Finance::Quote 并正常工作的情况下，你必须配置证券才能通过此功能自动获取价格信息更新。无论是创建新证券还是修改已创建的证券，都要使用〔工具〕→〔证券编辑器〕编辑证券，选中“〔获取在线报价〕”复选框。现在可以更改单选框选择〔报价源类型〕，通过下拉菜单指定特定的报价源和〔这些报价的时区〕。完成设置后，〔关闭〕证券编辑器，返回〔价格数据库〕，然后单击获取报价按钮使用互联网上的报价更新证券价格。

配置定期获取在线报价

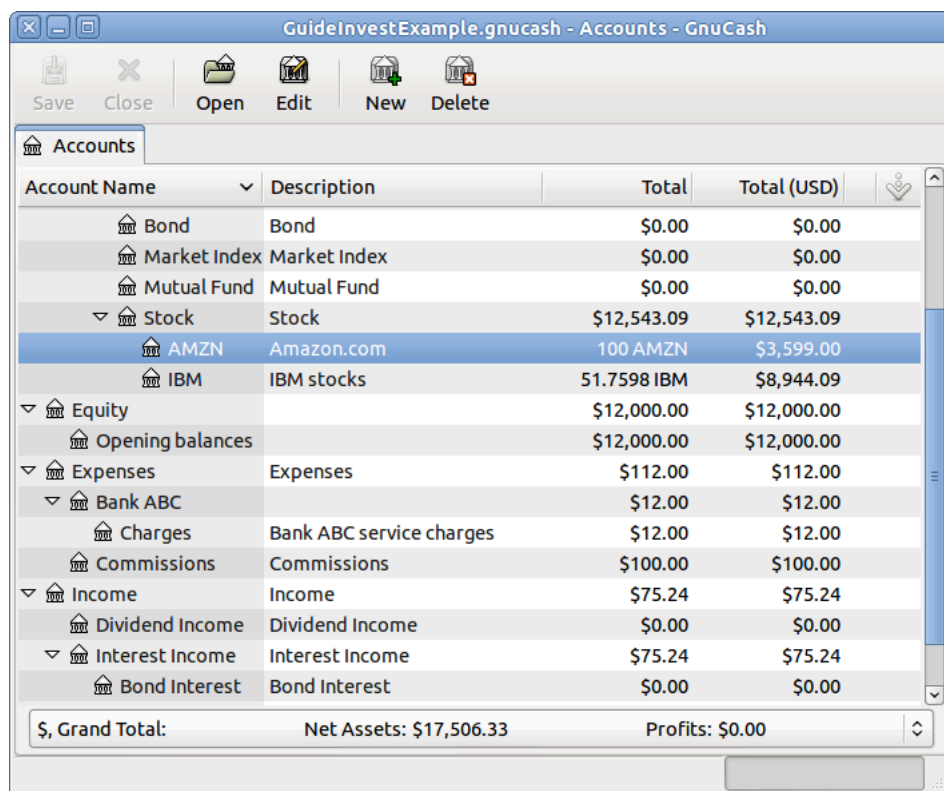
在验证能够正确获取商品的在线报价之后，你可以配置操作系统实现定期获取报价。参见手册中的使用 gnucash-cli 自动更新价格 [<https://www.gnucash.org/docs/v5/C/gnucash-manual/fq-command-line.html>]。

显示份额价值

主科目窗口默认仅在〔合计〕列下显示每种商品的持有数量。对股票而言，这种商品记录的是份额数量。然而，你常常想了解以货币单位表示的股票价值。进入主窗口，选择〔科目〕标签

页，点击表头右侧的下箭头，并勾选“合计(CNY)”选项。可以看到主窗口中新出现了名为〔合计(CNY)〕的一列，其下以报表货币单位给出了各类商品的价值。

9.15.



The screenshot shows the 'Accounts' window in GnuCash. The window title is 'GuideInvestExample.gnucash - Accounts - GnuCash'. The menu bar includes 'Save', 'Close', 'Open', 'Edit', 'New', and 'Delete'. The main table lists various accounts with their descriptions, totals, and totals in USD. The accounts are categorized by icons: Bond, Market Index, Mutual Fund, Stock, Equity, Expenses, Income, and Dividend Income. The 'Stock' category is expanded, showing 'AMZN' and 'IBM' with their respective quantities and values. The 'Equity' category shows 'Opening balances' and 'Expenses'. The 'Income' category shows 'Dividend Income' and 'Interest Income'. The 'Dividend Income' category shows 'Bond Interest'.

| Account Name | Description | Total | Total (USD) |
|------------------|--------------------------|-------------------------|-----------------|
| Bond | Bond | \$0.00 | \$0.00 |
| Market Index | Market Index | \$0.00 | \$0.00 |
| Mutual Fund | Mutual Fund | \$0.00 | \$0.00 |
| Stock | Stock | \$12,543.09 | \$12,543.09 |
| AMZN | Amazon.com | 100 AMZN | \$3,599.00 |
| IBM | IBM stocks | 51.7598 IBM | \$8,944.09 |
| Equity | | \$12,000.00 | \$12,000.00 |
| Opening balances | | \$12,000.00 | \$12,000.00 |
| Expenses | Expenses | \$112.00 | \$112.00 |
| Bank ABC | | \$12.00 | \$12.00 |
| Charges | Bank ABC service charges | \$12.00 | \$12.00 |
| Commissions | Commissions | \$100.00 | \$100.00 |
| Income | Income | \$75.24 | \$75.24 |
| Dividend Income | Dividend Income | \$0.00 | \$0.00 |
| Interest Income | Interest Income | \$75.24 | \$75.24 |
| Bond Interest | Bond Interest | \$0.00 | \$0.00 |
| \$, Grand Total: | | Net Assets: \$17,506.33 | Profits: \$0.00 |

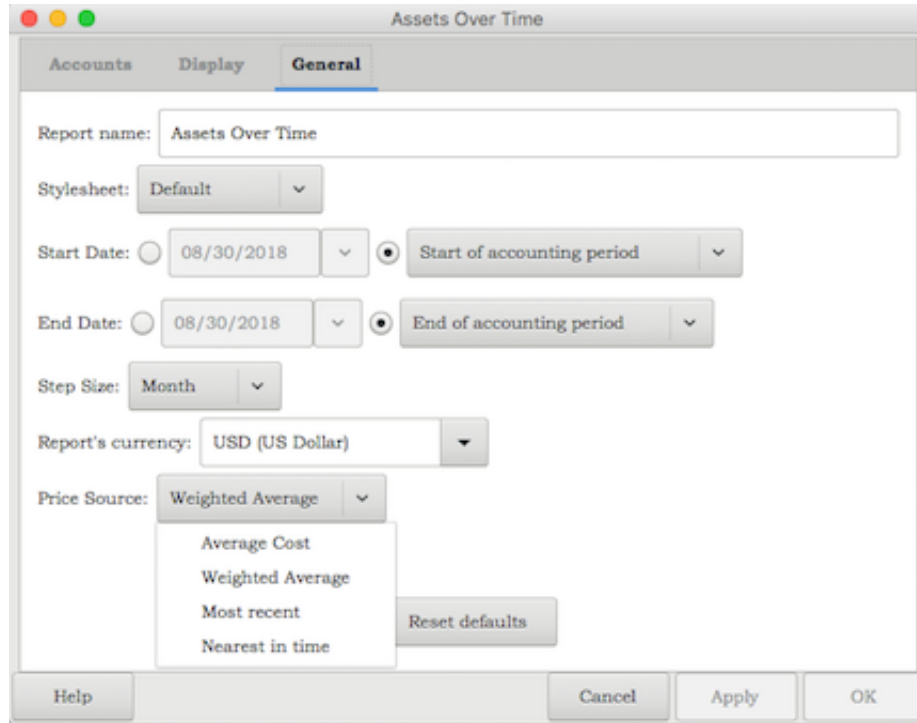
在主窗口中以报表货币单位查看股票总价值。

报表中的“价格来源”

多数GnuCash报表能够设置许多报表选项。选中报表标签页，点击菜单栏中的选项图标，或选择〔编辑〕→〔报表选项〕就可以打开选项对话框。〔价格来源〕指定了以不同于报表货币的商品为单位的科目怎样换算为报表货币。根据报表类型的不同，该选项可能出现在〔报表选项〕对话框中的〔常规〕、〔商品〕或〔显示〕标签页下。

在此示例中，选择菜单〔报表〕→〔资产负债〕→〔余额预测〕打开报表进行自定义。

9.16.



在报表中设置价格来源选项，确定非报表货币的股票商品或货币的价格。

〔加权平均〕

计算商品数量之和，以及各科目中以该商品为单位的各分录金额之和，其中不含数量为零的分录，最后总金额除以总数量得到价格。例如，如果你有一项买入交易，以2000元的金额买入200份甲乙丙股票，后以1300元卖出100份，则加权平均算法会计算 $3300/300$ ，即份额单价11元。

份额数量为0的损益分录不纳入计算。

〔平均成本〕

计算商品数量之和，以及各科目中以该商品为单位的各分录金额之和，其中包含数量为零的分录。在上面的例子中，如果有额外一项分录（可以是卖出交易的分录，也可构成独立交易）记录了收益，份额为0，收益300元，则平均成本为 $1000/100$ （2000元的初始成本 - 1300元卖出所得 + 300元收益）/（200 - 100 份额），即份额单价10元。

- 损益分录纳入计算。
- 这是唯一一种能够与〔试算平衡表〕相符的〔价格来源〕算法，要实现两者相符，你必须正确记录资本损益。

〔最新价格〕

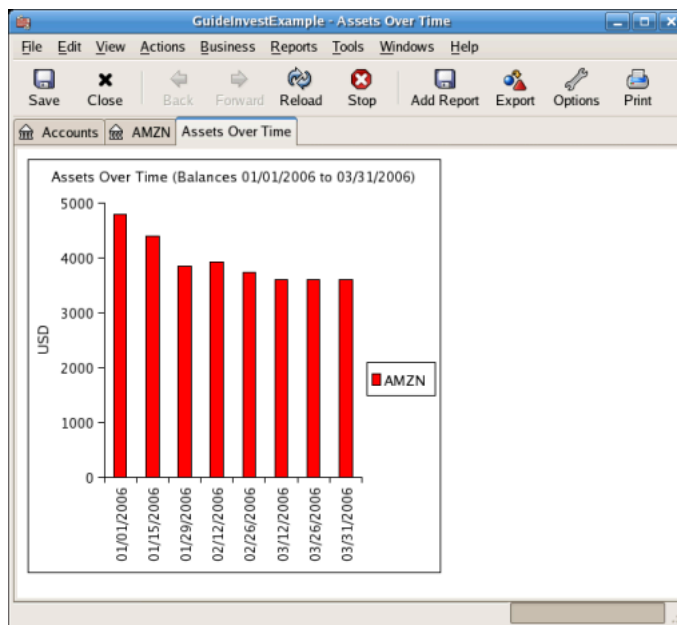
使用价格数据库中的最新价格。

〔时间临近〕

使用价格数据库中时间上最接近报表日期（如〔资产时变〕等时序报表中各数据点相应日期）的价格记录。

由给定日期确定的临近日期不一定在其之前。

9.17. " "



跟踪当前持有的股票所应有的价值。

卖出份额

录入投资出售交易与录入投资购入交易（参见“新购份额”一节）的操作类似，只是录入〔份额〕列的数额为负并在〔出售〕列中录入交易进款。交易进款应当录入为从份额科目到银行或经纪商科目的转账。

有关处理佣金与使用净价格或毛价格有关信息，请参见“处理佣金和手续费”一节。如果你打算记录出售带来的资本损益，请参见第 11 章 资本利得和第 16 章 Depreciation 以获取更多信息。要使用 GnuCash 的根据批次自动计算资本损益功能，请参见“使用批次卖出份额并自动计算资本损益”一节，否则请继续阅读下文。

卖出份额时手动计算资本损益

在GnuCash里，为了向下文所述交易中添加与资产:股票:某股票科目关联的零份额、零价格分录，你必须按下 `<Tab>` 将光标从当前分录中移出。如果你使用 `<Enter>`，GnuCash会将分录换算为商品份额。

在下文展示的交易示例中，会使用下述符号：

份额数量

要卖出的份额数量。

佣金

经纪商收取的佣金或其它费用。

毛卖出单价

未减去佣金的卖出份额的单价。

净卖出单价

已减去佣金的卖出份额的单价。

毛买入单价

未减去佣金的买入份额的单价。

净买入单价

已减去佣金的买入份额的单价。

毛买入总额

买入份额的总额，不含佣金，等于 份额数量 \times 毛买入单价。

净买入总额

买入份额时的出账金额，包含佣金，等于 毛买入总额 + 佣金。

毛卖出总额

买入份额的总价，不含佣金，等于 份额数量 \times 毛卖出单价。

净卖出总额

卖出份额时的入账金额，等于 毛卖出总额 - 佣金。

毛利润

未减去佣金的因卖出得到的利润。

净利润

已减去佣金的因卖出得到的利润。

有两种手动记录资本损益的方式。资本损益可以与卖出份额合并为一项交易，也可分别录入到独立的交易中。

合并卖出份额与资本损益为一项交易

这种方式的优点在于将卖出过程所涉及的所有资金流动记录在一起。这种方式与对消功能（参见“使用批次卖出份额并自动计算资本损益”一节）不兼容。如果你可能以后在特定的证券上使用对消功能，那现在就考虑在独立的交易中录入资本损益，以便减轻以后的一些工作量。

当资本损益与卖出份额分录合并为一项交易时，证券科目下同一项交易中会存在两项分录，所以交易必须在〔自动分录账簿〕或〔交易日志〕视图下的证券账簿中录入。证券账簿下一项分录记录卖出份额，另一项分录记录资本损益。用于记录资本损益的证券科目分录必须将份额数量录入为0并将份额单价录入为0，以阻止自动重新计算这两列。

用于处理损益的科目依然为收入：资本利得或费用：资本损失。

合并交易、毛价格

9.1.

| 科目 | 份额 | 份额单价 | 买入总额 | 卖出总额 |
|------------|-------|-------|-------|-------|
| 资产:农业银行 | | | 净卖出总额 | |
| 资产:股票:股票名称 | 0 | 0 | 毛利润 | (毛亏损) |
| 费用:佣金 | | | 佣金 | |
| 资产:股票:股票名称 | 一份额数量 | 毛卖出单价 | | 毛卖出总额 |
| 收入:资本利得 | | | (毛亏损) | 毛利润 |

9.1.

在此示例中，我们会使用“股票科目示例”一节创建的浪潮信息科目。

你以20元每股的价格买入100股浪潮信息的股票，然后以36元每股的价格全部卖出，并支付佣金75元。分录交易方案如下所示：

- 毛买入总额为20元（原始买入价），
- 份额数量为100，

- 毛买入总额为2,000元（买入花费的原始金额），
- 毛卖出总额为3,600元，
- 所以最终毛利润为1,600元（毛卖出总额 – 毛买入总额）。

9.2.

| 科目 | 份额 | 价格 | 买入 | 卖出 |
|------------------|------|-------|---------|---------|
| 资产:农业银行 | | | 3525.00 | |
| 资产:经纪商科目:股票:浪潮信息 | 0 | 0 | 1600.00 | |
| 费用:佣金 | | | 75.00 | |
| 资产:经纪商科目:股票:浪潮信息 | -100 | 36.00 | | 3600.00 |
| 收入:长期资本利得:浪潮信息 | | | | 1600.00 |

9.18.

The screenshot shows the GnuCash interface with a trial balance for the AMZN account. The table displays transactions from 2005 to 2017, including opening balances, stock purchases, sales, and commissions. The trial balance is as follows:

| Actic | Memo | Account | R | Shares | Price | Buy | Sell |
|------------|--|--------------------------------------|---|--------|-------|----------|----------|
| 01/01/2005 | Opening Balance | Equity:Opening Balances | n | 100 | 20 | 2,000.00 | 100 |
| 03/21/2006 | Sell stock for profit, Gross Pricing | -- Split Transaction -- | n | | | 1,600.00 | 100 |
| 03/21/2006 | Sell stock for profit, Gross Pricing | | n | -100 | | | 100 |
| | Net Sale = Gross Sale 3600 - Commission 75 = \$3525 | Assets:Bank ABC | n | | | 3,525.00 | |
| | Gross Profit = Gross Sale 3600 - Gross Buy 2000 = \$1600 | Assets:Brokerage Account:Stock:AMZN | n | | | 1,600.00 | |
| | Broker Commission | Expenses:Commission:AMZN | n | | | 75.00 | |
| | Sell Gross Sale 100 @ 36 = 3600 | Assets:Brokerage Account:Stock:AMZN | n | -100 | 36 | | 3,600.00 |
| | Capital Gain (Long) = Gross Profit | Income:Capital Gain (Long Term):AMZN | n | | | | 1,600.00 |
| 01/23/2017 | | Account | R | Shares | 1 | Buy | Sell |
| | | | n | | 1 | | |

Shares: 0 AMZN Current Value: \$0.00
Enter a description of the split

上方截图显示在2006年3月21日有两项交易。这是因为账簿处于〔自动分录账簿〕视图，而其中一项交易中包含有两项分录。〔交易日志〕视图可能显示得更清晰，详见“简单交易还是分录交易？”一节。因为账簿的卖出交易包含两项分录，因此交易必须在〔自动分录账簿〕或〔交易日志〕视图下录入，而无法在〔基本账簿〕视图下录入。

9.19.

The screenshot shows the GnuCash 'Accounts' window for a file named 'GuideInvestExample.gnucash'. The 'AMZN' account is selected. The table displays a hierarchical view of accounts with columns for 'Account Name', 'Description', 'Total', and 'Total (USD)'. The 'Stock' account is expanded, showing 'AMZN' (Amazon.com) with a balance of 0 and 'IBM' (IBM stocks) with a balance of 51.7598. Other accounts include 'Equity', 'Expenses', 'Income', and 'Interest Income'. The 'Grand Total' at the bottom shows 'Net Assets: \$13,907.33' and 'Profits: \$0.00'.

| Account Name | Description | Total | Total (USD) |
|------------------|--------------------------------|-------------------------|-----------------|
| Stock | Stock | \$8,944.09 | \$8,944.09 |
| AMZN | Amazon.com | 0 AMZN | \$0.00 |
| IBM | IBM stocks | 51.7598 IBM | \$8,944.09 |
| Equity | Equity | \$12,000.00 | \$12,000.00 |
| Opening Balances | Opening Balances | \$12,000.00 | \$12,000.00 |
| Expenses | Expenses | \$187.00 | \$187.00 |
| Bank ABC | Bank ABC | \$12.00 | \$12.00 |
| Charges | Bank ABC service charges | \$12.00 | \$12.00 |
| Commissions | Commissions | \$175.00 | \$175.00 |
| Income | Income | \$1,675.24 | \$1,675.24 |
| Cap Gain (Long) | Cap Gain (Long) | \$1,600.00 | \$1,600.00 |
| Dividend Income | Dividend Income | \$0.00 | \$0.00 |
| Interest Income | Interest Income | \$75.24 | \$75.24 |
| Bond Interest | Bond Interest | \$0.00 | \$0.00 |
| CD | 6 month CD interest | \$50.21 | \$50.21 |
| Savings | Interest on savings - Bank ABC | \$25.03 | \$25.03 |
| \$, Grand Total: | | Net Assets: \$13,907.33 | Profits: \$0.00 |

合并交易、净价格

9.3.

| | | | | |
|----------|-------|-------|--------|-------|
| 科目 | 份额数量 | 份额单价 | 买入总额 | 卖出总额 |
| 资产:农业银行 | | | 净卖出总额 | |
| 资产:股票:简称 | 0 | 0 | 净利润 | (净亏损) |
| 资产:股票:简称 | -份额数量 | 净卖出单价 | | 净卖出总额 |
| 收入:资本利得 | | | (资本损失) | 净利润 |

9.2.

在此示例中，我们会使用在“股票科目示例”一节中创建的浪潮信息科目。

你以每股20元的价格买入了100股浪潮信息（包含佣金），后来又以每股36元的价格全部卖出，佣金共75元。分录交易相关数值选取如下：

- 份额数量为100，
- 净买入单价为20元（原始买入价格），
- 净买入总额为2,000元（原始买入花费），

- 净卖出单价为35.25元，即 $(3600 - 75) / 100$ ，
- 毛卖出总额为3,600元，净卖出总额为3,525元，
- 最后的净利润为1,525元，即净卖出总额 - 净买入总额。

9.4.

| 科目 | 份额 | 单价 | 买入 | 卖出 |
|-----------------|------|-------|---------|---------|
| 资产:农业银行 | | | 3525.00 | |
| 资产:经纪科目:股票:浪潮信息 | 0 | 0 | 1525.00 | |
| 资产:经纪科目:股票:浪潮信息 | -100 | 35.25 | | 3525.00 |
| 收入:长期资本利得:浪潮信息 | | | | 1525.00 |

9.20.

| Date | Memo | Account | R | Shares | Price | Buy | Sell |
|------------|--|--------------------------------------|---|--------|-------|----------|----------|
| 01/01/2007 | Buy 100 @ 20 ea = \$2000 (including commission) | Assets:Brokerage Account:Stock:AMZN | n | 100 | 20 | 2,000.00 | |
| | Buy | Assets:Bank ABC | n | | | | 2,000.00 |
| 03/21/2008 | Sell stock for profit, Net Pricing | Assets:Bank ABC | n | -100 | | | 100 |
| | Net Sale = Gross Sale 3600 - Commission 75 = 3525 | Assets:Bank ABC | n | | | 3,525.00 | |
| | Net Profit = Net Sale 3525 - Net Buy 2000 = 1525 | Assets:Brokerage Account:Stock:AMZN | n | | | 1,525.00 | |
| | Sell Net Sale = Sell 100 @ 36 Ea = 3600 - Commission 75 = 3525 | Assets:Brokerage Account:Stock:AMZN | n | -100 | 35.25 | | 3,525.00 |
| | Capital Gain (Long) = Net Profit | Income:Capital Gain (Long Term):AMZN | n | | | | 1,525.00 |
| 01/23/2017 | | | n | 1 | 0.00 | | |

Shares: 0 AMZN Current Value: \$0.00

分离卖出份额与资本损益交易

这是使用对消功能自动创建资本损益交易所必需的。否则对消功能不会检测到这些交易，创建的资本损益交易金额也会是错误的。

分离交易、毛价格

9.5.

| 科目 | 份额数量 | 份额单价 | 买入总额 | 卖出总额 |
|----|------|------|------|------|
|----|------|------|------|------|

| | | | | |
|----------|-------|-------|-------|-------|
| 资产:农业银行 | | | 净卖出总额 | |
| 费用:佣金 | | | 佣金 | |
| 资产:股票:简称 | -份额数量 | 毛卖出单价 | | 毛卖出总额 |

9.6.

| | | | | |
|----------|------|------|-------|-------|
| 科目 | 份额数量 | 份额单价 | 买入总额 | 卖出总额 |
| 资产:股票:简称 | 0 | 0 | 毛收益 | (毛损失) |
| 收入:资本利得 | | | (毛损失) | 毛收益 |

9.3.

示例使用“股票科目示例”一节创建的浪潮信息科目。

你以每股20元的价格买入了100股浪潮信息股票，后来又以每股36元的价格全部卖出，佣金75元。分离交易形式如下：

- 毛买入单价为20元（原始买入单价），
- 份额数量为100，
- 毛买入总额为为2,000元（原始买入花费），
- 毛卖出总额为3,600元，
- 最终毛利润为1,600元（毛卖出总额－毛买入总额）。

9.7.

| | | | | |
|-----------------|------|-------|---------|---------|
| 科目 | 份额 | 单价 | 买入 | 卖出 |
| 资产:农业银行 | | | 3525.00 | |
| 费用:佣金 | | | 75.00 | |
| 资产:经纪科目:股票:浪潮信息 | -100 | 36.00 | | 3600.00 |

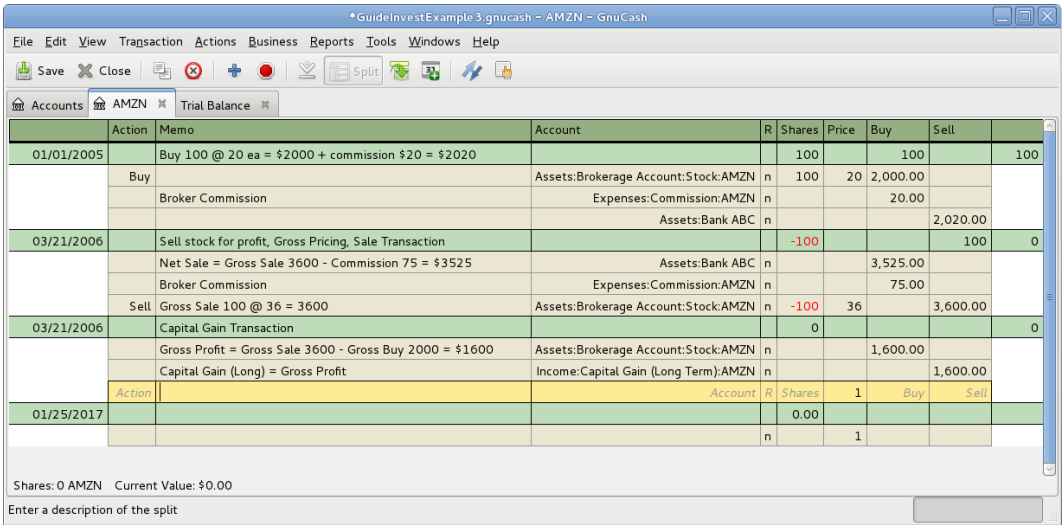
9.8.

| | | | | |
|-----------------|----|----|---------|---------|
| 科目 | 份额 | 价格 | 买入 | 卖出 |
| 资产:经纪科目:股票:浪潮信息 | 0 | 0 | 1600.00 | |
| 收入:长期资本利得:浪潮信息 | | | | 1600.00 |

志]

9.21.

[交易日



分离交易、净价格

9.9.

| | | | | |
|----------|-------|-------|-------|-------|
| 科目 | 份额数量 | 份额单价 | 买入总额 | 卖出总额 |
| 资产:农业银行 | | | 净卖出总额 | |
| 资产:股票:简称 | 一份额数量 | 净卖出单价 | | 净卖出总额 |

9.10.

| | | | | |
|----------|------|------|-------|-------|
| 科目 | 份额数量 | 份额单价 | 买入总额 | 卖出总额 |
| 资产:股票:简称 | 0 | 0 | 净利润 | (净亏损) |
| 收入:资本利得 | | | (净亏损) | 净利润 |

9.4.

此示例使用“股票科目示例”一节中创建的浪潮信息科目。

你以20元每股的价格买入了100股（计入手续费），后来又以36元每股的价格全部卖出，佣金75元。分离交易形式如下：

- 份额数量为100,
- 净买入单价为20元（原始买入价格），
- 净买入总额为2,000元（原始买入花费），
- 净卖出单价为35.25元 $((3600 - \$75) / 100)$),
- 毛卖出总额为3,600元，净卖出总额为3,525元,

· 最终净利润为1,525元 (净卖出总额 – 净买入总额)。

9.11.

| 科目 | 份额 | 单价 | 买入 | 卖出 |
|-----------------|------|-------|---------|---------|
| 资产:农业银行 | | | 3525.00 | |
| 资产:经纪科目:股票:浪潮信息 | -100 | 35.25 | | 3525.00 |

9.12.

| 科目 | 份额 | 单价 | 买入 | 卖出 |
|-----------------|----|----|---------|---------|
| 资产:经纪科目:股票:浪潮信息 | 0 | 0 | 1525.00 | |
| 收入:长期资本利得:浪潮信息 | | | | 1525.00 |

9.22.

[交易日

志]

| Action | Memo | Account | R | Shares | Price | Buy | Sell | Balance |
|------------|--|--------------------------------------|---|--------|-------|----------|----------|---------|
| 01/01/2005 | Buy 100 @ 20 ea = \$2000 + commission \$20 = \$2020 | Assets:Brokerage Account:Stock:AMZN | n | 100 | 20 | 2,000.00 | | 100 |
| | Buy | Assets:Brokerage Account:Stock:AMZN | n | 100 | 20 | 2,000.00 | | |
| | Broker Commission | Expenses:Commission:AMZN | n | | | 20.00 | | |
| | | Assets:Bank ABC | n | | | | 2,020.00 | |
| 03/21/2006 | Sell stock for profit, Gross Pricing, Sale Transaction | Assets:Bank ABC | n | -100 | | | 100 | 0 |
| | Net Sale = Gross Sale 3600 - Commission 75 = \$3525 | Assets:Bank ABC | n | | | 3,525.00 | | |
| | Broker Commission | Expenses:Commission:AMZN | n | | | 75.00 | | |
| | Sell Gross Sale 100 @ 36 = 3600 | Assets:Brokerage Account:Stock:AMZN | n | -100 | 36 | | 3,600.00 | |
| 03/21/2006 | Capital Gain Transaction | Assets:Brokerage Account:Stock:AMZN | n | 0 | | | | 0 |
| | Gross Profit = Gross Sale 3600 - Gross Buy 2000 = \$1600 | Assets:Brokerage Account:Stock:AMZN | n | | | 1,600.00 | | |
| | Capital Gain (Long) = Gross Profit | Income:Capital Gain (Long Term):AMZN | n | | | | 1,600.00 | |
| | Action | Account | R | Shares | Price | Buy | Sell | |
| 01/25/2017 | | | n | 0.00 | | | | |
| | | | n | | | 1 | | |

Shares: 0 AMZN Current Value: \$0.00

Enter a description of the split

使用批次卖出份额并自动计算资本损益

介绍

维基百科 [<https://en.wikipedia.org/wiki/Lot>]对批次给出了如下定义：

拍卖中一组共同出售的商品，或一定数量的金融票据。

GnuCash内置批次管理工具，能够跟踪由卖出证券带来的资本损益。买卖交易对应至各批次，以便计算所卖出证券的成本价。具体而言，批次是用来关联特定的买入和卖出交易的。批次的创建与关联可以自动完成也可以手动执行。通过批次，GnuCash能够自动计算出资本损益，并根据卖出金额同卖出证券成本价的差值创建交易。GnuCash将此过程称为对消。

之所以用对消这个词，是因为需要在卖出证券之后整理证券科目，确保为证券支付的成本价与由卖出获得的金额之间的差值记为资本损益。如果资本损益不正确，〔试算平衡表〕的底栏中借方合计与贷方合计不会平衡。要查看〔试算平衡表〕，选择〔报表〕→〔收入与支出〕→〔试算平衡表〕。

如果你操作出错，可以删除批次与资本损益交易并重试。在删除资本损益交易前，确保已经删除了批次，或起码取消了卖出交易同批次的关联。否则，〔科目中的批次〕页面会在你选择此批次时重新创建资本损益交易。

如果你不熟悉先入先出、后入先出或平均成本算法，请参阅维基百科的 FIFO and LIFO accounting [https://en.wikipedia.org/wiki/FIFO_and_LIFO_accounting] and Average cost method [https://en.wikipedia.org/wiki/Average_cost_method].

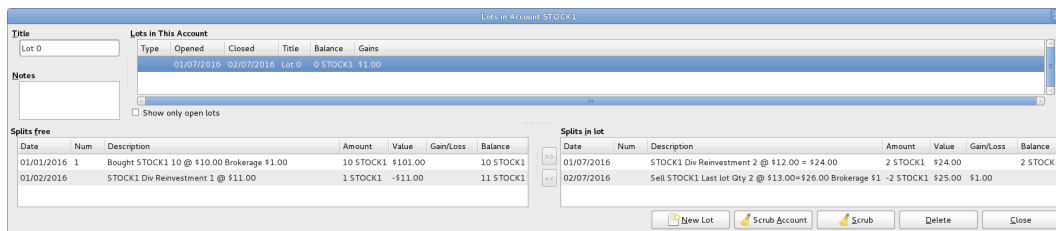
GnuCash批次管理工具是一项有用的功能，能够减少手动计算，特别是当红利再投已经进行了数年，涉及的成本价格各异的时候。它可以使用先入先出成本算法自动将买入交易关联到卖出交易，或手动将买入交易关联到卖出交易以实现后入先出成本算法。如果使用先入先出或后入先出成本计算方法，高级投资组合报表中的成本价和损益会与通过对消计算得到的成本价和损益一致。

科目窗口中的批次

〔科目SSSS中的批次〕窗口（其中SSSS是一证券科目）用于手动或自动关联证券交易分录至批次，并在科目中根据买入证券的成本与卖出证券所得金额之间的差值创建资本损益交易。

要打开〔科目批次〕窗口，打开证券科目账簿，并选择〔功能〕→〔查看批次〕。

9.23.



科目批次窗口。

参考帮助手册，第8章 工具与向导 - 科目批次 [https://www.gnucash.org/docs/v5/C/gnucash-manual/tool-lots.html]中的有关内容详细了解科目批次窗口的各项功能。

操作总结

使用批次管理工具自动计算资本损益的一般步骤：

1. 使用净价格录入卖出交易（但暂时不要录入资本损益交易，因为此交易会通过对消创建）。参阅 例 9.4 “卖出份额获利：手动计算损益、分离卖出与获利、净价格”。
2. “手动创建并关联批次”一节（部分成本核算方法需要）
3. “自动创建资本损益交易”一节
4. “将孤立的收益-CCC更改为损益科目”一节
5. “运行试算平衡”一节 确保贷方总额与借方总额平衡。

手动创建并关联批次

使用此功能前，确保你已经读过“注意事项”一节。

可以通过此功能手动关联特定的买入与卖出交易。如果想使用除自动关联方法（先进先出）之外的其它成本核算方法，就需要执行手动关联。实际上，如果希望高级投资组合报表中显示的成本价和资本损益与由对消功能创建的资本损益交易相符，则只有在使用后进先出或“个别计价法”时才需要手动创建批次。这是由于对消功能可以自动实现先进先出的关联，所以无需手动操作。另外，对消功能无法用于平均成本法。详见例 9.5 “手动创建并关联批次”。

自动创建资本损益交易

如果你没有使用先进先出或后进先出方法核算成本，则不要使用该功能。详见“注意事项”一节。

GnuCash能够自动计算并创建卖出证券时的资本损益交易。批次用于关联买入交易分录与卖出交易分录，以正确确定卖出证券的成本价。GnuCash会使用已经存在的批次，并为任何尚未关联到已有批次的买入交易创建新批次。买入与卖出交易分录使用先进先出方法关联到批次。

参见：

- 例 9.6 “自动创建资本损益交易”，
- 例 9.7 “自动创建资本损益交易，一次进行两笔卖出。”，
- 例 9.8 “自动创建资本利得交易 - 简单的拆股之后”。

将孤立的收益-CCC更改为损益科目

由对消创建的资本损益交易时，因为不知道应该用哪个资本损益账户，所以使用自动创建的通用孤立的收益-CCC账户（其中CCC是为证券交易货币）。执行对消后，应编辑孤立的收益-CCC交易分录，将科目重新指定为有意义的收入或费用类损益科目（如，收入：长期资本利得：浪潮信息）。

参见例 9.9 “将孤立的收益-CCC更改为损益科目”。

运行试算平衡

在创建资本损益交易后生成试算平衡报表（ [报表] → [收入与费用] → [试算平衡表] ）。试算平衡报表能够初步检查资本损益能否正确补足买入证券时支付的成本与卖出证券时收到的金额间的差值。在报告末尾，借方总额应等于贷方总额。

由于某些问题，试算平衡表可能没有平衡。请按照下述指引确定不平衡是否由错误记录的资本损益引起：

1. 如有必要，临时将卖出交易和资本损益交易的日期改为其它交易没有使用的日期，确保该日期下仅存在这两项交易。然后在前一天运行试算平衡。若试算平衡仍给出相同的金额，则问题不是由错误的资本损益记录引起。
2. 如果发现先前的试算平衡同样无法做到借贷平衡，则在不同的日期上运行试算平衡报表，直到找到最开始未能平衡的日期。临时将每项交易的日期更改为问题日期的后一天，然后一项交易一项交易地将日期改回正确日期，每次都运行试算平衡，直到定位到出现问题的交易。在更改证券卖出交易日期时，也要更改对应的资本损益交易的日期，因为只有这两项交易一起才能使试算平衡报表平衡。

注意事项

在使用批次管理工具前请注意一些要点。

1. 自动关联买入交易与卖出交易时，GnuCash只实现了先进先出成本计算方法，即最早买入的证券总是最先卖出。在对消前将最近的证券买入交易手动关联至卖出交易，则可以使用后进先出成本计算方法。
2. 高级投资组合报表在计算成本时不使用批次信息，而仅使用证券交易分录。报表根据选择的基本计算方法报表选项（平均、先进先出或后进先出）计算成本价和损益。若希望高级投资组合报表中的成本和损益与由对消功能创建的资本损益一致，则应在两处使用相同的成本计算方法。
3. 对消无法识别佣金或手续费，故计算损益时不会将这些款项纳入考虑。因此，若想使用对消，必须使用净价格而非毛价格。详见“处理佣金和手续费”一节。
4. 对消不会识别卖出交易下手动录入的资本损益分录。因此确保之前的卖出操作均各自记为两项交易：

9.13.

| 科目 | 总份额 | 单价 | 买入（借方） | 卖出（贷方） |
|--------|-----|----|--------|--------|
| 经纪商或银行 | | | 借 | |

| 证券科目 | -卖出数量 | 卖出单价 | | 卖出总额 |
|------|-------|------|--|------|
|------|-------|------|--|------|

9.14.

| 科目 | 总份额 | 单价 | 买入（借方） | 卖出（贷方） |
|------|-----|----|--------|--------|
| 资本亏损 | | | 借方金额 | |
| 证券科目 | 0 | 0 | | 贷方金额 |

5. GnuCash实现的资本损益自动计算功能能够处理直接的买入、卖出和资本退回交易，但任何改变份额数量的交易，即使是简单的分录，也会使其生成错误的结果，所以遇到这些情况必须手动处理。

示例

9.5.

下面给出一个使用后进先出方法出售持有的部分证券的示例。此示例中，最近买入的交易（2016年7月1日的红利再投）与卖出交易一起关联到同一批次，然后使用GnuCash的对消功能计算资本损益并创建资本损益交易。

1. 打开证券科目账簿。

9.24.

| Date | Num | Description | Tot Shares | Tot Buy | Tot Sell | Balance |
|------------|------|--|------------|---------|----------|---------|
| 01/01/2016 | 1 | Bought STOCK1 10 @ \$10.00 Brokerage \$1.00 | 10 | 10 | | 10 |
| | | Buy | 10 | 10.1 | | 101.00 |
| | | Assets:Investments:Broker1:STOCK1 | n | | | |
| | | Assets:Investments:Broker1 | n | | 101.00 | |
| 01/02/2016 | | STOCK1 Div Reinvestment 1 @ \$11.00 | 1 | 1 | | 11 |
| | | Buy Div qty 10 @ CPS \$2.00 = \$20.00 | 1 | 11 | | 11.00 |
| | | Increase in Residual \$20.00 - \$11.00 = \$9.00 | | | 9.00 | |
| | | Assets:Investments:DRP Residual:STOCK1 | n | | | |
| | | Income:Dividend:Franked:STOCK1 | n | | 20.00 | |
| 01/07/2016 | | STOCK1 Div Reinvestment 2 @ \$12.00 = \$24.00 | 2 | 2 | | 13 |
| | | Buy Div qty 11 @ CPS \$2.00 = \$22.00 | 2 | 12 | | 24.00 |
| | | Decrease in Residual = (\$9.00 - \$7.00) = \$2.00 | | | 2.00 | |
| | | Assets:Investments:DRP Residual:STOCK1 | n | | | |
| | | Income:Dividend:Franked:STOCK1 | n | | 22.00 | |
| 02/07/2016 | | Sell STOCK1 Last lot Qty 2 @ \$13.00=\$26.00 Brokerage \$1 | -2 | | 2 | 11 |
| | | Net Sale | | | 25.00 | |
| | | Assets:Investments:Broker1 | n | | | |
| | | Sell Cost qty 2 @ \$12.00 = \$24.00 | -2 | 12.5 | | 25.00 |
| | | Assets:Investments:Broker1:STOCK1 | n | | | |
| 23/02/2017 | Num: | | 0.00 | Tot Buy | Tot Sell | Balance |
| | | | n | 1 | | |

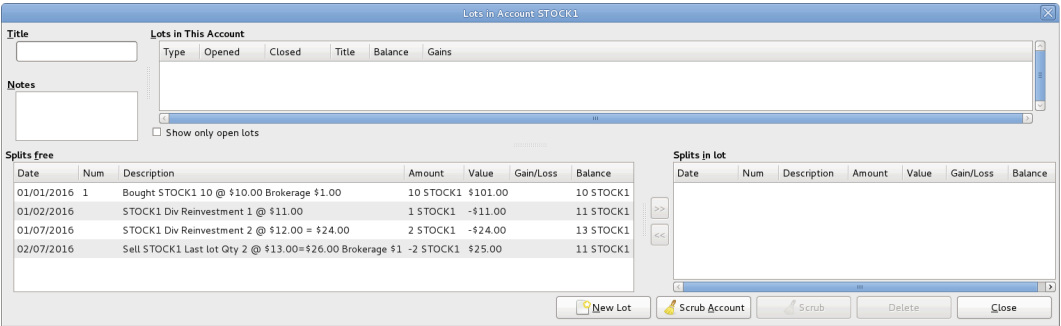
Shares: 11 STOCK1 Current Value: \$143.00

Enter a description of the transaction

对消批次前以交易日志视图显示的证券科目账簿。

2. 确保所有之前的资本损益交易独立于记录份额数量减少与收到回款的卖出交易。
3. 选择 [功能] → [查看批次] 打开 [科目SSSS中的批次] 窗口，其中SSSS为证券科目。

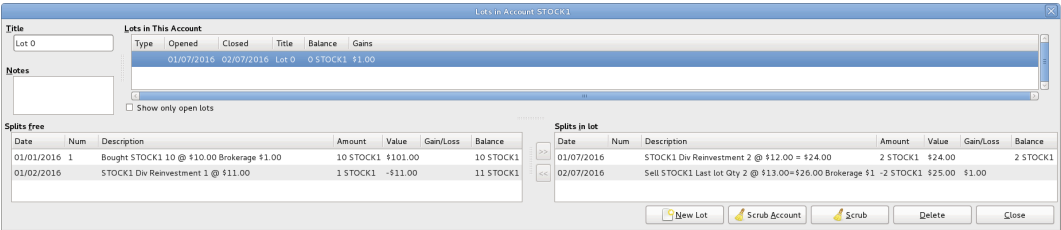
9.25. - -



对消批次前的科目中的批次窗口。

- 4. 使用新建批次按钮创建新批次。最开始，此批次没有关联到任何买入或卖出分录。
- 5. 在〔科目中的批次〕面板中选中新创建的批次。
- 6. 在〔未关联交易〕面板中，选中将要卖出的证券对应的交易分录（日期为2016年7月1日）。
- 7. 点击〔>>〕按钮将此买入分录关联到选中的批次。此分录会从〔未关联分录〕面板移动到〔批次中的分录〕面板。
- 8. 重复上述两步处理其它应当关联到各自批次的买入分录。（在此示例中，只有一项买入分录会对应卖出。）
- 9. 在〔未关联分录〕面板中选中卖出分录。
- 10. 点击〔>>〕按钮将此卖出分录关联到选中的批次。
- 11. 检查批次〔余额〕是否合乎预期。在示例中，批次余额应当为零，因为该批次中卖出证券的数量等于买入证券的数量。

9.26. - -



对消单个批次前的科目中的批次窗口。

- 12. 点击对消按钮（注意不是对消整个科目按钮）。
使用对消按钮后，〔科目中的批次〕窗口没有变化，因此这里不再给出屏幕截图。
- 13. 关闭〔科目SSSS中的批次〕窗口，回到证券科目窗口账簿。

9.27.

| GuideInvestExample.gnucash - STOCK1 - GnuCash | | | | | | | | | |
|--|-----|--|--|------------|---------|----------|---------|---------|------------------|
| File Edit View Transaction Actions Business Reports Tools Windows Help | | | | | | | | | |
| Save Close Split | | | | | | | | | |
| Accounts Advanced Portfolio LIFO STOCK1 | | | | | | | | | |
| Date | Num | Description | | Tot Shares | Tot Buy | Tot Sell | Balance | | |
| 01/01/2016 | 1 | Bought STOCK1 10 @ \$10.00 Brokerage \$1.00 | | 10 | 10 | | 10 | | |
| | | Buy | Assets:Investments:Broker1:STOCK1 | n | 10 | 10.1 | 101.00 | | |
| | | | Assets:Investments:Broker1 | n | | | 101.00 | | |
| 01/02/2016 | | STOCK1 Div Reinvestment 1 @ \$11.00 | | 1 | 1 | | 11 | | |
| | | Buy Div qty 10 @ CPS \$2.00 = \$20.00 | Assets:Investments:Broker1:STOCK1 | n | 1 | 11 | 11.00 | | |
| | | Increase in Residual \$20.00 - \$11.00 = \$9.00 | Assets:Investments:DRP Residual:STOCK1 | n | | 9.00 | | | |
| | | | Income:Dividend:Franked:STOCK1 | n | | | 20.00 | | |
| 01/07/2016 | | STOCK1 Div Reinvestment 2 @ \$12.00 = \$24.00 | | 2 | 2 | | 13 | | |
| | | Buy Div qty 11 @ CPS \$2.00 = \$22.00 | Assets:Investments:Broker1:STOCK1 | n | 2 | 12 | 24.00 | | |
| | | Decrease in Residual = (\$9.00 - \$7.00) = \$2.00 | Assets:Investments:DRP Residual:STOCK1 | n | | | 2.00 | | |
| | | | Income:Dividend:Franked:STOCK1 | n | | | 22.00 | | |
| 02/07/2016 | | Sell STOCK1 Last lot Qty 2 @ \$13.00=\$26.00 Brokerage \$1 | | -2 | | 2 | 11 | | |
| | | Net Sale | Assets:Investments:Broker1 | n | | 25.00 | | | |
| | | Sell Cost qty 2 @ \$12.00 = \$24.00 | Assets:Investments:Broker1:STOCK1 | n | -2 | 12.5 | 25.00 | | |
| 02/07/2016 | | Realized Gain/Loss | | 0 | | | 11 | | |
| | | Realized Gain/Loss | Assets:Investments:Broker1:STOCK1 | n | | 1.00 | | | |
| | | Realized Gain/Loss | Orphaned Gains-AUD | n | | 1.00 | | | |
| 23/02/2017 | Num | | | 0.00 | | | | Tot Buy | Tot Sell Balance |
| | | | | n | 1 | | | | |

Shares: 11 STOCK1 Current Value: \$143.00

Enter a description of the transaction

对消单个批次后的科目账簿。

14. 继续“将孤立的收益-CCC更改为损益科目”一节。

9.6.

