

# Ledger-SMB Manual v. 1.2

The LedgerSMB Core Team

April 26, 2007

Copyright ©2006 The LedgerSMB Core Team. Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.2 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. A copy of the license is included in the section entitled "GNU Free Documentation License" (Appendix E).

## Contents

<b>I</b>	<b>Ledger-SMB and Business Processes</b>	<b>7</b>
<b>1</b>	<b>Introduction to Ledger-SMB</b>	<b>7</b>
1.1	Why Ledger-SMB . . . . .	7
1.1.1	Advantages of Ledger-SMB . . . . .	7
1.1.2	Key Features . . . . .	7
1.2	Limitations of Ledger-SMB . . . . .	9
1.3	System Requirements of Ledger-SMB . . . . .	9
<b>2</b>	<b>User Account and Database Administration Basics</b>	<b>9</b>
2.1	Companies and Datasets . . . . .	10
2.2	How to Create a User . . . . .	10
2.3	Permissions . . . . .	10
2.4	User Account Types . . . . .	10
2.5	Other Features . . . . .	10
<b>3</b>	<b>Chart of Accounts</b>	<b>10</b>
3.1	Introduction to Double Entry Bookkeeping . . . . .	11
3.1.1	Business Entity . . . . .	11
3.1.2	Double Entry . . . . .	11
3.1.3	Accounts . . . . .	11
3.1.4	Debits and Credits . . . . .	12
3.1.5	Accrual . . . . .	12
3.1.6	References . . . . .	12
3.2	General Guidelines on Numbering Accounts . . . . .	13
3.3	Adding/Modifying Accounts . . . . .	13
3.4	Listing Account Balances and Transactions . . . . .	13

<b>4</b>	<b>Administration</b>	<b>14</b>
4.1	Taxes, Defaults, and Preferences . . . . .	14
4.1.1	Adding A Sales Tax Account . . . . .	14
4.1.2	Setting a Sales Tax Amount . . . . .	14
4.1.3	Default Account Setup . . . . .	14
4.1.4	Currency Setup . . . . .	14
4.1.5	Sequence Settings . . . . .	14
4.2	Audit Control . . . . .	15
4.2.1	Explaining transaction reversal . . . . .	15
4.2.2	Close books option . . . . .	15
4.2.3	Audit Trails . . . . .	15
4.3	Departments . . . . .	16
4.3.1	Cost v Profit Centers. . . . .	16
4.4	Warehouses . . . . .	16
4.5	Languages . . . . .	16
4.6	Types of Businesses . . . . .	16
4.7	Misc. . . . .	16
4.7.1	GIFI . . . . .	16
4.7.2	SIC . . . . .	16
4.7.3	Overview of Template Editing . . . . .	17
4.7.4	Year-end . . . . .	17
4.8	Options in the ledger-smb.conf . . . . .	17
<b>5</b>	<b>Goods and Services</b>	<b>17</b>
5.1	Basic Terms . . . . .	18
5.2	The Price Matrix . . . . .	18
5.3	Pricegroups . . . . .	18
5.4	Groups . . . . .	18
5.5	Labor/Overhead . . . . .	18
5.6	Services . . . . .	18
5.6.1	Shipping and Handling as a Service . . . . .	18
5.7	Parts . . . . .	18
5.8	Assemblies and Manufacturing . . . . .	19
5.8.1	Stocking Assemblies . . . . .	19
5.9	Reporting . . . . .	19
5.9.1	All Items and Parts Reports . . . . .	19
5.9.2	Requirements . . . . .	19
5.9.3	Services and Labor . . . . .	19
5.9.4	Assemblies . . . . .	19
5.9.5	Groups and Pricegroups . . . . .	20
5.10	Translations . . . . .	20
5.11	How Cost of Goods Sold is tracked . . . . .	20
<b>6</b>	<b>AP</b>	<b>20</b>
6.1	Basic AP Concepts . . . . .	20
6.2	Vendors . . . . .	20
6.3	AP Transactions . . . . .	20
6.4	AP Invoices . . . . .	21
6.4.1	Correcting an AP Invoice . . . . .	21
6.5	Cash payment And Check Printing . . . . .	21
6.5.1	Rapid Payment Entry Screen . . . . .	21
6.6	Transaction/Invoice Reporting . . . . .	22