按如下步骤创建资本利得交易：

1. 打开证券科目账簿。
2. 确保所有之前手动录入的资本损益交易独立于卖出交易。
3. 选择 [功能] → [查看批次] 打开 [科目SSSS中的批次] 窗口，其中SSSS为证券科目。
4. 若使用后进先出方法，按上文介绍的“手动创建并关联批次”一节步骤，为每项卖出交易创建批次，并将卖出交易和构成本次卖出的每项买入交易关联到同一批次。
5. 点击 [对消整个科目] 按钮，用以：
 - 为所有尚未关联到批次的买入交易创建批次，并使用先进先出方法将这些批次与卖出交易分录关联。由于一项交易分录只能关联到一个批次，如果卖出交易需要关联到多个批次，则卖出交易分录会自行拆分为多个子分录。在多个子分录的情况下，[未关联分录] 和 [批次中的分录] 面板都能够列出来自相同交易下的不同分录。
 - 为每个批次创建独立的资本损益交易。
6. 继续“将孤立的收益-CCC更改为损益科目”一节。

9.7.

下面是先进先出对消的示例，不涉及手动创建批次。在此示例中，一次对消了两项卖出交易，不过通常对消会在每次卖出之后进行。一次对消两笔卖出的一个原因是同一天存在多项卖出。

9.28.

| *GuideInvestExample.gnucash - STK2 - GnuCash | | | | | | | | | |
|--|------|---|--------------------------------------|--------|-----|---------|----------|---------|--|
| File Edit View Transaction Actions Business Reports Tools Windows Help | | | | | | | | | |
| Save Close Split | | | | | | | | | |
| Accounts STK2 | | | | | | | | | |
| Date | Num | Description | | Tot Sh | | Tot Buy | Tot Sell | Balance | |
| 01/01/2016 | | Bought 10 @ \$10.00 Brokerage \$1.00 | | 10 | | 10 | | 10 | |
| | Buy | | Assets:Investments:Broker1:STK2 | n | 10 | 10.1 | 101.00 | | |
| | | | Assets:Investments:Broker1 | n | | | 101.00 | | |
| 01/02/2016 | | Div Reinvestment 1 @ \$11.00 | | 1 | | 1 | | 11 | |
| | Buy | Div qty 10 @ CPS \$2.00 = \$20.00 | Assets:Investments:Broker1:STK2 | n | 1 | 11 | 11.00 | | |
| | | Increase in Residual \$20.00 - \$11.00 = \$9.00 | Assets:Investments:DRP Residual:STK2 | n | | | 9.00 | | |
| | | | Income:Dividend:Franked:STK2 | n | | | 20.00 | | |
| 01/07/2016 | | Div Reinvestment 2 @ \$12.00 = \$24.00 | | 2 | | 2 | | 13 | |
| | Buy | Div qty 11 @ CPS \$2.00 = \$22.00 | Assets:Investments:Broker1:STK2 | n | 2 | 12 | 24.00 | | |
| | | Decrease in Residual = (\$9.00 - \$7.00) = \$2.00 | Assets:Investments:DRP Residual:STK2 | n | | | 2.00 | | |
| | | | Income:Dividend:Franked:STK2 | n | | | 22.00 | | |
| 02/07/2016 | | Sell FIFO Qty 11 @ \$13.00=\$143.00 | | -11 | | | 11 | 2 | |
| | | Net Sale: \$143.00 - Commission \$1.00 = \$142.00 | Assets:Investments:Broker1 | n | | | 142.00 | | |
| | Sell | Cost (10@10.10)+(1@11.00) = \$112.00 | Assets:Investments:Broker1:STK2 | n | -11 | 12.9091 | 142.00 | | |
| 01/08/2016 | | Div Reinvestment 1 @ \$14.00 = \$14.00 | | 1 | | 1 | | 3 | |
| | Buy | Div qty 2 @ CPS \$4.00 = \$8.00 | Assets:Investments:Broker1:STK2 | n | 1 | 14 | 14.00 | | |
| | | Decrease in Residual = (\$7.00 - \$1.00) = \$6.00 | Assets:Investments:DRP Residual:STK2 | n | | | 6.00 | | |
| | | | Income:Dividend:Franked:STK2 | n | | | 8.00 | | |
| 02/08/2016 | | Sell FIFO Qty 3 @ \$15.00=\$45.00 | | -3 | | | 3 | 0 | |
| | | Net Sale: \$45.00 - Commission \$1.00 = \$44.00 | Assets:Investments:Broker1 | n | | | 44.00 | | |
| | Sell | Cost (2@12.00)+(1@14.00) = \$38.00 | Assets:Investments:Broker1:STK2 | n | -3 | 14.6667 | 44.00 | | |
| 28/02/2017 | Num | | | 0.00 | | | | | |
| | | | | | | 1 | | | |

执行〔对消整个科目〕前，以交易日志视图显示的证券科目账簿。

1. 选择 〔功能〕→〔查看批次〕 打开〔科目SSSS中的批次〕窗口，其中SSSS为证券科目。

9.29.

The screenshot shows the 'Lots in Account STK2' window. It has a 'Title' field, a 'Notes' field, and a 'Show only open lots' checkbox. The main table displays transaction history with columns: Date, Num, Description, Amount, Value, Gain/Loss, and Balance. The 'Splits free' section shows a list of transactions, and the 'Splits in lot' section is empty.

| Date | Num | Description | Amount | Value | Gain/Loss | Balance |
|------------|-----|--|----------|----------|-----------|---------|
| 01/01/2016 | | Bought 10 @ \$10.00 Brokerage \$1.00 | 10 STK2 | \$101.00 | | 10 STK2 |
| 01/02/2016 | | Div Reinvestment 1 @ \$11.00 | 1 STK2 | -\$11.00 | | 11 STK2 |
| 01/07/2016 | | Div Reinvestment 2 @ \$12.00 = \$24.00 | 2 STK2 | -\$24.00 | | 13 STK2 |
| 02/07/2016 | | Sell FIFO Qty 11 @ \$13.00=\$143.00 | -11 STK2 | \$142.00 | | 2 STK2 |
| 01/08/2016 | | Div Reinvestment 1 @ \$14.00 = \$14.00 | 1 STK2 | -\$14.00 | | 3 STK2 |
| 02/08/2016 | | Sell FIFO Qty 3 @ \$15.00=\$45.00 | -3 STK2 | \$44.00 | | |

执行〔对消整个科目〕前的科目批次窗口。

2. 点击〔对消整个科目〕按钮。

9.30.

The screenshot shows the 'Lots in Account STK2' window after clicking the 'Scrub Account' button. The 'Title' field now contains 'Lot 0'. The 'Splits free' section shows a list of transactions, and the 'Splits in lot' section is empty.

| Date | Num | Description | Amount | Value | Gain/Loss | Balance |
|------------|------------|-------------|--------|---------|-----------|---------|
| 01/01/2016 | 02/07/2016 | Lot 0 | 0 STK2 | \$28.09 | | |
| 01/02/2016 | 02/07/2016 | Lot 1 | 0 STK2 | \$1.91 | | |
| 01/07/2016 | 02/08/2016 | Lot 2 | 0 STK2 | \$5.33 | | |
| 01/08/2016 | 02/08/2016 | Lot 3 | 0 STK2 | \$0.67 | | |

执行〔对消整个科目〕后的科目批次窗口。

在使用〔对消整个科目〕按钮后，只会显示最后一个批次，因此上图中关闭又重新打开了科目批次窗口以便显示出所有批次。

3. 关闭〔科目SSSS中的批次〕窗口，返回证券科目账簿。

9.31.

| *GuideInvestExample.gnucash - STK2 - GnuCash | | | | | | | | | |
|--|------|---|--------------------------------------|--------|-----|---------|----------|---------|--|
| File Edit View Transaction Actions Business Reports Tools Windows Help | | | | | | | | | |
| Save Close Split | | | | | | | | | |
| Accounts STK2 | | | | | | | | | |
| Date | Num | Description | | Tot Sh | | Tot Buy | Tot Sell | Balance | |
| 01/01/2016 | | Bought 10 @ \$10.00 Brokerage \$1.00 | | 10 | | 10 | | 10 | |
| | Buy | | Assets:Investments:Broker1:STK2 | n | 10 | 10.1 | 101.00 | | |
| | | | Assets:Investments:Broker1 | n | | | 101.00 | | |
| 01/02/2016 | | Div Reinvestment 1 @ \$11.00 | | 1 | | 1 | | 11 | |
| | Buy | Div qty 10 @ CPS \$2.00 = \$20.00 | Assets:Investments:Broker1:STK2 | n | 1 | 11 | 11.00 | | |
| | | Increase in Residual = \$20.00 - \$11.00 = \$9.00 | Assets:Investments:DRP Residual:STK2 | n | | | 9.00 | | |
| | | | Income:Dividend:Franked:STK2 | n | | | 20.00 | | |
| 01/07/2016 | | Div Reinvestment 2 @ \$12.00 = \$24.00 | | 2 | | 2 | | 13 | |
| | Buy | Div qty 11 @ CPS \$2.00 = \$22.00 | Assets:Investments:Broker1:STK2 | n | 2 | 12 | 24.00 | | |
| | | Decrease in Residual = (\$9.00 - \$7.00) = \$2.00 | Assets:Investments:DRP Residual:STK2 | n | | | 2.00 | | |
| | | | Income:Dividend:Franked:STK2 | n | | | 22.00 | | |
| 02/07/2016 | | Sell FIFO Qty 11 @ \$13.00=\$143.00 | | -11 | | | 11 | 2 | |
| | | Net Sale: \$143.00 - Commission \$1.00 = \$142.00 | Assets:Investments:Broker1 | n | | | 142.00 | | |
| | Sell | Cost (10@10.10)+(1@11.00) = \$112.00 | Assets:Investments:Broker1:STK2 | n | -10 | 12.909 | 129.09 | | |
| | Sell | Cost (10@10.10)+(1@11.00) = \$112.00 | Assets:Investments:Broker1:STK2 | n | -1 | 12.91 | 12.91 | | |
| 02/07/2016 | | Realized Gain/Loss | | 0 | | | | 2 | |
| | | Realized Gain/Loss | Assets:Investments:Broker1:STK2 | n | | | 1.91 | | |
| | | Realized Gain/Loss | Orphaned Gains-AUD | n | | | 1.91 | | |
| 02/07/2016 | | Realized Gain/Loss | | 0 | | | | 2 | |
| | | Realized Gain/Loss | Assets:Investments:Broker1:STK2 | n | | | 28.09 | | |
| | | Realized Gain/Loss | Orphaned Gains-AUD | n | | | 28.09 | | |
| 01/08/2016 | | Div Reinvestment 1 @ \$14.00 = \$14.00 | | 1 | | 1 | | 3 | |
| | Buy | Div qty 2 @ CPS \$4.00 = \$8.00 | Assets:Investments:Broker1:STK2 | n | 1 | 14 | 14.00 | | |
| | | Decrease in Residual = (\$7.00 - \$1.00) = \$6.00 | Assets:Investments:DRP Residual:STK2 | n | | | 6.00 | | |
| | | | Income:Dividend:Franked:STK2 | n | | | 8.00 | | |
| 02/08/2016 | | Sell FIFO Qty 3 @ \$15.00=\$45.00 | | -3 | | | 3 | 0 | |
| | | Net Sale: \$45.00 - Commission \$1.00 = \$44.00 | Assets:Investments:Broker1 | n | | | 44.00 | | |
| | Sell | Cost (2@12.00)+(1@14.00) = \$38.00 | Assets:Investments:Broker1:STK2 | n | -2 | 14.665 | 29.33 | | |
| | Sell | Cost (2@12.00)+(1@14.00) = \$38.00 | Assets:Investments:Broker1:STK2 | n | -1 | 14.67 | 14.67 | | |
| 02/08/2016 | | Realized Gain/Loss | | 0 | | | | 0 | |
| | | Realized Gain/Loss | Assets:Investments:Broker1:STK2 | n | | | 5.33 | | |
| | | Realized Gain/Loss | Orphaned Gains-AUD | n | | | 5.33 | | |
| 02/08/2016 | | Realized Gain/Loss | | 0 | | | | 0 | |
| | | Realized Gain/Loss | Assets:Investments:Broker1:STK2 | n | | | 0.67 | | |
| | | Realized Gain/Loss | Orphaned Gains-AUD | n | | | 0.67 | | |
| 28/02/2017 | Num | | | 0.00 | | Tot Buy | Tot Sell | Balance | |
| | | | | n | | 1 | | | |

Shares: 0 STK2 Current Value: \$0.00

Enter a description of the transaction

〔对消整个科目〕后的证券科目账簿。

卖出交易下的证券交易分录已经被拆分为数个子分录，一个子分录关联至一个批次。同时，已经为每项卖出交易下的每项证券子分录创建了资本利得交易。

4. 继续“将孤立的收益-CCC更改为损益科目”一节。

9.8.

下面是不涉及手动创建并关联批次的先进先出对消示例，先前已经使用股票分割向导进行了简单的股票分割。在此示例中，以每股10元的价格买入了100股XYZ股票，后来又进行了1换2的

股票分割且没有额外支付费用，因此现在持有200股，每股5元。最后以每股6元的价格卖出所有这200股。

9.32.

| | Action | Memo | Account | R | Shares | Price | Buy | Sell | |
|------------|------------------------------------|------|--------------------------------|---|--------|-------|----------|----------|-----|
| 01/01/2008 | Buy qty 100 @ \$10.00 | | | | 100 | | 100 | | 100 |
| | Buy | | Assets:Investments:Broker1:XYZ | n | 100 | 10 | 1,000 | | |
| | | | Assets:Current Assets:Bank1 | n | | | | 1,000.00 | |
| | | | | | | 1 | | | |
| 01/01/2009 | Stock Split | | | | 100 | | 100 | | 200 |
| | Split | | Assets:Investments:Broker1:XYZ | n | 100 | | | | |
| 01/07/2009 | Sell 200 for \$6.00 ea = \$1200.00 | | | | -200 | | | 200 | 0 |
| | | | Assets:Current Assets:Bank1 | n | | | 1,200.00 | | |
| | Sell | | Assets:Investments:Broker1:XYZ | n | -100 | 6 | | 600.00 | |
| | Sell | | Assets:Investments:Broker1:XYZ | n | -100 | 6 | | 600.00 | |
| 01/07/2009 | Realized Gain/Loss | | | | 0 | | | | 0 |
| | Realized Gain/Loss | | Assets:Investments:Broker1:XYZ | n | | | 600.00 | | |
| | Realized Gain/Loss | | Orphaned Gains-AUD | n | | | | 600.00 | |
| 01/07/2009 | Realized Gain/Loss | | | | 0 | | | | 0 |
| | Realized Gain/Loss | | Orphaned Gains-AUD | n | | | 400.00 | | |
| | Realized Gain/Loss | | Assets:Investments:Broker1:XYZ | n | | | | 400.00 | |

Shares: 0 XYZ Current Value: \$0.00

执行〔对消整个科目〕后，以交易日志视图显示的证券科目账簿。

9.33.

0

| Lots in This Account | | | | | |
|----------------------|------------|------------|-------------|---------|-----------|
| Type | Opened | Closed | Title | Balance | Gains |
| | 01/01/2008 | 01/07/2009 | Lot 0 0 XYZ | | -\$400.00 |
| | 01/01/2009 | 01/07/2009 | Lot 1 0 XYZ | | \$600.00 |

☐ Show only open lots

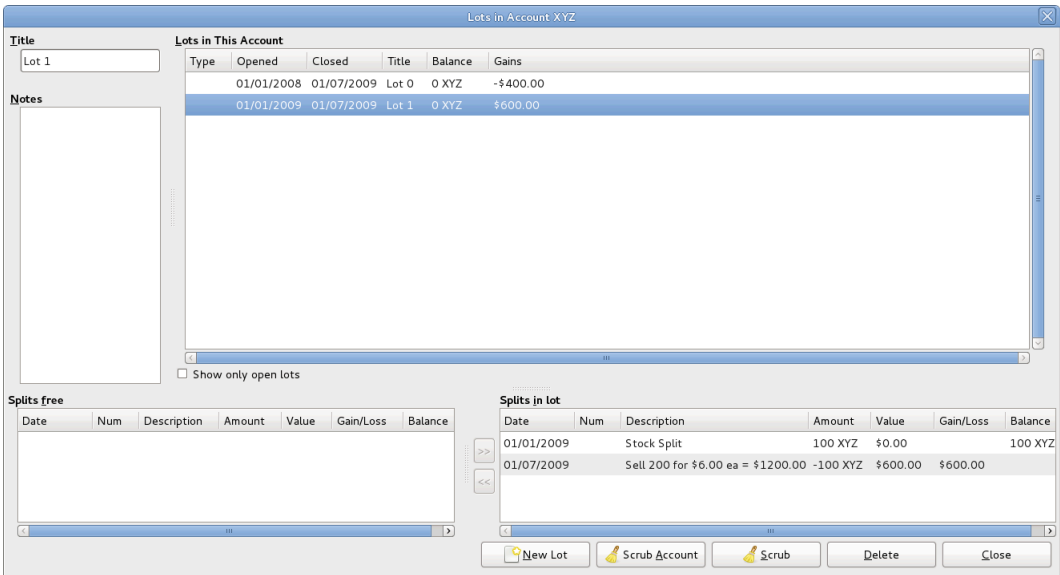
| Splits free | | | | | | |
|-------------|-----|-------------|--------|-------|-----------|---------|
| Date | Num | Description | Amount | Value | Gain/Loss | Balance |
| | | | | | | |

| Splits in lot | | | | | | |
|---------------|-----|------------------------------------|----------|------------|-----------|---------|
| Date | Num | Description | Amount | Value | Gain/Loss | Balance |
| 01/01/2008 | | Buy qty 100 @ \$10.00 | 100 XYZ | \$1,000.00 | | 100 XYZ |
| 01/07/2009 | | Sell 200 for \$6.00 ea = \$1200.00 | -100 XYZ | \$600.00 | -\$400.00 | |

Buttons: New Lot, Scrub Account, Scrub, Delete, Close

对消整个科目后批次0的详细信息。

9.34. - - 1



对消整个科目后批次1的详细信息。

上面的截图表明对消创建了：

- 两项批次。每笔买入各创建了一个独立的批次。本质是将股票分割视为零花费买入100股的分录。
- 两项资本损益交易（每个批次一项交易），日期为卖出交易的日期：
 - 批次0：2009年7月1日，亏损400元（卖出价600元 – 成本价1000元）
 - 批次1：2009年7月1日，盈利600元（卖出价600元 – 成本价0元）

总盈利200元是正确的。这笔收益是全部归类于长期资本利得，还是分两部分分别归类于长期资本利得和短期资本利得，以及两者是否需要区别，都要根据当地的税法和分录的组织结构确定。若计算起来遇到困难，请咨询专业人士。

9.9. -CCC

1. 关闭打开的〔科目SSSS中的批次〕窗口，返回证券科目账簿。

9.35. - -

| Date | Num | Description | | Tot Shares | Tot Buy | Tot Sell | Balance |
|------------|-----|--|--|------------|---------|----------|---------|
| 01/01/2016 | 1 | Bought STOCK1 10 @ \$10.00 Brokerage \$1.00 | | 10 | 10 | | 10 |
| | | Buy | Assets:Investments:Broker1:STOCK1 n | 10 | 101.00 | | |
| | | | Assets:Investments:Broker1 n | | | 101.00 | |
| 01/02/2016 | | STOCK1 Div Reinvestment 1 @ \$11.00 | | 1 | 1 | | 11 |
| | | Buy Div qty 10 @ CPS \$2.00 = \$20.00 | Assets:Investments:Broker1:STOCK1 n | 1 | 11.00 | | |
| | | Increase in Residual \$20.00 - \$11.00 = \$9.00 | Assets:Investments:DRP Residual:STOCK1 n | | 9.00 | | |
| | | | Income:Dividend:Franked:STOCK1 n | | | 20.00 | |
| 01/07/2016 | | STOCK1 Div Reinvestment 2 @ \$12.00 = \$24.00 | | 2 | 2 | | 13 |
| | | Buy Div qty 11 @ CPS \$2.00 = \$22.00 | Assets:Investments:Broker1:STOCK1 n | 2 | 24.00 | | |
| | | Decrease in Residual = (\$9.00 - \$7.00) = \$2.00 | Assets:Investments:DRP Residual:STOCK1 n | | 2.00 | | |
| | | | Income:Dividend:Franked:STOCK1 n | | | 22.00 | |
| 02/07/2016 | | Sell STOCK1 Last lot Qty 2 @ \$13.00=\$26.00 Brokerage \$1 | | -2 | | 2 | 11 |
| | | Net Sale | Assets:Investments:Broker1 n | | 25.00 | | |
| | | Sell Cost qty 2 @ \$12.00 = \$24.00 | Assets:Investments:Broker1:STOCK1 n | -2 | 12.5 | 25.00 | |
| 02/07/2016 | | Realized Gain/Loss | | 0 | | | 11 |
| | | Realized Gain/Loss | Assets:Investments:Broker1:STOCK1 n | | 1.00 | | |
| | | Realized Gain/Loss | Orphaned Gains-AUD n | | | 1.00 | |
| 23/02/2017 | Num | | | 0.00 | Tot Buy | Tot Sell | Balance |
| | | | n | | 1 | | |

Shares: 11 STOCK1 Current Value: \$143.00

Enter a description of the transaction

对消单个批次后的证券科目账簿。

2. 找到证券科目账簿中新创建的每项已实现损益交易，这些交易的日期与卖出交易的日期相同。
3. 编辑孤立的收益-CCC交易分录，将收入科目重新指定为命名更有意义的用于损益的收入或费用科目，如 收入:长期资本利得:浪潮信息。

可以根据当地税法将资本利得拆分为应税和免税两部分。

分红

一些企业或共同基金定期向份额持有人支付红利，即分红。分红一般有两种方式：要么红利自动再投资到份额中，即红利再投；要么以现金支付，即现金分红。共同基金往往实行红利再投，而普通股通常实行现金分红。

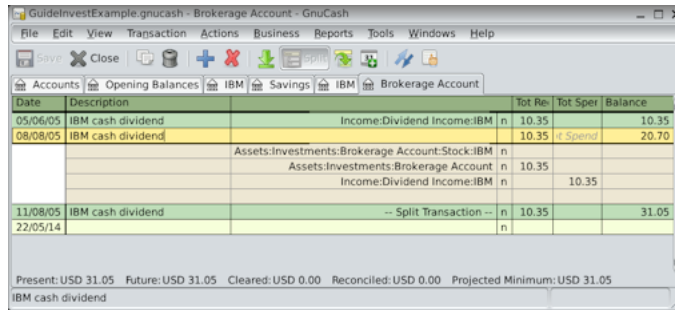
现金分红

如果分红是以现金形式支付的，你应当在收到分红款的资产科目中录入交易，收入来自收入:分红科目。如果你想将现金红利关联到持有的某一特定股票上，则向股票科目添加一条数量为0、单价为1、总价为0的伪交易分录。

如果想跟踪每支股票的分红，则需要为每支持有的支付分红的股票创建收入：分红：股票名称科目。

思考下述示例，其中红利以现金付款到经纪科目，并将分红交易关联到股票科目。

9.36.



| Date | Description | | Tot Re | Tot Sper | Balance |
|----------|-------------------|--|--------|----------|---------|
| 05/06/05 | IBM cash dividend | Income:Dividend Income:IBM | n | 10.35 | 10.35 |
| 08/08/05 | IBM cash dividend | | 10.35 | 4 Spend | 20.70 |
| | | Assets:Investments:Brokerage Account:Stock:IBM | n | 10.35 | |
| | | Assets:Investments:Brokerage Account | n | 10.35 | |
| | | Income:Dividend Income:IBM | n | 10.35 | |
| 11/08/05 | IBM cash dividend | -- Split Transaction -- | n | 10.35 | 31.05 |
| 22/05/14 | | | n | | |

Present: USD 31.05 Future: USD 31.05 Cleared: USD 0.00 Reconciled: USD 0.00 Projected Minimum: USD 31.05
IBM cash dividend

此图展示了一系列现金分红后的经纪科目账簿。

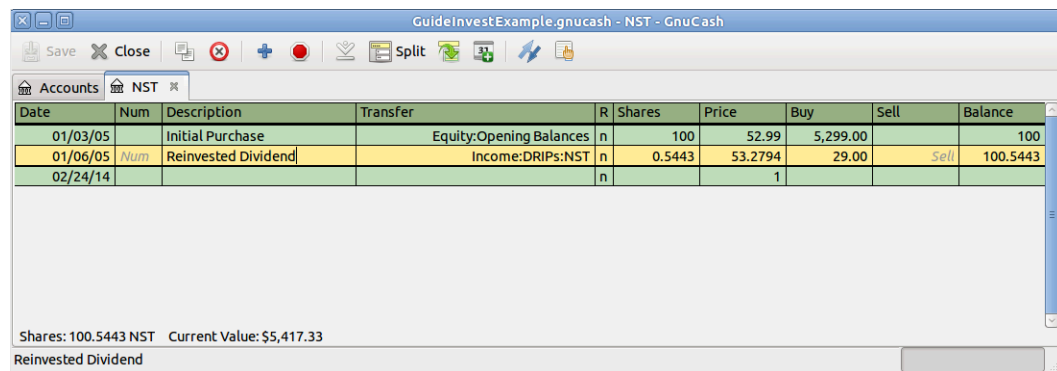
红利再投

如果以自动再投资的形式进行分红，则该分红对应的交易应当在股票或共同基金科目中录入，收入来自收入：分红科目，再投份额设为合适的数量。这类再投资科目常称为红利再投资计划。

思考下述通过红利再投买入重庆燃气股票的示例。共同基金的红利再投与之同理。

最初，在2005年1月3日买入100股，全部分红都会再投资，同时创建一个科目跟踪对特定股票的红利再投。GnuCash能够在交易行内执行计算，可以简化录入过程。若首次分红为每股0.29元，则在份额〔单价〕列输入**53.28**（买入单价+分红），在〔买入〕列输入**100*0.29**。GnuCash会自动计算相应的〔份额〕数量。

9.37.



| Date | Num | Description | Transfer | R | Shares | Price | Buy | Sell | Balance |
|----------|-----|---------------------|-------------------------|---|--------|---------|----------|------|----------|
| 01/03/05 | | Initial Purchase | Equity:Opening Balances | n | 100 | 52.99 | 5,299.00 | | 100 |
| 01/06/05 | Num | Reinvested Dividend | Income:DRIPs:NST | n | 0.5443 | 53.2794 | 29.00 | Sell | 100.5443 |
| 02/24/14 | | | | n | | | 1 | | |

Shares: 100.5443 NST Current Value: \$5,417.33
Reinvested Dividend

红利再投后的重庆燃气科目账簿。

资本退回

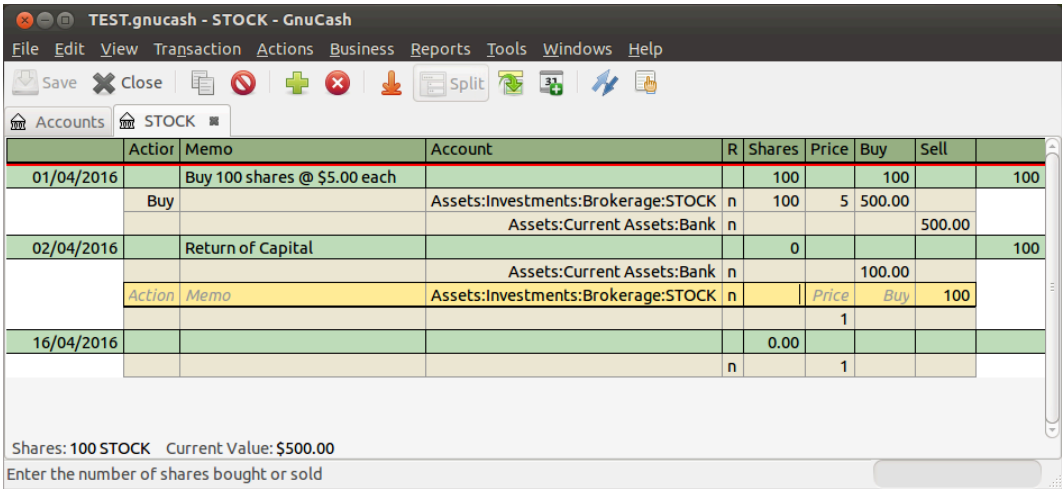
指一种投资标的向投资人退回资本金，同时除降低成本价外无任何会计事务影响的交易，前后持有的份额数量不会改变。

在股票科目账簿中，可以通过如下方式录入股票科目交易记录资本退回：

- 份额：0
- 单价：0
- 卖出：资本退回金额

复式记账的对侧通常对经纪科目记借。

9.38.



The screenshot shows the GnuCash interface with the STOCK window open. The transaction table contains the following entries:

| Date | Action | Memo | Account | R | Shares | Price | Buy | Sell | |
|------------|--------|------------------------------|------------------------------------|---|--------|-------|--------|--------|-----|
| 01/04/2016 | | Buy 100 shares @ \$5.00 each | | | 100 | | 100 | | 100 |
| | Buy | | Assets:Investments:Brokerage:STOCK | n | 100 | 5 | 500.00 | | |
| | | | Assets:Current Assets:Bank | n | | | | 500.00 | |
| 02/04/2016 | | Return of Capital | | | 0 | | | | 100 |
| | | | Assets:Current Assets:Bank | n | | | 100.00 | | |
| | Action | Memo | Assets:Investments:Brokerage:STOCK | n | | Price | Buy | 100 | |
| | | | | | | | | | |
| 16/04/2016 | | | | | 0.00 | | | | |
| | | | | n | | 1 | | | |

Shares: 100 STOCK Current Value: \$500.00
Enter the number of shares bought or sold

录入资本退回交易后的股票科目账簿。

无法使用股票分割向导录入此类交易。

如果你不小心在股票交易分录中输入了非零单价，GnuCash可能会创建多余的价格数据库记录，并导致报表显示错误。使用 [工具] → [价格数据库] 在价格数据库中核对并移除多余的记录。

拆股与合股

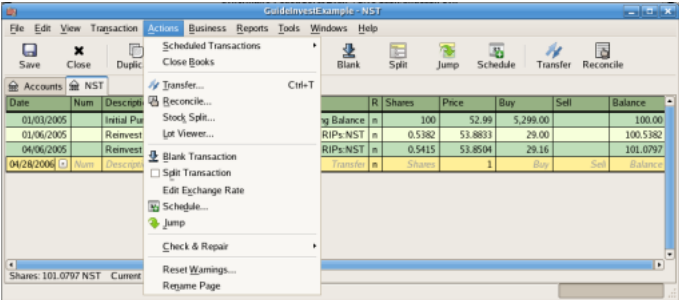
企业可能会出于各种各样的理由实行股票拆分，即拆股，但最常见的原因还是股票价格已经上涨得比管理部门认为对许多投资者合理的价格还要高。一些拆股是简单的置换，如，1换2、2

换3。但还有一些情况由于涉及现金分配而更加复杂。如果置换比例反转，股票拆分也会使持有的份额减少，即合股，如，3换1、1换0.75。

简单的拆股

作为示例，持有的重庆燃气股票发出公告在2005年6月6日实行1换2的拆股。录入此交易的过程如下。首先，选择〔功能〕→〔股票拆分〕启动向导。

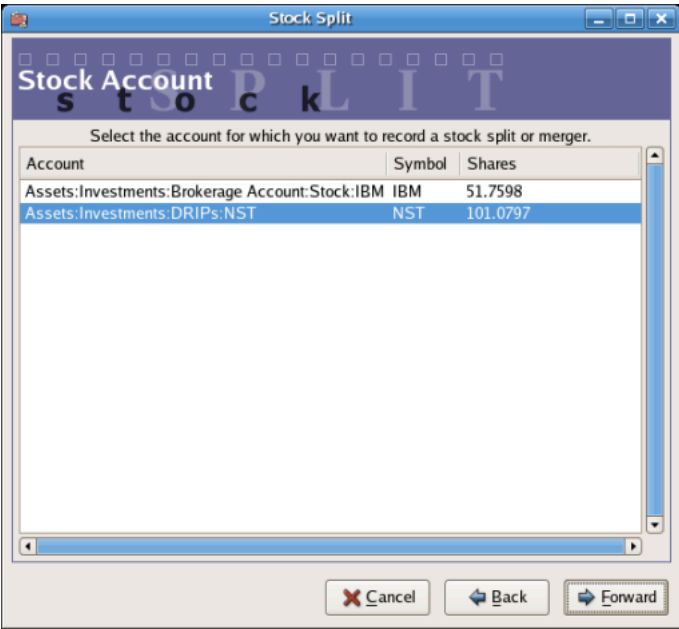
9.39. “ ”



选择股票拆分向导。

第一页是介绍。点击前进浏览并选择要拆分的科目和股票。需要为拆股前后的每对科目：股票创建条目。

9.40. “ ”



股票拆分向导第二步——选择科目与股票。

选择资产：投资：红利再投：重庆燃气并点击 前进。

下一页的股票拆分详情窗口给出了五个输入框：

日期

拆股发生的日期。

份额

拆股交易中增加（或减少）的份额数量。示例中为1换2，所以增加的份额数量为账簿中当前的份额数量。

描述

给出对交易的简要解释。

拆股后价格

如果需要，可以填入拆分后股票的最新价格。

货币

必须选择交易使用的货币单位。应当与买入股票时使用的的货币单位相同。

点击前进按钮。

9.41. “ ”

Stock Split

Stock Split Details

S T O C K S P L I T

Enter the date and the number of shares you gained or lost from the stock split or merger. For stock mergers (negative splits) use a negative value for the share distribution. You can also enter a description of the transaction, or accept the default one.

Date: 06/06/2005

Shares: 101.0797

Description: NST 2:1 Stock Split

If you want to record a stock price for the split, enter it below. You may safely leave it blank.

New Price:

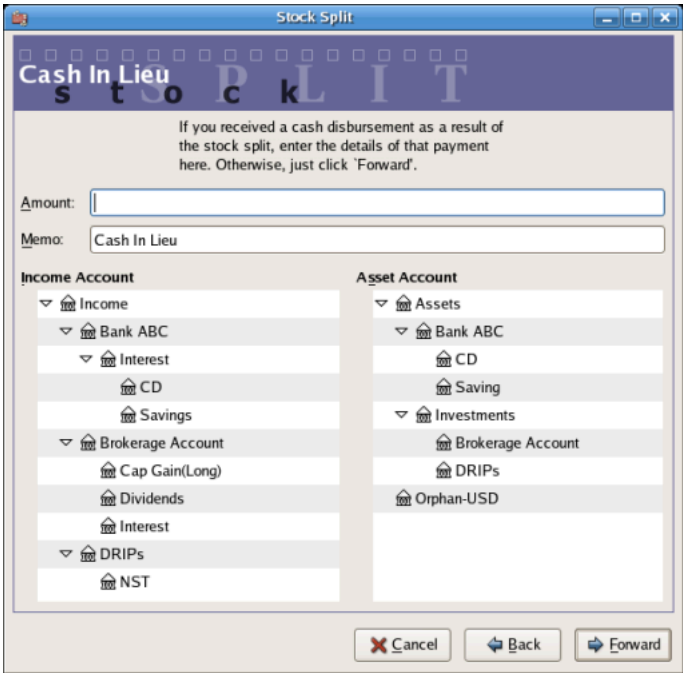
Currency: USD (US Dollar)

Cancel Back Forward

股票拆分向导第三步——拆分详情。

此示例中，下一页会被跳过，因为不存在“现金代付”。

9.42. " " " "



股票拆分向导第四步——现金代付。

最后，〔完成〕页面提供取消、后退（修改录入的任何数据）和应用（按录入的数据完成拆股）三个最终选项。

9.43.

简单拆股交易后的资产:投资:红利再投:重庆燃气科目账簿。

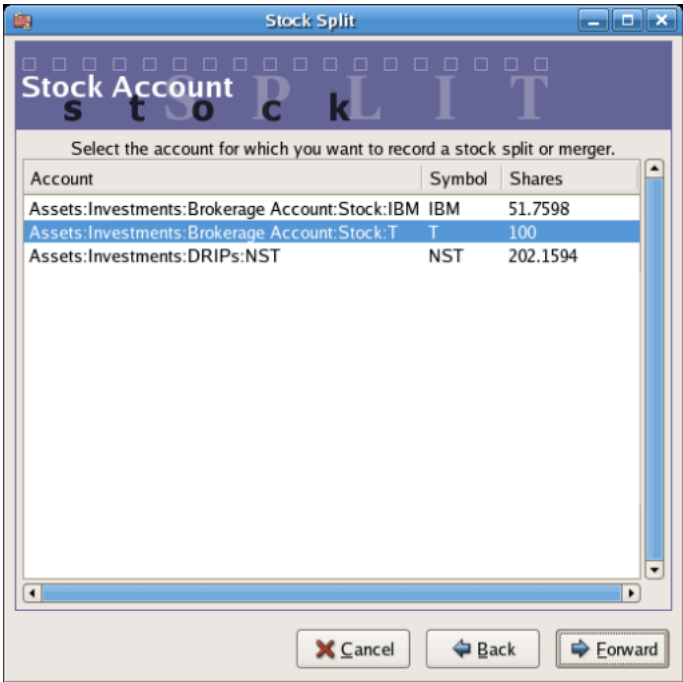
较为复杂的合股

此示例中，假定在2015年5月20日中国南车对中国北车并购案期间持有中国北车股票。你已于2015年4月1日买入了中国北车股票，后续分红均以现金支付，因此没有影响中国北车的股票账簿。

合并条件为1.1股的中国北车股票换取1股中国南车股票。合并后公司的股票名称变更为中国中车。

录入此交易的过程在详情页面前与简单拆股一致。你需要在每个存在拆股的投资科目:股票组合中创建分录。

9.44. “ ”



股票拆分向导第二步——选择科目与股票（投资：中国北车）。

选择资产：投资：经纪：股票：中国北车并点击 前进。

下一页的股票拆分详情窗口给出了五个输入框：

日期

输入拆股日期。此处输入2015年5月20日。

份额

交易中份额数量的增加（或减少）。此示例中为1.1股换1股，所以份额数量会自账簿当前份额数量减少。你可以借助GnuCash对输入框内容执行计算的能力，直接向输入框填入（100#100*1.1），会自动计算出股票拆分中份额的减少。

描述

描述应当给出对交易的简要解释。

新价格

如果需要，可以填入拆分后股票的新价格。

货币

必须选择交易使用的货币单位。应当与买入股票时的货币单位相同。

点击前进按钮。

9.45. “ ”

Stock Split

Stock Split Details

Enter the date and the number of shares you gained or lost from the stock split or merger. For stock mergers (negative splits) use a negative value for the share distribution. You can also enter a description of the transaction, or accept the default one.

Date:

11/18/2005

Shares:

(.77942*100)-100

Description:

SBC / AT&T merger

If you want to record a stock price for the split, enter it below. You may safely leave it blank.

New Price:

24.38

Currency:

USD (US Dollar)

Cancel

Back

Forward

股票拆分向导第三步——拆分详情。

此示例中，下一页会被跳过，因为不存在“现金代付”。

最后，〔完成〕页面提供后退（修改录入的任何数据）和应用（按录入的数据完成拆股）选项。

9.46.

Guidinstyest.sample.split - T

File Edit View Transaction Actions Business Reports Tools Windows Help

Save Close Duplicate Delete Enter Cancel Blank Split Jump Schedule Transfer Reconcile

Accounts Advanced Portfolio T

| Action | Memo | Account | R | Shares | Price | Buy | Sell | |
|------------|-------------------|--------------------------------|---|---------|-------|----------|--------|--------|
| 04/01/2005 | Initial Purchase | Equity Opening Balance | n | 100 | 23.66 | 2,366.00 | | 100.00 |
| 11/18/2005 | SBC / AT&T merger | | n | -22.058 | | | 22.058 | 77.942 |
| | Split Memo | Assets Investments Brokerage A | n | -22.058 | Price | Buy | Sell | |
| | | | | | | 1 | | |
| 04/01/2005 | | | n | | 1 | | | |

Shares: 77.942 T Current Value: \$1,900.23

Assets Investments Brokerage Account Stock T

经减少份额数量的股票拆分交易后的投资：中国北车科目账簿。

183

第 10 章 Reports

GnuCash is a powerful double entry accounting software package that allows users to enter and track their money in a reliable manner. However, putting this information into GnuCash is only a part of the process. To be truly helpful, you need to be able to extract this information in meaningful ways. GnuCash's reporting features allow you to just that.

GnuCash's reporting features allow you to display nearly any group of transactions in a wide variety of formats. This makes it easy to answer questions about your finances, such as “How much did I spend on groceries last month?” or “How much did I earn in the previous six months?”

GnuCash includes a number of common report types, which can be modified to meet your specific needs. If these common reports are insufficient, it is possible to modify or even write your own custom reports (although this is not recommended for beginners).

Overview

There are many standard reports pre-built in GnuCash, all available from the [Reports] pulldown menu in the main account window.

When you select a report from the list of reports, that report is first run with its default settings. Once you have opened the report, you can modify its parameters by clicking on the Options button on the toolbar. Under [Options], you will see the different settings that you can change for each report. Note that for different reports, the options will be different.

Saved Reports Configurations Concepts

Once you have modified a report to meet your needs, you may wish to save that report for reuse at a later point. GnuCash allows custom reports to be saved, using the [Save Report Configuration] command.

10.1. Save a report configuration

1. Go to the [Reports] Menu and choose the desired report.
2. Change the settings on the report until it includes what is needed.
3. Go to the [General] tab of the report's options and change the Report Name to something meaningful (Do not confuse this with the Report Title.)

4. Apply the changes and close the dialog.
5. Click the Save Report Configuration or Save Report Configuration As... button.

This will store the report options in a file in your home directory.

The first time you save a report with a name that has not already been saved, you can use either the Save Report Configuration or the Save Report Configuration As... button. You can modify the report name before saving it.

Once a report has been saved with the current name, the Save Report Configuration button will immediately update the saved report configuration. Use the Save Report Configuration As... button to save the current report configuration with a new name.

Saved report configurations are available for use under the [Reports] → [Saved Report Configurations] entry. They will also be available for use on multicolumn reports.

Saved report configurations can be deleted in the [Saved Report Configurations] dialog by clicking the trashcan icon.

To edit saved report configurations, open the report via [Reports] → [Saved Report Configurations] , edit and apply the new options, and click Save Report Configuration.

Standard Reports Overview

The standard reports that are included in GnuCash are presented in this section. For each report, a short description is given, which explains what the report is intended to show and its primary purpose. We begin with a description of the Transaction Report, which is an all-purpose report with many features and settings that allow sophisticated examination of the data in a file.

Transaction Report

The Transaction Report is a fundamental report that allows users to retrieve a wide variety of useful information from the financial records. The report includes many features that allow powerful and flexible reporting from the single basic report.

- Debit and Credit subtotals can be displayed, and within a grouping section, if debit totals > credit totals, the group subtotal will be displayed in the debit column.
- Multiple data columns enable display of transaction amounts in report currency and original currency in separate columns.
- Additional sortkeys e.g. sort by reconcile status (unreconciled -> cleared -> reconciled), sort by weeks, allow additional views into the financial records.

- Full text filter enabled for description, notes and memo, and for account full names. This allows easier selection of source account names and tailored reporting. Filters can optionally use full POSIX regular expressions.
- Indenting of columns will indent the report columns for easier understanding of grouping and sorting. (This is now enabled by default for GnuCash 3.0 onwards)
- Subtotal table is useful to understand the relationship in the data. This feature compiles the primary and secondary subtotals into a table view, allowing additional insight into financial records, e.g. account subtotals in different time series.

For further ideas on how to use the Transaction Report to produce common reports, see Using the Transaction Report.

Assets & Liabilities Group

Reports in this group provide general information about assets and liabilities.

Advanced Portfolio

The Advanced Portfolio produces a report on commodity accounts (stock and mutual fund type accounts) using price information stored in the GnuCash price database and within the GnuCash transaction data. If you do not have stock price information in your file, the report will indicate an error. This report includes extended information about commodity holdings, including information about the basis, gain, and return of individual commodities.

Advanced Portfolio Capital Gains

The Advanced Portfolio report doesn't use the capital gain splits to calculate capital gains. It calculates the gains from the various buy and sell transactions in the account without regard to whether the gains and losses are recorded or not. Any realized gain splits are ignored. Realized gain splits are recognized as two splits, one in the stock's account with a zero number of shares and a non-zero value, the other in an income or expense account with a value that is the negative of the split in the stock account. These two splits can be in a separate transaction (as created by scrubbing) or in the same transaction as the sale (this will cause incorrect future scrubbing). The income or expense split can be split into multiple splits, say for taxable/non-taxable or short/long term gains, without affecting this report.

Asset Barchart

The Asset Barchart presents the value of assets on a monthly basis in barchart form. By default, the report displays the eight largest accounts that have specific asset types assigned to them, and it displays bars for the current financial period.

Several settings on this report can greatly affect the information included.

- On the Accounts tab, the "Show Accounts until level" option changes how the report aggregates account totals. Change this value to see information at deeper levels of the account structure.
- On the Display tab, the "Maximum Bars" option will display more bars in the chart, allowing information for more accounts to display. Additionally, the "Show table" option enables the display of chart information in tabular form below the chart.
- On the General tab, the "Price Source" option can significantly affect the reported value of various commodities included in the report.

Asset Piechart

The Asset Piechart presents the value of assets on a monthly basis in piechart form. By default, the report shows the seven largest accounts, that have specific asset types assigned to them, arranged in descending order by value as of the end of the current accounting period.

Several settings on this report can greatly affect the information included.

- On the Accounts tab, the "Show Accounts until level" option changes how the report aggregates account totals. Change this value to see information at deeper levels of the account structure.
- On the Display tab, the "Maximum Slices" option will display more slices in the chart, allowing information for more accounts to display.
- On the General tab, the "Price Source" option can significantly affect the reported value of various commodities included in the report.

Average Balance

The Average Balance report displays monthly averages for the current accounting period.

Balance Sheet

The Balance Sheet lists Asset, Liability, and Equity account balances for all such accounts, and provides totals as of a given date. Balance sheets are commonly run for the last day of each fiscal year to give an overall sense of the financial condition of the entity.

General Journal

The General Journal produces a register of all transactions (beginning to end) in order by date, showing the accounts and the amounts involved, and totals the Net Change by all currencies and assets.

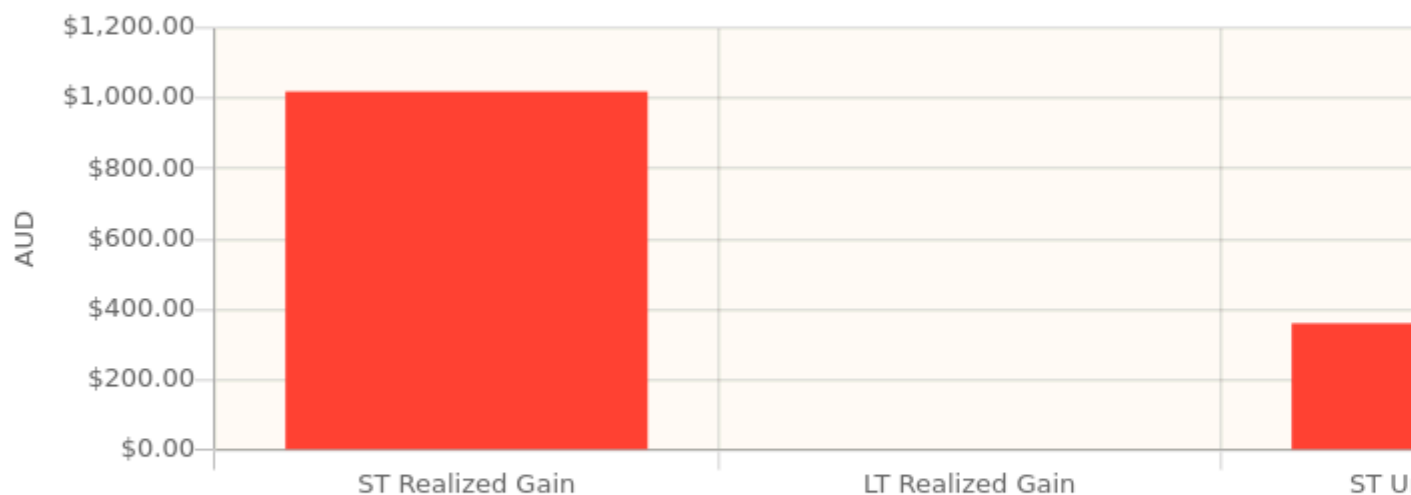
This report is not customizable by date or account, though you can include more or fewer details about the individual transactions, and whether to include running balances and totals for the credits and debits. If you need a report restricted to particular accounts, consider the Transaction Report or open a particular account and choose the Account Transaction Report.

General Ledger

The General Ledger produces information about all transactions for a selected set of accounts. When first run, this report loads no data, and the report options must be changed to retrieve information from the file.

Investment Lots, 03/07/20 to 24/11/22

Account Lot Gain
03/07/20 to 24/11/22



Broker:RIO

End price: \$180.00 on 08/10/22

| Lot Title | Opened | Closed | Bought Amount | Bought Value (Basis) | Bought Average Price | Sold Splits | Sold Amount | So |
|-----------|-----------------|-----------------|---------------|----------------------|----------------------|-------------|-------------|----|
| Lot 1 | <u>01/02/22</u> | <u>04/02/22</u> | 10 RIO | \$1,000.00 | \$100.00 | 2 | 10 RIO | \$ |
| Lot 0 | <u>02/02/22</u> | | 12 RIO | \$1,440.00 | \$120.00 | 1 | 6 RIO | |

Account Lots Total **22 RIO \$2,440.00 \$110.91** **3 16 RIO \$1,120.00**

Warning: 5 split(s) are not assigned to a lot. Do lots need to be scrubbed?

| | Bought Value (Basis) | Sold Splits | So |
|--------------------|----------------------|-------------|-------------------|
| Grand Total | \$2,440.00 | 3 | \$1,120.00 |

The Investment Lots report shows details about lots in a selected set of accounts of type Stock or Mutual Fund (see “使用批次卖出份额并自动计算资本损益” 一节). It includes an optional capital gains summary chart and a table with columns for dates, purchases, sales, end balances, capital gains (realized and unrealized, short term and long term), and return on investment (ROI). It also performs optional lot validation checks.

The report can help answer questions such as:

- Which lots are best candidates for tax loss/gain harvesting?
- What capital gains have been realized for a particular time period (i.e. this year, last year, etc)? Are they long term or short term gains?
- Which lots held longer than 1+ years have the largest unrealized gains, and would therefore be most advantageous (tax-wise) if donated to charity?
- Are the lots (and related accounts and splits) well-formed?

For more details, see the Investment Lots [<https://www.gnucash.org/docs/v5/C/gnucash-manual/report-classes.html#investment-lots-report>] report documentation in the Manual.

Investment Portfolio

The Investment Portfolio produces a report of commodity accounts (that is, accounts with type "Stock" or "Mutual Fund"), giving holdings, price and value information about commodities in the file.

Liability Barchart

The Liability Barchart presents the value of liabilities on a monthly basis in barchart form. By default, the report displays the eight largest accounts that have specific asset types assigned to them, and it displays bars for the current financial period.

Several settings on this report can greatly affect the information included.

- On the Accounts tab, the "Show Accounts until level" option changes how the report aggregates account totals. Change this value to see information at deeper levels of the account structure.
- On the Display tab, the "Maximum Bars" option will display more bars in the chart, allowing information for more accounts to display. Additionally, the "Show table" option enables the display of chart information in tabular form below the chart.
- On the General tab, the "Price Source" option can significantly affect the reported value of various commodities included in the report.

Liability Piechart

The Liability Piechart presents the value of liabilities on a monthly basis in piechart form. By default, the report shows the seven largest accounts, that have specific asset types assigned to them, arranged in descending order by value as of the end of the current accounting period.

Several settings on this report can greatly affect the information included.

- On the Accounts tab, the "Show Accounts until level" option changes how the report aggregates account totals. Change this value to see information at deeper levels of the account structure.
- On the Display tab, the "Maximum Slices" option will display more slices in the chart, allowing information for more accounts to display.
- On the General tab, the "Price Source" option can significantly affect the reported value of various commodities included in the report.

Net Worth Barchart

The Net Worth Barchart summarizes Asset accounts, Liability accounts, and overall Net Worth as bars on a graph on a monthly basis for the current financial period. This report provides a graphic overview of the file over time.

Net Worth Linechart

The Net Worth Linechart summarizes Asset accounts, Liability accounts, and overall Net Worth as a line graph on a monthly basis for the current financial period. This report provides a graphic overview of the file over time.

Price Scatterplot

The Price Scatterplot displays the value of one commodity relative to another commodity, for example the value of a stock relative to a currency. When first run, this report loads no data, and the report options must be changed to retrieve information from the file. Specifically, the "Price of Commodity" setting on the Price options tab must be changed to a specific commodity.

Budget Group

Budget reports in GnuCash allow you to gather summary information related to budgets you may have created. In order for these reports to work, you must first create a budget. The reports in this group are specifically based on budget information. To use these reports, you need to have a budget saved in your file.

Budget Balance Sheet

Budget Barchart

Budget Flow

Budget Income Statement

Budget Profit & Loss

Budget Report

Business Group

Reports in this group provide general information about activities related to a business.

Customer Report

Customer Summary

Customer Summary is a customer profit report that can help with job analysis by comparing the income and expenses for a specific customer.

All invoices have an Owner in GnuCash, so invoices that are made will show a customer and show in the report. When creating a Bill, the Default Chargeback Customer is blank. To use the profit report, this field needs an entry, since this is the tag that decides the line to which to attach the expense. Left blank, the bill will be assigned to "No Customer." Similarly, when income is entered directly in a register rather than creating an invoice, it will also be assigned to "No Customer."

Thus, if this report includes an entry for "No Customer", this means that the report may be inaccurate, as the results are not all properly labeled.

Possible use scenarios include:

- Tracking retail sales from different cities
- Tracking rental properties
- Tracking types of business

- Tracking commission sales

Each of these scenarios assumes that the account structure includes breakdowns for individual tracked categories. Changing settings on the Income and Expense tabs under Options can hone the information displayed. By default all income and expense accounts are included; however, since GnuCash can't really predict the names and classification of income and expense accounts, it must group them all into the "No Customer" entry.

Note that inventory-based businesses won't benefit from this report because of its nature.

Useful options:

- The Expense Accounts tab allows the selection of one or more expense accounts.
- The Income Accounts tab allows the selection of one or more income accounts.
- The Display tab allows sorting by name, profit percentage, or amount of profit.

Easy Invoice

Employee Report

Fancy Invoice

Payable Aging

Printable Invoice

Receivable Aging

This report provides a listing of all customers, their current balance, and how much they have outstanding from invoices over different time periods—how much they owe from 0-30 days, from 31-60 days, from 61-90 days, and over 90 days. The report also contains links to each customer and to their current customer report.

Vendor Report

Income and Expense Group

Reports in this group provide information about Income and Expense

Cash Flow

This report shows the change in value for a set of accounts (the flow of cash) over a given period of time. By default, this report is based on accounts in Assets and Special Accounts, and covers the current financial period. The report enumerates all money coming in to and going out of the base accounts, broken down by the other account.

Equity Statement

This report can be seen as extension of the Balance Sheet report. The Balance Sheet states the balance of Assets, Liabilities and Equity at a specific point of time. The Equity Statement focuses on the Equity Accounts by showing the cash flow to and from them for a given period of time.

By balancing this cash flow with income, the report shows the available capital at the beginning and end of the selected time period.

Expense Barchart

The Expense Barchart presents the value of expenses on a monthly basis in barchart form. By default, the report displays the eight largest accounts that have specific expense types assigned to them, and it displays bars for the current financial period.

Several settings on this report can greatly affect the information included.

- On the Accounts tab, the "Show Accounts until level" option changes how the report aggregates account totals. Change this value to see information at deeper levels of the account structure.
- On the Display tab, the "Maximum Bars" option will display more bars in the chart, allowing information for more accounts to display. Additionally, the "Show table" option enables the display of chart information in tabular form below the chart.
- On the General tab, the "Price Source" option can significantly affect the reported value of various commodities included in the report.

Expense Piechart

The Expense Piechart presents the value of expenses on a monthly basis in piechart form. By default, the report shows the seven largest accounts, that have specific expense types assigned to them, arranged in descending order by value as of the end of the current accounting period.

Several settings on this report can greatly affect the information included.

- On the Accounts tab, the "Show Accounts until level" option changes how the report aggregates account totals. Change this value to see information at deeper levels of the account structure.

- On the Display tab, the "Maximum Slices" option will display more slices in the chart, allowing information for more accounts to display.
- On the General tab, the "Price Source" option can significantly affect the reported value of various commodities included in the report.

Expenses vs. Day of Week

Expenses vs. Day of Week presents a pie chart showing the totals for selected expense accounts totaled by the day of the week of the transaction. The report options enable you to make some adjustments (such as accounts, display options, and the date range) but the account selector only allows expense accounts to be chosen. The report aggregates expense transactions by day of week, not by any other period or category. Due to these limitations, the report may be considered a demonstration or an example to someone wanting to examine the source code for composing a useful custom report.

Income Barchart

The Income Barchart presents the value of income on a monthly basis in barchart form. By default, the report displays the eight largest accounts that have specific income types assigned to them, and it displays bars for the current financial period.

Several settings on this report can greatly affect the information included.

- On the Accounts tab, the "Show Accounts until level" option changes how the report aggregates account totals. Change this value to see information at deeper levels of the account structure.
- On the Display tab, the "Maximum Bars" option will display more bars in the chart, allowing information for more accounts to display. Additionally, the "Show table" option enables the display of chart information in tabular form below the chart.
- On the General tab, the "Price Source" option can significantly affect the reported value of various commodities included in the report.

Income & Expense Chart

Income Piechart

The Income Piechart presents the value of income on a monthly basis in piechart form. By default, the report shows the seven largest accounts, that have specific income types assigned to them, arranged in descending order by value as of the end of the current accounting period.

Several settings on this report can greatly affect the information included.

- On the Accounts tab, the "Show Accounts until level" option changes how the report aggregates account totals. Change this value to see information at deeper levels of the account structure.
- On the Display tab, the "Maximum Slices" option will display more slices in the chart, allowing information for more accounts to display.
- On the General tab, the "Price Source" option can significantly affect the reported value of various commodities included in the report.

Income Statement

This report lists Income and Expense account totals for a set period. By default, it shows all Expense and Income accounts down to 3 levels of sub-accounts for the current financial period.

An Income Statement is also called a "Profit and Loss" report or "Revenue Statement."

In earlier versions of GnuCash, this report was called "Profit & Loss," but with version 2, the report was renamed "Income Statement" to use more common accounting terminology.

The Income Statement helps show where money is coming from and where it is going for a given time period.

Income vs. Day of Week

Income vs. Day of Week presents a piechart showing the totals for selected income accounts totaled by the day of the week of the transaction. The report options enable you to make some adjustments (such as accounts, display options, and the date range) but the account selector only allows income accounts to be chosen. The report aggregates income transactions by day of week, not by any other period or category. Due to these limitations, the report may be considered a demonstration or an example to someone wanting to examine the source code for composing a useful custom report.

Trial Balance

Trial Balance lists the ending balances in all accounts as of a particular date. It is typically run at the end of an accounting period and is primarily used to ensure that the total of all debits equals the total of all credits.

Income and GST Statement

The Income and GST Statement is a specialised Transaction Report designed to print Business-related Sales and Receipts, as well as their GST or VAT components. This report is designed for use in Australia, but is also usable in any jurisdictions

implementing value-added taxes (or goods and services taxes) during regular business sales and receipts; and the business owner is expected to submit periodic reports of total sales, purchases, and taxes collected on sales from clients or paid to a supplier.

This report makes some assumptions upon the accounts used for sales, purchases, and taxes collected on sales, and paid on purchases.

- Net Sales must be an Income-type account
- Net Purchases must be an Expense-type account
- GST/VAT on Sales must be a Liability-type account
- GST/VAT on Purchases must be an Asset-type account
- There may be multiple GST accounts, e.g. reduced GST, standard GST etc. The GST accounts can be printed individually, or summarised for sales and purchases.
- There may be multiple sales and expenses accounts, e.g. Income:Sales, Income:Grants, Expenses:Suppliers, Expenses:Marketing, Expenses:Insurance. These amounts may be reported individually, or summarised for sales and purchases.
- The transactions may be entered manually as a multi-split transaction, or they may be entered via the Business Invoices and Bills tools.

As an example, consider a business has the following transactions: It incurs sales of \$1,000 plus 5% GST, receiving \$1,050. It also incurs GST-exempt sales of \$600. It purchases goods/services worth \$400 plus 5% = \$420. The following transactions will record the business activities:

10.1. Sample Chart of Accounts for Income and GST Statement

| Date | Description | Account | Debit | Credit |
|------------|-------------------------|--------------------------|---------|---------|
| 01/01/2018 | Sales \$1,000 + 5% GST | Income:Sales | | \$1,000 |
| | | GST on Sales [Liability] | | \$50 |
| | | Asset:Bank | \$1,050 | |
| 02/01/2018 | GST-Free Sales \$600 | Income:Sales | | \$600 |
| | | Asset:Bank | \$600 | |
| 03/01/2018 | Purchase \$400 + 5% GST | Expenses:Purchases | \$400 | |
| | | Asset:Bank | | \$420 |
| | | GST on Purchases [Asset] | \$20 | |

For the Income and GST Statement, the Accounts selected will be the Income and Expense Accounts, and the Tax accounts are used to select the GST accounts. The GST period is selected via the [General] tab. The resulting report is shown as follows:

10.2. Income and GST Statement

| Date | Description | Total Sales | Net Sales | Tax on Sales | Total Purchases | Net Purchases | Tax on Purchases |
|------------|-------------------------|-------------|-----------|--------------|-----------------|---------------|------------------|
| 01/01/2018 | Sales \$1,000 + 5% | \$1,050 | \$1,000 | \$50 | | | |
| 02/01/2018 | GST-Free Sales \$600 | \$600 | \$600 | | | | |
| 03/01/2018 | Purchase \$400 + 5% GST | | | | \$420 | \$400 | \$20 |
| Totals | | \$1,650 | \$1,600 | \$50 | \$420 | \$400 | \$20 |

This report would indicate that there was a total of \$50 GST collected, and \$20 GST on purchases, therefore a \$30 GST liability is payable to the authorities.

Sample & Custom Group

The reports in this group offer examples on how reports can be customized or modified to suit personal need.

Custom Multicolumn Report

This report provides a base that allows several standard and custom reports to be combined into one view. Note that this report opens with an empty window; you must open the options and designate which reports to include for display. Once the reports have been selected, the settings for individual reports in the multicolumn display can be edited.

Sample Report with Examples

This is a sample report that users can examine to learn how to write their own reports.

Welcome Sample Report

This report demonstrates how the Multicolumn Report can be use to create custom dashboard-type reports.

Other Reports

Several reports are included on the main Reports menu. These are described below.

Account Summary

This lists the balances of all accounts and subaccounts as of a particular date. By default, this report shows accounts and totals down to third-level subaccounts.

This report gives effectively the same information as the Chart of Accounts. You can use this report to export and print the Chart of Accounts.

To generate a report of account totals over a particular time period (especially if you do not close your books at regular intervals), you might consider using the Income Statement, or Cash Flow reports.

Balance Forecast

This creates a balance chart which aims to track asset (or any other account type) balances, including amounts from future scheduled transactions. The balances will include all regular booked amounts and future amounts from numeric scheduled transaction templates.

This report can also add additional data charts to forecast future minimum balances. Reserve line can be displayed to confirm if future minimum will dip below the reserve amount. A target line may also be displayed above the reserve line for financial planning.

Future Scheduled Transactions Summary

Tax Report & TXF Export

Generates a report and a downloadable .txf file of taxable income and deductible expenses for a particular accounting period. To download the report data, choose the Export button on the toolbar and choose between html and .txf downloadable versions.

To use this report, you must use Edit --> Tax Options to identify which form the taxing authority uses for each income or expense account. Note that you can see but not modify the "Tax related" checkbox in Edit --> Edit Account.

Saved Report Configurations

Selecting this will open a dialog with a list of available Saved Report Configurations. "Saved Report Configurations" means sets of customized settings for standard reports.

These sets must be saved by the user before they appear here. See Report Concepts above for instructions on how to save report configurations.

Account Report

The Account Report menu entry only appears when an account register is the active tab. This report produces a list of all transactions in the open register.

Note that if you conduct a search that retrieves several transactions, the results are displayed in a new search register, which can then be used to create a report for just those transactions.

Account Transaction Report

This report also only appears when an account register is the active tab. However, this report only lists transactions that have been selected (e.g. by mouse click) in the current register. If no transactions are selected, an empty report will be generated.

Using the Transaction Report

The Transaction Report can be heavily customised to produce most reports appropriate for personal finance and business bookkeeping purposes. It aims to retrieve and filter information from the database, and present transactions and totals, useful for the user and the accountant.

The following guide to the Transaction Report will assume the user has set up the chart of accounts according to conventional bookkeeping practice. For example, the following describes a typical household book with skeleton accounts. Further accounts will undoubtedly be necessary.

10.3. Sample Chart of Accounts for the Transaction Report

| Account Name | Account Type |
|-----------------------|-------------------------|
| Asset | ASSET (placeholder) |
| Asset:Bank | BANK |
| Asset:Property | ASSET |
| Liability | LIABILITY (placeholder) |
| Liability:Credit Card | CREDIT CARD |
| Liability:Home Loan | LIABILITY |
| Income | INCOME (placeholder) |
| Income:Salary | INCOME |
| Income:Interest | INCOME |

| Account Name | Account Type |
|-------------------------|-----------------------|
| Expense | EXPENSE (placeholder) |
| Expense:Groceries | EXPENSE |
| Expense:Auto | EXPENSE |
| Expense:Medical | EXPENSE |
| Equity | EQUITY (placeholder) |
| Equity:Opening Balances | EQUITY |

Conventionally, the oldest transaction would be equity transfers from Equity:Opening Balances to Asset and Liability Accounts. Most subsequent transactions would be transfers from Income/Expense accounts to Assets/Liability accounts (representing day-to-day activity e.g. receiving salary, paying utility bill), or transfers between assets and liability accounts (representing movements between loans and assets, e.g. paying credit card bill, receiving a loan).

The following use cases will be illustrated, and the options to be selected explained:

Using the Transaction Report to show previous year expenses

The Transaction Report can show how much was spent on expense accounts last year. This will usually be the most useful basic transaction report format. To create this report:

1. Open the report options, from the [Accounts] tab, click the Expense placeholder account and Select Children.
2. From the [General] tab, Choose relative dates “Start of Previous Year” and “End of Previous Year” .
3. From the [Sorting] tab, set [Primary Key] to Accounts, and enable [Primary Key Subtotal] .
4. Set the [Secondary Key] to Date, and set the [Secondary Key Subtotal] to None.

Secondary level grouping

The following will modify the above Transaction Report to use a secondary grouping strategy. The first grouping (i.e. Primary Key = Accounts, Subtotal = enabled) will group transactions from the same accounts, whereas the secondary grouping (i.e. Secondary Key = Date, Subtotal = Monthly) will also calculate monthly sums within each account.

1. Select options as above
2. From the [Sorting] tab, set [Secondary Key Subtotal] to Monthly

Using secondary grouping for periodic comparison

The Display / Subtotal Table adds a subtotal summary table, which will ease comparison of accounts across date periods. This displays the same subtotal calculations as the main table, presented in a grid structure.

In addition, if there are multiple primary-key groups (e.g. Date grouping with monthly subtotals, across multiple months) the subtotal table will also compute and display the primary-key grouping Average. The average is computed as the total amount per row, divided by the number of columns. This may cause confusion when computing monthly spend averages, for instance, if a month has no transactions and is not displayed in the table, it will not create a corresponding column in the table, therefore the average amount will not reflect a true monthly average.

1. Select options as above
2. From the [Display] tab, enable the [Subtotal Table] .

Use filtering to limit accounts and transactions

Accounts and transactions can be filtered for reporting according to account full name (e.g. ":Business:" will select account structure Income:Business:Sales, Income:Business:Grants, Expenses:Business:Rent, Expenses:Business:Utilities) or transaction description/notes/memo (select transactions tagged #gift). The filtering text may be either a standard string, or a POSIX regular expression.

1. From the [Filter] tab, complete the [Account Name Filter] , or [Transaction Filter] . Optionally toggle the regular expression check boxes to toggle full POSIX regex matching. Regular expressions will allow more complex queries e.g. the transaction filter can be set to (#gift|#travel) which will filter transactions with either #gift or #travel. They can be complicated and are best learned from external guides.

Using the Transaction Report to generate a reconciliation statement.

This sortkey 'reconcile status' will separate reconciled, cleared and unreconciled transactions. This report may be useful in producing a printable reconciliation report.

Alternatively, the [Reconciliation Report] will preset these defaults and requires only the Accounts selection.

1. From the [Accounts] tab, select the relevant bank or credit card account
2. From the [General] tab, choose an appropriate date range e.g. past 3 months
3. From the [Sorting] tab, set the [Primary Key] to Reconcile Status, [Primary Key Subtotal] to enabled, and set the [Secondary Key] to Date. The [Secondary Date Subtotal] is left to None

第 11 章 资本利得

本章会介绍一些方法，用于跟踪未实现损益和已实现损益，也即众所周知的资本利得。

基本概念

像是股票、债券、房屋或汽车这样的可再次出售的资产可能会随时间发生价值的变动。一些资产的价值可能会增加，而另一些资产的价值可能会减少。能够跟踪这些随时间变化的资产价值是十分重要的，本章会介绍具体做法。

你拥有的一切东西都可能会随时间发生价值的增减。所以问题在于到底要跟踪哪些资产的价值变化。一个简单的回答是你只需跟踪那些将来可能出售以换取现金，或涉及缴税的资产。

显然，消耗品或一次性用品（如食品、汽油或打印纸）无需跟踪其价值变化。因此，尽管你刚买的新衣服确实会折旧，你也不应该跟踪衣服的折旧，因为你不会想着重新出售这件衣服，以后也不会为这件衣服缴税。所以，对这个例子而言，买新衣服应该记录为纯粹的费用——你花出去了一些钱，于是一切就结束了。

资产增值指资产价值随时间增加。当你持有的资产价值增加时，初始购买价格与当前价值间的差额就是所说的未实现损益。当你出售资产时，你赚取的利润就是所谓的已实现损益或资本利得。有很多可以产生未实现损益并最终实现资本利得的资产，例如上市公司的股票。

估值

正如前文简介提到的那样，资本利得是由出售资产得到的利润。下面这一节会介绍怎样在 GnuCash 中记录资本利得。

处理资产增值会计方法与折旧稍有不同，因为一般你只关心卖出资产并实现资本利得的时刻。资本利得是税务领域中的重要主题，因为政府往往对使用各种方法就资本利得收税特别感兴趣。

事情总会有例外情况。如果你持有到期一次性结息的债券，税务部门常常会要求将每年的利息划分清楚，并拒绝将之视为资本利得。研究清楚适合的税务代码，以便为每种你拥有并可能涉及资本利得税的资产确定最合适的应对方法。

评估资产的价值增加通常并不简单，因为直到售出那一刻之前都难以知晓资产的确切价值。

在股票交易所等公开市场每日交易的证券或许是最容易预估其价值的一类资产，以市场价出售这类资产可能简单到只需要给经纪人打电话下单。另一方面，估计房屋的价值更加困难。房屋的出售量常常比股票少，出售时往往还要花很多时间和精力进行协商，这意味着对价值的估计不会很准确。类似地，出售二手汽车也涉及协商的过程，同样会使价格不易预测。

估计珠宝、艺术品、棒球卡等收藏品的价值更是难上加难。收藏品市场比证券市场封闭得多，与房屋市场相比也更不透明。更糟糕的是那些独一无二的资产。常常工厂里会有为生产特定产品而花几十、几百万元定制的挤压设备或模具，这类设备可能离开某个特定的场景就毫无价值了。这种情况下，资产可能会被评估出几个相互冲突的估值，其中没有哪个是清晰准确的。

估计未实现损益的一般性经验法则是你只应该为股票这样买卖方便、估值准确的资产重新估值。对于所有其它资产，最好等到出售的时候再具体确定资本利得。当然，这方面也没什么硬性准则，实际上不同会计师的处理习惯也不尽相同。

建立科目

下面会创建未实现损益科目，为了便于理解未实现损益的概念，会在这个科目中录入估计的未实现损益。然而，如果不是在实际会计事务中采用国际财务报告准则的大企业，那么记录未实现损益不是必要的。使用GnuCash的小型企业用户不应该记录未实现损益，只要在确定已实现损益之后再记录就足够了。

与多数会计实操一样，建立资本利得科目也有很多不同的方式。这里只会介绍常用的方法，但多数情况下也足够灵活了。需要的第一个科目是名为成本的资产类科目（GnuCash科目类型为〔资产〕），这个科目简单地用来记录最初的资产购入。资产购入通常记录为从银行科目转出资金的交易。

跟踪资产增值还需要另外三个科目。第一个科目是未实现损益资产类科目，用于累计增值金额。未实现损益资产科目借助未实现损益收入科目平衡账目，用于定期记录所有资产增值收入。最后还有一个收入科目是必需的，称为已实现损益，用于记录出售资产时实际的资本利得。

下面是一个用于跟踪物品甲和物品乙两项资产的通用科目结构。资产:固定资产:物品甲:成本科目借助资产:流动资产:储蓄存款科目实现平衡；资产:固定资产:物品甲:未实现损益科目借助收入:未实现损益科目实现平衡。物品乙与之同理。

```
-资产
  -流动资产
    -储蓄存款
  -固定资产
    -物品甲
      -成本
      -未实现损益
    -物品乙
      -成本
      -未实现损益
-收入
  -已实现损益
  -未实现损益
```

示例

假定你购入了一项资产，比如德加的画作，希望它能够增值，同时想跟踪其价值变化。（保险公司很关心这个事情，即使其它人都不会在意。）

首先仿照“建立科目”一节建立科目结构，只不过将物品甲换成德加并删去物品乙科目。假定德加的画作初始价值为十万元。一开始你的银行账户中有100,000元，然后将这笔钱从银行账户转到资产：固定资产：德加：成本科目，即资产购入交易。现在主科目窗口应该如下所示：

11.1.

| 科目名称 | 类型 | 商品 | 合计 |
|---------|-------|------|---------------|
| ▼ 收入 | 收入 | 人民币元 | CN¥0.00 |
| 未实现损益 | 收入 | 人民币元 | CN¥0.00 |
| 已实现损益 | 收入 | 人民币元 | CN¥0.00 |
| ▶ 所有者权益 | 所有者权益 | 人民币元 | CN¥100,000.00 |
| ▼ 资产 | 资产 | 人民币元 | CN¥100,000.00 |
| ▼ 固定资产 | 资产 | 人民币元 | CN¥100,000.00 |
| ▼ 德加 | 资产 | 人民币元 | CN¥100,000.00 |
| 成本 | 资产 | 人民币元 | CN¥100,000.00 |
| 未实现损益 | 资产 | 人民币元 | CN¥0.00 |
| ▼ 流动资产 | 资产 | 人民币元 | CN¥0.00 |
| 储蓄存款 | 资产 | 人民币元 | CN¥0.00 |

主窗口的初始显示。

未实现损益

过了一个月，你有理由认为画作的价值增长了10,000元（未实现损益）。为了记录增值，从收入：未实现损益科目转出10,000元到资产：固定资产：德加：未实现损益科目。主窗口现在如下所示：

11.2.

| 科目名称 | 类型 | 商品 | 合计 |
|---------|-------|------|---------------|
| ▼ 收入 | 收入 | 人民币元 | CN¥10,000.00 |
| 未实现损益 | 收入 | 人民币元 | CN¥10,000.00 |
| 已实现损益 | 收入 | 人民币元 | CN¥0.00 |
| ▶ 所有者权益 | 所有者权益 | 人民币元 | CN¥100,000.00 |
| ▼ 资产 | 资产 | 人民币元 | CN¥110,000.00 |
| ▼ 固定资产 | 资产 | 人民币元 | CN¥110,000.00 |
| ▼ 德加 | 资产 | 人民币元 | CN¥110,000.00 |
| 成本 | 资产 | 人民币元 | CN¥100,000.00 |
| 未实现损益 | 资产 | 人民币元 | CN¥10,000.00 |
| ▼ 流动资产 | 资产 | 人民币元 | CN¥0.00 |
| 储蓄存款 | 资产 | 人民币元 | CN¥0.00 |

添加未实现损益后的科目表。

出售

假定又过了一个月，德加的画作价格又涨了一些，你估计这次增值有20,000元。你将这20,000元及时记录为未实现损益收入，就像上面那样，并随后决定出售这份画作。

现在有三种可能的情况。你估计的未实现损益可能是估计准确的，可能被高估，也可能被低估了。

1. 未实现损益估计准确。

你对画作的乐观估值是正确的。首先你需要将盈利记录为已实现损益，而非未实现损益。具体操作是将收入由收入：未实现损益转至收入：已实现损益。

其后，你需要按画作的出售价格贷记银行科目。这些钱刚好来自于资产：固定资产：德加的子科目。将资产：固定资产：德加：成本的全部金额转到资产：流动资产：储蓄存款，再将资产：固定资产：德加：未实现损益的全部金额转到资产：流动资产：储蓄存款。

这些交易现在应该如下所示：

11.1.

| 科目 | 转账至 | 交易金额 | 科目余额 |
|----------|----------|----------|------|
| 收入:未实现损益 | 收入:已实现损益 | ¥ 30,000 | ¥ 0 |

| | | | |
|------------------|--------------|-----------|-----|
| 资产:固定资产:德加:成本 | 资产:流动资产:储蓄存款 | ¥ 100,000 | ¥ 0 |
| 资产:固定资产:德加:未实现损益 | 资产:流动资产:储蓄存款 | ¥ 30,000 | ¥ 0 |

现在，资产:流动资产:储蓄存款科目总计130,000元，收入:已实现损益科目总计30,000元。

11.3.

| 科目名称 | 类型 | 商品 | 合计 |
|---------|-------|------|---------------|
| ▼ 收入 | 收入 | 人民币元 | CN¥30,000.00 |
| 未实现损益 | 收入 | 人民币元 | CN¥0.00 |
| 已实现损益 | 收入 | 人民币元 | CN¥30,000.00 |
| ▶ 所有者权益 | 所有者权益 | 人民币元 | CN¥100,000.00 |
| ▼ 资产 | 资产 | 人民币元 | CN¥130,000.00 |
| ▼ 固定资产 | 资产 | 人民币元 | CN¥0.00 |
| ▼ 德加 | 资产 | 人民币元 | CN¥0.00 |
| 成本 | 资产 | 人民币元 | CN¥0.00 |
| 未实现损益 | 资产 | 人民币元 | CN¥0.00 |
| ▼ 流动资产 | 资产 | 人民币元 | CN¥130,000.00 |
| 储蓄存款 | 资产 | 人民币元 | CN¥130,000.00 |

录入已实现损益后的科目表。

2. 未实现损益被高估。

你对画作的价值过于乐观了。你认为画作值130,000元，而得到的报价只有120,000元。但你还是决定出售，因为在你看来这120,000元比画作更值钱。估值稍有变化，但差距也不算夸张。

交易现在应该如下所示（注意最后一项交易平衡了未实现损益科目）：

11.2.

| 科目 | 转账至 | 交易金额 | 科目余额 |
|----|-----|------|------|
|----|-----|------|------|

| | | | |
|------------------|--------------|-----------|----------|
| 收入:未实现损益 | 收入:已实现损益 | ¥ 20,000 | ¥ 10,000 |
| 资产:固定资产:德加:成本 | 资产:流动资产:储蓄存款 | ¥ 100,000 | ¥ 0 |
| 资产:固定资产:德加:未实现损益 | 资产:流动资产:储蓄存款 | ¥ 20,000 | ¥ 10,000 |
| 资产:固定资产:德加:未实现损益 | 收入:未实现损益 | ¥ 10,000 | ¥ 0 |

现在, 资产:流动资产:储蓄存款科目总额120,000元, 同时收入:已实现损益总额20,000元。

3. 未实现损益被低估。

你成功地把画作卖了一个你做梦都想不到的价格, 150,000元。额外的价值同样记录为收益, 也就是收入。

交易现在应该如下所示 (注意最后一项交易平衡了未实现损益科目) :

11.3.

| 科目 | 转账至 | 交易金额 | 科目余额 |
|------------------|------------------|-----------|-----------|
| 收入:未实现损益 | 收入:已实现损益 | ¥ 50,000 | ¥ -20,000 |
| 资产:固定资产:德加:成本 | 资产:流动资产:储蓄存款 | ¥ 100,000 | ¥ 0 |
| 资产:固定资产:德加:未实现损益 | 资产:流动资产:储蓄存款 | ¥ 50,000 | ¥ -20,000 |
| 收入:未实现损益 | 资产:固定资产:德加:未实现损益 | ¥ 20,000 | ¥ 0 |

现在, 资产:流动资产:储蓄存款科目总额150,000元, 同时收入:已实现损益总额50,000元。

缴税

不同国家间的税收政策差别迥异, 所以介绍全世界通用的相关知识几乎是不可能的。然而共同的一点是, 资本利得收入在资产被实际卖出之前不涉及缴税, 不过也有例外。北美住宅拥有者通常会发现, 当他们出售个人住所的时候, 资本利得免于征税。其它一些国家对住宅出售的处理

似乎有所不同，会对这样的收益收税。例如，德国政府会在持有房产少于十年的情况下会对这些收益征税。

克里斯·布朗分享了这样一个故事，在他职业报税日的时候，有一家人想要出售农场，本以为会收到金额巨大的税单，但由于在1971年之前就拥有了这块地产，结果缴税额几乎为零。同时，由于是一家奶牛场，还获得了一些奇奇怪怪的金额扣除。故事的关键在于尽管相关事实十分简单，但税务机制经常会把事情搞得复杂得可怕……

第 12 章 多货币

本章会介绍怎样配置GnuCash科目管理多种货币。

基本概念

GnuCash支持从安道尔法郎到津巴布韦元在内的100多种货币。例如，您可以使用欧元建立一个银行科目，而使用港元建立另一个银行科目。

管理多种货币时，你的脑海中或许会浮现出一些疑问：

- 怎样在使用不同货币的科目之间转账？
- 怎样计算使用不同货币的科目的总金额？
- 怎样浏览包含不同货币的报表？

你既可以按“手动记录外币交易”一节所述，手动录入交易记录多种货币的兑汇损益；也可以按随后的章节所述，利用GnuCash在2.3.14版本中引入的外币贸易科目自动记录兑汇损益。GnuCash的贸易科目功能默认不启用，必须按“使用贸易科目自动记录外币交易”一节手动启用。

手动记录外币交易

建立科目

本位币由〔编辑〕→〔首选项〕(macOS中为〔GnuCash〕→〔首选项〕)下的〔科目〕标签页设定。

类似地，GnuCash也提供选项设置用于展示报表的首选货币，称为〔默认报表货币〕，位于〔GnuCash首选项〕窗口中的〔报表〕标签页下。建议开始使用GnuCash时就将这两个选项都设置好，例如假设科目均以港元记账，而生成的报表都显示为人民币，报表就会提示“所选时段内没有交易数据或交易金额为零”。

创建新科目时，可以选择用作科目记账单位的商品。可以为以货币为单位的科目指定任何GnuCash支持的货币，只需在货币商品列表中选择需要的货币。默认选中的货币为该新建科目的父科目使用的货币。

例如，现在创建一套银行科目，你大多使用人民币记账，但也拥有使用欧元的欧洲银行账户，以及使用港元的香港银行账户。因此需要建立三个银行科目，一个科目单位为人民币，一个为欧元，而第三个为港元。一种适用此情况的科目结构可以是：