6.6.1	Transactions Report . . . . .	22
6.6.2	Outstanding Report . . . . .	22
6.6.3	AP Aging Report . . . . .	22
6.6.4	Tax Paid and Non-taxable Report . . . . .	22
6.7	Vendor Reporting . . . . .	22
6.7.1	Vendor Search . . . . .	22
6.7.2	Vendor History . . . . .	22
<b>7</b>	<b>AR</b>	<b>23</b>
7.1	Customers . . . . .	23
7.1.1	Customer Price Matrix . . . . .	23
7.2	AR Transactions . . . . .	23
7.3	AR Invoices . . . . .	23
7.4	Cash Receipt . . . . .	23
7.4.1	Cash Receipts for multiple customers . . . . .	23
7.5	AR Transaction Reporting . . . . .	23
7.5.1	AR Transactions Report . . . . .	24
7.5.2	AR Aging Report . . . . .	24
7.6	Customer Reporting . . . . .	24
<b>8</b>	<b>Projects</b>	<b>24</b>
8.1	Project Basics . . . . .	24
8.2	Timecards . . . . .	24
8.3	Projects and Invoices . . . . .	24
8.4	Reporting . . . . .	24
8.4.1	Timecard Reporting . . . . .	24
8.4.2	Project Transaction Reporting . . . . .	24
8.4.3	List of Projects . . . . .	25
8.5	Possibilities for Using Projects . . . . .	25
<b>9</b>	<b>Quotations and Order Management</b>	<b>25</b>
9.1	Sales Orders . . . . .	25
9.2	Quotations . . . . .	25
9.3	Shipping . . . . .	25
9.4	AR Work Flow . . . . .	25
9.4.1	Service Example . . . . .	25
9.4.2	Single Warehouse Example . . . . .	26
9.4.3	Multiple Warehouse Example . . . . .	26
9.5	Requests for Quotation (RFQ) . . . . .	26
9.6	Purchase Orders . . . . .	29
9.7	Receiving . . . . .	29
9.8	AP Work Flow . . . . .	29
9.8.1	Bookkeeper entering the received items, order completed in full . . . . .	29
9.8.2	Bookkeeper entering received items, order completed in part . . . . .	29
9.8.3	Receiving staff entering items . . . . .	31
9.9	Generation and Consolidation . . . . .	31
9.9.1	Generation . . . . .	31
9.9.2	Consolidation . . . . .	32
9.10	Reporting . . . . .	32
9.11	Shipping Module: Transferring Inventory between Warehouses . . . . .	32
<b>10</b>	<b>HR</b>	<b>32</b>

<b>11 POS</b>	<b>32</b>
11.1 Sales Screen . . . . .	32
11.2 Possibilities for Data Entry . . . . .	32
11.3 Hardware Support . . . . .	33
11.4 Reports . . . . .	33
11.4.1 Open Invoices . . . . .	33
11.4.2 Receipts . . . . .	33
<b>12 General Ledger</b>	<b>33</b>
12.1 GL Basics . . . . .	33
12.1.1 Paper-based accounting systems and the GL . . . . .	33
12.1.2 Double Entry Examples on Paper . . . . .	34
12.1.3 The GL in Ledger-SMB . . . . .	34
12.2 Cash Transfer . . . . .	34
12.3 GL Transactions . . . . .	35
12.4 Payroll as a GL transaction . . . . .	35
12.5 Reconciliation . . . . .	35
12.6 Reports . . . . .	36
12.6.1 GL as access to almost everything else . . . . .	36
<b>13 Recurring Transactions</b>	<b>36</b>
<b>14 Financial Statements and Reports</b>	<b>36</b>
14.1 Cash v. Accrual Basis . . . . .	37
14.2 Viewing the Chart of Accounts and Transactions . . . . .	37
14.3 Trial Balance . . . . .	37
14.3.1 The Paper-based function of a Trial Balance . . . . .	37
14.3.2 Running the Trial Balance Report . . . . .	37
14.3.3 What if the Trial Balance doesn't Balance? . . . . .	37
14.3.4 Trial Balance as a Summary of Account Activity . . . . .	37
14.3.5 Trial Balance as a Budget Planning Tool . . . . .	38
14.4 Income Statement . . . . .	38
14.4.1 Uses of an Income Statement . . . . .	38
14.5 Balance Sheet . . . . .	39
14.6 What if the Balance Sheet doesn't balance? . . . . .	39
14.7 No Statement of Owner Equity? . . . . .	39
<b>15 The Template System</b>	<b>39</b>
15.1 Text Templates . . . . .	40
15.2 HTML Templates . . . . .	40
15.3 L <sup>A</sup> T <sub>E</sub> X Templates . . . . .	41
15.3.1 What is L <sup>A</sup> T <sub>E</sub> X ? . . . . .	41
15.3.2 Using L <sub>Y</sub> X to Edit L <sup>A</sup> T <sub>E</sub> X Templates . . . . .	41
15.4 Customizing Logos . . . . .	42
15.5 How are They Stored in the Filesystem? . . . . .	42
15.6 Upgrade Issues . . . . .	42
<b>16 An Introduction to the CLI</b>	<b>42</b>
16.1 Conventions . . . . .	42
16.2 Preliminaries . . . . .	42
16.3 First Script: lsmb01-cli-example.sh . . . . .	43
16.3.1 Script 1 (Bash) . . . . .	43
16.4 Second Script: lsmb02-cli-example.pl . . . . .	44

16.4.1	Script 2 (Perl)	45
<b>II</b>	<b>Technical Overview</b>	<b>49</b>
<b>17</b>	<b>Basic Architecture</b>	<b>49</b>
17.1	The Software Stack	49
17.2	Capacity Planning	50
17.2.1	Scalability Strategies	50
17.2.2	Database Maintenance	50
17.2.3	Known issues	51
<b>18</b>	<b>Customization Possibilities</b>	<b>51</b>
18.1	Brief Guide to the Source Code	51
18.2	Data Entry Screens	52
18.2.1	Examples	52
18.3	Extensions	52
18.3.1	Examples	52
18.4	Templates	52
18.4.1	Examples	52
18.5	Reports	52
18.5.1	Examples	53
<b>19</b>	<b>Integration Possibilities</b>	<b>53</b>
19.1	Reporting Tools	53
19.1.1	Examples	53
19.2	Line of Business Tools on PostgreSQL	53
19.2.1	Known Issues	53
19.2.2	Strategies	53
19.2.3	Examples	54
19.3	Line of Business Tools on other RDBMS's	54
19.3.1	Strategies	54
19.3.2	Integration Products and Open Source Projects	54
<b>20</b>	<b>Customization Guide</b>	<b>54</b>
20.1	General Information	54
20.2	Customizing Templates	55
20.2.1	Page Breaks in L <sup>A</sup> T <sub>E</sub> X	55
20.2.2	Conditionals	55
20.2.3	Loops	55
20.2.4	File Inclusion	56
20.2.5	Cross-referencing and multiple passes of L <sup>A</sup> T <sub>E</sub> X	56
20.2.6	Variable Substitution	56
20.3	Customizing Forms	56
20.4	Customizing Modules	57
20.4.1	Database Access	57
20.5	CLI Examples	57
<b>III</b>	<b>Appendix</b>	<b>58</b>
<b>A</b>	<b>Where to Go for More Information</b>	<b>58</b>

<b>B Quick Tips</b>	<b>58</b>
B.1 Understanding Shipping Addresses and Carriers . . . . .	58
B.2 Handling bad debts . . . . .	58
<b>C Step by Steps for Vertical Markets</b>	<b>58</b>
C.1 Common Installation Errors . . . . .	58
C.2 Retail With Light Manufacturing . . . . .	58
<b>D Glossary</b>	<b>60</b>
<b>E GNU Free Documentation License</b>	<b>61</b>
1. APPLICABILITY AND DEFINITIONS . . . . .	61
2. VERBATIM COPYING . . . . .	62
3. COPYING IN QUANTITY . . . . .	62
4. MODIFICATIONS . . . . .	63
5. COMBINING DOCUMENTS . . . . .	64
6. COLLECTIONS OF DOCUMENTS . . . . .	65
7. AGGREGATION WITH INDEPENDENT WORKS . . . . .	65
8. TRANSLATION . . . . .	65
9. TERMINATION . . . . .	65
10. FUTURE REVISIONS OF THIS LICENSE . . . . .	65
ADDENDUM: How to use this License for your documents . . . . .	66

## List of Figures

1 Simple AR Service Invoice Workflow Example . . . . .	26
2 AR Workflow with Shipping . . . . .	27
3 Complex AR Workflow with Shipping . . . . .	28
4 Simple AP Workflow . . . . .	29
5 AP Workflow with Receiving . . . . .	30
6 Complex AP Workflow . . . . .	31
7 Payroll as a GL Transaction (Purely fictitious numbers) . . . . .	35
8 The Ledger-SMB software stack in a Typical Implementation . . . . .	49

## Part I

# Ledger-SMB and Business Processes

## 1 Introduction to Ledger-SMB

### 1.1 Why Ledger-SMB

#### 1.1.1 Advantages of Ledger-SMB

- Flexibility and Central Management
- Accessibility over the Internet (for some users)
- Relatively open data format
- Integration with other tools
- Excellent accounting options for Linux users
- Open Source
- Flexible, open framework that can be extended or modified to fit your business.
- Security-conscious development community.