- 资产 (CNY)
 - 流动资产 (CNY)
 - 中国银行 (CNY)
 - 欧洲银行 (EUR)
 - 香港银行 (HKD)
- 所有者权益 (CNY)
 - 期初余额 (CNY)
 - 人民币 (CNY)
 - 欧元 (EUR)
 - 港元 (HKD)

每个科目使用的货币标注在括号里。

由于这个示例中大多使用人民币记账，所以父科目均设为人民币（CNY）。自然，你如果大多使用欧元记账，则可将这些父科目的货币更换为欧元。在科目树窗口中显示的总额都会换算为相应科目的货币。注意，这里也建立了三个期初余额所有者权益科目，用于向三个银行科目添加初始金额。

也可以只建立一个期初余额科目，再通过跨货币转账向使用不同货币的科目添加初始交易，然而这是更进阶的做法，后文“使用外币买入资产”一节会详细介绍。

下面展示了按该示例得到的结果。你在三个银行科目中最初各有10,000元人民币、10,000欧元，以及10,000港元。注意，父科目总金额只包含与之货币相同的子科目金额。换句话说，资产总额和所有者权益总额只体现人民币金额，因为GnuCash目前还无法换算欧元或港元的金额。一旦设定了货币间的汇率，父科目就会换算出所有子科目的金额。具体操作方法见后文“记录并更新货币汇率”一节。

12.1.

| Account Name | P | Type | Total | Total (USD) |
|------------------|-------------------------------------|--------|--------------|-------------|
| Assets | <input checked="" type="checkbox"/> | Asset | \$10,000.00 | \$10,000.00 |
| Current Assets | <input checked="" type="checkbox"/> | Asset | \$10,000.00 | \$10,000.00 |
| European Bank | <input type="checkbox"/> | Bank | €10,000.00 | \$0.00 |
| HK Bank | <input type="checkbox"/> | Bank | HK\$1,000.00 | \$0.00 |
| US Bank | <input type="checkbox"/> | Bank | \$10,000.00 | \$10,000.00 |
| Equity | <input checked="" type="checkbox"/> | Equity | \$10,000.00 | \$10,000.00 |
| Opening Balances | <input checked="" type="checkbox"/> | Equity | \$10,000.00 | \$10,000.00 |
| EUR | <input type="checkbox"/> | Equity | €10,000.00 | \$0.00 |
| HKD | <input type="checkbox"/> | Equity | HK\$1,000.00 | \$0.00 |
| USD | <input type="checkbox"/> | Equity | \$10,000.00 | \$10,000.00 |

\$, Grand Total: Net Assets: \$10,000.00 Profits: \$0.00

最初建立的三个银行科目，货币各不相同。

注意此处显示了〔合计(CNY)〕列，可以在表头点击下箭头并勾选〔合计(CNY)〕。

自定义货币

我们谈到货币的时候，通常是指由政府提供信用担保的通货，或者更准确地说，是ISO 4217 [https://en.wikipedia.org/wiki/ISO_4217]中定义的货币。GnuCash不允许你创建自己的货币。要记录非ISO货币，有两种变通方法可供使用，你可以选择更符合需求的一种。

假设你想跟踪在某一企业集团购物时获得的会员积点。记录会员积点的科目为资产:其它:会员积点

1. 将会员积分视为证券——就像股票或基金一样。

在这种变通方法中，要定义一种新的证券，类型为〔基金〕，叫做会员积点。思路很简单：创建新的会员积点科目时，只需将科目类型设为〔股票〕或〔基金〕，点击〔证券/货币:〕输入框旁的选择...按钮，再点击新建，添加一种类型为〔基金〕的新证券。

这其实不是股票和基金科目类型的本意，但GnuCash还是允许这样做。缺点是每项涉及该科目的交易都需要输入“价格”，因为GnuCash需要价格信息算出积点的货币价值，即将这些积点视为资产的一部分。

2. 使用一种“伪货币”记录会员积点。

在这种变通方法中，你使用一种伪货币记录会员积点。这些货币是“XTS (测试用代码)”和“XXX (无货币)”。如果将其中一种用于会员积点科目，则当向科目录入交易时无需每次都输入份额价格。另外，可以一直使用这两种伪货币跟踪各种数量，目前本年已获得和已使用的的休假津贴、已获得和已使用的年假时长、已用和剩余的医疗保险额度，诸如此类。

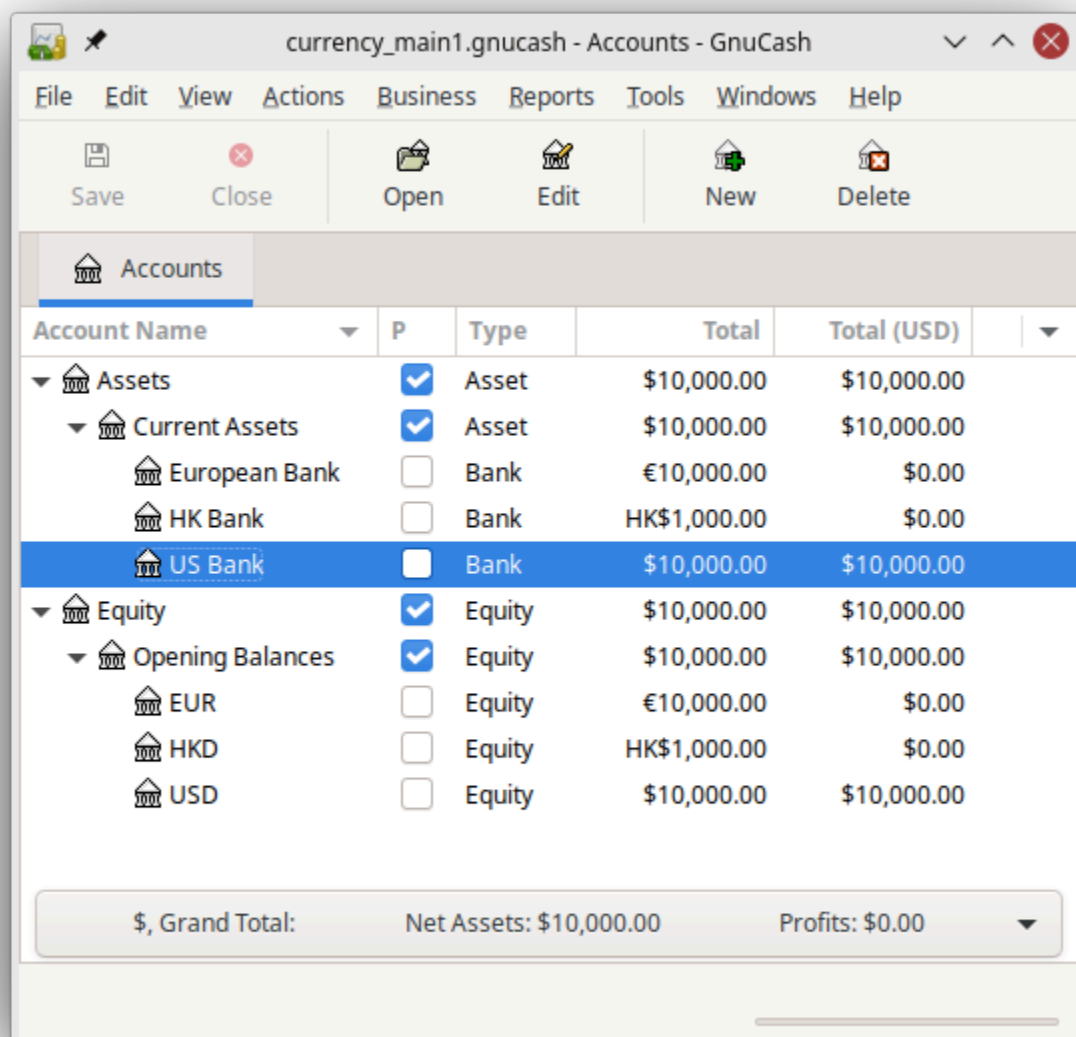
这种变通方法的缺点在于你无法定义伪货币的汇率，以将之换算为ISO货币。如果想实现这一功能，就应该使用第一种变通方法。

记录并更新货币汇率

你可以在GnuCash中更新货币汇率，手动或自动皆可，操作上与设定投资份额价格（详见“设置份额价格”一节）几乎相同。下面两小节会分别介绍两种方法。

开始前，先简单看一下科目表。

12.2.



| Account Name | P | Type | Total | Total (USD) |
|------------------|-------------------------------------|--------|--------------|-------------|
| Assets | <input checked="" type="checkbox"/> | Asset | \$10,000.00 | \$10,000.00 |
| Current Assets | <input checked="" type="checkbox"/> | Asset | \$10,000.00 | \$10,000.00 |
| European Bank | <input type="checkbox"/> | Bank | €10,000.00 | \$0.00 |
| HK Bank | <input type="checkbox"/> | Bank | HK\$1,000.00 | \$0.00 |
| US Bank | <input checked="" type="checkbox"/> | Bank | \$10,000.00 | \$10,000.00 |
| Equity | <input checked="" type="checkbox"/> | Equity | \$10,000.00 | \$10,000.00 |
| Opening Balances | <input checked="" type="checkbox"/> | Equity | \$10,000.00 | \$10,000.00 |
| EUR | <input type="checkbox"/> | Equity | €10,000.00 | \$0.00 |
| HKD | <input type="checkbox"/> | Equity | HK\$1,000.00 | \$0.00 |
| USD | <input type="checkbox"/> | Equity | \$10,000.00 | \$10,000.00 |

\$, Grand Total: Net Assets: \$10,000.00 Profits: \$0.00

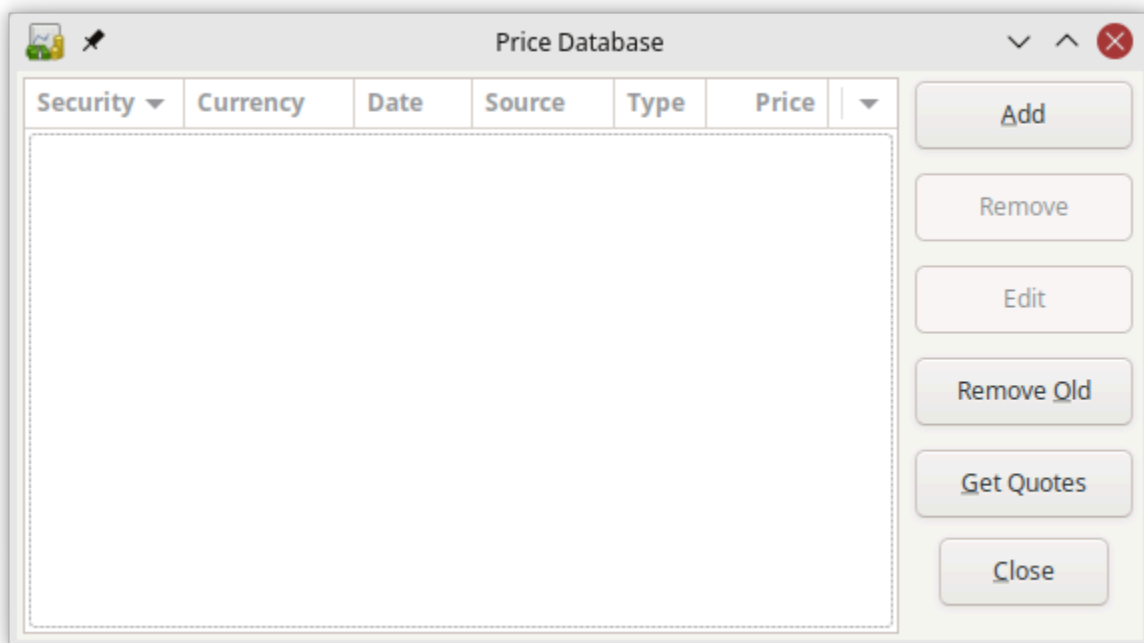
最初建立的三个银行科目，使用的货币各不相同。

如图所示，总余额没有反映目前持有的欧元或港元的价值。添加货币汇率会解决这个问题。

手动更新汇率

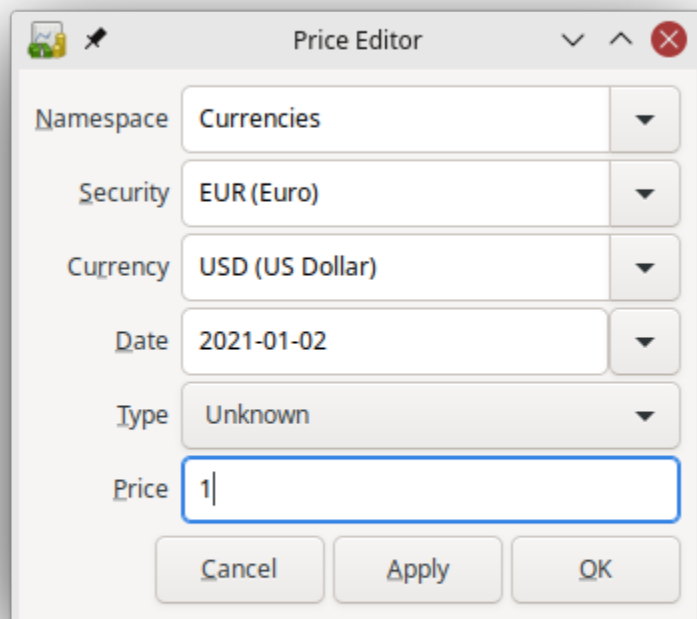
点击〔工具〕→〔价格数据库〕打开〔价格数据库〕。

12.3.



点击添加按钮添加一条新的货币汇率记录。可以在弹出的窗口中设定新汇率，如下图所示：

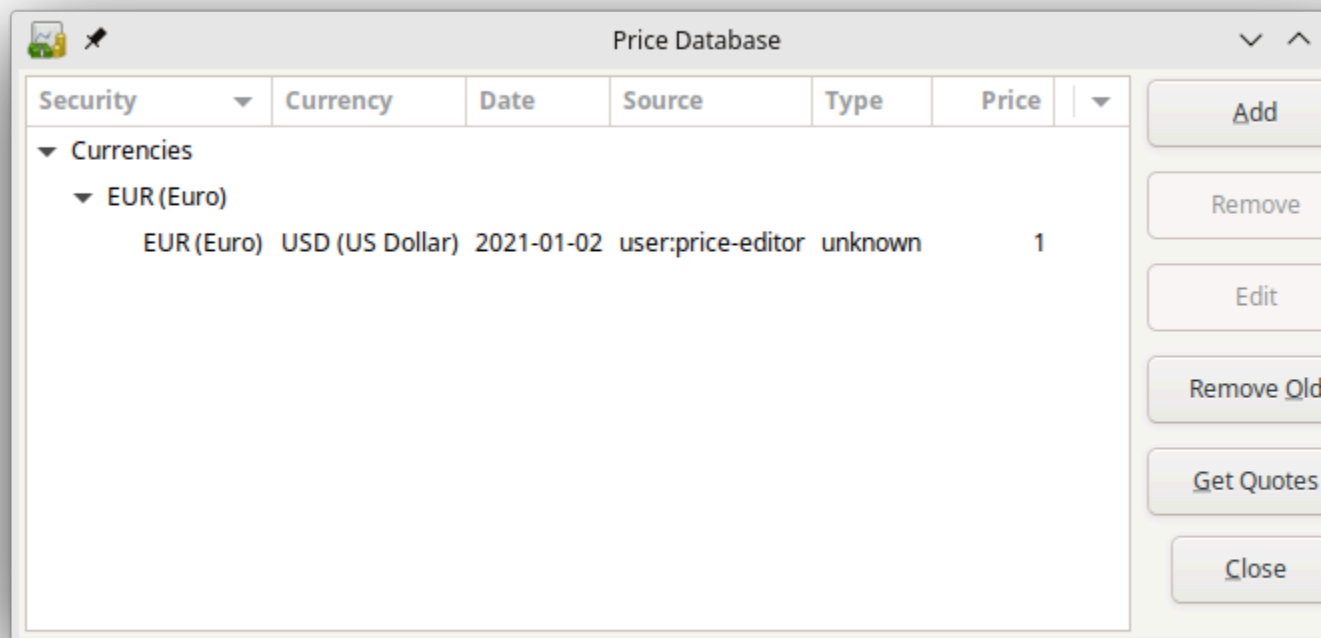
12.4.

A screenshot of a 'Price Editor' dialog box. It contains several dropdown menus and a text input field. The 'Namespace' dropdown is set to 'Currencies'. The 'Security' dropdown is set to 'EUR (Euro)'. The 'Currency' dropdown is set to 'USD (US Dollar)'. The 'Date' dropdown is set to '2021-01-02'. The 'Type' dropdown is set to 'Unknown'. The 'Price' text input field contains the number '1'. At the bottom, there are three buttons: 'Cancel', 'Apply', and 'OK'.

添加价格记录窗口

将〔命名空间〕设为〔货币〕，〔证券〕设为〔EUR（欧元）〕。然后设定选择的“证券”与选择的“货币”之间的汇率。在〔价格〕输入框设定购买一个单位的外币需花费多少个单位的本币。在这个情况下，就是指多少元人民币能够买入1欧元。在本例中，将汇率设为1欧元兑7.8元人民币。 Then set the exchange rate between the selected security and the selected currency. The〔Price〕 box defines how many units of currency are required to purchase one unit of the security. In this case, how many dollars it will take to purchase on 1 Euro. In this example, you will set the exchange rate to 1 EUR for 1 USD.

12.5.



12.6.

| Account Name | P | Type | Total | Total (USD) |
|------------------|-------------------------------------|--------|--------------|-------------|
| Assets | <input checked="" type="checkbox"/> | Asset | \$20,000.00 | \$20,000.00 |
| Current Assets | <input checked="" type="checkbox"/> | Asset | \$20,000.00 | \$20,000.00 |
| European Bank | <input type="checkbox"/> | Bank | €10,000.00 | \$10,000.00 |
| HK Bank | <input type="checkbox"/> | Bank | HK\$1,000.00 | \$0.00 |
| US Bank | <input type="checkbox"/> | Bank | \$10,000.00 | \$10,000.00 |
| Equity | <input checked="" type="checkbox"/> | Equity | \$20,000.00 | \$20,000.00 |
| Opening Balances | <input checked="" type="checkbox"/> | Equity | \$20,000.00 | \$20,000.00 |
| EUR | <input type="checkbox"/> | Equity | €10,000.00 | \$10,000.00 |
| HKD | <input type="checkbox"/> | Equity | HK\$1,000.00 | \$0.00 |
| USD | <input type="checkbox"/> | Equity | \$10,000.00 | \$10,000.00 |

\$, Grand Total: Net Assets: \$20,000.00 Profits: \$0.00

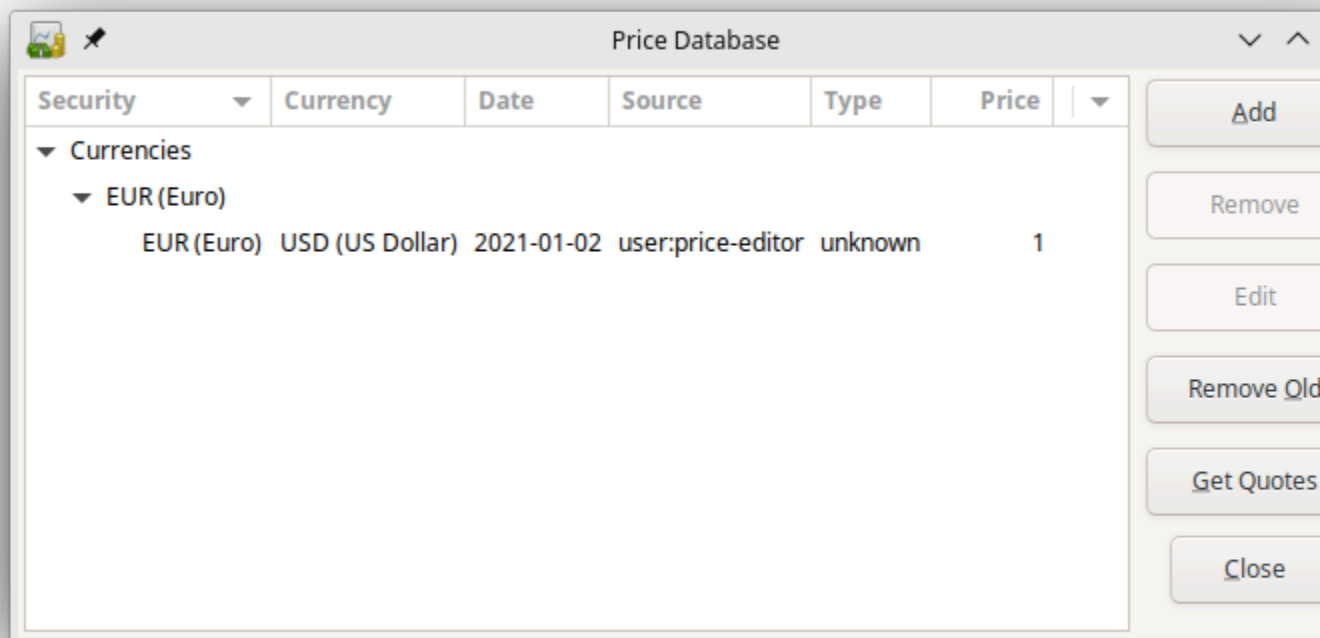
注意到由于没有录入港元汇率，GnuCash不会将港元科目余额换算为人民币。下一节会添加港元汇率。

自动更新汇率

上一节介绍了怎样手动添加新的货币汇率，但GnuCash也具备自动更新报价的功能，下面进行介绍。

点击〔工具〕→〔价格数据库〕打开〔价格数据库〕。

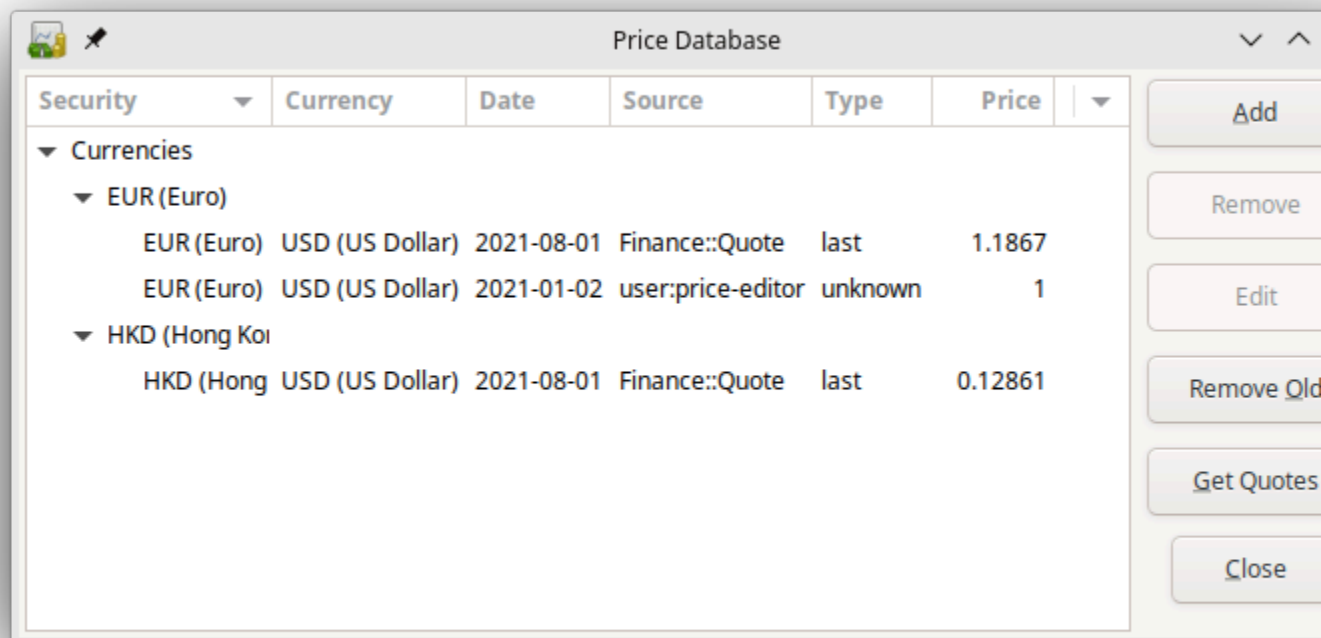
12.7.



点击获取报价按钮即可自动加载需要的多项汇率数据。

如果获取报价按钮不可用，则表明尚未安装 Perl 模块 Finance::Quote。有关如何安装该模块的信息参见“配置自动获取报价”一节。

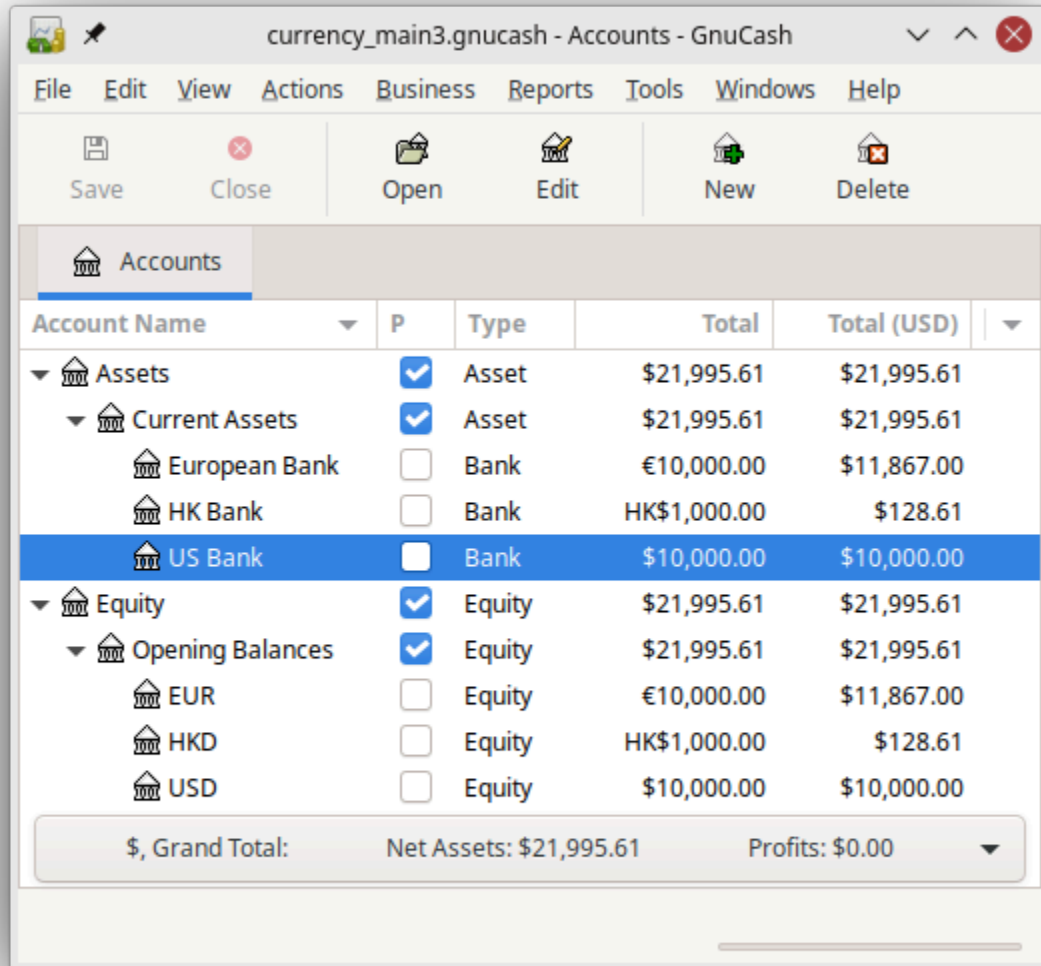
12.8.



每次点击获取报价或如“配置自动获取报价”一节所述配置GnuCash按要求下载报价的时候，GnuCash就会下载各科目使用的所有货币的汇率数据。

现在查看主科目表，可以看到GnuCash已经自动将港元金额换算为人民币金额，并计入以人民币记账的父科目中，也就是〔总计(CNY)〕中给出的金额。欧元科目也已经同样按最新的汇率更新了金额。

12.9.



| Account Name | P | Type | Total | Total (USD) |
|------------------|-------------------------------------|--------|--------------|-------------|
| Assets | <input checked="" type="checkbox"/> | Asset | \$21,995.61 | \$21,995.61 |
| Current Assets | <input checked="" type="checkbox"/> | Asset | \$21,995.61 | \$21,995.61 |
| European Bank | <input type="checkbox"/> | Bank | €10,000.00 | \$11,867.00 |
| HK Bank | <input type="checkbox"/> | Bank | HK\$1,000.00 | \$128.61 |
| US Bank | <input type="checkbox"/> | Bank | \$10,000.00 | \$10,000.00 |
| Equity | <input checked="" type="checkbox"/> | Equity | \$21,995.61 | \$21,995.61 |
| Opening Balances | <input checked="" type="checkbox"/> | Equity | \$21,995.61 | \$21,995.61 |
| EUR | <input type="checkbox"/> | Equity | €10,000.00 | \$11,867.00 |
| HKD | <input type="checkbox"/> | Equity | HK\$1,000.00 | \$128.61 |
| USD | <input type="checkbox"/> | Equity | \$10,000.00 | \$10,000.00 |

\$, Grand Total: Net Assets: \$21,995.61 Profits: \$0.00

禁用汇率获取

一旦创建了使用非本位币的科目，GnuCash就会自动开启该货币的汇率获取功能。然而，如果你之后删除那个科目，GnuCash不会自动停用该货币的汇率获取功能。 Whenever you create an account that uses a non-default currency, exchange rate retrieval will be automatically enabled for that currency. However, if you later delete that account, GnuCash will not automatically disable exchange rate retrieval for that currency.

你如果已经删除了最后一个使用某货币的科目，并且以后不想再获取该货币的汇率，则需进行如下操作：

1. 点击〔工具〕→〔证券编辑器〕打开证券窗口。
2. 确定已勾选〔显示国家货币〕。
3. 展开〔货币〕行。
4. 双击希望停用汇率获取的货币。
5. 取消勾选〔获取在线报价〕并点击确定。

记录通过外币的购买

使用外币的购买交易可以通过两种不同的方式处理。

A.

使用GnuCash内置的货币兑换功能录入交易。这主要用于单次交易，不会定期产生其它交易。

B.

使用单独的科目跟踪交易，所有涉及到的科目使用同种货币。这样跟踪和追加起来简单得多，因此是推荐的做法。使用这种方式时，先进行一次货币兑换交易，然后再使用本位币录入正常的交易。

本节接下来将主要基于“B”方式介绍操作流程。

使用外币买入资产

你使用的本位币为人民币，但你决定在越南买一艘船。相应地，你需要打开越南的银行科目，从中国转进一些钱，然后买下你的船。

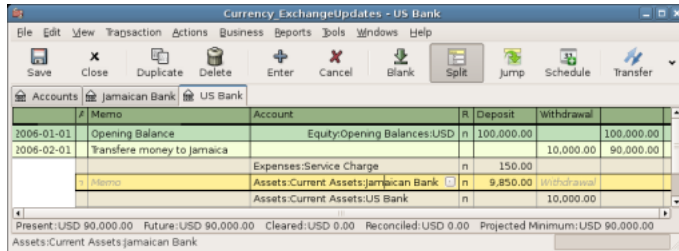
要在GnuCash中记录这项交易，可以使用如下基本科目结构：

- 资产 (CNY)
 - 流动资产 (CNY)
 - 中国银行 (CNY)
 - 越南银行 (VND)
 - 固定资产 (CNY)
 - 船 (VND)
- 所有者权益 (CNY)
 - 期初余额 (CNY)
 - 人民币 (CNY)

每个科目使用的货币标在了括号中。

首先，你需要向越南转账10,000元，使用的是中国银行科目（余额100,000元人民币）。银行给出的汇率是1 CNY = 3505 JMD，但收取了150元人民币的兑汇手续费。

12.10. “

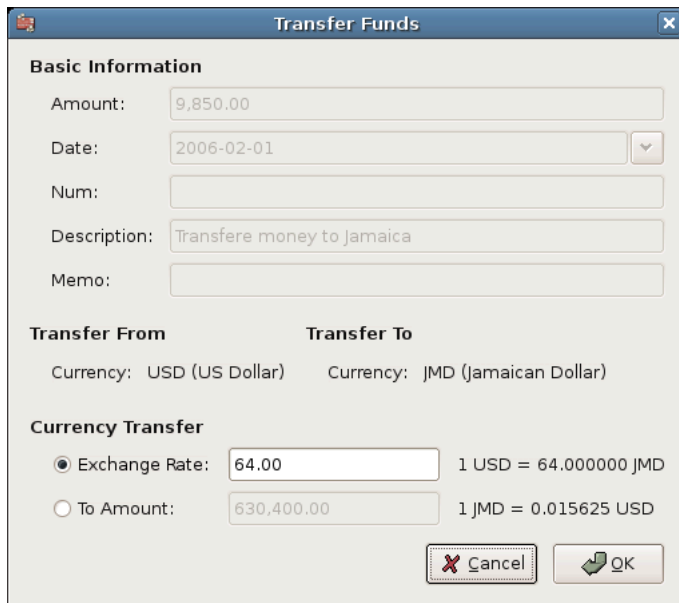


| # | Memo | Account | R | Deposit | Withdrawal |
|------------|----------------------------|-------------------------------------|---|------------|------------|
| 2006-01-01 | Opening Balance | Equity:Opening Balances:USD | n | 100,000.00 | 100,000.00 |
| 2006-02-01 | Transfere money to jamaica | | | 10,000.00 | 90,000.00 |
| | | Expenses:Service Charge | n | 150.00 | |
| | | Assets:Current Assets:jamaican Bank | n | 9,850.00 | |
| | | Assets:Current Assets:US Bank | n | | 10,000.00 |

Present: USD 90,000.00 Future: USD 90,000.00 Cleared: USD 0.00 Reconciled: USD 0.00 Projected Minimum: USD 90,000.00

选中兑换越南盾的交易行（9,850.00元），右击选择〔编辑汇率〕。

12.11.



Transfer Funds

Basic Information

Amount: 9,850.00

Date: 2006-02-01

Num:

Description: Transfere money to jamaica

Memo:

Transfer From **Transfer To**

Currency: USD (US Dollar) Currency: JMD (Jamaican Dollar)

Currency Transfer

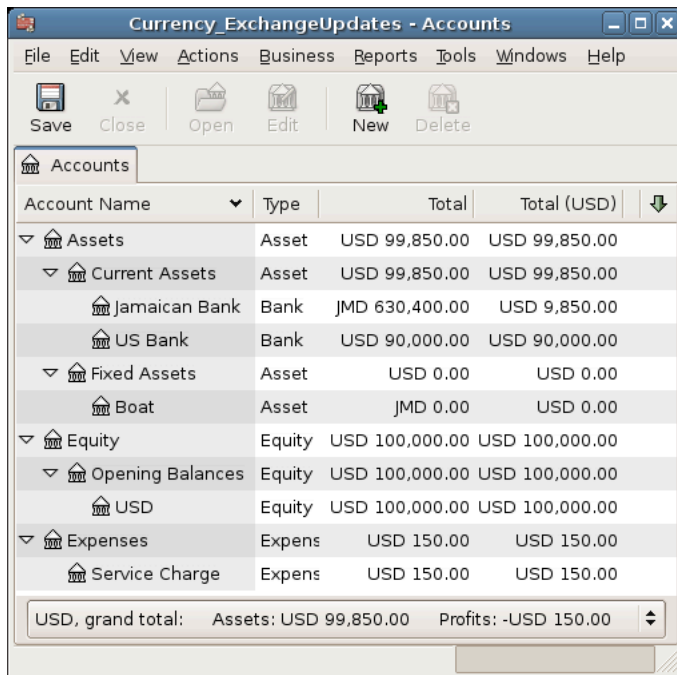
☒ Exchange Rate: 64.00 1 USD = 64.000000 JMD

☐ To Amount: 630,400.00 1 JMD = 0.015625 USD

在这个对话框中指定外币交易的汇率。

你输入1 CNY = 3505 VND 为汇率，因为这个汇率是银行给定的。在资金转账（编辑汇率）窗口中点击〔确定〕，保存分录交易。科目表中现在应当如下所示。

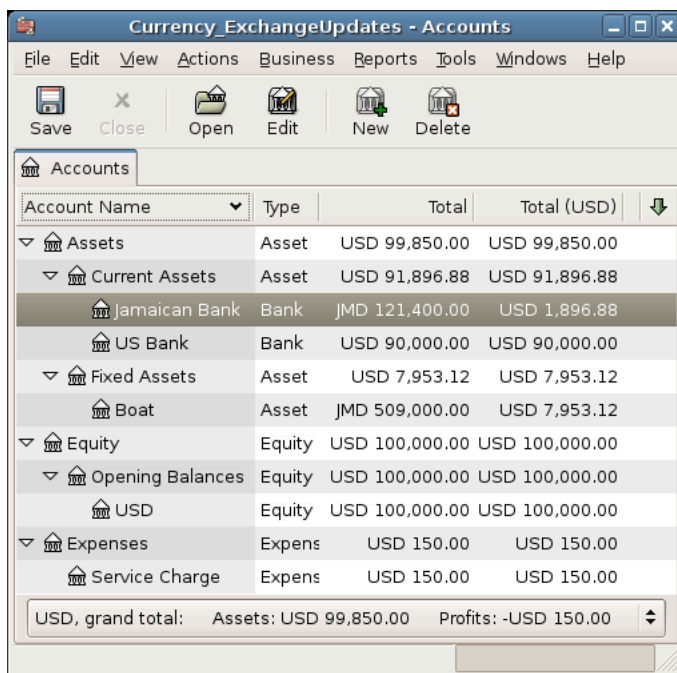
12.12.



| Account Name | Type | Total | Total (USD) |
|-------------------|--------|-----------------------|----------------------|
| Assets | Asset | USD 99,850.00 | USD 99,850.00 |
| Current Assets | Asset | USD 99,850.00 | USD 99,850.00 |
| Jamaican Bank | Bank | JMD 630,400.00 | USD 9,850.00 |
| US Bank | Bank | USD 90,000.00 | USD 90,000.00 |
| Fixed Assets | Asset | USD 0.00 | USD 0.00 |
| Boat | Asset | JMD 0.00 | USD 0.00 |
| Equity | Equity | USD 100,000.00 | USD 100,000.00 |
| Opening Balances | Equity | USD 100,000.00 | USD 100,000.00 |
| USD | Equity | USD 100,000.00 | USD 100,000.00 |
| Expenses | Expens | USD 150.00 | USD 150.00 |
| Service Charge | Expens | USD 150.00 | USD 150.00 |
| USD, grand total: | | Assets: USD 99,850.00 | Profits: -USD 150.00 |

你确定花509,000越南盾买船。要在GnuCash中记录这项交易，你需要在资产:流动资产:越南银行中将509,000越南盾转账至资产:固定资产:船只。

12.13.



| Account Name | Type | Total | Total (USD) |
|-------------------|--------|-----------------------|----------------------|
| Assets | Asset | USD 99,850.00 | USD 99,850.00 |
| Current Assets | Asset | USD 91,896.88 | USD 91,896.88 |
| Jamaican Bank | Bank | JMD 121,400.00 | USD 1,896.88 |
| US Bank | Bank | USD 90,000.00 | USD 90,000.00 |
| Fixed Assets | Asset | USD 7,953.12 | USD 7,953.12 |
| Boat | Asset | JMD 509,000.00 | USD 7,953.12 |
| Equity | Equity | USD 100,000.00 | USD 100,000.00 |
| Opening Balances | Equity | USD 100,000.00 | USD 100,000.00 |
| USD | Equity | USD 100,000.00 | USD 100,000.00 |
| Expenses | Expens | USD 150.00 | USD 150.00 |
| Service Charge | Expens | USD 150.00 | USD 150.00 |
| USD, grand total: | | Assets: USD 99,850.00 | Profits: -USD 150.00 |

现在科目表反映出你的银行科目已经减少了509,000越南盾，而固定资产船只科目增加了同样的数额。如果已在科目表的选择列中勾选了〔总计 (CNY)〕，则可以看到这些科目换算为人民币的金额。人民币金额总是根据自动或手动获取的最新货币汇率换算的。

购买境外股票

此示例会展示怎样购买以外币计价的股票。

假定你住在上海，即本位币为人民币。你决定买入在香港交易的、以港元计价的股票。另外，你也想跟踪属于每支股票和每家经纪商的各种收入和费用数额。

你决定买入港铁公司的股票，编号为0694.HK。由于想要跟踪各种收入和费用金额，所以需要建立如下科目结构：

- 资产:投资:经纪账户:宝盛证券:0694.HK (0694.HK)
- 资产:投资:经纪账户:宝盛证券:资金账户 (HKD)
- 所有者权益:期初余额:港元 (HKD)
- 费用:佣金:宝盛证券:0694.HK (HKD)
- 收入:投资:分红:宝盛证券:0694.HK (HKD)

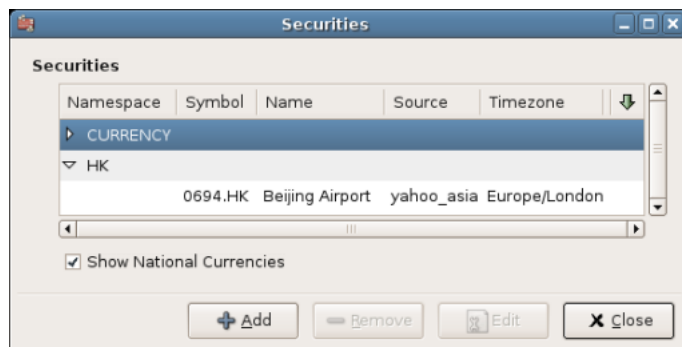
逐一创建所需科目后的科目表如下：

12.14.

| Account Name | Type | Commodity | Total | Total (USD) |
|-------------------|------------------|-------------------|-----------|-------------|
| Assets | Asset | US Dollar | USD 0.00 | USD 0.00 |
| Investments | Asset | US Dollar | USD 0.00 | USD 0.00 |
| Brokerage Account | Bank | US Dollar | USD 0.00 | USD 0.00 |
| Boom | Bank | Hong Kong Dollar | HKD 0.00 | USD 0.00 |
| 0694.HK | Stock | Beijing Airport | 0 0694.HK | USD 0.00 |
| Bank | Bank | Hong Kong Dollar | HKD 0.00 | USD 0.00 |
| Equity | Equity | US Dollar | USD 0.00 | USD 0.00 |
| Opening Balances | Equity | US Dollar | USD 0.00 | USD 0.00 |
| HKD | Equity | Hong Kong Dollar | HKD 0.00 | USD 0.00 |
| Expenses | Expense | US Dollar | USD 0.00 | USD 0.00 |
| Commissions | Expense | US Dollar | USD 0.00 | USD 0.00 |
| Boom | Expense | Hong Kong Dollar | HKD 0.00 | USD 0.00 |
| 0694.HK | Expense | Hong Kong Dollar | HKD 0.00 | USD 0.00 |
| Income | Income | US Dollar | USD 0.00 | USD 0.00 |
| Investments | Income | US Dollar | USD 0.00 | USD 0.00 |
| Dividend | Income | US Dollar | USD 0.00 | USD 0.00 |
| Boom | Income | Hong Kong Dollar | HKD 0.00 | USD 0.00 |
| USD, grand total: | Assets: USD 0.00 | Profits: USD 0.00 | | |

各支股票可通过证券编辑器（〔工具〕→〔证券编辑器〕）添加。

12.15.