#### 1.1.2 Key Features

- Accounts Receivable
  - Track sales by customer
  - Issue Invoices, Statements, Receipts, and more
  - Do job costing and time entry for customer projects
  - Manage sales orders and quotations
  - Ship items from sales orders
- Accounts Payable
  - Track purchases and debts by vendor
  - Issue RFQ's Purchase Orders, etc.
  - Track items received from purchase orders
- Budgeting
  - Track expenditures and income across multiple departments
  - Track all transactions across departments
- Check Printing
  - Customize template for any check form
- General Ledger
- Inventory Management

- Track sales and orders of parts
  - Track cost of goods sold using First In/First Out method
  - List all parts below reorder point
  - Track ordering requirements
  - Track, ship, receive, and transfer parts to and from multiple warehouses
- Localization
  - Provide Localized Translations for Part Descriptions
  - Provide Localized Templates for Invoices, Orders, Checks, and more
  - Select language per customer, invoice, order, etc.
- Manufacturing
  - Track cost of goods sold for manufactured goods (assemblies)
  - Create assemblies and stock assemblies, tracking materials on hand
- Multi-company/Multiuser
  - One isolated database per company
  - Users can have localized systems independent of company data set
- Point of Sale
  - Run multiple cash registers against main Ledger-SMB installation
  - Suitable for retail stores and more
  - Credit card processing via TrustCommerce
  - Supports some POS hardware out of the box including:
    - \* Logic Controls PD3000 pole displays (serial or parallel)
    - \* Basic text-based receipt printers
    - \* Keyboard wedge barcode scanners
    - \* Keyboard wedge magnetic card readers
    - \* Printer-attached cash drawers
- Price Matrix
  - Track different prices for vendors and customers across the board
  - Provide discounts to groups of customers per item or across the board
  - Store vendors' prices independent of the other last cost in the parts record
- Reporting
  - Supports all basic financial statements
  - Easily display customer history, sales data, and additional information
  - Open framework allows for ODBC connections to be used to generate reports using third party reporting tools.
- Tax
  - Supports Retail Sales Tax and Value Added Tax type systems
  - Flexible framework allows one to customize reports to change the tax reporting framework to meet any local requirement.



## 1.2 Limitations of Ledger-SMB

- No payroll module (Payroll must be done manually)
- Some integration limitations
- Further development/maintenance requires a knowledge of a relatively broad range of technologies

## 1.3 System Requirements of Ledger-SMB

- PostgreSQL
- A CGI-enabled Web Server (for example, Apache)
- Perl 5.8.x
- An operating system which supports the above software (usually Linux, though Windows, MacOS X, etc. do work)
- L<sup>A</sup>T<sub>E</sub>X (optional) is required to create PDF or Postscript invoices
- The following CPAN modules:
  - Data::Dumper
  - Locale::Maketext
  - Locale::Maketext::Lexicon
  - MIME::Base64
  - Digest::MD5
  - HTML::Entities
  - DBI
  - DBD::Pg
  - Math::BigFloat
  - IO::File
  - Encode
  - Locale::Country
  - Locale::Language
  - Time::Local
  - Cwd
  - Config::Std
  - MIME::Lite

## 2 User Account and Database Administration Basics

These functions are accessed via the admin.pl script in the installed directory of Ledger-SMB.

## 2.1 Companies and Datasets

Ledger-SMB stores its information in locale-specific data sets. When a dataset is created, it sets various defaults such as currency, a basic chart of accounts setup, and so forth. Note that the default setup is for Canada, where the author of the software resides.

Datasets are stored as PostgreSQL databases. The application is designed with the idea that each dataset will represent exactly one company. If a customer is working with multiple companies, he/she must create a dataset to for each.

When creating a dataset, the application asks for both a username and a superusername. If the superuser's information is not filled in, Ledger-SMB will attempt to populate an existing dataset, but if this information is filled in, the program will log into the PostgreSQL cluster with the superusername and password, create the database, and attempt to add Plpgsql to it.

## 2.2 How to Create a User

Users are created by going to the admin.pl page and clicking on "Add User." One then fills out the form and when it is saved, the user is created.