若尚未将资金（50,000港元）转入经纪商现金账户（资产:投资:经纪账户:宝盛证券:资金账户），则现在就该转入这笔资金了，可以使用所有者权益（港元）科目，或某一已经存在的银行科目（现金转账）。

有两种录入买入交易的方式，可以在现金科目中录入交易（下文会介绍），也可以在股票科目中录入交易。

如果在股票科目中录入交易，会假定股票以父科目使用的货币计价。

假设股票价格为每股28港元。要录入买入交易，打开经纪商的港元现金科目（资产:投资:经纪账户:宝盛证券:资金账户），并按下表录入：

| | | |
|-------------------------|----|------------------|
| 资产:投资:经纪账户:宝盛证券:资金账户 | 取出 | 50,000 |
| 费用:佣金:宝盛证券:0694.HK | 存入 | 500 |
| 资产:投资:经纪账户:宝盛证券:0694.HK | 存入 | 49,500 (16,500份) |

如果汇率对话框没有自动弹出，右击股票交易行，选择〔编辑汇率〕。在〔转出份额〕中录入份额数量（16,500）。

12.16.

Transfer Funds

Basic Information

Amount: 49,500.00

Date: 2006-01-02

Num:

Description: Purchase Beijing Airport

Memo:

Transfer From **Transfer To**

Currency: HKD (Hong Kong Dollar) Currency: 0694.HK (Beijing Airport)

Currency Transfer

☐ Exchange Rate: 0.333333 1 HKD = 0.333333 0694.HK

☒ To Amount: 16,500.00 1 0694.HK = 3.000000 HKD

在资金转账对话框中设定份额数量。

回到科目表后，可以看到买入的份额已经反映在了股票科目的总额中。

12.17.

| Account Name | Type | Commodity | Total | Total (USD) |
|-------------------|---------|------------------|------------------|-------------------|
| Assets | Asset | US Dollar | USD 0.00 | USD 0.00 |
| Investments | Asset | US Dollar | USD 0.00 | USD 0.00 |
| Brokerage Account | Bank | US Dollar | USD 0.00 | USD 0.00 |
| Boom | Bank | Hong Kong Dollar | HKD 0.00 | USD 0.00 |
| 0694.HK | Stock | Beijing Airport | 16,500 0694.HK | USD 0.00 |
| Bank | Bank | Hong Kong Dollar | HKD 0.00 | USD 0.00 |
| Equity | Equity | US Dollar | USD 0.00 | USD 0.00 |
| Opening Balances | Equity | US Dollar | USD 0.00 | USD 0.00 |
| HKD | Equity | Hong Kong Dollar | HKD 50,000.00 | USD 0.00 |
| Expenses | Expense | US Dollar | USD 0.00 | USD 0.00 |
| Commissions | Expense | US Dollar | USD 0.00 | USD 0.00 |
| Boom | Expense | Hong Kong Dollar | HKD 500.00 | USD 0.00 |
| Income | Income | US Dollar | USD 0.00 | USD 0.00 |
| USD, grand total: | | | Assets: USD 0.00 | Profits: USD 0.00 |

科目表现已包含境外股票。

不过，可能你已经注意到了，如果GnuCash还没有记录人民币与港元间的汇率数据，则人民币总额仍为零。要解决这个问题，前往〔工具〕→〔价格数据库〕点击获取报价按钮自动获取所需汇率数据。

需要重申，该示例展示了怎样在提供买入资金的现金科目账簿中录入交易，来以任意一种货币买入股票。也可以在股票科目账簿中录入买入交易，但要注意股票默认使用其父科目的货币记价。

在此示例中，股票科目的父科目（资产:投资:经纪账户:宝盛证券）以港元记价。由于父科目货币也就是股票的计价货币，因此可以放心地在股票科目账簿中录入买入交易。

跟踪外币投资

当你决定投资于其它国家的货币，并希望相对于本国货币升值时，你就是在进行外币投资。

向GnuCash录入这些交易时，需要考虑是记录得简单些，还是详细些。

如果你一点也不在意细节，下面这个相当简单的科目结构就足够了：

- 资产:投资:货币:银行 (CNY)
- 资产:投资:货币:XXX (XXX)

你只需录入两科目间的转账交易，并写明汇率。

但是，你如果想跟踪资本损益，以及任何形式的手续费，则无疑需要使用更复杂的科目结构，例如：

- 资产:投资:货币:银行 (CNY)
- 资产:投资:货币:外币银行:XXX (XXX)
- 费用:投资:货币:银行 (CNY)
- 收入:投资:货币:外币银行:资本利得:XXX (XXX)

买入外币

买入外币，就是以一定的汇率拿本币买入一定数量的外币，例如，你可以买入价值10,000元人民币的瑞士法郎，即 1 CNY = 5 THB，交易手续费150元人民币。

| | | |
|-------------------------|----|--------|
| 资产:投资:货币:银行 (CNY) | 取出 | 10,000 |
| 费用:投资:货币:银行 (CNY) | 费用 | 150 |
| 资产:投资:货币:外币银行:THB (THB) | 存入 | 49,250 |

在离开上述分录外币交易的最后一行时，汇率窗口会自动弹出。如果没有弹出窗口，右击该交易行或选择〔功能〕，再选择〔编辑汇率〕。在汇率窗口中设定由银行给出的汇率。

卖出外币投资

录入外币卖出的操作与外币买入相同，不过这里是将钱从外币科目转回本币银行科目（与“卖出份额”一节很类似）。

外币卖出要录入得合理，就必须明确记录已实现损益，这可以通过分录交易实现。分录交易必须体现来自收入:投资:货币:外币银行:资本利得:XXX科目的损益。要平衡这项收入，需要录入外币资产两次，一次记录实际发生的卖出（使用实际金额和实际汇率），另一次用来平衡收入获利（份额设为0）。

简而言之，外币卖出交易应当如下所示，也可以在资产:投资:货币:外币银行:XXX科目中看到这项交易。

12.1.

| | | |
|------------------------------|-----------------|-------------------|
| | / | / |
| 资产:投资:货币:银行 (CNY) | 卖出金额 - 兑汇手续费 | |
| 费用:投资:货币:银行 (CNY) | 兑汇手续费 | |
| 资产:投资:货币:外币银行:XXX (XXX) | | 卖出金额 (= 余额 + 获利) |
| 收入:投资:货币:外币银行:资本利得:XXX (XXX) | [亏损] | 获利 |
| 资产:投资:货币:外币银行:XXX (XXX) | 获利 (转 出份额=0) | [亏损 (转 出份额=0)] |

对账外币账单

对账外币账单的方法与对账当地银行账单的方法一样。如果你已经按照账单给出的条目，创建了相应的科目表结构，并将每个科目的货币设为该种外币，则对账操作与对账本币银行账单时几乎相同。

如果存在多种货币，那么可能需要在对账科目时手动将一种货币的金额换算至另一种货币。

使用贸易科目自动记录外币交易

贸易科目是在GnuCash 2.3.14 版本中引入的功能。其基本概念由Peter Selinger 在Tutorial on multiple currency accounting [https://www.mathstat.dal.ca/~selinger/accounting/tutorial]中提出，教程的第二部分Multiple currency accounting in GnuCash [https://www.mathstat.dal.ca/~selinger/accounting/gnucash]则介绍了怎样在早于2.3.14的版本中手动创建并使用贸易科目。 Trading accounts were introduced as a feature from . The conceptual basis for trading accounts is described in by . A second

tutorial describes the manual creation and use of trading accounts in GnuCash prior to version 2.3.14.

GnuCash默认不启用贸易科目。要启用贸易科目，前往〔文件〕→〔属性〕+〔科目〕标签页，勾选〔使用贸易科目〕复选框。

贸易科目的概念

在会计领域中，贸易科目是一种特殊的所有者权益科目，用于记录由两种或多种货币间交易导致的收入变动。GnuCash使用贸易这一顶级科目类型实现贸易科目的功能。启用贸易科目功能后，如果在创建两种或多种货币间的交易时，所需的贸易子科目还不存在，则会自动创建。

使用贸易科目拓展得到的会计等式如下： In the extended Accounting Equation, Trading accounts would appear as follows:

$$\text{资产} = \text{负债} + \text{所有者权益} (+\text{收入} - \text{费用}) + \text{贸易} \quad (12.1)$$

GnuCash会阻止用户直接手动向贸易顶级科目类别下的科目账簿添加条目。这些科目下的所有条目均为当用户在GnuCash科目结构中的其它科目账簿下创建外币交易时自动生成的。

贸易科目组织在顶级科目〔贸易〕下，该顶级科目具有子科目〔货币〕，在此之下又为每种相互产生交易的货币分配了各自的子科目。如之前“手动记录外币交易”一节中举的例子，启用贸易科目功能后，当录入不同货币间的交易时，会自动创建更多的贸易科目，具体为：

```
贸易
  货币
    EUR
    HKD
    CNY
```

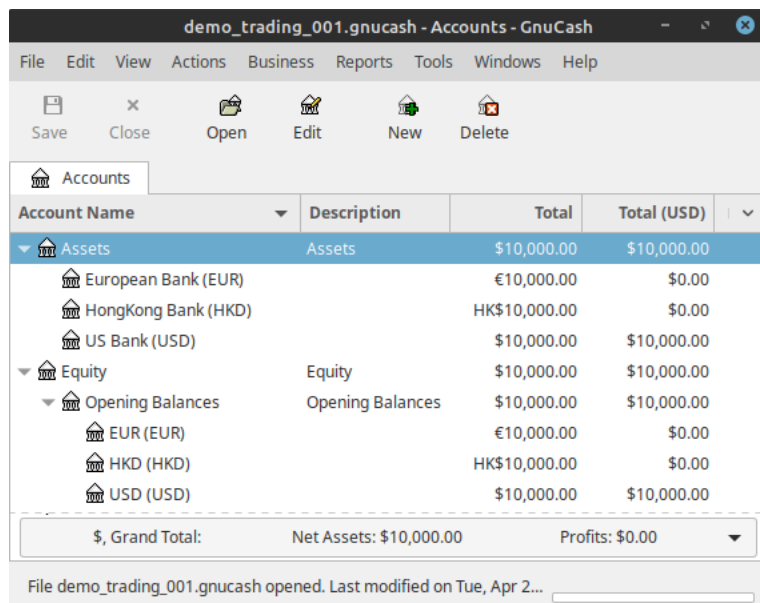
这些科目的名称也同样是自动指定的，如果不在GnuCash中关闭贸易科目功能，就无法由用户修改。

使用贸易科目记录外币交易

建立适用于贸易科目功能的科目结构

仍然继续采用前一小节手动记录外币交易中使用的科目结构。资产和所有者权益科目的初始结构如下：

12.18.



The screenshot shows the 'demo_trading_001.gnucash - Accounts - GnuCash' window. The 'Accounts' tab is active, displaying a tree view of accounts. The 'Assets' account is expanded, showing 'European Bank (EUR)', 'HongKong Bank (HKD)', and 'US Bank (USD)'. The 'Equity' account is also expanded, showing 'Opening Balances', 'EUR (EUR)', 'HKD (HKD)', and 'USD (USD)'. The 'Total' and 'Total (USD)' columns show the initial balances.

| Account Name | Description | Total | Total (USD) |
|---------------------|------------------|---------------|-------------|
| Assets | Assets | \$10,000.00 | \$10,000.00 |
| European Bank (EUR) | | €10,000.00 | \$0.00 |
| HongKong Bank (HKD) | | HK\$10,000.00 | \$0.00 |
| US Bank (USD) | | \$10,000.00 | \$10,000.00 |
| Equity | Equity | \$10,000.00 | \$10,000.00 |
| Opening Balances | Opening Balances | \$10,000.00 | \$10,000.00 |
| EUR (EUR) | | €10,000.00 | \$0.00 |
| HKD (HKD) | | HK\$10,000.00 | \$0.00 |
| USD (USD) | | \$10,000.00 | \$10,000.00 |

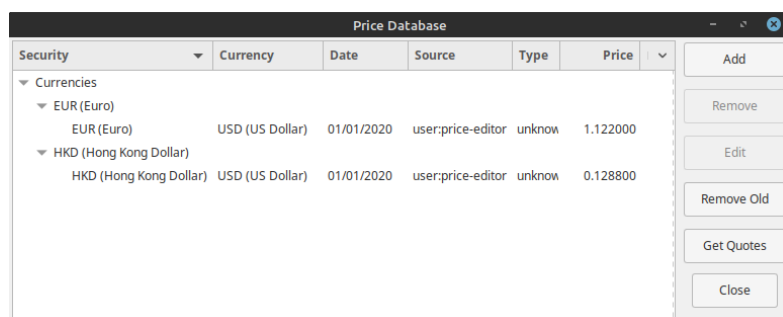
Grand Total: Net Assets: \$10,000.00 Profits: \$0.00

File demo_trading_001.gnucash opened. Last modified on Tue, Apr 2...

多货币场景下银行科目的初始结构。

到2020年1月1日，价格数据库（〔工具〕→〔价格数据库〕）中已录入了下列价格记录，并按此将外币科目余额换算为人民币：

12.19.



The screenshot shows the 'Price Database' window. It displays a table of exchange rates for EUR and HKD against the USD (US Dollar) as of 01/01/2020. The 'Price' column shows the exchange rate for each currency.

| Security | Currency | Date | Source | Type | Price |
|------------------------|-----------------|------------|-------------------|---------|----------|
| EUR (Euro) | USD (US Dollar) | 01/01/2020 | user:price-editor | unknown | 1.122000 |
| HKD (Hong Kong Dollar) | USD (US Dollar) | 01/01/2020 | user:price-editor | unknown | 0.128800 |

Buttons: Add, Remove, Edit, Remove Old, Get Quotes, Close

设定人民币、港元和欧元间初始汇率后的价格数据库窗口。

如图设定货币汇率后，现在〔科目〕标签页会在〔总计(CNY)〕列中以人民币显示外币科目的金额。

12.20.

| Account Name | Description | Total | Total (USD) |
|---------------------|------------------|-------------------------|-----------------|
| Assets | Assets | \$22,508.00 | \$22,508.00 |
| European Bank (EUR) | | €10,000.00 | \$11,220.00 |
| HongKong Bank (HKD) | | HK\$10,000.00 | \$1,288.00 |
| US Bank (USD) | | \$10,000.00 | \$10,000.00 |
| Equity | Equity | \$22,508.00 | \$22,508.00 |
| Opening Balances | Opening Balances | \$22,508.00 | \$22,508.00 |
| EUR (EUR) | | €10,000.00 | \$11,220.00 |
| HKD (HKD) | | HK\$10,000.00 | \$1,288.00 |
| USD (USD) | | \$10,000.00 | \$10,000.00 |
| \$, Grand Total: | | Net Assets: \$22,508.00 | Profits: \$0.00 |

File demo_trading_001.gnucash opened. Last modified on Tue, Apr 2...

设定各外币相对人民币汇率后的〔科目〕标签页。

兑换资金为外币

假定我们想在2020年1月1日通过香港银行科目花20,000港元在香港当地买些东西。香港银行科目中的流动资金不够，需要再从中国内地银行科目额外兑换10,200港元。兑汇时的汇率为 $1 \text{ CNY} = 1.0753 \text{ HKD}$ ，即需要花9,486元人民币购汇，同时银行收取40元人民币的购汇手续费。相关交易由〔资金转账〕对话框创建，如下所示：

12.21.

Transfer Funds

Basic Information

Amount: 1,309.64

Date: 01/02/2020

Num:

Description: Transfer USD to Hong Kong Bank Account

Notes:

Memo:

Debit Account

Assets

- European Bank (EUR)
- HongKong Bank (HKD)**

Currency: HKD (Hong Kong Dollar)

☐ Show Income/Expense

Credit Account

US Bank (USD)

- Equity
- Imbalance-HKD

Currency: USD (US Dollar)

☐ Show Income/Expense

Currency Transfer

☐ Exchange Rate: 7 + 10122751/12839607 1 USD = 7.788400 HKD

☒ Debit Amount: 10,200.00 1 HKD = 0.128396 USD

从中国内地银行科目到香港银行科目的转账

〔借方金额〕已由港元指定，而交易〔金额〕以人民币进行转账。无法在对话框中录入手续费。关闭对话框后，新创建的交易在中国银行科目中显示如下：

在〔资金转账〕对话框中录入的〔金额〕就是将要从选择的〔贷方科目〕转账到选择的〔借方科目〕的金额。如果选中〔借方金额〕单选框并输入外币金额，则记录的交易金额是精确的。然而如果选中〔汇率〕单选框并输入相应的汇率，则实际金额往往由于舍入误差而记录得不精确。

12.22.

The screenshot shows the GnuCash interface with a transaction split. The main window title is "demo_trading_001.gnucash - US Bank (USD) - GnuCash". The menu bar includes File, Edit, View, Transaction, Actions, Business, Reports, Tools, Windows, and Help. The toolbar has buttons for Save, Close, Duplicate, Delete, Enter, Cancel, Blank, Split, Jump, Schedule, and Transfer. The Accounts pane shows US Bank (USD), HongKong Bank (HKD), USD, and HKD. The transaction table is as follows:

| Date | Num | Description | | Tot Debit | Tot Credit | Balance |
|------------|-----|---|----------------------------|-----------------|-----------------|-----------|
| 01/01/2020 | | Opening Balance US Bank account | -- Split Transaction -- | n 10,000.00 | | 10,000.00 |
| 01/02/2020 | | Transfer of USD to Hong Kong Bank Account | | | 1,309.64 | 8,690.36 |
| | | | Assets:HongKong Bank (HKD) | n HK\$10,200.00 | | |
| | | | Trading:CURRENCY:USD | n 1,309.64 | | |
| | | | Assets:US Bank (USD) | | n 1,309.64 | |
| | | | Trading:CURRENCY:HKD | | n HK\$10,200.00 | |
| 04/05/2020 | | | | | | |

Present: \$8,690.36 Future: \$8,690.36 Cleared: \$0.00 Reconciled: \$0.00 Projected Minimum: \$8,690.36 Sort By: Standard Order

Transfer of USD to Hong Kong Bank Account

关闭转账对话框后，从中国银行科目转账至香港银行科目的交易。

在中国银行科目账簿中展开该交易后，可以向其中添加分录，用于记录手续费，如下所示：

12.23.

The screenshot shows the GnuCash interface with a transaction split. The main window title is "demo_trading_001.gnucash - US Bank (USD) - GnuCash". The menu bar includes File, Edit, View, Transaction, Actions, Business, Reports, Tools, Windows, and Help. The toolbar has buttons for Save, Close, Duplicate, Delete, Enter, Cancel, Blank, Split, Jump, Schedule, and Transfer. The Accounts pane shows US Bank (USD), HongKong Bank (HKD), USD, and HKD. The transaction table is as follows:

| Date | Num | Description | | Tot Debit | Tot Credit | Balance |
|------------|-----|---|------------------------------|-----------------|-----------------|----------|
| 01/02/2020 | | Transfer of USD to Hong Kong Bank Account | -- Split Transaction -- | n | 1,309.64 | 8,690.36 |
| 01/02/2020 | | Transfer of USD to Hong Kong Bank Account | | | 1,349.64 | 8,650.36 |
| | | | Assets:HongKong Bank (HKD) | n HK\$10,200.00 | | |
| | | | Trading:CURRENCY:USD | n 1,309.64 | | |
| | | | Expenses:Bank Service Charge | n 40.00 | | |
| | | | Assets:US Bank (USD) | | n 1,309.64 | |
| | | | Trading:CURRENCY:HKD | | n HK\$10,200.00 | |
| | | | Assets:US Bank (USD) | | n 40.00 | |
| 04/05/2020 | | | | | | |

Present: \$8,650.36 Future: \$8,650.36 Cleared: \$0.00 Reconciled: \$0.00 Projected Minimum: \$8,650.36 Sort By: Standard Order

Transfer of USD to Hong Kong Bank Account

添加兑汇手续费分录后，从中国银行科目转账至香港银行科目的交易。

回到〔科目〕标签页，可以看到由科目间转账交易引起的金额变动：

12.24.

The screenshot shows the GnuCash Accounts window for a demo trading account. The interface includes a menu bar (File, Edit, View, Actions, Business, Reports, Tools, Windows, Help) and a toolbar with icons for Save, Close, Open, Edit, New, and Delete. Below the toolbar, there are tabs for different accounts: Accounts, US Bank (USD), HongKong Bank (HKD), USD, and HKD. The main table displays the account hierarchy and balances.

| Account Name | Description | Total | Total (USD) |
|---------------------|------------------|-------------------------|-------------------|
| Assets | Assets | \$22,463.96 | \$22,463.96 |
| European Bank (EUR) | | €10,000.00 | \$11,220.00 |
| HongKong Bank (HKD) | | HK\$20,200.00 | \$2,593.60 |
| US Bank (USD) | | \$8,650.36 | \$8,650.36 |
| Equity | Equity | \$22,503.96 | \$22,503.96 |
| Opening Balances | Opening Balances | \$22,503.96 | \$22,503.96 |
| EUR (EUR) | | €10,000.00 | \$11,220.00 |
| HKD (HKD) | | HK\$10,000.00 | \$1,283.96 |
| USD (USD) | | \$10,000.00 | \$10,000.00 |
| Expenses | Expenses | \$40.00 | \$40.00 |
| Imbalance-HKD | | HK\$0.00 | \$0.00 |
| Imbalance-USD | | \$0.00 | \$0.00 |
| Income | Income | \$0.00 | \$0.00 |
| Liabilities | Liabilities | \$0.00 | \$0.00 |
| Trading | | \$0.00 | \$0.00 |
| CURRENCY | | \$0.00 | \$0.00 |
| HKD | | -HK\$10,200.00 | -\$1,309.64 |
| USD | | \$1,309.64 | \$1,309.64 |
| \$, Grand Total: | | Net Assets: \$22,463.96 | Profits: -\$40.00 |

完成从中国银行科目到香港银行科目的资金转账后，科目标签页中各科目的余额。

资产和所有者权益科目显示的金额已经根据2020年2月1日资金转账交易中录入的汇率进行了换算。

GnuCash已经自动创建了贸易科目，转账金额也已录入到相应的贸易科目中。其中的交易条目与录入该交易之前已经存在的期初余额均无关联。也就是，贸易科目只反映在其本身创建之后产生的损益。

到现在，还没有产生贸易损益。人民币和港元的贸易科目余额换算为人民币后是相等的。港元贸易科目有一条借方交易（余额为负），而人民币贸易科目有一条贷方交易（余额为正）。货币子科目的净余额为，也即贸易父科目自身的余额，仍然为零，表明目前没有记录到由货币兑换导致的损益。

原本去香港的购物之行未能成真。一个月后，即2020年3月1日，因其它安排，又需向中国银行科目汇集资金。为此执行一次从香港银行转回到中国银行，总额10,200.00港元的转账。进行这次转账时的汇率为 $1 \text{ CNY} = 7.7933 \text{ HKD}$ ，兑汇手续费20.00元人民币。记录这次兑汇的交易如下：

12.25.

demo_trading_001.gnucash - HongKong Bank (HKD) - GnuCash

File Edit View Transaction Actions Business Reports Tools Windows Help

Save Close Duplicate Delete Enter Cancel Blank Split Jump Schedule

Accounts US Bank (USD) HongKong Bank (HKD) USD HKD

| | Action | Memo | Account | R | Debit | Credit |
|------------|--------|------------------------------|-----------------------------------|---|------------|------------|
| 01/01/2020 | | Opening Balance Hong Kong E | Equity:Opening Balances:HKD (HKD) | n | 10,000.00 | 10,000.00 |
| 01/02/2020 | | Transfer of USD to Hong Kong | -- Split Transaction -- | n | 10,200.00 | 20,200.00 |
| 01/03/2020 | | Transfer HKD to US Bank Acco | | | 10,200.00 | 10,000.00 |
| | | | Assets:US Bank (USD) | n | \$1,308.82 | |
| | | | Trading:CURRENCY:HKD | n | 10,200.00 | |
| | | | Expenses:Bank Service Charge | n | \$20.00 | |
| | | | Assets:HongKong Bank (HKD) | n | | 10,200.00 |
| | | | Trading:CURRENCY:USD | n | | \$1,308.82 |
| | | | Assets:US Bank (USD) | n | | \$20.00 |
| | Action | | Account | R | Debit | Credit |
| 04/05/2020 | | | | n | | |

Present: HK\$10,000.00 Future: HK\$10,000.00 Cleared: HK\$0.00 Reconcil... HK\$0.00 Projecte... HK\$10,000.00 Sort By: Standard Order ^

Enter a description of the split

该交易从香港银行科目向中国银行科目转账10,200.00港元。

在资金转回中国银行科目后，价格数据库记录如下：

12.26.

Price Database

| Security | Currency | Date | Source | Type | Price |
|--------------------------|-----------------|------------|-------------------|--------|----------|
| ▼ Currencies | | | | | |
| ▶ EUR (Euro) | | | | | |
| ▼ HKD (Hong Kong Dollar) | | | | | |
| HKD (Hong Kong Dollar) | USD (US Dollar) | 01/03/2020 | user:price | unknow | 0.128315 |
| HKD (Hong Kong Dollar) | USD (US Dollar) | 01/02/2020 | user:xfer-dialog | unknow | 0.128396 |
| HKD (Hong Kong Dollar) | USD (US Dollar) | 01/01/2020 | user:price-editor | unknow | 0.128800 |

Add Remove Edit Remove Old Get Quotes Close

价格数据库显示了将资金从港元转回人民币时使用的汇率。

转回资金后，科目标签页中各科目余额如下：

12.27.

| Account Name | Description | Total | Total (USD) |
|---------------------|------------------|-------------------------|-------------------|
| Assets | Assets | \$22,442.33 | \$22,442.33 |
| European Bank (EUR) | | €10,000.00 | \$11,220.00 |
| HongKong Bank (HKD) | | HK\$10,000.00 | \$1,283.15 |
| US Bank (USD) | | \$9,939.18 | \$9,939.18 |
| Equity | Equity | \$22,503.15 | \$22,503.15 |
| Opening Balances | Opening Balances | \$22,503.15 | \$22,503.15 |
| EUR (EUR) | | €10,000.00 | \$11,220.00 |
| HKD (HKD) | | HK\$10,000.00 | \$1,283.15 |
| USD (USD) | | \$10,000.00 | \$10,000.00 |
| Expenses | Expenses | \$60.00 | \$60.00 |
| Imbalance-HKD | | HK\$0.00 | \$0.00 |
| Imbalance-USD | | \$0.00 | \$0.00 |
| Income | Income | \$0.00 | \$0.00 |
| Liabilities | Liabilities | \$0.00 | \$0.00 |
| Trading | | \$0.82 | \$0.82 |
| CURRENCY | | \$0.82 | \$0.82 |
| HKD | | HK\$0.00 | \$0.00 |
| USD | | \$0.82 | \$0.82 |
| \$, Grand Total: | | Net Assets: \$22,442.33 | Profits: -\$60.00 |

转回资金后各科目的状态。

现在，资产和所有者权益余额反映了按将资金转回中国银行科目时的汇率换算得到的科目总额。贸易和货币占位符科目表明，货币兑换导致了些许已实现亏损，总计0.82元人民币。相关的兑汇手续费体现在费用总额中。在现实生活中，这并非有利可图的做法，但仍然可以通过这个例子理解贸易科目的实际用处。

由贸易科目汇总得到的损益为已实现损益。

最终，在上述一系列交易后，资产负债表指出产生了贸易亏损：

12.28.

| demo_trading_001.gnucash - Balance Sheet - GnuCash | | | | |
|--|---------------|-------------|---------------|--------------------|
| File Edit View Actions Business Reports Tools Windows Help | | | | |
| Save | Close | Back | Forward | Reload |
| Stop | Options | Save Config | | |
| HongKong Bank (HKD) | USD | HKD | Balance Sheet | |
| Balance Sheet 31/12/2020 | | | | |
| Assets | | | | |
| Assets | | | | \$0.00 |
| European Bank (EUR) | €10,000.00 | \$11,220.00 | | |
| HongKong Bank (HKD) | HK\$10,000.00 | \$1,283.15 | | |
| US Bank (USD) | | \$9,939.18 | | |
| Imbalance-HKD | | | | \$0.00 |
| Imbalance-USD | | | | \$0.00 |
| Total Assets | | | | \$22,442.33 |
| Liabilities | | | | |
| Liabilities | | | | \$0.00 |
| Credit Card | | | \$0.00 | |
| Total Liabilities | | | | \$0.00 |
| Equity | | | | |
| Equity | | | | \$0.00 |
| Opening Balances | | | \$0.00 | |
| EUR (EUR) | €10,000.00 | \$11,220.00 | | |
| HKD (HKD) | HK\$10,000.00 | \$1,283.15 | | |
| USD (USD) | | \$10,000.00 | | |
| Retained Losses | | | | \$60.00 |
| Trading Losses | | | | \$0.82 |
| Total Equity | | | | \$22,442.33 |
| Total Liabilities & Equity | | | | \$22,442.33 |

经这些交易后，资产负债表反映产生了贸易亏损。 Balance Sheet after these transactions reflecting the trading loss incurred.

部分 III. 高级功能

目录

| | |
|---|-----|
| 13. Business Features | 242 |
| Introduction to Business Features | 242 |
| Business Setup | 242 |
| Account Setup | 242 |
| Setting up Sales Tax Tables | 244 |
| Enter Company Information | 246 |
| Setting Business Preferences | 248 |
| Setting Billing Terms | 248 |
| Accounts Receivable | 250 |
| System Components | 250 |
| Customers | 251 |
| Invoices | 252 |
| Customer Jobs | 256 |
| Process Payment | 257 |
| Changing the Invoice Appearance | 257 |
| Accounts Payable | 260 |
| System Components | 260 |
| Vendors | 261 |
| Bills | 263 |
| Vendor Jobs | 266 |
| Process Payment | 267 |
| Payroll | 267 |
| Basic Concepts | 267 |
| Account Setup | 268 |
| Payroll Protocol Sample | 269 |
| Business Payroll Example | 271 |
| 14. Budgets | 275 |
| Basic Concepts | 275 |
| Terminology | 275 |
| Creating a Budget | 276 |
| Choose Which Accounts To Budget For | 276 |
| Choosing a Budget Period | 276 |
| Getting Started | 276 |
| Entering Budget Values | 277 |
| Budget Reporting | 277 |
| Budgeted Income Statement | 277 |
| Budgeted Balance Sheet | 277 |
| 15. Other Assets | 278 |
| General Concepts | 278 |
| Other Assets Described | 278 |

| | |
|---|-----|
| Current Assets | 278 |
| Long-term (Fixed) Assets | 278 |
| Current Assets | 279 |
| Short-term Receivables | 279 |
| Reimbursable Expenses | 279 |
| Travel Advances | 281 |
| Prepaid Premiums or Prepaid Rent | 281 |
| Suspense or Wash Accounts | 282 |
| Short or Long-term Assets | 283 |
| Long-term (Fixed) Assets | 284 |
| Land | 284 |
| Buildings | 284 |
| Leasehold Improvements | 285 |
| Vehicles or Equipment | 285 |
| Intangibles | 285 |
| 16. Depreciation | 286 |
| Basic Concepts | 286 |
| Personal Finances | 286 |
| Business | 287 |
| Estimating Valuation | 288 |
| Depreciation Schemes | 288 |
| Account Setup | 290 |
| Example | 291 |
| 17. Python Bindings | 295 |
| 18. Importing Business Data | 296 |
| Importing Bills and Invoices | 296 |
| General | 296 |
| The format of the import file | 296 |
| Import your data | 298 |
| Feedback and statistics | 299 |
| What could go wrong? | 300 |
| Not supported invoice functionality | 302 |
| Importing Customers and Vendors | 302 |
| General | 302 |
| The format of the import file | 302 |
| Import your data | 304 |
| Feedback and statistics | 305 |
| What could go wrong? | 306 |
| Not supported customer/vendor functionality | 306 |

第 13 章 Business Features

Introduction to Business Features

The accounting needs of a business are quite different from that of a person. Businesses have customers that owe money, vendors which are owed money, employee payroll, more complex tax laws, etc. GnuCash offers business oriented features to facilitate these needs.

Accounts Receivable (A/R) are used by businesses to record sales for which they are not immediately paid. This is represented on the balance sheet as an asset, because the expectation is that you will receive payment soon.

Accounts Payable (A/P) record bills that businesses have received, but may not pay until later. This is represented on the balance sheet as a liability because you will have to pay for them.

A/R and A/P accounts are used primarily when you have a lot of bills and receipts flowing in and out, and do not want to lose track of them just because you do not pay or get paid right away. For most home users, A/R and A/P are too complicated to be worthwhile.

Business Setup

To set up GnuCash to handle accounts receivable or accounts payable for a company, these preliminary steps must be done.

- Build an appropriate Account Hierarchy.
- Set up Sales Tax Tables.
- Enter the company information in GnuCash.
- Set Business Preferences.
- Set up Billing Terms.

Account Setup

There are many different ways to set up a business account hierarchy. You can start with the Business Accounts setup which is available from the New Account Hierarchy assistant, or you could build one manually. To access the prebuilt Business Accounts,

start GnuCash and click on [File] → [New File] and proceed until you see the list of available accounts, select Business Accounts.

The prebuilt Business Account hierarchy will not meet your needs exactly. You will need make adjustments to the hierarchy so that it will meet your particular needs. It should be close enough, however, that it is recommended you begin with it.

To use GnuCash' s integrated accounts receivable system, you must first set up a special account (usually a sub-account under Assets) to hold transactions for receivables. This account must be defined with account type A/Receivable. GnuCash will use this account to place transactions related to the integrated accounts receivable system.

To use GnuCash' s integrated accounts payable system, you must first set up an account (usually a sub-account under Liabilities) to hold transactions for payables. This account must be defined with account type A/Payable. GnuCash will use this account to place transactions related to the integrated accounts payable system.

Basic A/R and A/P Account Hierarchy:

- Assets
 - Accounts Receivable
 - Checking
- Expenses
 - ...(as required)
- Income
 - Sales
- Liabilities
 - Accounts Payable
 - Tax
 - Tax on Purchases
 - Tax on Sales

You need to add additional accounts to this hierarchy for it to be useful.

You do not need to create an individual A/R account for each customer. GnuCash keeps track of customers internally and provides per-customer reports based on the internal tracking. The same applies to A/P and vendors.

If you deal with customers in more than one currency you will need a separate [Accounts Receivable] account for each currency.

If you deal with vendors in more than one currency you will need a separate [Accounts Payable] account for each currency.

Transactions involving an Accounts Receivable or Accounts Payable account should not be added, changed or deleted in any way other than by using Post/Unpost Bill/Invoice/Voucher or Process Payment!

Setting up Sales Tax Tables

Sales Tax Tables can be used to determine the tax for customer invoices (or vendor bills).

A tax table entry can be assigned to an invoice line or bill line.

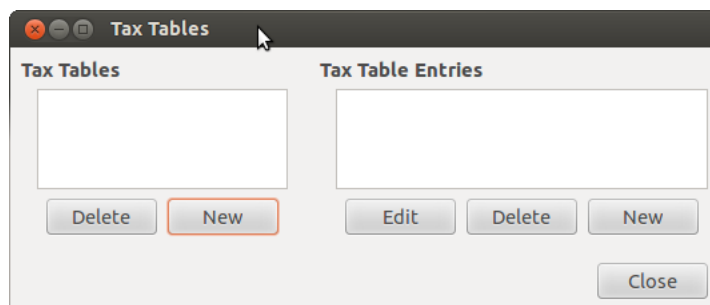
Set up distinct tax tables for customers and vendors.

The default invoice tax table entry can be assigned to each customer and the default bill tax table entry can be assigned to each vendor.

The default tax table entry for new customers or new vendors can be specified in the Book Options window which can be accessed by [File] → [Properties] → [Business tab.]

Sales Tax Tables are maintained using the Sales Tax Table editor which is accessed via menu [Business] → [Sales Tax Table] .

13.1. Sales Tax Tables Editor



13.2. New Sales Tax Table Entry

- [Name] This is the tax table name.
- [Type] Either [Percent %] or [Value \$] .
- [Value] This is the percentage or value depending on [Type] .
- [Account] This is the account to which tax will be posted. For tax collected from customers, this should probably be a Liability account as it must be paid to the government. For tax paid to vendors, if tax laws allow tax paid to vendors to offset tax collected from customers, this should probably also be a Liability account (even though it will usually have a debit balance) so that the net tax owed to the government can be easily observed.

If you set up Tax on Purchases and Tax on Sales as subaccounts of Liabilities:Tax then the net tax will be rolled up and can be seen in the GnuCash Accounts tab.

If unsure about tax law requirements, get professional advice.

The following charts illustrate sample tax tables and may be used as starting points to determine the setup appropriate for a particular jurisdiction.

13.1. Sample Tax Table Entries for EU country (e.g. 21% / 6% / 0% Belgium, 20% / 5% / 0% UK etc.) (2017)

| Tax Table | Tax Table Entries [Asset/ Liability] | Percentage or Amount |
|--------------------|---|----------------------|
| Standard VAT Sales | VAT:Sales:Standard [L] | 21% |

| Tax Table | Tax Table Entries [Asset/ Liability] | Percentage or Amount |
|--------------------------|---|----------------------|
| Reduced VAT Sales | VAT:Sales:Reduced [L] | 6% |
| Zero-Rated VAT Sales | VAT:Sales:Zero [L] | 0% |
| EC Sales | VAT:Sales:EC [L] | 21% |
| | VAT:Sales:Reverse EC [L] | -21% |
| Standard VAT Purchases | VAT:Purchases:Standard [A] | 21% |
| Reduced VAT Purchases | VAT:Purchases:Reduced [A] | 6% |
| Zero-Rated VAT Purchases | VAT:Purchases:Zero [A] | 0% |

13.2. Sample Tax Table Entries for Australia (2017)

| Tax Table | Tax Table Entries [Asset/ Liability] | Percentage or Amount |
|------------------------|---|----------------------|
| Standard GST Sales | GST:Sales:Standard [L] | 10% |
| GST-free Sales | GST:Sales:Zero [L] | 0% |
| Standard GST Purchases | GST:Purchases:Standard [A] | 10% |
| GST-free Purchases | GST:Purchases:Zero [A] | 0% |

13.3. Sample Tax Table Entries for Cook County, Illinois (2017)

| Tax Table | Tax Table Entries [Asset/ Liability] | Percentage or Amount |
|---------------------|---|----------------------|
| Chicago Sales Taxes | Taxes:Sales:State [L] | 6.25% |
| | Taxes:Sales:City [L] | 1.25% |
| | Taxes:Sales:County [L] | 1.75% |
| | Taxes:Sales:Region [L] | 1% |

Enter Company Information

After you have built the account structure and defined your tax tables, designate your company in the GnuCash file. To do this, select the [Business] tab in the [Book Options] window, which can be accessed from [File] → [Properties] .

13.3. Entering Company Information

The screenshot shows the 'Book Options' dialog box with the 'Business' tab selected. The dialog contains several input fields for company information, including Company Name, Company Address, Company Contact Person, Company Phone Number, Company Fax Number, Company Email Address, Company Website URL, and Company ID. It also features dropdown menus for Default Customer TaxTable and Default Vendor TaxTable, both set to 'None'. A Fancy Date Format section includes a dropdown set to 'UK (31/12/2001)', radio buttons for Months (Number selected), and a checked checkbox for Years (Include Century). A text field shows the format '%d/%m/%Y' and a sample date '29/08/2015'. A 'Reset defaults' button is located below the date format section. At the bottom of the dialog are buttons for 'Help', 'Cancel', 'Apply', and 'OK'.

Book Options

Accounts Budgeting **Business** Counters

Company Name:

Company Address:

Company Contact Person:

Company Phone Number:

Company Fax Number:

Company Email Address:

Company Website URL:

Company ID:

Default Customer TaxTable: None

Default Vendor TaxTable: None

Fancy Date Format: UK (31/12/2001)

Months: ☒ Number ☐ Abbreviation ☐ Name

Years: ☒ Include Century

Format: %d/%m/%Y

Sample: 29/08/2015

Reset defaults

Help Cancel Apply OK

Here you can:

- Enter the name of your company along with contact information such as your phone number, fax number, e-mail address and website URL.

- Enter your company's tax payer id number in the [Company ID] field.
- Select default tax tables applicable to your most common customers and vendors.

Setting Business Preferences

Set options on the Business tab of the GnuCash preferences, which is accessed via [Edit] → [Preferences] ([GnuCash] → [Preferences] on macOS). See GnuCash Manual chapter 10.3.3 Business Book Options Tab.

Setting Billing Terms

Billing Terms can be used to determine the payment due date and be a guide for determining discount for early payment of invoices (or vendor bills).

As of GnuCash 2.6.7, Billing Terms are only partially supported. Date due is calculated using the Billing Terms but discount amount is not.

Discount for early invoice payment is not implemented. There are 2 ways this may be done, although neither is recommended, and professional advice should be used to confirm that regulations are being complied with:

- After creating and posting a payment which pays the invoice in full, manually edit the payment transaction (usually strongly discouraged) and split the payment to reduce it by the amount of the discount and create a compensating split in an income (discount) account.
- Alternatively, after creating and posting a payment for the discounted amount, create a credit note for the discount using a specific negative sales income (discount) account for the transfer account.

You can specify the billing terms on each invoice/bill. Invoice billing terms will default from the customer billing terms. Bill billing terms will default from the vendor billing terms.

Billing Terms are maintained using the Billing Terms Editor which is accessed via menu [Business] → [Billing Terms Editor] .

13.4. Billing Terms Editor

Terms

Net-15
Net-30

Delete New

Term Definition

Description: Net 30 Days

Type: Days

Due Days: 30

Discount Days: 10

Discount %: 1.00

Edit Close

13.5. New Billing Term

GnuCash

New Billing Term

Name: Net-60

Term Definition

Description: Net 60 Days

Type: Days

Due Days: 60

Discount Days: 10

Discount %: 1.00

Cancel OK

- [Name] The internal name of the billing term. For some examples of billing term names and descriptions see https://wiki.gnucash.org/wiki/Payment_Terms.
- [Description] The description of the billing term, printed on invoices
- There are 2 types of billing terms, with different information to be entered
 - Type Days
 - [Due Days] The invoice or bill is due to be paid within this number of days after the post date

- [Discount Days] The number of days after the post date during which a discount will be applied for early payment
- [Discount %] The percentage discount applied for early payment
- Type Proximo
 - [Due Day] The day of the month bills are due
 - [Discount Day] The last day of the month for the early payment discount
 - [Discount %] The discount percentage applied if paid early
 - [Cutoff Day] The cutoff day for applying bills to the next month. After the cutoff, bills are applied to the following month. Negative values count backwards from the end of the month.

Accounts Receivable

Accounts Receivable (or A/R) refers to products or services provided by your company for which payment has not yet been received.

System Components

Transactions generated by the A/R system are recorded within the Accounts Receivable account. You should not work directly with this account. Instead, you will work with the four integrated GnuCash A/R application components available through the [Business] → [Customer] sub-menu. These four components are:

- [Customers] are people or companies to whom you sell products or services on credit.
- [Invoices] represent the physical invoice you send to a customer to request payment. This invoice contains an itemized list of things you sold.

In addition, GnuCash also has support for [Credit Notes] which represent the inverse of Invoices. A credit note is usually handed to a customer to correct items that were incorrectly invoiced or returned.

Both document types will be set up using the same menu items. Credit notes were introduced starting with GnuCash stable release 2.6.0.

- [Jobs] (optional) is where you register Customer Jobs. Jobs are a mechanism by which you can group multiple invoices to a particular customer.
- [Process Payments] is used to register payments you received from a customer.