## 2.3 Permissions

The permissions system is not rigorously enforced within Ledger-SMB, in the sense that the permissions API is generally not used in the application itself. Instead permissions are used to enable/disable menu options. Setting an enforcement of such permissions would require some custom programming at the present time. Most organizations, however, find that the current system is adequate.

The checkboxes which are marked enable menu entries. Those that are unchecked disable those entries on the menu.

## 2.4 User Account Types

- User is a general user of the system
- Managers often are able to see a larger amount of data
- Administrators have full access to the system

## 2.5 Other Features

- Lock System allows one to lock users out of the system while maintenance is performed. This is only necessary during upgrades or maintenance which results in the RDBMS being offline.
- Change Admin Password changes the administrative password.
- Logout terminates the admin session.

## 3 Chart of Accounts

The Chart of Accounts provides a basic overview of the logical structure of the accounting program. One can customize this chart to allow for tracking of different sorts of information.

## 3.1 Introduction to Double Entry Bookkeeping

In order to set up your chart of accounts in LedgerSMB you will need to understand a bit about double entry bookkeeping. This section provides a brief overview of the essential concepts. There is a list of references for further reading at the end.

### 3.1.1 Business Entity

You don't want to mix your personal expenses and income with that of the business or you will not be able to tell how much money it is making (if any). For the same reason you will want to keep track of how much money you put into and take out of the business so you will want to set up a completely separate set of records for it and treat it almost as if it had a life of its own.

### 3.1.2 Double Entry

Examples:

- When you buy you pay money and receive goods.
- When you sell you get money and give goods.
- When you borrow you get money and give a promise to pay it back.
- When you lend you give money and get a promise to pay it back.
- When you sell on credit you give goods and get a promise to pay.
- When you buy on credit you give a promise to pay and get goods.

You need to record both sides of each transaction: thus double entry. Furthermore, you want to organize your entries, recording those having to do with money in one place, value of goods bought and sold in another, money owed in yet another, etc. Hence you create accounts, and record each half of each transaction in an appropriate account. Of course, you won't have to actually record the amount in more than one place yourself: the program takes care of that.

### 3.1.3 Accounts

**Assets** Valuable stuff the business owns such as money and goods available for sale

**Liabilities** Debts owned by the business such as bank loans and unpaid bills

**Equity or Capital** What would be left for the owner if all the assets were converted to money and all the liabilities paid off ("Share Capital" on the LedgerSMB default chart of accounts: not to be confused with "Capital Assets".)

**Revenue** Income from business activity: increases Equity

**Expense** The light bill, the cost of goods sold, etc: decreases Equity

All other accounts are subdivisions of these. The relationship between the top-level accounts is often stated in the form of the Accounting Equation (don't worry: you won't have to solve it):

$$\text{Assets} = \text{Liabilities} + \text{Equity} + (\text{Revenue} - \text{Expense})$$

You won't actually use this equation while doing your bookkeeping, but it's a useful tool for understanding how the system works.

### 3.1.4 Debits and Credits

Traditional paper accounting systems used a two-column form in which all increases went in one column and all decreases in the other. For asset and expense accounts increases went in the left column and decreases in the right. For liability and capital accounts decreases went in the left column and increases in the right. Looking at the accounting equation we see that assets are on the left, so it is logical that asset increases would go on the left. Liabilities, capital, and revenue are on the right so it is logical that their increase would go on the right. Expenses, however, are on the right, so why do their increases go on the left? Because expenses are subtracted from the right side of the equation and so expense increases decrease the right side of the equation.

Entries in the left column of the traditional form are called debits, while entries on the right are called credits. Neither is "negative".

- Debits increase assets
- Debits increase expense
- Credits increase liabilities
- Credits increase capital
- Credits increase revenue

Examples:

You go to the bank and make a deposit. The teller tells you that he is going to credit your account. This is correct: your account is money the bank owes you and so is a liability from their point of view. Your deposit increased this liability and so they will credit it. They will make an equal debit to their cash account. When you return you will debit your bank deposits account because you have increased that asset and credit cash on hand because you have decreased that one.

### 3.1.5 Accrual

Early accounting systems were usually run on a cash basis. One generally did not consider money owed to affect the financial health of a company, so expenses posted when paid as did income.

The problem with this approach is that it becomes very difficult or impossible to truly understand the exact nature of the financial health of a business. One cannot get the full picture of the financial health of a business because outstanding debts are not considered. Furthermore, this does not allow for revenue to be tied to cost effectively, so it becomes difficult to assess how profitable a given activity truly is.