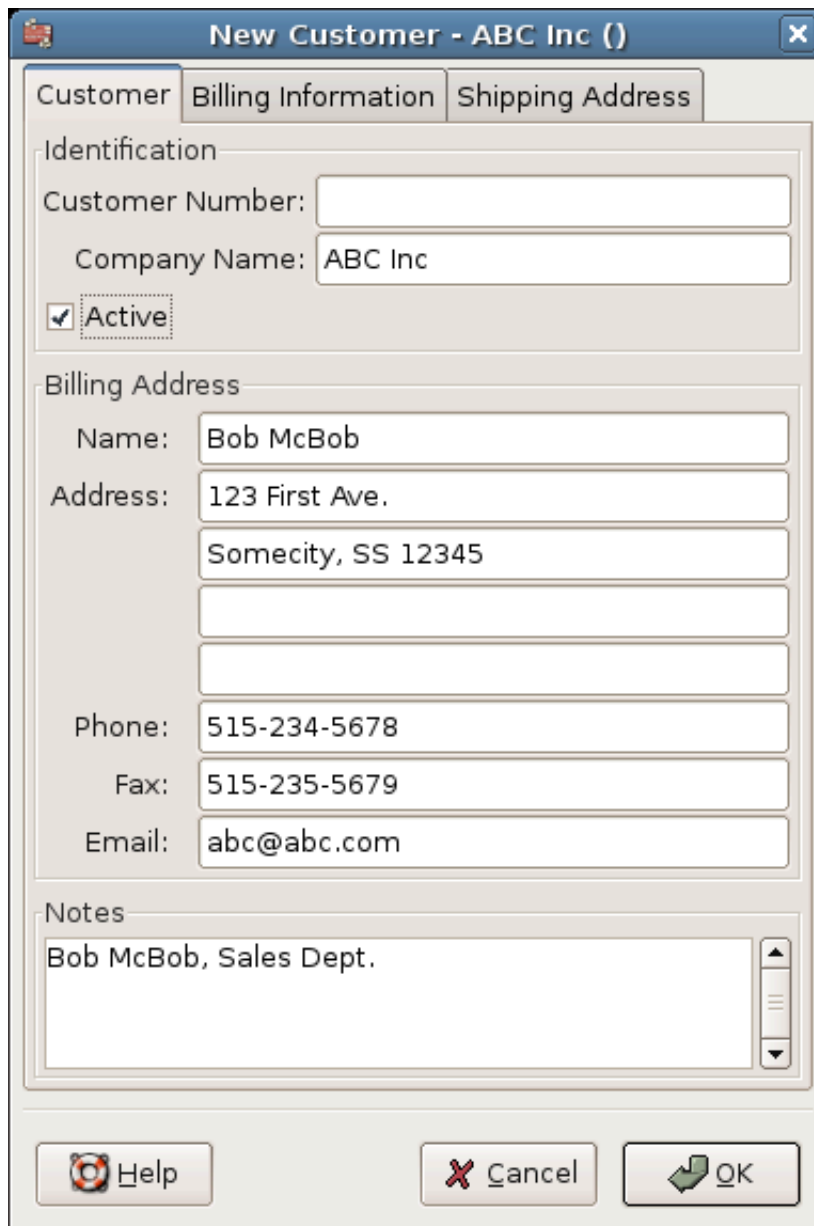
Customers

Customers are people or companies to whom you sell goods or services. They must be registered within the A/R system.

New

To register a new customer, enter the menu [Business] → [Customer] → [New Customer] . Fill in customer information, such as Company Name, Address, Phone, Fax, etc.

13.6. New Customer Registration Window



New Customer - ABC Inc ()

Customer | Billing Information | Shipping Address

Identification

Customer Number:

Company Name:

☒ Active

Billing Address

Name:

Address:

Phone:

Fax:

Email:

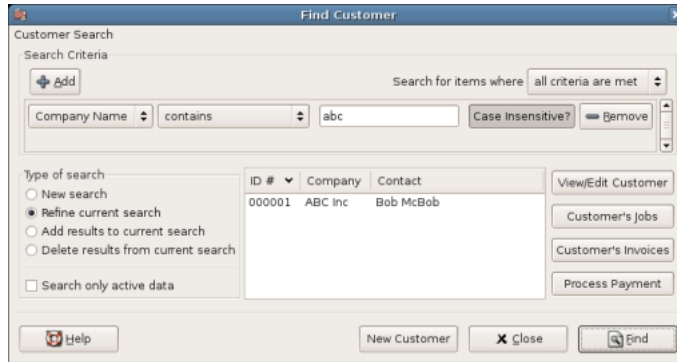
Notes

Help Cancel OK

Find and Edit

To search for an existing customer, use the [Business] → [Customer] → [Find Customer] window. You select a customer to [View/Edit] from the results of the search. This window is also used to look up customers when creating invoices and processing payments.

13.7. Find Customer Window



If many customers match the search criteria you provide, the search can be refined by running an additional search within the current results. The current result set is searched when the [Refine Current Search] radio button is selected. In fact, GnuCash selects this option for you after you run the initial search.

If the customer you are searching for does not match the supplied search criteria, change the search criteria, click the [New Search] radio button and then the [Find] button. The relevant step is the [New Search] selection. If the customer is not in the result of the original search, and you only search within this set, the customer cannot be found, regardless of new search criteria.

To return a list of all registered active customers, set the search criterion to [matches regex] , and place a single dot "." in the text field area. Make sure [Search only active data] is checked, then click Find. The regular expression "." means to match anything.

Invoices

An invoice is the paperwork you send to a customer to request payment for products or services rendered. GnuCash can generate and track invoices.

A credit note is the paperwork you send to a customer to correct products or services rendered that were incorrectly invoiced. GnuCash can generate and track credit notes via the same menu entries as invoices.

This section applies to both invoices and credit notes. In order to avoid repeating everything twice and to keep the text easier to read it will refer only to invoices. You can apply it equally to credit notes. Only where the behavior of credit notes differs from invoices this will be explicitly mentioned.

New

To send an invoice to a customer you must first create the new document. To create an invoice use [Business] → [Customer] → [New Invoice] . The New Invoice window must be filled in appropriately.

13.8. Creating a New Invoice

The 'New Invoice' window is a standard application dialog. It features a title bar with the text 'New Invoice' and a close button (X). The main content area is divided into three sections. The first section, 'Invoice Information', contains two fields: 'Invoice ID' and 'Date Opened', which is set to '04/19/06'. The second section, 'Billing Information', contains four fields: 'Customer' (set to 'ABC Inc'), 'Job', 'Billing ID' (set to 'ABC Purchase Order # 12988'), and 'Terms' (set to '30 Days'). The 'Customer' and 'Job' fields have 'Select...' buttons next to them. The third section, 'Notes', contains a text area with the text 'Your personal notes go here. Notes do not appear on printed invoices.' At the bottom of the window are three buttons: 'Help' (with a lifebuoy icon), 'Cancel' (with a red X icon), and 'OK' (with a green checkmark icon).

When you click the OK button, the [Edit Invoice] window opens.

Edit

From the Edit Invoice window you can enter an itemized list of goods and services you sold on this invoice in a manner similar to how the account register works. For credit notes you enter an itemized list of goods and services you refunded instead.

13.9. Edit Invoice Window

When you have finished entering all the items, you can [Post] and print the invoice.

Post

When you finish editing an invoice and are ready to print, you must [Post] the invoice. The invoice does not have to be posted immediately. In fact, you should only post an invoice when you are ready to print it. Posting an invoice places the transactions in an accounts receivable account.

13.10. Post Invoice Window

Find

To find an existing invoice, use the [Business] → [Customer] → [Find Invoice] menu item. From the results of the search, you can select an invoice to edit or view.

Before you can edit a posted invoice, you will need to [Unpost] it.

One of the design goals in GnuCash's Account Receivable system was to allow different processes to get to the same state, so you can reach an invoice from different directions based on the way you think about the problem:

- You can search for the customer first, then list their invoices.
- You can search for invoices by number or by company name.
- You can list invoices associated with a customer job.

Print

After you post an invoice, you should print it and send it to your customer. To print an invoice use [File] → [Print Invoice] menu item.

13.11. Invoice Print Output

| Invoice #000001 | | | | | | | |
|---------------------------------------|-------------|-------------|----------|---------------------|----------|---------|-----------------|
| Invoice Date: 04/19/06 | | | | My Company | | | |
| Due Date: 05/19/06 | | | | 9876 Second St. | | | |
| | | | | Othercity, SS 12345 | | | |
| | | | | April 19, 2006 | | | |
| ABC Inc | | | | | | | |
| 123 First Ave. | | | | | | | |
| Somecity, SS 12345 | | | | | | | |
| Reference: ABC Purchase Order # 12988 | | | | | | | |
| Terms: 30 Days - same as cash | | | | | | | |
| Date | Description | Charge Type | Quantity | Unit Price | Discount | Taxable | Total |
| 04/19/06 | Nails | Material | 1,000.00 | \$0.10 | 0.00% | | \$100.00 |
| 04/19/06 | Hammer | Material | 1.00 | \$500.00 | 5.00% | | \$475.00 |
| Subtotal | | | | | | | \$575.00 |
| Tax | | | | | | | \$0.00 |
| Amount Due | | | | | | | \$575.00 |
| Thank you for your patronage | | | | | | | |

You can modify the appearance of the invoice, IE: add a company logo, etc.

To do so, see the “Changing the Invoice Appearance” 一节.

Invoices can also be printed from the main window by selecting [Reports] → [Business Reports] → [Printable Invoice] from the main menu. The resulting report window states that no valid invoice is selected. To select the invoice to print:

1. Use the Options Toolbar button or select [Edit] → [Report Options] from the main menu.

2. Select the [General] tab of the report options dialog.
3. Click the Select button next to the [Invoice Number] field.
4. Search for the invoice as usual.

You can also print invoices from within the Process Payment dialog. See the “Process Payment” 一節 for instructions on how to do so.

Assign Starting Invoice Number

By default, GnuCash starts with invoice number 1 and increments from there. You can manually type an invoice number into the text box each time you create an invoice, but this gets tiring and sooner or later leads to duplicate numbers.

You can change the starting invoice number if it is important you. Use [File] → [Properties] , access the [Counters] tab, change the [Invoice number] value to be one less than your desired starting invoice number and click the OK button or the Apply button.

Customer Jobs

Customer Jobs are used to group multiple invoices and credit notes to the same customer. Use of the Customer Jobs feature is optional. The feature is useful when you have multiple [jobs] for the same customer, and would like to view all the invoices and credit notes related to a single job.

13.12. New Customer Job



New Job - ABC Deck Construction ()

Job Information

Job Number

Job Name

Owner Information

Customer

Billing ID

Active

☒ Job Active

To use customer jobs, you must create them using the [Business] → [Customer] → [New Job] menu item. You will see the [New Job] window.

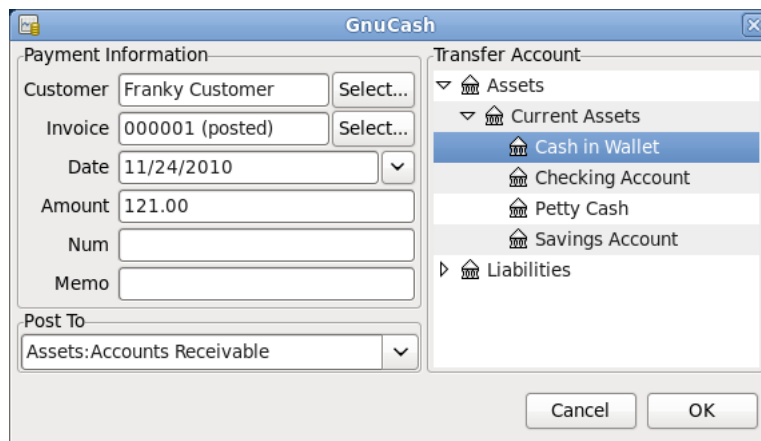
To edit an existing customer job, use the [Business] → [Customer] → [Find Job] menu item. Select the desired job in the search results, and click the [View/Edit Job] button.

To select from the invoices and credit notes associated with a given job, use [Business] → [Customer] → [Find Job] menu item. Select the desired job in the search results and click the [View Invoices] button. A window listing invoices and credit notes associated with this job appears. Select an invoice or credit note and click the [View Invoice] button to open an invoice editor in the main application window.

Process Payment

Eventually, you will receive payment from your customers for outstanding invoices. To register these payments, use the Process Payment application found in [Business] → [Customer] → [Process Payment] .

13.13. Process Payment From Customer



Changing the Invoice Appearance

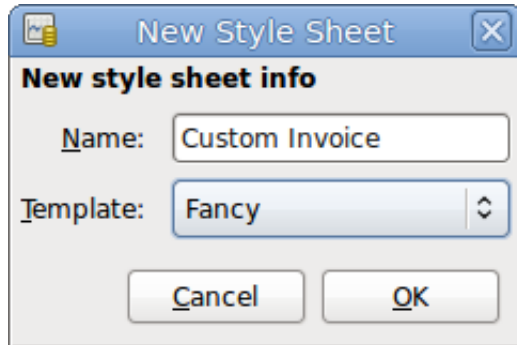
The default Invoice style, as shown in “Print” 一节, is fairly barren. The default invoice style leaves the top part of the form blank, so you can print on company letterhead paper. There are some things you can do to change invoice appearance.

Use [File] → [Properties] to enter your Company information in the [Business] tab of the [Book Options] window. Some of the entered information is printed on the right side of invoices.

To add a customized background, heading banner or logo to invoices, modify the invoice style sheets. To do this, go to [Edit] → [Style Sheets] and select the New

button in the [Select HTML Style Sheet] window that will appear. You will then see a window like this:

13.14. New Style Sheet Window

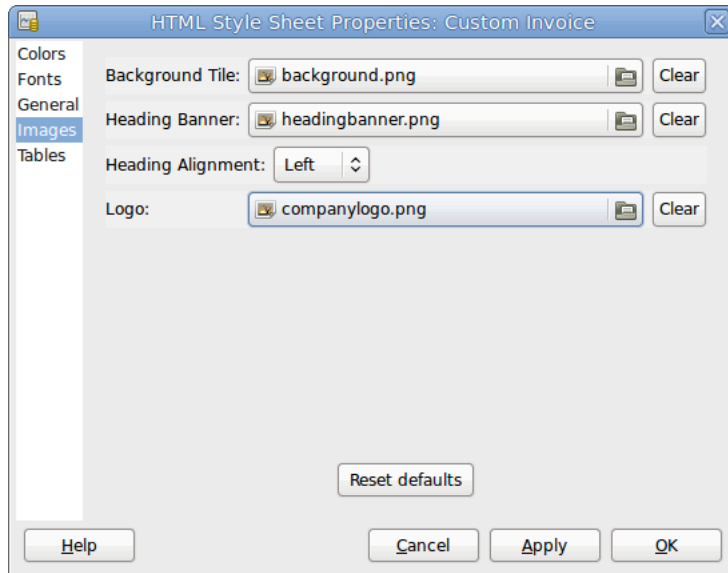


Give a [Name] to the new style sheet (e.g. [Custom Invoice]) and select the [Fancy] [Template]. When you click the OK button, the [HTML Style Sheet Properties] window is displayed. This window presents you five sections listed in the left pane: [Colors], [Fonts], [General], [Images], and [Tables]. The [Colors] section allows you to change the colors of various items of the invoice. The [Fonts] section lets you set fonts type and dimensions. The [General] section allows you to set the [Preparer] and [Prepared for] information, and to [Enable Links]. The [Images] section allows you to import graphics into the style sheet. The [Tables] section allows you to adjust the spacing around the tables which make up the invoice.

To include a company logo, banner heading and background image, use your favorite graphics application such as The Gimp or Apache OpenOffice/LibreOffice Draw to save the images in either GIF or PNG format. Then import them into the style sheet using the [Images] section described above.

Below is an example that imports all three types of images.

13.15. HTML Style Sheet Example Window



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

The images are placed in the invoice as follows. The [Background Tile] is tiled to become the background image, the [Heading Banner] goes to above the invoice text, and the [Logo] is placed in the upper left corner of the invoice to the left of the Heading Banner. You will probably have to try a few different sized images until you get the invoices to print nicely. Some sizing suggestions are that the Logo should be 1 square cm (~0.5 inch), and the Heading Banner should be 15 cm (~6 inches) wide and 1 cm (~0.5 inch) tall.

With the style sheet configured, when you print the invoice, you select the style sheet to use from the [Options] menu. Below is the resultant invoice after applying the style sheet demonstrated above.

13.16. HTML Style Sheets Example Output

My Company Logo **Heading Banner**

Invoice #000001
 Invoice Date: 04/19/06
 Due Date: 05/19/06

ABC Inc
 123 First Ave.
 Somecity, SS 12345

My Company
 9876 Second St.
 Othercity, SS 12345
 April 23, 2006

Reference: ABC Purchase Order # 12988
 Terms: 30 Days - same as cash

| Date | Description | Charge Type | Quantity | Unit Price | Discount | Taxable | Total |
|-------------------|-------------|-------------|----------|------------|----------|---------|-----------------|
| 04/19/06 | Nails | Material | 1,000.00 | \$0.10 | 0.00% | | \$100.00 |
| 04/19/06 | Hammer | Material | 1.00 | \$500.00 | 5.00% | | \$475.00 |
| Subtotal | | | | | | | \$575.00 |
| Tax | | | | | | | \$0.00 |
| Amount Due | | | | | | | \$575.00 |

Thank you for your patronage

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

Accounts Payable

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

System Components

GnuCash has an integrated accounts payable system. The transactions generated by the A/P system are placed within the Accounts Payable account, as a record of what occurs. Generally you do not directly work with this account but use the four integrated GnuCash A/P application components. The A/P components are available from the [Business] → [Vendor] sub-menu. These A/P components are:

- [Vendors] are people or companies from which you buy products or services on credit.
- [Bills] represent the physical bills vendors send to request payment from you. A bill contains an itemized list of things you purchased.

In addition, GnuCash also has support for [Credit Notes] which represent the inverse of Bills. A credit note is usually received from a vendor to correct items that were erroneously billed or returned.

Both document types will be set up using the same menu items.

- [Jobs] (optional) is where you register Vendor Jobs. Jobs are mechanism by which you can group multiple bills from a particular vendor.

- [Process Payments] is where you register payments to a vendor to whom you owe money.

The following sections introduce the individual Accounts Payable application components.

Vendors

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

New

To register a new vendor, select the [Business] → [Vendor] → [New Vendor] menu item. Fill in general information about the vendor, such as Company Name, Address, Phone, Fax, etc. Below is a list of the other options:

This is what the New Vendor registration window looks like:

13.17. New Vendor Registration Window

New Vendor - BCA Corp ()

Vendor | **Payment Information**

Identification

Vendor Number:

Company Name:

☒ Active

Payment Address

Name:

Address:

Phone:

Fax:

Email:

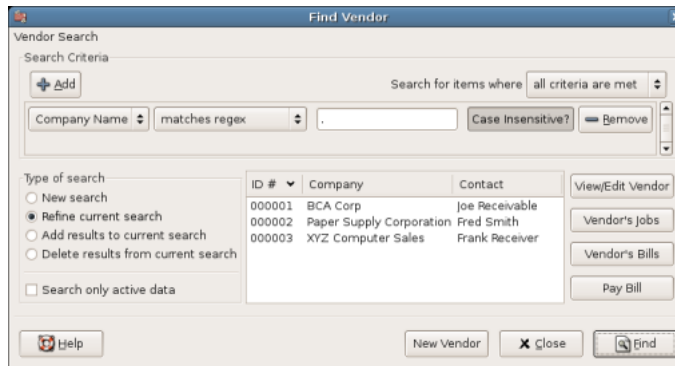
Notes

Help Cancel OK

Find and Edit

To search for an existing vendor, use the [Business] → [Vendor] → [Find Vendor] window. You select a vendor to View/Edit from the results of the search. This window is also used to look up a vendor when entering bills and processing payments.

13.18. Find Vendor Window



If many vendors match the search criteria you provide, the search can be refined by running an additional search within the current results. The current result set is searched when the [Refine Current Search] radio button is selected. In fact, GnuCash selects this option for you after you run the initial search.

If the vendor you are searching for does not match the supplied search criteria, change the search criteria, click the [New Search] radio button and then the Find button. The relevant step is the [New Search] selection. If the vendor is not in the result of the original search, and you only search within this set, the vendor cannot be found, regardless of new search criteria.

To return a list of all registered active vendors, set the search criterion to [matches regex] , and place a single dot “.” in the text field area. Make sure [Search only active data] is checked, then click Find. The regular expression “.” means to match anything.

Bills

A bill is a request for payment you receive from a vendor. GnuCash can track bills.

A credit note is the document you receive from a vendor to correct products or services rendered that you were incorrectly charged for on a bill. GnuCash can generate and track credit notes via the same menu entries as bills.

This section applies to both bills and credit notes. In order to avoid repeating everything twice and to keep the text easier to read it will refer only to bills. You can apply it equally to credit notes. Only where the behaviour of credit notes differs from bills this will be explicitly mentioned.

New

When you receive a bill from a vendor and want to enter it into GnuCash, you must create a new bill. To create a new bill use the [Business] → [Vendor] → [New Bill] menu item, and fill in the resulting window appropriately.

13.19. New Bill Registration Window

The 'New Bill' window is a standard dialog box with a title bar. It is organized into four main sections, each with a label and a group box:

- Invoice Information:** Contains an 'Invoice ID' text field and a 'Date Opened' date field set to '05/18/06' with a dropdown arrow.
- Billing Information:** Contains a 'Vendor' field with the text 'Paper Supply Corporatic' and a 'Select...' button, a 'Job' field with a 'Select...' button, a 'Billing ID' text field, and a 'Terms' dropdown menu currently set to 'None'.
- Default Chargeback Project:** Contains a 'Customer' field with a 'Select...' button and a 'Job' field.
- Notes:** A large text area with the placeholder text 'Additional notes about the bill go here.' and vertical scrollbars.

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

When you click the OK button the [Edit Bill] window opens.

Edit

From the Edit Bill window you can enter an itemized list of goods and services you purchased, in a manner similar to how the account register works. For credit notes you enter an itemized list of goods and services the vendor refunded instead.

13.20. Edit Bill Window

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

Post

When you finish editing a bill, you should [Post] the bill. You do not have to post the bill, you can close it and return to it later. You have to post the bill eventually. Posting a bill places its transactions into an accounts payable account. The Post Bill window appears and asks you to enter information:

13.21. Post Bill Window

Find

To find an existing bill, use the [Business] → [Vendor] → [Find Bill] menu item. From the results of the search, you can select a bill to edit, or view.

Before you can edit a posted bill, you will need to [Unpost] it.

There are other ways to access an existing bill. These are similar to accessing invoices for your customers. See “Find” 一節 for more information.

Vendor Jobs

Vendor Jobs are used to group multiple bills and credit notes from a single vendor. Use of the vendor jobs feature is optional. The feature is useful when you have multiple [jobs] for the same vendor, and would like to view all the bills and credit notes for a single job.

To use vendor jobs, you must create them using the [Business] → [Vendor] → [New Job] menu item. You will see the [New Job] window.

13.22. New Vendor Job

The screenshot shows a dialog box titled "New Job - 2006 Computer Upgrades". It has a standard Windows-style title bar with a close button (X). The dialog is divided into three main sections. The first section, "Job Information", contains two text boxes: "Job Number" (empty) and "Job Name" (containing the text "2006 Computer Upgrades"). The second section, "Owner Information", contains a "Vendor" text box (containing "XYZ Computer Sales") and a "Select..." button, followed by a "Billing ID" text box (empty). The third section, "Active", contains a checkbox labeled "Job Active" which is checked. At the bottom of the dialog are three buttons: "Help" (with a question mark icon), "Cancel" (with a red X icon), and "OK" (with a green checkmark icon).

To edit an existing vendor job, use the [Business] → [Vendor] → [Find Job] menu item. Select the desired job in the search results, and click the [View/Edit Job] button.

To select from the bills associated with a given job, use [Business] → [Vendor] → [Find Job] menu item. Select the desired job in the search results and click the [View Invoices] button. A window listing bills and credit notes associated with this job appears. Select a bill or credit note and click the [View Invoice] button to open a bill editor in the main application window.

Process Payment

Eventually, you need to pay your bills. To do so, use the Process Payment application found in {Business} → {Vendor} → {Process Payment} .

Below is the GnuCash Accounts Payable payment window.

13.23. Process Payment To Vendor

The screenshot shows the 'GnuCash' application window for processing a payment to a vendor. The 'Payment Information' section on the left includes fields for Vendor (ABC Inc.), Bill (000001 (posted)), Date (11/24/2010), Amount (60.50), Num, Memo, and Post To (Liabilities:Accounts Payable). The 'Transfer Account' section on the right displays a tree view of accounts, with 'Checking Account' selected under 'Current Assets'. The window has 'Cancel' and 'OK' buttons at the bottom.

Payroll

Payroll is a financial record of wages, net pay, paid vacations, and deductions for an employee. This section demonstrates how to track payroll using GnuCash.

Basic Concepts

Payroll is a financial record of wages, net pay, paid vacations, and deductions for an employee. Basically, anything that relates to giving money or benefits to an employee. Payroll is one of the more complex tasks in accounting, because there are many different accounts, people, and agencies involved in paying salaries.

Payroll is typically accounted for as an expense. Sometimes accountants “store” some payroll expenses in a short term liability account. This is useful for things such as payroll taxes, which may be paid at a different time than the employee. The employee might get paid biweekly, while taxes are paid quarterly. This chapter presents a methodology which expenses payroll immediately for salaries, but stores taxes in liability accounts.

GnuCash does not have an integrated payroll system. While you can track payroll expenses in GnuCash, the calculation of taxes and deductions has to be done outside of GnuCash.

Account Setup

Local tax law must be considered when setting up accounts. Because there are many different ways payroll taxes are handled throughout the world, this section presents a very simple structure. From this, you should be able to adapt your particular payroll deductions setup.

Assume that you must pay 2 taxes, Tax1 and Tax2, and that each has an employee contribution and an employer contribution.

The employee's salary and these two taxes are expense accounts. The tax components are liability accounts. The tax liability accounts are where you accumulate the taxes withheld for all of your employees. The taxes are later paid to the appropriate government agency.

Simple Payroll Account Layout:

```
-Assets
--Checking
-Liabilities
--Tax1 (short term "storage" account)
--Tax2 (short term "storage" account)
-Expenses
--Salaries
--Tax1
--Tax2
```

Resist the temptation to create per-employee sub-accounts to track individual salaries. Creating a sub-account for each employee leads to unmanageably large lists of accounts. Imagine the account structure after a few years of employees coming and going. It is much simpler to keep all of your employees' payroll records within a single account (`Expenses:Salaries` for example) and use reports to view per-employee information.

The Transaction report can be used to sort and total by description or memo (but not by part of them).

To report salary transactions for a specific employee where the employee name or code is entered in the transaction description or memo, use the `[Find Transaction] assistant ([Edit] → [Find])` to select the transactions, and then report on them using `[Reports] → [Account Report]`. Further formatting or analysis may be done by copying and pasting the report into a spreadsheet. See Find Transaction [<https://www.gnucash.org/docs/v5/C/gnucash-manual/tool-find.html?tool-find-txn>] in the GnuCash Manual.

Payroll Protocol Sample

GnuCash does not have an integrated payroll system. GnuCash can track your payroll expenses, but you need to develop a payroll protocol and perform the calculations outside of GnuCash, in a spreadsheet for example. In this section, one such protocol is presented. You can use the sample protocol as a model.

Step 1: Deductions list

The first step to the payroll protocol is to create a list of all the possible taxes and deductions for each employee. Each entry should include definitions and formulas for calculating each value. Once the protocol is established it needs to be changed only when payroll laws or tax rates change.

In the proposed scenario, such a list would look like this:

- E_GROSS_SALARY - Employee gross salary
- E_TAX1 - Employee contribution to tax1 (X% of E_GROSS_SALARY)
- E_TAX2 - Employee contribution to tax2 (X% of E_GROSS_SALARY)
- C_TAX1 - Company contribution to tax1 (X% of E_GROSS_SALARY)
- C_TAX2 - Company contribution to tax2 (X% of E_GROSS_SALARY)

The employee's net salary (E_NET_SALARY) is defined as $E_GROSS_SALARY - E_TAX1 - E_TAX2$ and need not be placed in this list since it is composed of items that already exist.

Place the actual formulas for calculating each deduction in this list. Sometimes these formulas are quite complex, and sometimes they simply say "look it up in table XYZ of the tax codes".

Notice that you can calculate some interesting values using the above definitions. One such value is the total cost to the company: $E_GROSS_SALARY + C_TAX1 + C_TAX2$.

Step 2: Create the Transaction Map

When you record payroll in GnuCash, do so with a single split transaction. This split transaction populates the appropriate expense and liability accounts. If you need to look the payroll details at a later time, open the split transaction.

With the deductions list from above, an employee split transaction map can be generated. Each of the items in the list is mapped to a GnuCash account.

13.4. Payroll Transaction Map

| Account | Increase | Decrease |
|-----------------|----------|--------------|
| Assets:Checking | | E_NET_SALARY |

| Account | Increase | Decrease |
|-------------------|----------------|----------|
| Expenses:Salaries | E_GROSS_SALARY | |
| Liabilities:Tax1 | | E_TAX1 |
| Liabilities:Tax2 | | E_TAX2 |
| Expenses:Tax1 | C_TAX1 | |
| Liabilities:Tax1 | | C_TAX1 |
| Expenses:Tax2 | C_TAX2 | |
| Liabilities:Tax2 | | C_TAX2 |

Note that the C_TAX1 and C_TAX2 components have entries in the both the liability and expense accounts. The company component of each tax is expensed at the time of payroll, but remains a liability until taxes are due.

Step 3: Pay the Employee

Go to the account from which the employee will be paid, for example your `Assets:Checking` account. Open a split transaction and enter the real values using the Transaction Map above as a guide. Repeat this for all employees.

This manual process is tedious, especially if you have a large number of employees.

One GnuCash tool you certainly want use when entering employee payroll is duplicate transaction (use the `Duplicate Toolbar` button). This saves you from having to enter all the transaction splits for each employee. You still need to change the amounts of money to match each employee's real payroll values, but you will not have to build the split for each employee.

If payroll transactions do not change significantly every pay period, you can also use the duplicate transaction feature to duplicate each employee's most recent payroll transaction for the current pay period. If you find you are doing so all the time, read about the `Schedule Transactions` feature and save even more time!

Step 4: Pay the Government

The final thing to do is to pay the taxes to the government. The liability accounts have been collecting the taxes for various government agencies, and periodically you need to send a check to the government to pay this charge. To do so, you simply enter a 2 account transaction in (for example) your checking account to pay off the tax liability.

The transaction is between the checking account and the liability account, no expense account is involved. The expense accounts are charged at the time the tax liability is recorded.

Business Payroll Example

Using the account setup seen previously, let's go through an example. Assume that there are 2 employees (E1 and E2) who each earn \$1000 per month gross salary. The employee contribution to Tax1 and Tax2 are 10% and 5% respectively. The company contribution to Tax1 and Tax2 are 15% and 10% each on top of the employee's gross salary.

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

13.24. Payroll Example: Initial Setup

| Account Name | Type | Total |
|-------------------|-----------|-------------|
| Assets | Asset | \$50,000.00 |
| Checking | Asset | \$50,000.00 |
| Equity | Equity | \$50,000.00 |
| Starting Balances | Equity | \$50,000.00 |
| Expenses | Expense | \$0.00 |
| Salaries | Expense | \$0.00 |
| Tax1 | Expense | \$0.00 |
| Tax2 | Expense | \$0.00 |
| Liabilities | Liability | \$0.00 |
| Tax1 | Liability | \$0.00 |
| Tax2 | Liability | \$0.00 |

\$, grand total: Assets: \$50,000.00 Profits: \$0.00

The deductions list for employee 1 are:

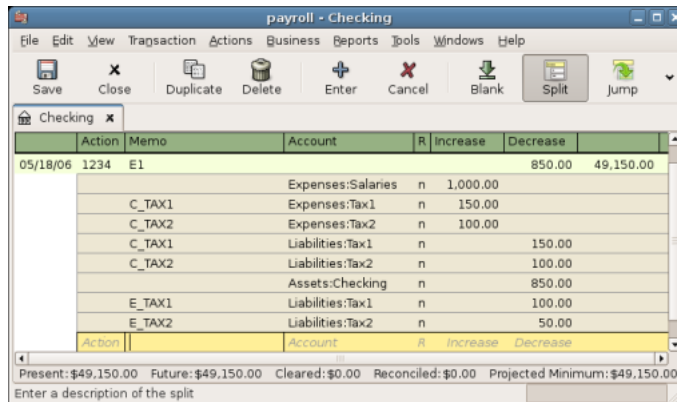
- E_GROSS_SALARY - Employee gross salary - \$1000
- E_TAX1 - Employee contribution to tax1 - \$100 (10% of E_GROSS_SALARY)
- E_TAX2 - Employee contribution to tax2 - \$50 (5% of E_GROSS_SALARY)
- C_TAX1 - Company contribution to tax1 - \$150 (15% of E_GROSS_SALARY)
- C_TAX2 - Company contribution to tax2 - \$100 (10% of E_GROSS_SALARY)

13.5. Payroll Transaction Map for Employee 1

| Account | Increase | Decrease |
|-------------------|----------------------------|----------------------|
| Assets:Checking | | \$850 (E_NET_SALARY) |
| Expenses:Salaries | \$1000 (E_GROSS_SALARY) | |
| Liabilities:Tax1 | | \$100 (E_TAX1) |
| Liabilities:Tax2 | | \$50 (E_TAX2) |
| Expenses:Tax1 | \$150 (C_TAX1) | |
| Liabilities:Tax1 | | \$150 (C_TAX1) |
| Expenses:Tax2 | \$100 (C_TAX2) | |
| Liabilities:Tax2 | | \$100 (C_TAX2) |

From the checking account, enter the split transaction for employee 1. It should look like this:

13.25. Payroll Example: Employee Split Transaction

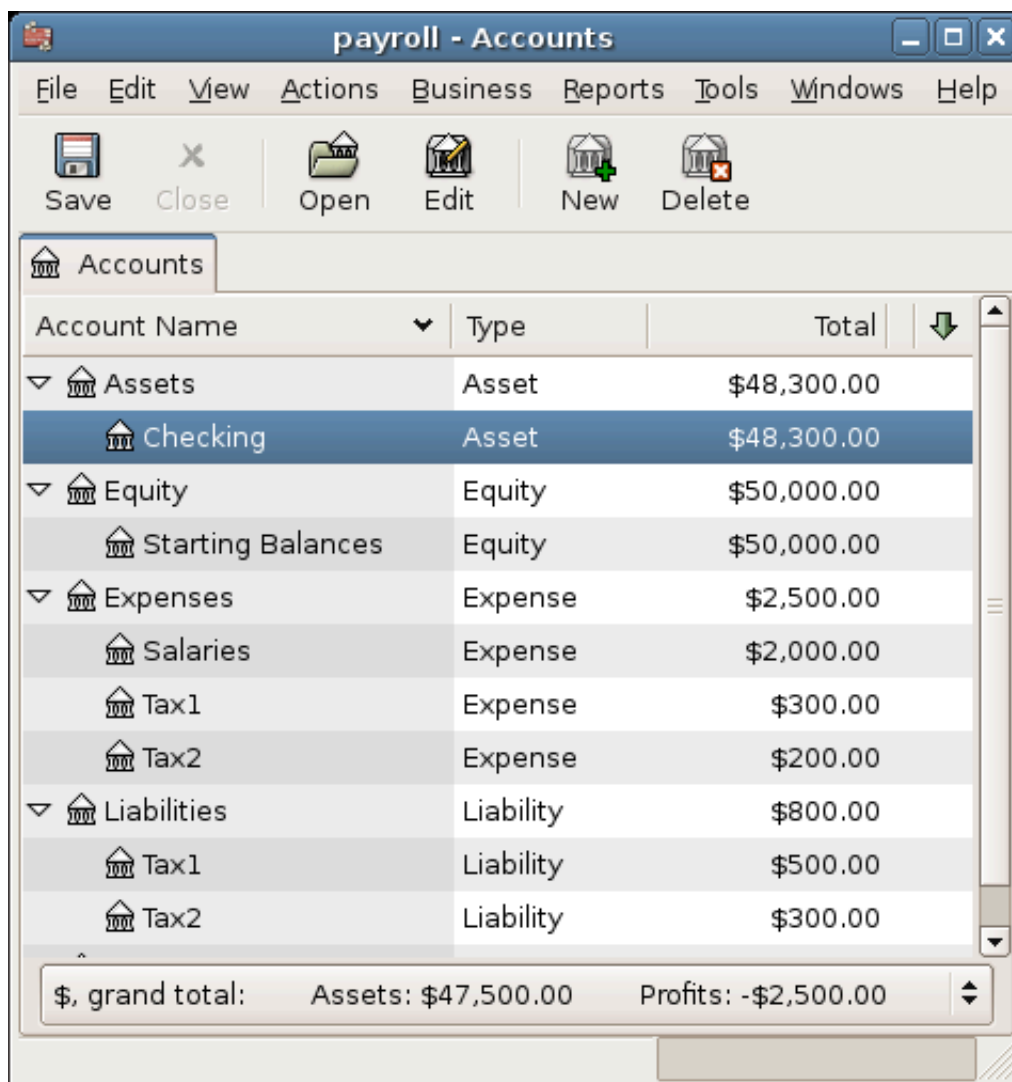


The Split Transaction for Employee 1

When paying employees, enter only the employee name in the Description area. If you decide to use GnuCash's check printing capabilities, the check is automatically made out to the correct employee. If you want to record other information in the transaction besides the employee name, use the Notes area, available when viewing the Register in double-line mode.

Repeat this for the second employee, which leaves the account hierarchy looking like this:

13.26. Payroll Example: Accounts After Salaries Paid

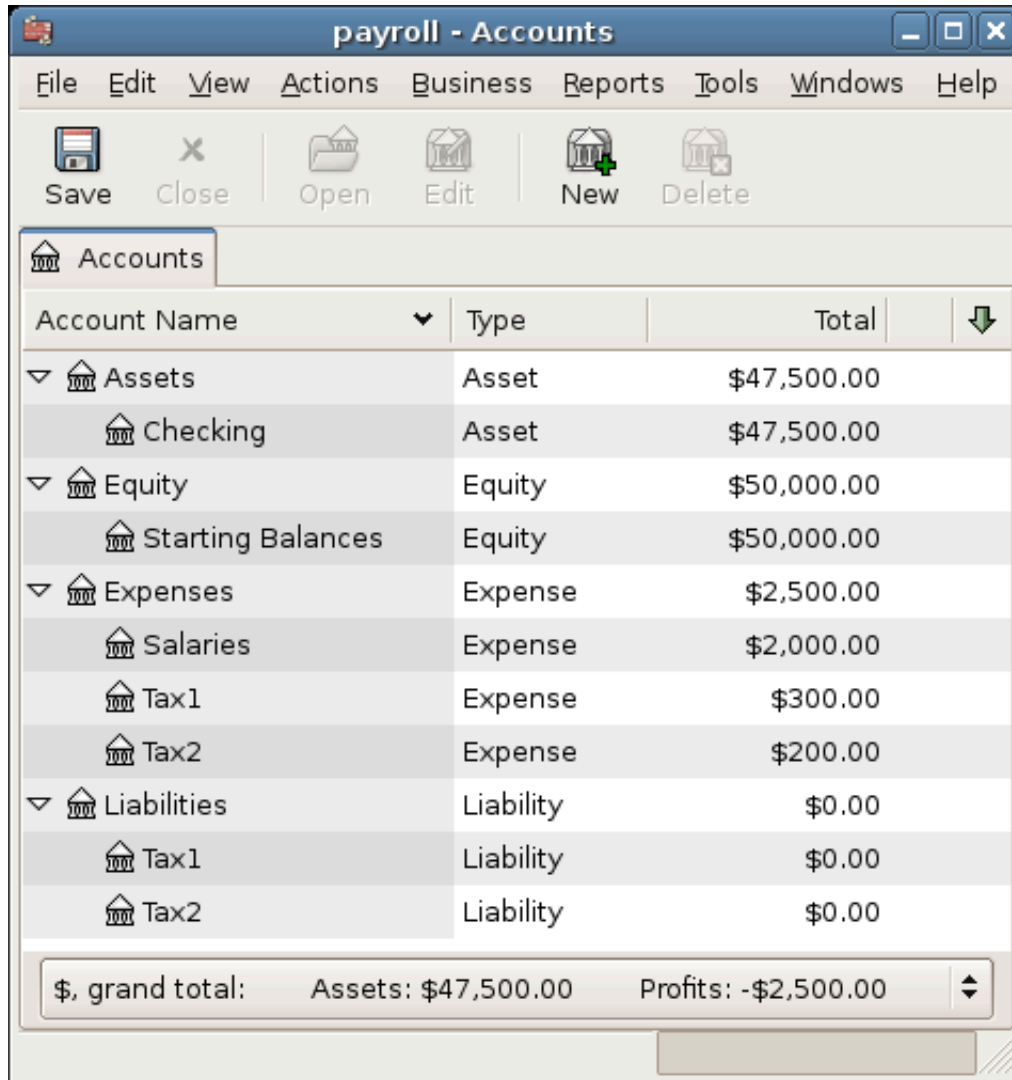


| Account Name | Type | Total |
|-------------------|-----------|---|
| Assets | Asset | \$48,300.00 |
| Checking | Asset | \$48,300.00 |
| Equity | Equity | \$50,000.00 |
| Starting Balances | Equity | \$50,000.00 |
| Expenses | Expense | \$2,500.00 |
| Salaries | Expense | \$2,000.00 |
| Tax1 | Expense | \$300.00 |
| Tax2 | Expense | \$200.00 |
| Liabilities | Liability | \$800.00 |
| Tax1 | Liability | \$500.00 |
| Tax2 | Liability | \$300.00 |
| \$, grand total: | | Assets: \$47,500.00 Profits: -\$2,500.00 |

Now, you will enter the company contributions. The `Liabilities:Tax1` and `Liabilities:Tax2` accounts track how much you must pay to the government for each tax type. When it is time to pay these agencies, you will make a transaction from the

checking account to these liability accounts. No expense accounts are involved. The main account will then appear like this:

13.27. Payroll Example: Accounts After Paying Government



The screenshot shows a software window titled "payroll - Accounts". It has a menu bar with "File", "Edit", "View", "Actions", "Business", "Reports", "Tools", "Windows", and "Help". Below the menu bar is a toolbar with icons for "Save", "Close", "Open", "Edit", "New", and "Delete". The main area displays a table of accounts. The table has columns for "Account Name", "Type", and "Total". The accounts are grouped by type: Assets, Equity, Expenses, and Liabilities. The "Assets" group includes "Checking" with a total of \$47,500.00. The "Equity" group includes "Starting Balances" with a total of \$50,000.00. The "Expenses" group includes "Salaries" (\$2,000.00), "Tax1" (\$300.00), and "Tax2" (\$200.00). The "Liabilities" group includes "Tax1" (\$0.00) and "Tax2" (\$0.00). At the bottom of the window, there is a summary bar showing "\$, grand total: Assets: \$47,500.00 Profits: -\$2,500.00".

| Account Name | Type | Total |
|-------------------|-----------|-------------|
| Assets | Asset | \$47,500.00 |
| Checking | Asset | \$47,500.00 |
| Equity | Equity | \$50,000.00 |
| Starting Balances | Equity | \$50,000.00 |
| Expenses | Expense | \$2,500.00 |
| Salaries | Expense | \$2,000.00 |
| Tax1 | Expense | \$300.00 |
| Tax2 | Expense | \$200.00 |
| Liabilities | Liability | \$0.00 |
| Tax1 | Liability | \$0.00 |
| Tax2 | Liability | \$0.00 |

\$, grand total: Assets: \$47,500.00 Profits: -\$2,500.00

第 14 章 Budgets

This chapter explains how to create and use budgets with GnuCash.

Basic Concepts

A budget is a tool for estimating expected income and expenses. You can use it to help you plan how you intend for your finances to change over a period of time, and to examine how your actual financial transactions for the period compare to your planned transactions.

The budgeting concept is quite general, so GnuCash offers a budgeting tool that is both simple and flexible. You, the user, have to decide how complex or simple you want to make your budget. This guide will help you make some of those decisions.

Terminology

There are a few helpful terms listed below that will be used to discuss budgeting.

- **Budget** - A financial plan describing the expected revenues and/or disbursements for a particular time period
- **Cash Budget** - A budget planning for expected cash receipts and cash disbursements. This type of budget tracks cash flow -- where your money comes from, where it goes, and, of course, how much.
- **Expense Budget** - A budget chiefly for planning what you spend your money on. This type of budget tracks your expenses. It is typically not concerned with things like appreciation or repayment of liabilities. However, it would account for interest charges. For example, if you buy \$100 worth of groceries with your credit card, you incur an \$100 expense for groceries, and a \$100 liability to your credit card company. When you pay the credit card bill for \$110, you are incurring an additional interest expense of \$10. An expense budget plans for the transaction of buying the groceries and paying the interest, but not the transaction of repaying the credit card company.
- **Capital Budget** - A budget that describes a plan for paying for a large future expense, often through a combination of saving and borrowing money. Note: Capital budgets can sometimes get quite complex because they can try to answer the question "Can we afford to do such-and-such?" by exploring various hypothetical scenarios that can involve hypothetical accounts.
- **Budget Period** - The period of time during which the plan is expected to take place. The most common budget periods are annual and monthly. Sometimes, you may budget for several consecutive periods at once, for convenience or for finer-grained planning. For example, an annual budget may include 12 monthly budget periods.

Creating a Budget

Even before you begin to make a budget, it's important to have given some thought to your account hierarchy. For example, if you want to budget a certain amount for your electric bill and a certain amount for your water bill, you can't have only an `Expenses:Utilities` account. Your accounts must be at least as specific as your budget.

Choose Which Accounts To Budget For

The first step in creating a budget is to decide what it is you want to plan for. This decision will affect which accounts you include in your budget. For example, if you are only interested in tracking your expenses, you may create an expense budget by only entering amounts for expense accounts. On the other hand, if you want to track all of your cash flow, you may create a cash flow budget by entering amounts for asset, liability, income and expense accounts.

Before you begin to create your budget, you need to make two decisions: What accounts do I want to budget for? and When do I want my budget to be for? You can always change your mind later, after you've created a budget, but you need to start with something.

As a rule of thumb, if you mostly care about what you spend your money on, you may want to make an expense report. If you're also concerned about having enough money in the right places at the right times, you may want to use a cash-flow budget.

Choosing a Budget Period

Before creating a budget you must also decide what period of time you want to plan for. The most common budget periods are monthly and annual. If you want your budget to plan for changes in financial patterns over time, then you should include multiple budget periods in your budget. For example, if you want to plan on having higher utility expenses in the winter than in the summer, then you might break your annual budget into 4 quarters or even 12 months, and budget a higher value for the winter periods than for the summer periods.

Getting Started

To create your first budget click on `[Actions] → [Budget] → [New Budget]`. You will immediately see a new budget with the default settings and no entries. Then click on the Options button. The most important options are the budget period and the number of periods. For the budget period, choose the beginning date and the smallest period

of time that you want to plan for. Then, for the number of periods, choose how many periods you want to plan for.

The budget page now shows a list of accounts with a column for each budget period. The date shown in the title of each column is the beginning of that budget period.

Entering Budget Values

Now, you must enter the budget values - the amounts that you expect the account balances to change during the budget period. There are two ways to enter budget values. The first way is to simply click on the cell and enter an amount.

If you have past transactions recorded in GnuCash, the second way is to let GnuCash estimate the budget values by looking at those transactions. First, select the accounts you want GnuCash to estimate. Then click on the `Estimate Toolbar` button. In the `[Estimate Budget Values]` dialog, select the date past which GnuCash should look for past transactions. GnuCash will start at that date and look forward for the duration of your budget. For example, if you are making an annual budget, and you select Jan. 1, 2005, GnuCash will look at all the transactions in that account from Jan. 1, 2005 through Dec. 31, 2005.

Budget Reporting

You've already done the hardest part - creating your budget. But now you want to know how your actual financial transactions compare to your plan. You need to run the Budget Report.

Click on `[Reports] → [Budget] → [Budget Report]`. For each account, the Budget Report will show the budgeted and the actual amounts in two adjacent columns for each period in the budget. If you have created multiple budgets, you can use the Budget Report Options to select which budget to use in the report.

Two other types of budget reports are commonly used in the small business setting. They are the Budgeted Income Statement and the Budgeted Balance Sheet.

Budgeted Income Statement

The budgeted income statement is similar to the income statement. Both show the revenues and expenses for a given period as well as the profit, which is the difference revenue - expenses. The income statement is based on historical data, but the budgeted income statement is based on the predictions made in the budget.

Budgeted Balance Sheet

The budgeted balance sheet is similar to the balance sheet. Both show the assets, liabilities, and equity. The difference is that the balance sheet is based on historical data, and the budgeted balance sheet is based on the predictions made in the budget.

第 15 章 Other Assets

General Concepts

This chapter presents many additional accounting treatments for frequently encountered business and less-frequently found personal activities that need recording in accounting books. The explanations below cover both the description and purpose of the activity, and they include also the usual accounting treatments (bookings or recordings) for these transactions.

These concepts have evolved over centuries of experience by those keeping accounting records and will help you maximize your record keeping' s utility and meaningfulness.

This section introduces categorization of assets in the balance sheet based on time or the asset' s useful life (current and long-term). Sometimes assets are also considered from the standpoint of their *liquidity* , which is regarded as how close or distant the asset is from being turned into cash. Near-cash assets are relatively quickly converted to cash (e.g., accounts receivable), while assets requiring rather a long time to convert to cash are considered to be relatively *fixed* in their non-cash state (e.g., heavy equipment, buildings, land). (Fixed does not mean they were repaired!)

You should find that current assets parallel those with more liquidity, while long-term and fixed assets are those with much less liquidity. Finally, below you will find a few assets that could be either current or long-term based on the nature of the facts constituting them.

Other Assets Described

Current Assets

Current Assets are those activities whose normal expected life would be one year or less. Such activities could be tracking reimbursable expenses, travel advances, short-term loans to a friend or family member, prepaid expenses, annual insurance premium amortization, and so on. The individual entity could have many other kinds of short term activities that reflect what it is doing. (These asset types are explained individually below.)

Long-term (Fixed) Assets

Long-term (Fixed) Assets are those activities whose normal expected life exceeds one or more years. This grouping covers both tangible and intangible assets. Examples of tangible assets are land, buildings, and vehicles (cars, trucks, construction

equipment, factory presses, etc.) Intangible assets include such things as patents, copy rights, goodwill, etc. Because the lives of some of these assets show wear and tear and deterioration in value over time, businesses and individuals can allow for that diminution in value by calculating depreciation on such assets. For example, land normally does not depreciate, but buildings do, as do equipment and vehicles. (These asset types are explained individually below.)

Current Assets

This section explains short-term receivables, reimbursable expenses, travel advances, prepaid premiums, prepaid rent, suspense or wash accounts.

Short-term Receivables

This kind of account is useful to reflect an agreement made with someone you trust. Suppose you lent someone \$500 and he agreed to repay you \$50 a month. If he paid on time, the loan you made would be paid off within a year, which is why it is classified as a short-term receivable. So you could record that loan initially in this account tree: `OtherAssets:Current Assets:LoanToJoe`. At the time you give him the money: your entry is debit (increase) `LoanToJoe` \$500 and credit (decrease) `Bank` \$500. Each time you receive Joe's payment you record \$50 debit (increase) to `Bank` and credit (decrease) `LoanToJoe`.

Don't become confused by the use of the word "Loan". "Loan-To" is the tipoff that you really have a receivable, that is, you will receive from Joe, the money you previously loaned. Until he actually pays the money owed you, you reflect his debt in your books by an account describing your expectation—you will receive the money owed you, hence the word "receivable".

Reimbursable Expenses

This kind of activity is one in which you spend your own money on behalf of someone else (your employer, perhaps) and later you receive repayment of what you spent. The case might be a business trip. The employer has a policy of covering (paying for) all authorized expenses. After the trip is over, the employee submits a report listing dates and amounts spent with receipts for all the expenditures. The employer reviews the report and pays for all items that it considers as having a valid business reason. (Normally, employees know in advance what the employer will reimburse, so only those items are recorded as a reimbursable expense on the employee's books.)

Because a business trip can involve different kinds of expenditures (air travel, lodging, transportation at the destination, etc.), different kinds of expenditures would be recorded in the one account as long as the expenditures all related to the same trip. In other words, if a second trip is made before the first is fully settled, a second account

for a different event could be set up. It would make sense to do this, if it would help to keep separate all the details of one trip from those of another. It is up to the person making the trip to decide how much trouble it would be to put separate trips in separate accounts or to put them all in the same account. The trip taker should remember that the account must be reconciled in order to know with certainty that all expenses have been reimbursed.

Recording the expenditures on the trip would be much the same. That is, if you paid trip expenses by cash you would debit (increase) the reimburseable expense account for the money paid in cash, because it is a receivable to you until it has been reimbursed to you. The credit offsetting your expenditure would decrease the account that shows the cash in your pocket or the account from which you drew the cash for the payment made. If you paid by credit card, the debit side would be the same as just described, but the credit would be an increase to the credit card company account on your books.

When you received your reimbursement, then the journal entry (or transaction) to record receipt of the funds from the employer would be: debit (increase) Bank for the check amount and credit (decrease) the reimbursable expense account for the check amount.

If it turns out that the reimbursable expense account is not zero balance after processing the employer's payment, then it means that there is a difference between you and the employer in handling the expense, which needs to be investigated. If the balance is a debit (a positive balance), your account has some money that was not reimbursed. If the balance is a credit (a negative balance), you were paid for more than what you recorded as due you. In both of those situations you should reconcile the difference between what you recorded and what was paid. That effort should disclose exactly what is causing the discrepancy. You will need to contact the employer's bookkeeper to know what was paid, if the reimbursement check was not accompanied by a detailed list of the items being paid you.

In the event the employer refused to reimburse you for an expenditure, that effectively makes it your expense. In that case, you would make this entry: debit (increase) your own Expense (appropriately named) and credit (decrease) the Reimbursable Expense account. That entry should result in a zero balance in the Reimbursable Expense account. If not, reconcile until you identify the difference.

Sometimes there are small differences that don't match an individual entry. In those cases divide the amount by 2 or by 9. If the unresolved amount is divisible by two, it suggests that both you and the employer entered the item in the same manner: both as debits or both as credits. If it is divisible by 9, then likely one of you transposed adjoining numbers; e.g., one entered 69 and the other entered 96. If the difference is divisible neither by 2 or by 9, then it could be that more than one error is present.

Travel Advances

These are very similar to Reimbursable Expenses. The difference is that someone gives you money first; you spend it, and then you give a report accounting for what you spent it on. The report is supported by invoices establishing who, what, where, when, and how much for each expenditure. In the Reimbursable Expense case, you spent your money first and later recovered it.

In the Travel Advance case when you receive the advance, you record on your books this entry: debit (increase) Bank for the travel advance amount received (say, \$500); credit (increase) the short-term liability Travel Advance (\$500). This is a liability, because you are not gifted with the money, but only loaned it for the purpose of having funds to spend when doing the employer's business.

Frequently, the way these monetary arrangements work is that at the beginning of for example a salesperson's employment, he or she receives the advance and monthly (or more frequently) turns in a report about who, what, where, when, and how much he spent. The money in the report is reimbursed if approved.

During the period after receiving the advance and before filing a request for reimbursement report, the salesperson can record his or her expenditures into the advance liability account. In that case, the balance in the account will show how much of the advance has not yet been spent (assuming the Travel Advance balance is a credit). If no mistakes have been made and all expenses are approved, then the sum of the unspent account balance and the reimbursing check amount will equal the original travel advance amount.

It makes sense for the salesperson to record the travel expenses to this advance account (and not to his or her own expense accounts), because the money is being spent on behalf of the employer, for the employer's authorized expenses. It is not the employee's own money, and therefore not his or her own expense.

When the salesperson receives the report reimbursement (say, \$350), he or she debits (increases) Bank, and credits (increases) again the Travel Advance liability account, assuming that previously he or she had been recording expenditures to the travel advance account. Tracking activity in this manner causes the account to always show the amount that is owed the employer.

See "Reimbursable Expenses" 一节 above for what to do if the employer does not accept an item the employee put on the travel advance reimbursement request report. The difference resolution effort is essentially the same for both types of accounts.

Prepaid Premiums or Prepaid Rent

Some types of expenses are usually billed as semi-annual or annual amounts. For example, the insurance industry will bill home insurance annually, while car insurance premiums can be annual or semi-annual. For those that pay an amount that covers

several months or a full year, the proper accounting treatment is to reflect in each accounting period the amount that expresses the benefit applying to that period.

In the case of someone who pays a full-year's insurance premium at the beginning of the insurance period, the entry to record this is debit (increase) Prepaid Insurance Premium for say, \$1,200, and credit (decrease) Bank for \$1,200.

Then a monthly recurring journal entry (scheduled transaction) is created that debits (increases) Insurance Expense \$100 and credits (decreases) Prepaid Insurance Premium \$100. This technique spreads the cost over the periods that receive the insurance coverage benefit. Businesses following generally accepted accounting practices would normally use this technique, especially if they had to present financial statements to banks or other lenders. Whether individuals do depends on the person and how concerned they are to match cost with benefit across time periods. Another factor influencing use of this technique would be the number of such situations the person encounters. It is relatively easy to remember one or two, but more difficult if having to manage 10 to 20. You would set up as many or as few as proved useful and important to you.

Suspense or Wash Accounts

The purpose of these accounts is to provide a device to track "change of mind" situations. The objective of these accounts is to provide a temporary location to record charges and credits that are not to be included permanently in your books of record. When the transactions reflected in these accounts have been fully completed, Wash/Suspense accounts will normally carry a zero balance.

For example, say in the grocery store you see canned vegetables on sale, so you buy 6 cans at \$1 per can. Say that the total purchases were \$50. When you come home and are putting things in the cupboard you discover you already had 12 cans. You decide to return the 6 you just bought. Some persons in this situation would charge (increase) the whole bill to Grocery Expense; and when they returned the cans, they would credit (decrease) Grocery Expense. That is one way of handling that. The effect of this method is to leave recorded on your books the cost of items that you really did not purchase from a permanent standpoint. It is only when the items have actually been returned and the vendor's return receipt has also been recorded that the distortion this method generates will then be removed.

Actually, there are several treatments, depending on when and how the original transaction was booked/recorded and when you decided to return the items purchased. Basically, did you change your mind before you recorded the transaction or after doing so?

If you decided to return the items after recording the purchase transaction, you may originally have charged Grocery Expense for the full amount (\$50) of all items. In that scenario, what you kept and the amount of the items to be returned were grouped into one account. You could edit the original transaction and restate the amount charged to

the Grocery Expense account to be the difference (\$44) between the total paid (\$50) for groceries and the value of the items to be returned. That leaves the returned-item value as the amount (\$6) you should record to the Suspense account.

Obviously, if you decided to return items before you recorded your purchase, then you would book the original entry as a charge to Grocery Expense for the amount kept (\$44) and as a charge to Suspense for the amount returned (\$6). The off-setting credit (\$50) to cash or credit card is not affected by these treatments.

When there are several persons shopping and at different vendors, there can be a case where there are several returns happening at once and in overlapping time frames. In that case the Wash Account is charged (increased) at time of changing the mind, and either Bank or Credit Card is credited. When the return occurs, the reverse happens: Bank or Credit Card is debited for the cash value of the returned items and the Wash/Suspense Account is credited in the same amount.

If the wash account has a non-zero balance, scanning the debit and credit entries in the account will show the non-matched items. That is, debits not matched by offsetting credits indicate items intended to be returned but not actually returned yet. The reverse (credits not matched by offsetting debits) indicates that returns were made but the original charge was not recorded in the Wash Account.

These differences can be cleared up by returning unreturned items or recording charges (debits) for items already returned. The mechanics of doing that likely will be finding the original expense account the item was charged to and making an entry like: debit Wash Account, credit original expense. It also could be as described above where the original recording is adjusted by adding a charge to Wash/Suspense account and decreasing the amount charged to the original account.

Short or Long-term Assets

This section explains why some types of assets may be short or long-term and presents an example.

An example is deposits (e.g., utility, rental, security). If the deposit agreement contains a provision to recover the deposit at the end of a year, the treatment could be that of a short-term asset. However, when the agreement is that the deposit holder returns the funds only upon successful inspection at the end of the relationship, then at the start of the relationship or agreement, the person paying the deposit has to decide whether to write it off as a current expense or to track it for eventual recovery at the end of the agreement (not infrequently, moving to a new location).

Whichever decision is made, the accounting treatment is to debit (increase) expense (assuming the write-off decision) or debit (increase) Deposits Receivable (assuming the intent is to recover the deposit in the future) and credit (decrease) Bank for the amount of the deposit (if paid by cash) or credit (increase) credit card if paid using that payment method.

Long-term (Fixed) Assets

This section illustrates long-term assets (those whose useful lives exceed a year) and discusses these types: land, buildings, leasehold improvements, intangibles, vehicles and other equipment.

Land

Land is not a wasting asset. That is, it does not get used up over time and rarely suffers damage such that it loses value. For that reason, it usually is recorded at cost at the time of purchase. Appreciation in its value over decades is not recorded and is not recognized in any way on the books of the owner. It is only after land has been sold that sale price and purchase cost are compared to calculate gain or loss on sale.

Land is frequently sold/purchased in combination with structures upon it. That means that the cost has to become separated from the cost of structures on it. Land valuation is usually part of the transfer of ownership process and its value is shown on the purchase documents separately from that of any structures it supports.

Land values shown on purchase documents frequently arise from the process of value determination managed by assessors whose job it is to assign values to land for tax purposes. Local and regional areas of a state or province use the values determined by assessors in their tax formulas, which provide revenues for local and regional governing authorities to finance their required community services.

Should land be acquired in a situation not subject to a history of land valuation by a formal valuation system, then the purchaser can appeal to real estate agents and an examination of recent sale transactions for information that would allow calculating a reasonable amount to express the value of the land.

Buildings

Buildings are the man-made “caves” in which much of life’s human interaction occurs. These structures are wasting assets, because in their use they or their components gradually wear. Over time they begin to lose some of their function and they can suffer damage due to planetary elements or human action.

Accepted accounting practice is to record the cost of the building determined at time of ownership transfer (purchase) or at conclusion of all costs of construction. Because buildings are frequently used for decades, and due to the need to be able to calculate gain or loss on sale, accounting practice preserves the original cost by not recording declines in value in the account containing the original purchase or construction cost.

Instead, the depreciation technique is used to show (in the balance sheet) the structure’s net book value (original cost reduced by accumulated depreciation). Depreciation is a separate topic treated elsewhere in this Guide.

Leasehold Improvements

When a business does not own the building where it operates, and instead has a long-term lease, it is not uncommon for the business tenant to make improvements to the premises so that the structure obtains both function and appearance that enhances conducting its business activities.

In these cases, the expenditures that the business incurs are recorded in a Leasehold Improvements account: increase (debit) Leasehold Improvements, decrease (credit) Bank or increase (credit) a suitable liability account (which could be a liability to a contractor or a bank or a credit card, etc.).

Vehicles or Equipment

Vehicles or Equipment of all kinds usually last for several years, but their useful lives are much shorter than that of assets that have little movement in their functioning. Because they do wear out over time, common accounting practice in business is to record depreciation using life spans and depreciation methods appropriate to the nature and use of the asset. Frequently, the life and depreciation methods chosen are influenced by what is permitted per national tax regulations for the kind of asset being depreciated.

Usually, businesses depreciate their assets. Individuals can do so as well to the degree that taxing authorities permit. Very wealthy persons employ accountants and attorneys to track and manage their investments and assets holdings to take advantage of all tax benefits permitted by law.

Intangibles

The mechanics of accounting (debiting and crediting appropriate accounts) for these assets are relatively simple, much the same as for any of the above assets. Where the difficulty lies is in their valuation, which is an advanced topic and not something that individual persons and small businesses would likely encounter. For that reason further discussion of items such as patents, copyrights, goodwill, etc. are left out of this Guide.

第 16 章 Depreciation

This chapter will introduce the concept of depreciation in accounting and give some real life examples for using it.

Basic Concepts

Depreciation is the accounting method for expensing capital purchases over time. There are two reasons that you may want to record depreciation; you are doing bookkeeping for your own personal finances and would like to keep track of your net worth, or you are doing bookkeeping for a small business and need to produce a financial statement from which you will prepare your tax return.

The method of recording depreciation is the same in either case, but the end goal is different. This section will discuss the differences between the two. But first, some terminology.

- **Accumulated depreciation** - the accumulated total of book depreciation taken over the life of the asset. This is accumulated in the depreciation account in the asset section.
- **Book depreciation** - this is the amount of depreciation that you record in your financial statements per accounting period.
- **Fair market value** - the amount for which an asset could be sold at a given time.
- **Net book value** - this is the difference between the original cost and the depreciation taken to date.
- **Original cost** - this is the amount that the asset cost you to purchase. It includes any cost to get the asset into a condition in which you can use it. For example - shipping, installation costs, special training.
- **Salvage value** - this is the value that you estimate the asset can be sold for at the end of its useful life (to you).
- **Tax depreciation** - this is the amount of depreciation that you take for income tax purposes.

Personal Finances

Depreciation is used in personal finances to periodically lower an asset's value to give you an accurate estimation of your current net worth. For example, if you owned a car you could keep track of its current value by recording depreciation every year. To accomplish this, you record the original purchase as an asset, and then record

a depreciation expense each year (See “Example” 一节 for an example). This would result in the net book value being approximately equal to the fair market value of the asset at the end of the year.

Depreciation for personal finance has no tax implications, it is simply used to help you estimate your net worth. Because of this, there are no rules for how you estimate depreciation, use your best judgement.

For which assets should you estimate depreciation? Since the idea of depreciation for personal finances is to give you an estimate of your personal net worth, you need only track depreciation on assets of notable worth that you could potentially sell, such as a car or boat.

Business

As opposed to personal finance where the goal is tracking personal worth, business is concerned with matching the expense of purchasing capital assets with the revenue generated by them. This is done through book depreciation. Businesses must also be concerned with local tax laws covering depreciation of assets. This is known as tax depreciation. The business is free to choose whatever scheme it wants to record book depreciation, but the scheme used for tax depreciation is fixed. More often than not this results in differences between book and tax depreciation, but steps can be taken to reduce these differences.

Now, what purchases should be capitalized? If you expect something that you purchase to help you earn income for more than just the current year, then it should be capitalized. This includes things like land, buildings, equipment, automobiles, and computers - as long as they are used for business purposes. It does not include items that would be considered inventory. So if you made a purchase with the intent to resell the item, it should not be capitalized.

In addition to the purchase of the asset itself, any costs associated with getting the asset into a condition so that you can use it should be capitalized. For example, if you buy a piece of equipment and it needs to be shipped from out of town, and then some electrical work needs to be done so you can plug the machine in, and some specialized training is needed so you know how to use the machine, all these costs would be included in the cost of the equipment.

You also need to know the estimated salvage value of the asset. Generally, this is assumed to be zero. The idea behind knowing the salvage value is that the asset will be depreciated until the net book value (cost less depreciation) equals the salvage value. Then, when the asset is written off, you will not have a gain or loss resulting from the disposal of the asset.

The last step is to determine the method of depreciation that you want to use. This will be discussed on the next few pages.

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 be expensed over time. For every asset which undergoes depreciation, you will need to decide on a depreciation scheme. An important point to keep in mind is that, for tax purposes, you will need to depreciate your assets at a certain rate. This is called tax depreciation. For financial statement purposes you are free to choose whatever method you want. This is book depreciation. Most small businesses use the same rate for tax and book depreciation. This way there is less of a difference between your net income on the financial statements and your taxable income.

This section will present 3 of the more popular depreciation schemes: linear, geometric, and sum of digits. To simplify the examples, we will assume the salvage value of the asset being depreciated is zero. If you choose to use a salvage value, you would stop depreciating the asset once the net book value equals the salvage value.

1. **Linear depreciation** diminishes the value of an asset by a fixed amount each period until the net value is zero. This is the simplest calculation, as you estimate a useful lifetime, and simply divide the cost equally across that lifetime.

Example: You have bought a computer for \$1500 and wish to depreciate it over a period of 5 years. Each year the amount of depreciation is \$300, leading to the following calculations:

16.1. Linear Depreciation Scheme Example

| Year | Depreciation | Remaining Value |
|------|--------------|-----------------|
| 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%.

16.2. Geometric Depreciation Scheme Example

| Year | Depreciation | Remaining Value |
|------|--------------|-----------------|
| 0 | - | 1500 |
| 1 | 450 | 1050 |
| 2 | 315 | 735 |
| 3 | 220.50 | 514.50 |
| 4 | 154.35 | 360.15 |
| 5 | 108.05 | 252.10 |

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:

16.3. Sum of Digits Depreciation Scheme Example

| Year | Depreciation | Remaining Value |
|------|--------------|-----------------|
| 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 `Expenses:Depreciation` account.

```
-Assets
  -Fixed Assets
    -ITEM1
```

```
-Cost                (Asset Cost account)
-Depreciation        (Accumulated Depreciation account)
-ITEM2
  -Cost              (Asset Cost account)
  -Depreciation      (Accumulated Depreciation account)
-Current Assets
  -Bank
-Expense
  -Depreciation      (Depreciation Expense account)
```

One of the features of the account hierarchy shown above is that you can readily see some important summary values about your depreciating asset. The `Assets:Fixed Assets:ITEM1` account total shows you the current estimated value for item1, the `Assets:Fixed Assets:ITEM1:Cost` shows you what you originally paid for item1, `Assets:Fixed Assets:ITEM1:Depreciation` shows you your accrued depreciation for item1, and finally, `Expenses:Depreciation` demonstrates the total accrued depreciation of all your assets.

It is certainly possible to use a different account hierarchy. One popular account setup is to combine the `Asset Cost` and `Accrued Depreciation` asset accounts. This has the advantage of having fewer accounts cluttering your account hierarchy, but with the disadvantage that to determine some of the summary details mentioned in the paragraph above you will have to open the account register windows. As with most things, there are many ways to do it, find a way that works best for you.

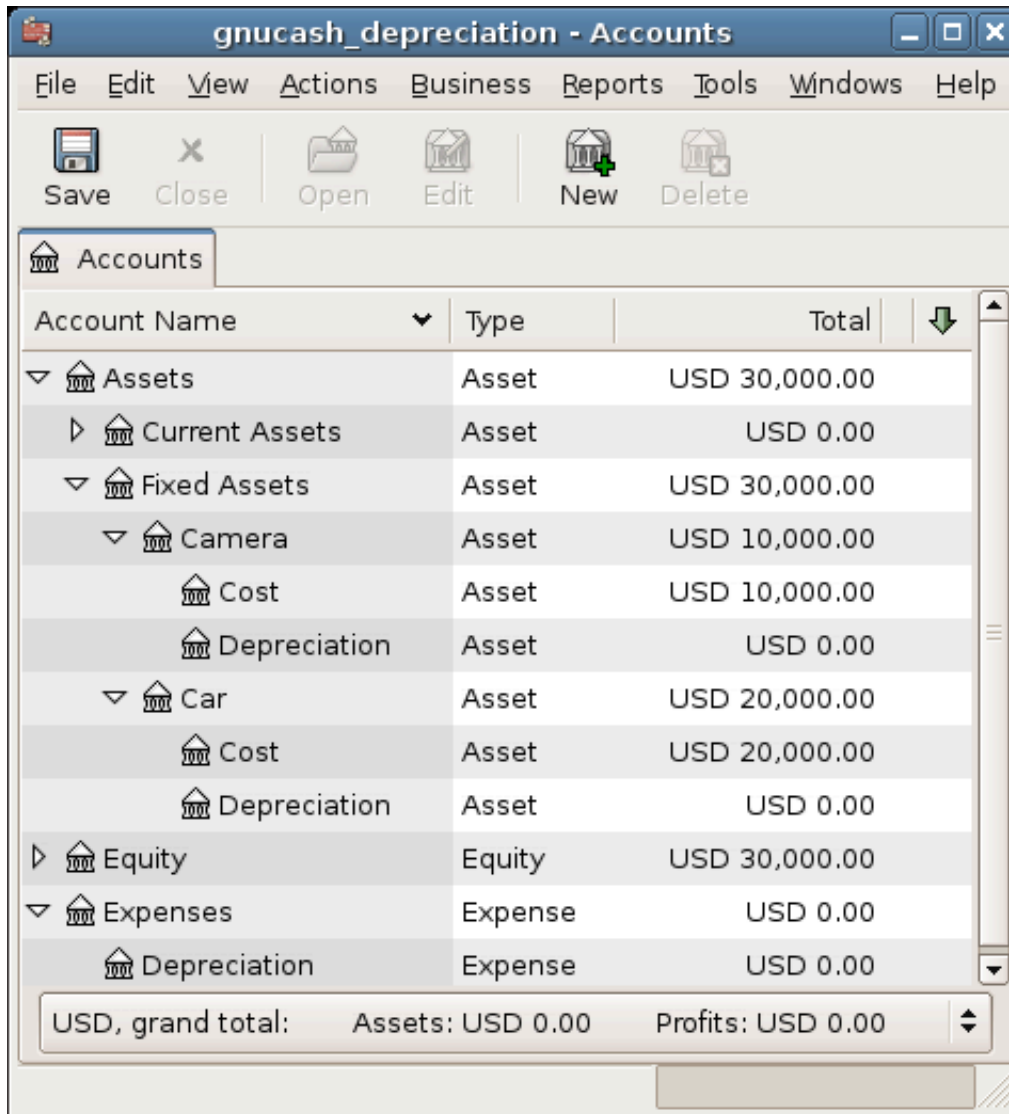
The actual input of the depreciation amounts is done by hand every accounting period. There is no way in GnuCash (as of yet) to perform the depreciation scheme calculations automatically, or to input the values automatically into the appropriate accounts. However, since an accounting period is typically one year, this really is not much work to do by hand.

Example

Let's go ahead and step through an example. Imagine you are a photographer and you use a car and an expensive camera for your personal business. You will want to track the depreciation on these items, because you can probably deduct the depreciation from your business taxes.

The first step is to build the account hierarchy (as shown in the previous section, replace `ITEM1` and `ITEM2` with "car" and "camera"). Now, record the purchase of your assets by transferring the money from your bank account to the appropriate `Asset Cost` accounts for each item (eg: the `Assets:Fixed Assets:Car:Cost` account for the car). In this example, you start with \$30k in the bank, the car cost \$20k and the camera cost \$10k and were both purchased on January 1, 2000.

16.1. Asset Accounts Before Depreciation



The screenshot shows the 'gnucash_depreciation - Accounts' window. It features a menu bar (File, Edit, View, Actions, Business, Reports, Tools, Windows, Help) and a toolbar with icons for Save, Close, Open, Edit, New, and Delete. Below the toolbar is a tab labeled 'Accounts'. The main area contains a table with columns 'Account Name', 'Type', and 'Total'. The table lists the following accounts and their balances:

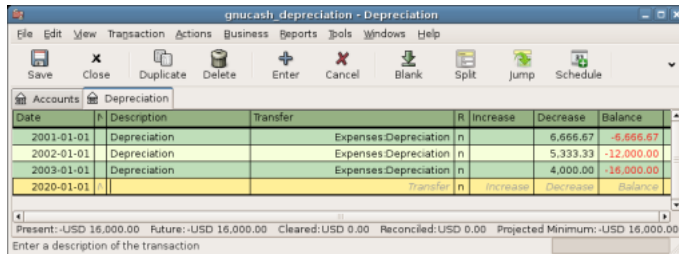
| Account Name | Type | Total |
|-------------------|------------------|-------------------|
| Assets | Asset | USD 30,000.00 |
| Current Assets | Asset | USD 0.00 |
| Fixed Assets | Asset | USD 30,000.00 |
| Camera | Asset | USD 10,000.00 |
| Cost | Asset | USD 10,000.00 |
| Depreciation | Asset | USD 0.00 |
| Car | Asset | USD 20,000.00 |
| Cost | Asset | USD 20,000.00 |
| Depreciation | Asset | USD 0.00 |
| Equity | Equity | USD 30,000.00 |
| Expenses | Expense | USD 0.00 |
| Depreciation | Expense | USD 0.00 |
| USD, grand total: | Assets: USD 0.00 | Profits: USD 0.00 |

The asset accounts still only have the investments stored.

Looking at the tax codes, we realize that we must report depreciation on these items using the “sum of digits” scheme, over a 5 year period. So, the yearly depreciation amounts for the car come to \$6667, \$5333, \$4000, \$2667, \$1333 for years 1 to 5 respectively, rounded to the nearest dollar. The yearly depreciation amounts for the camera are \$3333, \$2667, \$2000, \$1333, \$667. Consult the previous section on Depreciation Schemes for the formula for calculating these values.

For each accounting period (IE: fiscal year) you record the depreciation as an expense in the appropriate Accrued Depreciation account (eg: the Assets:Fixed Assets:Car:Depreciation account for the car). The two windows below show your car’s accrued depreciation account and the main window after the third year (IE: three periods) of depreciation using this [sum of digits] scheme.

16.2. Asset Depreciation In The Register

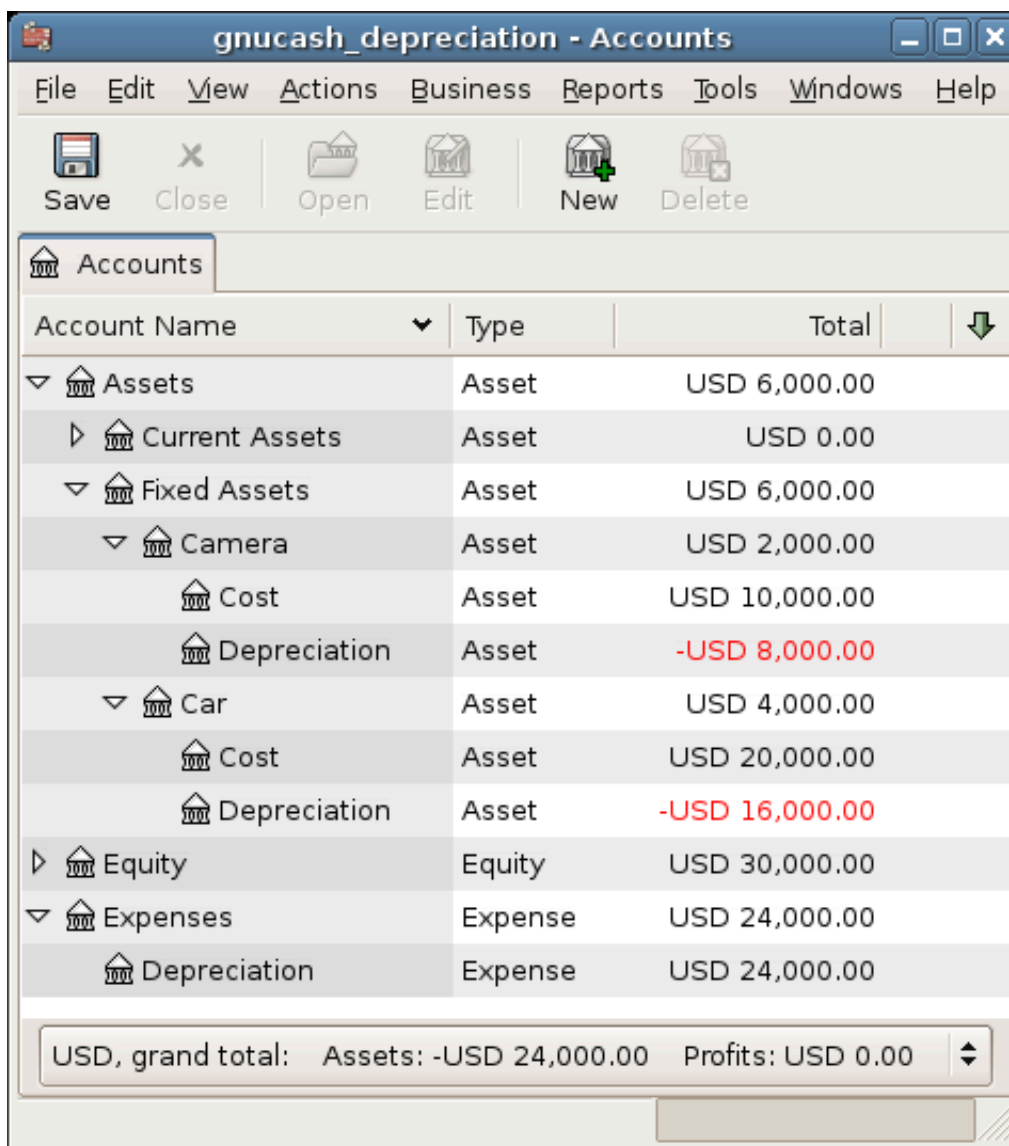


| Date | Description | Transfer | R | Increase | Decrease | Balance |
|------------|--------------|-----------------------|---|----------|------------|---------|
| 2001-01-01 | Depreciation | Expenses Depreciation | n | 6,666.67 | -6,666.67 | |
| 2002-01-01 | Depreciation | Expenses Depreciation | n | 5,333.33 | -12,000.00 | |
| 2003-01-01 | Depreciation | Expenses Depreciation | n | 4,000.00 | -16,000.00 | |
| 2020-01-01 | | Transfer | n | | | |

Present: USD 16,000.00 Future: USD 16,000.00 Cleared: USD 0.00 Reconciled: USD 0.00 Projected Minimum: -USD 16,000.00

This account shows a list of depreciation transactions.

16.3. Accounts Tab After Asset Depreciation



| Account Name | Type | Total |
|----------------|---------|----------------|
| Assets | Asset | USD 6,000.00 |
| Current Assets | Asset | USD 0.00 |
| Fixed Assets | Asset | USD 6,000.00 |
| Camera | Asset | USD 2,000.00 |
| Cost | Asset | USD 10,000.00 |
| Depreciation | Asset | -USD 8,000.00 |
| Car | Asset | USD 4,000.00 |
| Cost | Asset | USD 20,000.00 |
| Depreciation | Asset | -USD 16,000.00 |
| Equity | Equity | USD 30,000.00 |
| Expenses | Expense | USD 24,000.00 |
| Depreciation | Expense | USD 24,000.00 |

USD, grand total: Assets: -USD 24,000.00 Profits: USD 0.00

The asset accounts got decreased by the depreciation.

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.

第 17 章 Python Bindings

GnuCash historically has always been a traditional application in the sense that you open it, use it to manipulate your financial data via the windows it presents, save your data and close the windows again. This has the inherent limitation that you can only do whatever the windows, menus and toolbars allow you to do.

Sometimes you might need a little more flexibility. For example, you need a report with just a little different information than what the built-in reports provide, or you want to automate a frequently recurring action. Such custom manipulations are ideal candidates to write in one or the other scripting language.

Starting with GnuCash version 2.4 you can write Python [[https://en.wikipedia.org/wiki/Python_\(programming_language\)](https://en.wikipedia.org/wiki/Python_(programming_language))] scripts to manipulate your financial data.

The Python extensions are an optional feature which creates additional dependencies. To be able to use Python scripts, GnuCash must have been built with the `cmake -DWITH_PYTHON=ON ...` option enabled, otherwise all what follows won't work. At present this option is not enabled by default, so if you need this, you may have to compile GnuCash from source yourself. But some distributions offer it also as a separate package with a name like `python[version]-GnuCash`

The Python extensions come with a couple of ready to use scripts. This chapter will show you how to use some of them.

This chapter is not about how to write your own Python scripts. Refer to the developer documentation [https://code.gnucash.org/docs/STABLE/python_bindings_page.html] for that instead.

第 18 章 Importing Business Data

Importing Bills and Invoices

General

This functionality creates invoices or bills from a csv import file containing rows of invoice entry data. The import file may contain rows for new and/or existing invoices. If an invoice already exists, GnuCash adds the imported entries to the invoice (unless the invoice is already posted). If the import file contains posting data for an invoice, then GnuCash will also attempt to post the invoice. If any row of an invoice contains an error, GnuCash will ignore all rows of the same invoice.

The field separator in the csv file must be either a comma or a semicolon; field values may be enclosed in double quotes.

For the sake of readability, in this chapter the term “invoice” by itself is used to refer to both customer invoices and vendor bills.

The format of the import file

The import file should contain rows of invoice entry data. Any row contains both header and entry fields, but GnuCash takes the invoice header data from the first row of an invoice ID. For informational purposes, the header data may be repeated for each subsequent row of the same invoice.

There is no information in the file to indicate whether it concerns customer invoice or vendor bill data. Instead, a user option in the import dialog makes that distinction.

Each row should contain the fields listed below, in the same sequence, separated by a comma or a semicolon. The fields are listed here by their technical name, which GnuCash uses in the preview of the import data.

- `id` - The invoice ID. If the invoice ID is blank, GnuCash replaces it with the invoice ID from the previous row. If the invoice ID already exists, GnuCash will add the entries to the existing invoice (unless it is already posted).
- `date_opened` - Use the same date format as defined in Preferences. Defaulted to today's date if left blank, or if the date provided is not valid.
- `owner_id` - Customer or vendor number. Mandatory in the first data row of an invoice. If not provided, all rows of the same invoice will be ignored.
- `billingid` - Billing ID. Optional

- `notes` - Invoice notes. Optional.
- `date` - The date of the entry. Defaulted to date opened if left blank, or if the date provided is not valid.
- `desc` - Description. Optional
- `action` - Action. Optional
- `account` - Account for the entry. Mandatory in each row. If not provided or invalid, all rows of the same invoice will be ignored.
- `quantity` - Quantity. Defaulted to 1 if left blank.
- `price` - Price. Mandatory for each row. If not provided, all rows of the same invoice will be ignored.
- `disc_type` - Type of discount. Optional. Only relevant for invoices, not for bills. Use “%” or blank for percentage value, anything else for monetary value.
- `disc_how` - Discount how. Optional. Only relevant for invoices, not for bills. Use “>” for discount applied after tax, “=” for discount and tax applied before tax, and “<” , blank or anything else for discount applied before tax.
- `discount` - Amount or percentage of discount. Optional. Only relevant for invoices, not for bills
- `taxable` - Is this entry taxable? Optional. Use “Y” or “X” for yes, “N” or blank for no.
- `taxincluded` - Is tax included in the item price? Optional. Use “Y” or “X” for yes, “N” or blank for no.
- `tax_table` - Tax table. Optional. If the tax table provided does not exist, it will be blank in the invoice.
- `date_posted` - Date posted. Optional. Use the same date format as defined in Preferences. If you provide a date posted for the first row of an invoice, GnuCash will attempt to also post the invoice (as opposed to only saving or updating it).
- `due_date` - Due date. Optional. Use the same date format as defined in Preferences. Defaulted to date posted, if left blank. Only relevant in the first row of an invoice, if the invoice is posted.
- `account_posted` - Post to account, for vendor or customer posting. Only mandatory in the first row of an invoice, if the invoice is posted.
- `memo_posted` - Memo. Optional. Only relevant in the first row of an invoice, if the invoice is posted.

- `accu_splits` - Accumulate splits? Optional. Use “Y” or “X” for yes, “N” or blank for no. Only relevant in the first row of an invoice, if the invoice is posted. If you use a spreadsheet program to create the import file, it is advised not to use blank for no, because a final column with only blanks may not be recognized as relevant data when the spreadsheet program creates the csv file.

Note on double quotes

If you use the field separator character within a field, the field value should be enclosed in double quotes. Only for the fields description (desc) and notes, can you also include a double quote within a quoted field value, by doubling the double quote. E.g. "This field value uses the separator, and a ""quoted"" word", would be imported as This field value uses the separator, and a "quoted" word.

Example content for two bills; one of 2 entries, and one of 3 entries. The first is saved and posted, the second only saved. Using comma field separator, decimal point and dd/mm/yyyy date format.

```
1204;15/12/2018;2001;P0 210220;Special delivery;16/12/2018;Pride and Prejudice;pc;Expenses:Books;1;10.01;;;;;;;;;;
1204;15/12/2018;2001;P0 210220;Special delivery;16/12/2018;Electronic principles;pc;Expenses:Books;1;10.01;;;;;;;;;;
1205;15/12/2018;2044;P0 21099;;16/12/2018;Ultimate Guide;pc;Expenses:Books;1;10.01;;;;;;;;;;
1205;15/12/2018;2044;P0 21099;;16/12/2018;Dinner & drinks;pc;Expenses:Dining;1;10.01;;;;;;;;;;
1205;15/12/2018;2044;P0 21099;;16/12/2018;UG course;pc;Expenses:Education;1;10.01;;;;;;;;;;
```

Example content for one customer invoice, with one entry, including tax and discount. Using comma field separator, decimal point and dd/mm/yyyy date format. The the value of the description field contains the separator character.

```
20221;16/12/2018;1001;Order 3378;Discount as agreed;4/12/2018;"Accounting part 1, 2";ea;Income
```

Import your data

To import your invoice data, navigate to [File] → [Import] → [Import Bills & Invoices...] to open a new import dialog, and provide the necessary information.

- [1. Choose the file to import] - Select your import file, or manually type the path and file name.
- [2. Select import type] - Select the import type, either [Bill] or [Invoice] .
- [3. Select import options] - Select your csv format. Use the [with quotes] options if your file contains fields enclosed in double quotes. These options also match fields not enclosed in quotes; except for the fields for description and notes, fields should

not contain the quote character itself. See Note on double quotes above. Use one of the other options if your file does not have fields enclosed in quotes; any quote characters in the file will be imported as is.

- [4. Preview] - Once you have selected your import file and csv format, GnuCash shows you a preview of the data. You can verify if your data is listed in the correct columns. If you do not see any rows in the preview, then GnuCash was not able to match your import data rows to the selected csv format. See “What could go wrong?” 一节 below.
- [5. Afterwards] - You can choose if GnuCash should open tabs for the invoices after the import. Either for all invoices, or for the invoices that are saved but not posted, or for none of the invoices. Opening tabs slows down the import process considerably.
- Start the import - If you are satisfied with your selections, hit the OK button to start the import.

If your data file contains invoice IDs that already exist, then GnuCash will ask you (once per import session) to confirm that you want to update existing invoices. If not confirmed, all rows for existing invoices will be ignored.

Internally, GnuCash uses so called regular expressions to match the import rows to the data fields. The import option [Custom regular expression] offers the option to use your own regular expression for this matching process. Obviously, this option requires that you are well versed in regular expressions. When you choose the option [Custom regular expression], GnuCash opens a window in which you can edit the GnuCash regular expression, or replace it with your own. Your regular expression should contain a named subpattern for each of the fields of the csv file (using the technical names). A custom regular expression could be useful if the rows of your source data file contain all required fields, but in a different order or format. E.g. if the format of your source data file starts with customer number, followed by invoice ID, followed by the due date, and uses | as separator, your regular expression would start with something like this:

```
^(?<owner_id>[^\|]*)\|(?<id>[^\|]*)\|(?<due_date>[^\|]*)
```

With a custom regular expression, GnuCash could import your source data files, without the need to convert them to the GnuCash import format.

Feedback and statistics

GnuCash executes the import process in three steps:

- **Import** - Imports the data file and attempts to match each row to the data fields.
- **Validation and adjustment** - Validates the data fields and replaces data with defaults if applicable.
- **Processing** - Handles the currency related validations, and creates, updates and posts the invoices.

After all steps have finished, GnuCash issues information about the result of the process. The initial dialog shows the informational or error messages from the validation and processing steps. The second dialog shows the statistics of the process:

- **Import - rows ignored**: the number of rows that could not be matched to the data fields.
- **Import - rows imported**: the number of rows that were successfully matched to the data fields.
- **Processing and validation - rows fixed**: the number of rows for which a default value was used for a field.
- **Processing and validation - rows ignored**: the number of rows for that were not processed because of a validation error.
- **Processing and validation - invoices created**: the number of invoices created.
- **Processing and validation - invoices updated**: the number of invoices that were updated.

If there were unmatched rows in the import step, a final dialog shows the actual rows that could not be matched.

What could go wrong?

Errors in the import step

If the statistics show unmatched rows under “Import - rows ignored” , then there is some issue with the format of your import file. Verify that you use and select the correct separator. Verify that your data rows have exactly 21 separator characters (1 for each field, except for the last). Verify whether you use the separator character within a data field; if so, enclose the field in double quotes.

If you use one of the [with quotes] import options, verify if you use the double quote character in any of the data field values; if within the description or notes fields, make sure that the field value is quoted, and precede each double quote within the field with an extra double quote; if within any other field, remove the double quote character.

Errors in the validation step

The following errors can occur in the validation step. Any error in a data row will cause all rows of the same invoice to be ignored.

In versions 3.4 and earlier, an error in a data row would cause just that row to be ignored, as opposed all rows of the same invoice.

- The field `ID` is blank. GnuCash replaces a blank invoice ID with the invoice ID from the previous row. But the first row of the import file should always have an invoice ID.
- The field `owner_id` is blank. Every first row of an invoice should have an `owner_id`.
- The customer or vendor number in field `owner_id` does not exist. The `owner_id` in the first row of an invoice should be an existing customer (for invoices) or vendor (for bills).
- The date in field `date_posted` is not a valid date. If you provide a value for `date_posted` in the first row of an invoice, it should be a valid date. Did you use the date format as set in Preferences?
- The account in the field `account_posted` does not exist. If you provide a value for the field `date_posted` in the first row of an invoice, the field `account_posted` should be an existing account.
- The account in the field `account_posted` is not of type Accounts Receivable (for invoices) or Accounts Payable (for bills). If you provide a value for the field `date_posted` in the first row of an invoice, the field `account_posted` should be an account of the correct type.
- The field `price` is blank. Every row should have a value for the field `price`.
- The account in the field `account` does not exist. Every row should have an existing account in the field `account`.

Any error in the validation step is listed after the overall import process completes. Correct your data file accordingly.

Errors in the processing step

The following errors can occur in the processing step.

- The invoice cannot be updated because it is already posted. All rows of the same invoice will be ignored. If you want to update the existing invoice, unpost it first in GnuCash.

- The currency of the invoice differs from the currency of the account posted (“Invoice x NOT posted because currencies don't match”). GnuCash determines the currency of the invoice either from the customer or vendor master data (for a new invoice) or from the invoice itself (for an existing invoice). The currency of the invoice must agree with the currency of the post to account in the field `account_posted`. GnuCash creates the invoice but cannot post it. Manually correct the invoice in GnuCash.
- The invoice requires currency conversion. (“Invoice x NOT posted because it requires currency conversion”). The invoice contains entries on accounts with different currencies, or the currency of the entries differs from the currency of the post to account. For such an invoice, GnuCash needs exchange rates to translate the currency amounts. GnuCash creates the invoice but cannot post it. Post the invoice manually in GnuCash, and provide the requested exchange rates.

Not supported invoice functionality

Currently the invoice import function does not support (at least) the following:

- Import of billing terms and job.
- Import of customer and job in default chargeback project for bills.
- Application of billing terms from customer or vendor master data.
- Automatic numbering of invoices.
- Credit notes.

Importing Customers and Vendors

General

This functionality creates and updates customers and vendors from a csv import file containing rows of vendor/customer master data. The import file may contain rows for new and/or existing customers/vendors. If a customer/vendor already exists, GnuCash updates the existing customer/vendor.

The format of the import file

The import file should contain rows of customer/vendor data, one row for each customer/vendor. The customer/vendor is identified by the customer/vendor number in the field `id` of the data rows. If the field is blank, GnuCash will use the next number from the relevant counter (set in the [Counters] tab under [File] → [Properties]).

There is no data in the file to indicate whether it concerns customer or vendor master data. Instead, a user option in the import dialog makes that distinction.

Each row should contain the fields listed below, in the same sequence, separated by a comma or a semicolon. The fields are listed here by their technical name, which GnuCash uses in the preview of the import data.

- `id` - The customer/vendor number. If it is for an existing customer/vendor, GnuCash will update the customer/vendor. Note that in GnuCash e.g. '000010' is a different customer number than '10'. If the `id` field is empty, GnuCash will use the next number from the relevant counter.
- `company` - The company name. If it is left blank, it is defaulted to the value of field `name`. If that is also blank, then the row is ignored.
- `name` - Billing address - Name. Optional.
- `addr1` - Billing address - Address line 1. At least one of the four address lines of the billing address must be filled. If not, then the row is ignored.
- `addr2` - Billing address - Address line 2.
- `addr3` - Billing address - Address line 3.
- `addr4` - Billing address - Address line 4.
- `phone` - Billing address - Phone. Optional
- `fax` - Billing address - Fax. Optional
- `email` - Billing address - Email. Optional
- `notes` - Notes. Optional
- `shipname` - Shipping information - Name. Optional. Not relevant for vendors.
- `shipaddr1` - Shipping information - Address line 1. Optional. Not relevant for vendors.
- `shipaddr2` - Shipping information - Address line 2. Optional. Not relevant for vendors.
- `shipaddr3` - Shipping information - Address line 3. Optional. Not relevant for vendors.
- `shipaddr4` - Shipping information - Address line 4. Optional. Not relevant for vendors.
- `shipphone` - Shipping information - Phone. Optional. Not relevant for vendors.
- `shipfax` - Shipping information - Fax. Optional. Not relevant for vendors.
- `shipmail` - Shipping information - Email. Optional. Not relevant for vendors.

Example content for a customer with a separate shipping address. Using a semicolon for separator.


```
2201;A11      Star      Company;A11      Star      Company;Union      Avenue      776;San
Juan;CA;;0482938838;;contact@allstar.com;Last      contacted      on      4/4/2018.;A11
Star      Company;      John      Alderman,      Office      456;Union      Avenue      777;San
Juan;CA;78998766;;alderman@allstar.com
```

Example content for a vendor; no ID given, so GnuCash will take the next number from the counter. Using a comma for separator.

```
,Johnson      Supplies,Johnson      Supplies,Electric      Park
56,Plains,VA,,0482986538,,jack@johnson.com,Discount negotiated,,,,,,,,,
```

All fields by technical name in the required sequence.

```
id, company, name, addr1, addr2, addr3, addr4, phone, fax, email, notes, shipname,
shipaddr1, shipaddr2, shipaddr3, shipaddr4, shipphone, shipfax, shipmail
```

Import your data

To import your customer or vendor data, navigate to [File] → [Import] → [Import Customer & Vendors...] to open a new import dialog, and provide the necessary information.

- [1. Choose the file to import] - Select your import file, or manually type the path and file name.
- [2. Select import type] - Select the import type, either [Customer] or [Vendor] .
- [3. Select import options] - Select your csv format. Use the [with quotes] options if your file contains fields enclosed in double quotes. These options also match fields not enclosed in double quotes, but fields should not contain the double quote character itself. Use one of the other options if your file does not have fields enclosed in quotes; any double quote characters in the file will then be imported as is.
- [4. Preview] - Once you have selected your import file and csv format, GnuCash shows you a preview of the data. You can verify if your data is listed in the correct columns. If you do not see any rows in the preview, then GnuCash was not able to match your import data rows to the selected csv format. See “What could go wrong?” 一节 below.
- Start the import - If you are satisfied with your selections, hit the OK button to start the import.

Internally, GnuCash uses so called regular expressions to match the import rows to the data fields. The import option [Custom regular expression] offers the option to use your own regular expression for this matching process. Obviously, this option requires that you are well versed in regular expressions.

When you choose the option `[Custom regular expression]`, GnuCash opens a window in which you can edit the GnuCash regular expression, or replace it with your own. Your regular expression should contain a named subpattern for each of the fields of the csv file (using the technical names). A custom regular expression could be useful if the rows of your source data file contain all necessary fields, but in a different order or format. E.g. if the format of your source data file starts with customer number, followed by company name, name, and one address field, and that is all you want to import, then your custom regular expression would be something like this (using comma as a separator):

```
^(?<company>[^\,]*),(?<id>[^\,]*),(?<name>[^\,]*),(?<addr1>[^\,]*),(?<addr2>[^\,]*),(?<addr3>[^\,
```

With a custom regular expression, GnuCash could import your source data files, without the need to convert them to the GnuCash import format.

Feedback and statistics

GnuCash executes the import process in three steps:

- **Import** - Imports the data file and attempts to match each row to the data fields.
- **Validation and adjustment** - Validates the data fields and replaces data with defaults if applicable.
- **Processing** - Creates or updates the vendor or customer master data.

After all steps have finished, GnuCash issues information about the result of the process. The initial dialog shows the statistics of the process:

- **Import results - lines ignored:** the number of rows that could not be matched to the data fields.
- **Import results - lines imported:** the number of rows that were successfully matched to the data fields.
- **Import results - customers/vendors fixed:** the number of rows for which a default value was used for a field.
- **Import results - customers/vendors ignored:** the number of rows for that were not processed because of a validation error.
- **Import results - customers/vendors created:** the number of customers/vendors created.
- **Import results - customers/vendors updated:** the number of customers/vendors that were updated.

If there were unmatched rows in the import step, a final dialog shows the actual rows that could not be matched.

What could go wrong?

Errors in the import step

If the statistics show unmatched rows under “Import results - lines ignored” , then there is some issue with the format of your import file. Verify that you use and select the correct separator. Verify that your data rows have exactly 18 separator characters (1 for each field, except for the last). Verify whether you use the separator character within a data field; if so, enclose the field in double quotes.

If you use one of the `[with quotes]` import options, verify if you use the double quote character in any of the data field values. If so, remove them; importing double quotes as is, is not supported when using the `[with quotes]` import options.

Errors in the validation step

If the statistics show rows under “Import results - customers/vendors ignored” , then data rows were ignored because of one of the errors below:

- The field `company` and the field `name` are both blank. The field `company` is mandatory; if it is blank, then it is defaulted to the value of the field `name`, but if both are blank, then the data row cannot be processed.
- The fields `addr1`, `addr2`, `addr3` and `addr4` are all blank. At least one of these fields must have a value, otherwise the data row cannot be processed.

Not supported customer/vendor functionality

Currently the customer/vendor import function does not support (at least) the following:

- Import of any of the fields in the customer tab for billing information: currency, terms, discount, credit limit, tax included and tax table.
- Import of any of the fields in the vendor tab for payment information: currency, terms, tax included and tax table.

部分 IV. 附录

目录

| | |
|---|-----|
| GnuCash 术语表 | 309 |
| A. Migration Guide | 314 |
| Using Accounts vs. Categories | 314 |
| Importing Data | 314 |
| Import of Financial Data by QIF Files | 314 |
| Import of Business Data by CSV Files | 315 |
| Converting XML GnuCash File | 315 |
| B. Auxiliary File Formats | 317 |
| Check Format Files ("*.chk") | 317 |
| Overview | 317 |
| Example file | 317 |
| Field Descriptions | 318 |
| Creating Check Format Files | 332 |
| C. GNU Free Documentation License | 333 |
| 0. PREAMBLE | 333 |
| 1. APPLICABILITY AND DEFINITIONS | 333 |
| 2. VERBATIM COPYING | 334 |
| 3. COPYING IN QUANTITY | 335 |
| 4. MODIFICATIONS | 335 |
| 5. COMBINING DOCUMENTS | 337 |
| 6. COLLECTIONS OF DOCUMENTS | 338 |
| 7. AGGREGATION WITH INDEPENDENT WORKS | 338 |
| 8. TRANSLATION | 338 |
| 9. TERMINATION | 339 |
| 10. FUTURE REVISIONS OF THIS LICENSE | 339 |
| Addendum | 339 |

GnuCash 术语表

下面列出了 GnuCash中出现的常见术语。其中一些条目引自维基百科 [https://en.wikipedia.org/wiki/Main_Page]。

| | |
|-------------|--|
| 科目 | 科目用于跟踪你的资产、负债、花费和收入。 |
| 会计期间 | 会计期间指编制一个实体的会计账簿所参照的期间。常见的会计期间有月度、季度和年度期间。 |
| AqBanking | 一个实现德国家庭银行计算机接口（HBCI）和电子银行互联网通信标准（EBICS）、OFX Direct Connect和Paypal的库。它是openHBCI的继承者。 |
| 资产 | 资产是你拥有的某些东西。任何有形的或无形的，可以拥有或控制以产生积极的经济价值的东西都可被视为资产。 |
| 资产负债表 | 资产负债表是个人或组织财务余额的汇总。它总结了一家公司在特定时间点的资产、负债和所有者权益。 |
| 账簿 | 账簿是对个人或组织所有交易的记录。在 GnuCash中，每个文件包含一个账簿。 |
| 资本利得 / 资本损失 | 指在出售投资品时，投资品的买入价和卖出价之间的差额。又称 实现获利 / 实现损失。在投资品出售之前，现价与买入价间的差额称为 未实现收益 / 未实现损失。 参见Realized Gain/Loss, Unrealized Gain/Loss. |
| 股本 | 见Stock. |
| 科目汇总表 | 数据文件中所有可用科目的列表。科目汇总表包括资产负债表科目（资产、负债、股东权益）和损益表科目（收入、费用、收益、损失）。 |
| 佣金 | 支付给经纪人买卖证券的费用。 |
| 商品 | 商品是易于交易或出售的有价值的东西，比如货币、股票、债券、粮食、铜、石油都是商品。 |
| 普通股 | 见Stock. |
| 复利 | 利息重新投入后可以产生它自己的利息（利息产生的利息）。 |
| 成本基准? | Cost basis is the original cost of property, adjusted for various factors. Its primary use is for tax purposes. Factors that affect the cost basis include: stock splits, dividends, depreciation and return of capital distributions. Cost basis is used to determine the capital gain. |

| | |
|--|--|
| CSV | Stands for Comma Separated Values. CSV files are used to store data in plain text. Each line of the file is a record and each record can be comprised of multiple fields separated by commas. CSV is one import format that GnuCash supports. |
| 股息 | Dividends are cash payments a company makes to shareholders. The amount of this payment is usually determined as some amount of the profits of the company. Not all common stocks give dividends. 股息是一家公司向持股人支付的现金。股息金额通常取决于公司利润。不是所有普通股都会支付股息。 |
| Equities | Equities are investments in which the investor becomes part (or whole) owner in something. |
| Financial Transaction Services (FinTS) | A banking protocol used by German banks. Developed by the German Central Banking Committee ZKA (Zentraler Kredit-Ausschuss). The standard is used only by German banks. Prior 2002, it was called Home Banking Computer Interface (HBCI). FinTS is one import format that GnuCash supports. 参见Home Computer Banking Interface (HBCI). |
| GSettings | Since version 2.5 the tool that stores user configuration data. It uses the native data store of the operating system: Linux DConf since Gnome 3.0 dropped GConf macOS Defaults. Use the command-line defaults(1) to modify prefs when GnuCash isn't running. Windows Registry. |
| Home Computer Banking Interface (HBCI) | A bank-independent online banking protocol used by German banks. Home Banking Computer Interface (HBCI) was developed by the German Central Banking Committee ZKA (Zentraler Kredit-Ausschuss). Since 2002, it has been called Financial Transaction Services (FinTS). HBCI is one import format that GnuCash supports. 参见Financial Transaction Services (FinTS). |
| 利息 | 借款人为使用贷款人的钱而支付的费用。通常表示为每年本金的百分比。例如，一个利率为1%的储蓄账户，每年每存入100美元，就会付给你1美元。 |
| 负债 | 负债是个人或组织所欠的债务。 |

| | |
|--------------------|---|
| LibOFX | <p>An open source library for OFX. It was created by one of the GnuCash developers, originally to be the OFX interface for GnuCash.</p> <p>参见OFX.</p> |
| 流动性 | <p>衡量一项投资转换为现金的难易程度。储蓄账户里的钱流动性很强，而投资在房子里的钱流动性很低，因为卖房子需要时间。</p> |
| Lot | <p>A lot is a means of grouping a commodity so that you later identify that the item bought in one transaction is the same as one sold in a different transaction. Lots are often implemented with stocks, where capital gain can depend on which item is being sold at a given time.</p> |
| MT940 | <p>A financial information standard defined by SWIFT and used by several European banks. It is also used internally in HBCI. Unfortunately, you can't download the MT940 standard, but some banks publish it on their websites. MT940 is one import format that GnuCash supports.</p> |
| OFX | <p>The Open Financial eXchange format. This is a financial information exchange standard used by many institutions. OFX is one import format that GnuCash supports.</p> |
| Price Database | <p>The price database contains a store of price quotes for stocks, mutual funds, and currencies.</p> |
| Principal | <p>The original amount of money invested or borrowed.</p> |
| QFX | <p>The Quicken Financial eXchange format is a proprietary financial information exchange standard promoted by Quicken and used by many institutions. QFX is one import format that GnuCash supports.</p> |
| QIF | <p>The Quicken Interchange Format (QIF) is an open specification for reading and writing financial data to files. This is an older format that is still used by many institutions. QIF is one import format that GnuCash supports.</p> |
| Realized Gain/Loss | <p>The difference between the purchase and selling prices of an investment at the time the investment is sold. Also known as Capital Gain/Loss. Before an investment is sold, the difference in value is referred to as Unrealized Gain/Loss.</p> <p>参见资本利得 / 资本损失, Unrealized Gain/Loss.</p> |
| Reconciliation | <p>Reconciliation is a verification process in which the user compares their records with those of the financial institution to</p> |

ensure that the records in each system agree at some particular point in time. During reconciliation, the user checks individual transactions in their file against the institution's statement; upon completion, the reported balances of the institution and the reconciled account will match at that point in the register.

回报

计入资本利得或资本损失的总收入。另见“收益率”。

Reversing Transaction

In formal accounting, a transaction is never deleted. Therefore, when a mistake is made, the original transaction remains in the ledger, and a transaction is added to the ledger that reverses the original. This reversing transaction duplicates the original transaction, but with debit and credit amounts reversed. This removes the effect of the erroneous transaction from the books. After entering the reversing transaction, a corrected transaction can then be entered. Reversing transactions are not commonly used in personal accounting.

风险

投资回报与预期不同的可能性。投资通常可以从低风险（例如储蓄帐户或政府债券）到高风险（例如普通股或垃圾债券）分类。一般来说，风险越高，可能的回报也就越高。

计划交易

计划交易提供了一个框架，用于记住有关设置为将来发生一次或定期发生的交易的信息。
参见收益率。

Scheme

Scheme is a functional programming language based on a dialect of LISP. Reports in GnuCash use the Scheme programming language; thus, users wishing to customize reports must write Scheme.

Shareholder

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

Split

A split, or Ledger Entry, is the fundamental accounting unit. Each split consists of an amount, the value of that amount expressed in a (possibly) different currency, a Memo, a pointer to the parent Transaction, a pointer to the debited Account, a reconciled flag and timestamp, an action field, and a key-value frame which can store arbitrary data.

Stock

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

| | |
|----------------------|--|
| Stock split | Occurs when a company offers to issue some additional multiple of shares for each existing stock. For example, a “2 for 1” stock split means that if you own 100 shares of a stock, you will receive an additional 100 shares at no cost to you. The unit price of the shares will usually be adjusted so there is no net change in the value, so in this example the price per share will be halved. Note that such transactions affect the cost basis per share owned. |
| Transaction | A transaction consists of a date, a description, an ID number, a list of two or more splits, and a key-value frame. Transactions embody the notion of "double entry" accounting. |
| Unrealized Gain/Loss | Unrealized gain or loss is the difference in value between a purchase price and the current value of a given asset. These gains remain unrealized until the asset is sold, at which point they become <code>Realized gains or losses</code> . 参见资本利得 / 资本损失, <code>Realized Gain/Loss</code> . |
| Valuation | The process of determining the market value or the price the investment would sell at in a “reasonable time frame” . |
| XML | 可扩展标记语言是一种国际通用的标记标准。 GnuCash 默认使用 XML 存储数据。 |
| 收益率 | 对从投资中赚取收益的度量，也就是你从投资中挣到了多少钱。一般将收益率记为相对本金金额的百分比。资本利得或损失不计入收益率（另见“回报”。例如，价格100元，每年分红2元的股票收益率为2%。 |

附录 A. Migration Guide

This appendix is to help current users of other financial software packages in their migration to GnuCash. We address the conceptual differences between the layout of GnuCash accounts versus other software packages.

Using Accounts vs. Categories

If you are familiar with other personal finance programs, you are already accustomed to tracking your income and expenses as categories. Since GnuCash is a double-entry system (see “会计概念”一节), income and expenses are tracked in accounts. The basic concept is the same, but the account structure allows more consistency with accepted business practices. So, if you are a business user as well as a home user, GnuCash makes it easy to keep track of your business as well as your personal accounts.

Income and expense accounts give you the same information you would get with categories, but they also give you more flexibility in entering your transactions. In GnuCash, you have the option to enter transactions directly into income and expense accounts through their account registers. Other programs that use categories do not offer this option, because there is no “account register” for a category.

You also have the option in GnuCash to treat income and expense accounts exactly as you would treat categories, if you are more comfortable with that method. In Quicken® and similar programs, transactions require an account and a category. Substitute an income or expense account name in GnuCash where you would normally enter a category name in the other programs, and the result should be the same. We will discuss transaction entry in “交易”一节 in greater detail.

Importing Data

If you want to import data from your previous application, you should distinguish between financial data and other data. Probably the best way to import financial data is the “Quicken Interchange Format” QIF . It is a specific format for financial data, which most financial applications for the private sector know and can use for export.

For other data we suggest the use of the “Comma Separated Value” CSV format.

Import of Financial Data by QIF Files

See https://en.wikipedia.org/wiki/Quicken_Interchange_Format for it's details.

Some GnuCash users collected their knowledge about the best use in our https://wiki.gnucash.org/wiki/Quicken_Migration.

For other formats and more details see 第 3 章 Importing Data into GnuCash.

Import of Business Data by CSV Files

If you want to import customers and vendors or bills and invoices see 第 18 章 Importing Business Data.

Converting XML GnuCash File

The GnuCash XML data file can be transformed to almost any other data format (e.g., QIF, CSV...) quite easily if one is familiar with the “Extensible Stylesheet Language Transformations” XSLT [<https://en.wikipedia.org/wiki/XSLT>]. The GnuCash data file is well-formed XML, and it can therefore be run through an XSLT parser with an associated stylesheet. This allows one to transform the file to just about any format that can be designed, given a properly written stylesheet.

A few steps need to be followed. The writing of a stylesheet is a task for a different time, but if you can get one written, here’ s what you need to do:

1. Copy the GnuCash XML data file to a working file.

If the file was last modified by a version of GnuCash older than 2.0, then before you continue to the next step you will need to modify the working file’ s `<gnc-v2>` tag to read something like this:

```
<gnc-v2 xmlns:cd="http://www.gnucash.org/XML/cd"
xmlns:book="http://www.gnucash.org/XML/book"
xmlns:gnc="http://www.gnucash.org/XML/gnc"
xmlns:cmdty="http://www.gnucash.org/XML/cmdty"
xmlns:trn="http://www.gnucash.org/XML/trn"
xmlns:split="http://www.gnucash.org/XML/split"
xmlns:act="http://www.gnucash.org/XML/act"
xmlns:price="http://www.gnucash.org/XML/price"
xmlns:ts="http://www.gnucash.org/XML/ts"
xmlns:slot="http://www.gnucash.org/XML/kvpslot"
xmlns:cust="http://www.gnucash.org/XML/cust"
xmlns:entry="http://www.gnucash.org/XML/entry"
xmlns:lot="http://www.gnucash.org/XML/lot"
xmlns:invoice="http://www.gnucash.org/XML/invoice"
xmlns:owner="http://www.gnucash.org/XML/owner"
xmlns:job="http://www.gnucash.org/XML/job"
xmlns:billterm="http://www.gnucash.org/XML/billterm"
xmlns:bt-days="http://www.gnucash.org/XML/bt-days"
xmlns:sx="http://www.gnucash.org/XML/sx"
xmlns:fs="http://www.gnucash.org/XML/fs"
```

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

You can put pretty much anything you want behind the equal signs, but a unique URL is what is typically used.

2. Create an XSLT stylesheet containing the transformation your desire, or obtain one that's already written.

User Contributed XSLT Stylesheets

- in our repository: <https://github.com/Gnucash/gnucash/tree/stable/contrib/xslt>
 - in our wiki: https://wiki.gnucash.org/wiki/De/export_to_excel_xls_transform
3. Install an XSLT processor such as Saxon (https://en.wikipedia.org/wiki/Saxon_XSLT) or Xalan (https://en.wikipedia.org/wiki/Apache_Xalan). Any conforming processor [https://en.wikipedia.org/wiki/Category:XSLT_processors] 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 Apache OpenOffice/LibreOffice Calc (or vice-versa, for that matter). Such things as QIF ought to be a little less work.

Benefits are that you don't need to write a Scheme module or a new C routine to do this transformation. Anyone who knows or can learn XML and XSLT can perform this task. Not much harder, really, than writing a Web page....

Anyhow, I just wanted this tidbit to be captured somewhere permanently. The process works on 5.12 datafiles, and ought to work on earlier versions, too.

附录 B. Auxiliary File Formats

These are the formats of some auxiliary files used by GnuCash.

Check Format Files (`"*.chk"`)

Overview

The check format file is used to tell GnuCash how to print a check or checks onto a page of paper. This file first describes the overall layout of a page (number of checks, orientation, etc) and then describes the layout of the specific items on a single check. The file is organized as a typical Key/Value file used by many Linux applications. Keys/values pairs are grouped into sections that begin with the group name enclosed in square brackets.

GnuCash looks for check format files in two different locations when you bring up the check printing dialog. The first location is typically `/usr/share/gnucash/checks`, where check files distributed with the application can be found. The second location is the user private `~/.local/share/gnucash/checks`¹ directory. Users may add check formats at any time (even while GnuCash is running) simply by dropping a new `"*.chk"` file in this directory. The next time the check printing dialog is opened the new check format will appear in the list of available check formats.

Printing functions differently depending on the version of GTK that is installed on your system. When GnuCash is using a version of GTK prior to 2.10 all offsets are measured from the lower left corner of the page or check. When using GTK 2.10 or later, all offsets are measured from the upper left corner of the page or check.

Example file

A typical GnuCash check file is presented below. The contents of this file will be described in the next sections.

```
[Top]
Guid = 67b144d1-96a5-48d5-9337-0e1083bbf229
Title = Quicken/QuickBooks (tm) US-Letter
Rotation = 0.0
Translation = 0.0;4.0
Show_Grid = false
Show_Boxes = false
```

¹Up to GnuCash 2.6.21 it was `~/.gnucash/checks`

```
[Check Positions]
Height = 252.0
Names = Top;Middle;Bottom

[Check Items]
Type_1 = PAYEE
Coords_1 = 90.0;102.0;400.0;20.0

Type_2 = AMOUNT_WORDS
Coords_2 = 90.0;132.0

Type_3 = AMOUNT_NUMBER
Blocking_Chars_3 = true
Coords_3 = 500.0;102.0

Type_4 = DATE
Coords_4 = 500.0;67.0

Type_5 = NOTES
Coords_5 = 50.0;212.0
```

Field Descriptions

Top Group

This section of the check file describes the overall layout of a page of checks (or check) that goes into the printer.

B.1. Overall Page Description Fields

| Name | Type | Required | Description |
|------|--------|-----------|--|
| Guid | string | mandatory | The guid is used to uniquely identify a check format to GnuCash. It must be unique across the entire set of application supplied and user supplied check formats. If you copy an application check file as the basis of your own |

| Name | Type | Required | Description |
|----------------|---------|-----------|--|
| | | | check format, you must change this value. The <code>uuidgen</code> program may be used to generate these identifiers. |
| Title | string | mandatory | The title is used to uniquely identify a check format to the user. This value is presented verbatim in the check format list of the check printing dialog. If you copy an application check file as the basis of your own check format, you should change this value. The title may be any utf-8 string. |
| Font | string | optional | If supplied, this is the default font used to print all text items on this check. This field can contain any string that is acceptable by gtk as a font specifier. If this field is omitted, the default font is the font specified in the GnuCash preferences dialog. A typical string would be "sans 12" . |
| Blocking_Chars | boolean | optional | If supplied, this is the default used |

| Name | Type | Required | Description |
|------------|---------|----------|---|
| | | | when printing all TEXT items on this check. When set to true, will print *** before and after each text field on the check. Blocking characters are printed to protect check fields from alteration. For example, the amount field may be printed as ***100.00*** |
| DateFormat | boolean | optional | If supplied, this is the default used when printing all DATE items on this check. When set to true, will print the format of the DATE in 8 point type, centered and below the actual DATE. For example DDMMYYYY. |
| Rotation | double | optional | This value specified the rotation of the entire page (in degrees) around the origin point. For gtk versions prior to 2.10, the origin point is in the lower left corner of the page and rotation values increase in the counter-clockwise direction. For gtk version 2.10 and |

| Name | Type | Required | Description |
|-------------|-------------------|----------|---|
| | | | later, the origin point is in the upper left corner of the page and rotation values increase in the clockwise direction. Rotation of the page is applied before translation. |
| Translation | list of 2 doubles | optional | These values specify the x and y translation of the entire page (in points) relative to the origin point. For gtk versions prior to 2.10, the origin point is in the lower left corner of the page and translation values increase moving up and to the right. For gtk version 2.10 and later, the origin point is in the upper left corner of the page and translation values increase moving down and to the right. Rotation of the page is applied before translation. |
| Show_Grid | boolean | optional | If this value is set to true then GnuCash will draw a grid on the page, starting at the origin with the lines spaced every |

| Name | Type | Required | Description |
|------------|---------|----------|---|
| | | | 50 points. This can be helpful when creating a check format file. |
| Show_Boxes | boolean | optional | If this value is set to true then for each item where the width and height have been specified, GnuCash will draw a box showing location and maximum size of that item . This can be helpful when creating a check format file. |

The Blocking_Chars and DateFormat options are defined for all check formats in Edit->Preferences->Printing. It is recommended that these global options be set to false (the default), and that the options be set for individual Check Items as described below.

Check Positions Group

This group of items specifies how multiple checks are laid out on the same sheet of paper, and gives names to each of these check locations so that a user can specify which check location that GnuCash should print. This entire group of key/value pairs is optional, and should be omitted if the format file only specifies a single check per page of paper.

B.2. Multiple Checks Per Page Fields

| Name | Type | Required | Description |
|--------|--------|-----------|--|
| Height | double | mandatory | This field specifies the height of a single check on the page. If there are multiple checks per page then this item is mandatory. If there |

| Name | Type | Required | Description |
|-------|-----------------|-----------|--|
| | | | is only a single check per page, this entire section should be omitted. |
| Names | list of strings | mandatory | This field specifies the names of the check locations that can be printed on each page. These names represent the check positions starting from the top of the page and moving downward. The names are presented verbatim in the check position list of the check printing dialog. A typical value for this field is "Top;Middle;Bottom", but it could also be "First;Second;Third" or any other set of strings that clearly identify the check locations. If there are multiple checks per page then this item is mandatory. If there is only a single check per page, this entire section should be omitted. |

Check Items Group

This section specifies the individual items that are printed on the check. There is no limit to the number of items that may be present in this section, and any given type of item can be repeated multiple times. This allows for the printing of checks that have a

side stub, or for the one-per-page business checks that have both the check and multiple check stubs on the same page. For example, to print the payee name on a business check and on both stubs, simply specify three payee items with differing print coordinates.

Each key names in this section explicitly includes the item number to which it applies. E.G. The key named `[Type_1]` applies to the first item to be printed, and the key `[Coords_3]` applies to the third item to be printed. Item numbers start at one and increase sequentially. Any gap in the numbering sequence is interpreted by GnuCash as the end of the item list. Items are printed in the order of their item numbers, not in the order in which they appear in the file.

Each item specified must include a type declaration. The rest of the parameters for that item depend upon the particular type of that item. See 表 B.4 “Individual Check Item Types” for a list of valid item types and their required parameters.

B.3. Individual Check Item Fields

| Name | Type | Required | Description |
|----------|------------------------|-----------|---|
| Type_n | string | mandatory | This field specifies the type of a single item to be printed on a check. See 表 B.4 “Individual Check Item Types” for a list of valid item types. |
| Coords_n | list of 2 or 4 doubles | mandatory | This field specifies the coordinates where the item should be placed on a check, and optionally also specifies the width and height of the item. The numbers in order are the X and Y offset of the lower left corner of the item, and optionally the width and height of the item. If the width is supplied then the height must also be supplied, |

| Name | Type | Required | Description |
|--------|--------|----------|---|
| | | | <p>so this field will always contain two or four numbers. For gtk versions prior to 2.10, the origin point is in the lower left corner of the page and translation values increase moving up and to the right. For gtk version 2.10 and later, the origin point is in the upper left corner of the page and translation values increase moving down and to the right.</p> <p>Regardless of whether the origin is at the top or the bottom of the page, the coordinates always specify the lower left point of the item.</p> |
| Font_n | string | optional | <p>If supplied, this is the font used to print this specific text item. This field can contain any string that is acceptable by gtk</p> |

| Name | Type | Required | Description |
|------------|--------|----------|--|
| | | | as a font specifier. If this field is omitted, the default font is the font specified in the Top section of the check description file, or if that was omitted the font specified in the GnuCash preferences dialog. This field is only recognized when using gtk version 2.10 or later. |
| Align_n | string | optional | If supplied, this is the alignment used to print this specific text item. This field must contain one of the strings “left” , “center” or “right” . If this field is omitted, the text will be left aligned. This field is only recognized when using gtk version 2.10 or later. |
| Text_n | string | optional | This field is only used when the item type is TEXT. It specifies the utf-8 text that should be printed on the check. |
| Filename_n | string | optional | This field is only used when the item type is PICTURE. It specifies the |

| Name | Type | Required | Description |
|------------------|---------|----------|---|
| | | | filename of the image that should be printed on the check. The string may specify either an absolute path name or as a relative path name. If a relative path name is specified, GnuCash first looks in the application check format folder (typically <code>"/usr/share/gnucash/checks"</code>) for the image file, and if it is not found there then it looks in the user private <code>"~/.local/share/gnucash/checks"</code> ¹ directory for the image. This field is only recognized when using gtk version 2.10 or later. |
| Blocking_Chars_n | boolean | optional | If supplied, this will set the print <code>Blocking_Chars</code> option for this item. |
| DateFormat_n | boolean | optional | If supplied, this will set the print <code>DateFormat</code> option for this item. |

These are the individual items that can be printed on a check. All items require the coordinates on the page where the item should be printed. The majority of these items result in text being printed on the page, and these items may have individual font and alignments specified. For example, the numerical amount of a check could be printed

right justified while everything else is printed left justified. Other types may have unique parameters.

B.4. Individual Check Item Types

| Name | Required Fields | Optional Fields | Description |
|--------------|-----------------|---|---|
| PAYEE | Coords | Font Align Blocking_Chars | This type value tells GnuCash to print the check payee name at the specified coordinates. |
| DATE | Coords | Font Align Blocking_Chars DateFormat | This type value tells GnuCash to print the check date at the specified coordinates. |
| NOTES | Coords | Font Align Blocking_Chars | This type value tells GnuCash to print the transaction notes field at the specified coordinates. |
| CHECK_NUMBER | Coords | Font Align Blocking_Chars | This type value tells GnuCash to print the check number at the specified coordinates. The check number reflects the book option selection under [File] → [Properties] for number source (transaction number or anchor-split action - see Use Split Action Field for Number [https://www.gnucash.org/docs/v5/C/gnucash-manual/book-options.html#num-action-book-option]) |

| Name | Required Fields | Optional Fields | Description |
|--------|-----------------|---------------------------------|--|
| | | | in the Book Options section of the GnuCash Manual). |
| MEMO | Coords | Font Align Blocking_Chars | This type value tells GnuCash to print the split memo field at the specified coordinates. |
| ACTION | Coords | Font Align Blocking_Chars | <p>This type value tells GnuCash to print the split action field at the specified coordinates.</p> <p>However, the printed field reflects the book option selection under [File] → [Properties] for number source (transaction number or anchor-split action - see Use Split Action Field for Number [https://www.gnucash.org/docs/v5/C/gnucash-manual/book-options.html#num-action-book-option] in the Book Options section of the GnuCash Manual).</p> <p>If number source for the book is specified as anchor-split action, this field will instead print the transaction number field.</p> |

| Name | Required Fields | Optional Fields | Description |
|----------------|-----------------|---------------------------------|---|
| AMOUNT_WORDS | Coords | Font Align Blocking_Chars | This type value tells GnuCash to print the check amount in words at the specified coordinates. The amount will appear similar to the string "One thousand, two hundred thirty four and 56/100". |
| AMOUNT_NUMBER | Coords | Font Align Blocking_Chars | This type value tells GnuCash to print the check amount in numbers at the specified coordinates. The amount will appear similar to the number "\$1,234.56". |
| ADDRESS | Coords | Font Align Blocking_Chars | This type value tells GnuCash to print the address at the specified coordinates. |
| SPLITS_ACCOUNT | Coords | Font Align Blocking_Chars | This type value tells GnuCash to print the account names for each split entry stating at the specified coordinates. See the note on splits printing. |
| SPLITS_AMOUNT | Coords | Font Align Blocking_Chars | This type value tells GnuCash to print the amount for each split entry stating at the specified coordinates. |

| Name | Required Fields | Optional Fields | Description |
|-------------|------------------|---------------------------------|---|
| | | | Amounts are printed with currency symbols. See the note on splits printing. |
| SPLITS_MEMO | Coords | Font Align Blocking_Chars | This type value tells GnuCash to print the memo text for each split entry stating at the specified coordinates. See the note on splits printing. |
| TEXT | Coords, Text | Font Align Blocking_Chars | This type value tells GnuCash to print an arbitrary string at the specified coordinates. The string to be printed is specified with the <code>Text_n</code> key. |
| PICTURE | Coords, Filename | (none) | This type value tells GnuCash to print an image at the specified coordinates. The image to be printed is specified with the <code>Filename_n</code> key. This type is only recognized when using gtk version 2.10 or later. |

SPLIT items include all split entries for the transaction except for the split that applies to the current account register (referred to as the anchor-split). This is usually the last split listed when splits are displayed in the register. The coordinate location defines the lower left location for the split information.

Creating Check Format Files

Creating your own check format file is a fairly simple task. The easiest way to start is to copy an existing check format file from the application directory (typically `/usr/share/gnucash/checks`) to the directory `~/.local/share/gnucash/checks`¹. Make sure to change the guid so the new file will be accepted by gnucash, and change the title to something descriptive. Then change or add individual item fields as necessary. You can also create a new check file by clicking the Save Format button on the [Custom format] page of the check printing dialog.

Key names are case sensitive. If you are having problems with a check format file, ensure that all key names have capital letters as documented above.

附录 C. GNU Free Documentation License

Version 1.1, March 2000

版权 © 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.

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 [<http://www.gnu.org/fsf/fsf.html>] 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/> [<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 [<https://www.gnu.org/copyleft/gpl.html>], to permit their use in free software.