To solve this problem, accrual-based systems were designed. The basic principle is that income and expense should be posted as they are incurred, or accrued. This allows one to track income relative to expense for specific projects or operations, and make better decisions about which activities will help one maximize profitability.

To show how these systems differ, imagine that you bill a customer for time and materials for a project you have just completed. The customer pays the bill after 30 days. In a cash based system, you would post the income at the time when the customer pays, while in an accrual system, the income is posted at the time when the project is completed.

### 3.1.6 References

<http://www.accounting-and-bookkeeping-tips.com/learning-accounting/accounting-basics-credit.htm>

Discussion of debits and credits as well as links to other accounting subjects.

<http://www.computer-consulting.com/accttips.htm>  
Discussion of double entry bookkeeping.

<http://www.minnesota.com/~tom/sql-ledger/howtos/>  
A short glossary, some links, and a FAQ (which makes the "credit=negative number" error). The FAQ focuses on SQL-Ledger, LedgerSMB's ancestor.

<http://bitscape.com/pub2/etp/sql-ledger-notes#expenses>  
Some notes on using SQL-Ledger (LedgerSMB's ancestor).

[http://en.wikipedia.org/wiki/List\\_of\\_accounting\\_topics](http://en.wikipedia.org/wiki/List_of_accounting_topics)  
Wikipedia articles on accounting.

<http://www.bized.ac.uk/learn/accounting/financial/index.htm>  
Basic accounting tutorial.

[http://www.asset-analysis.com/glossary/glo\\_index.html](http://www.asset-analysis.com/glossary/glo_index.html)  
Financial dictionary and glossary.

<http://www.geocities.com/chapleaucree/educational/FinanceHandbook.html>  
Financial glossary.

<http://www.quickmba.com/accounting/fin/>  
Explanation of fundamentals of accounting, including good discussions of debits and credits and of double-entry.

### **3.2 General Guidelines on Numbering Accounts**

In general, most drop-down boxes in Ledger-SMB order the accounts by account number. Therefore by setting appropriate account numbers, one can affect the default values.

A second consideration is to try to keep things under each heading appropriate to that heading. Thus setting an account number for a bank loan account in the assets category is not generally advisable.

### **3.3 Adding/Modifying Accounts**

These features are listed under System->Chart of Accounts. One can list the accounts and click on the account number to modify them or click on the "add account" option to create new accounts.

- Headings are just broad categories and do not store values themselves, while accounts are used to store the transactional information.
- One cannot have an account that is a summary account (like AR) and also has another function.
- GIFI is mostly of interest to Canadian customers but it can be used to create reports of account hierarchies.

### **3.4 Listing Account Balances and Transactions**

One can list the account balances via the Reports->Chart of Accounts report. Clicking on the account number will provide a ledger for that account.

## 4 Administration

This section covers other (non-Chart of Accounts) aspects to the setup of the Ledger-SMB accounting package. These are generally accessed in the System submenu.

### 4.1 Taxes, Defaults, and Preferences

One of the new features in 1.2 is the modular sales tax system and the simple sales tax module. This allows one to install different tax modules and then select which taxes are applied by which programming modules. The sales tax module has access to everything on the submitted form so it is able to make complex determinations on what is taxable based on arbitrary criteria.

The tax rules drop-down box allows one to select any installed tax module (LedgerSMB 1.2 ships only with the simple module), while the ordering is an integer which allows one to specify a tax run which occurs on the form after any rules with lower entries in this box. This allows for compounding of sales tax (for example, when PST applies to the total and GST as well).

#### 4.1.1 Adding A Sales Tax Account

Sales Tax is collected on behalf of a state or national government by the individual store. Thus a sales tax account is a liability— it represents money owed by the business to the government.

To add a sales tax account, create an account in the Chart of Accounts as a liability account, check all of the "tax" checkboxes.

Once this account is created, one can set the tax amount.

#### 4.1.2 Setting a Sales Tax Amount

Go to System->Defaults and the tax account will be listed near the bottom of the page. The rate can be set there.

#### 4.1.3 Default Account Setup

These accounts are the default accounts for part creation and foreign exchange tracking.

#### 4.1.4 Currency Setup

The US accounts list this as USD:CAD:EUR. One can add other currencies in here, such as IDR (Indonesian Rupiah), etc. Currencies are separated by colons.

#### 4.1.5 Sequence Settings

These sequences are used to generate user identifiers for quotations, invoices, and the like. If an identifier is not added, the next number will be used.

A common application is to set invoices, etc. to start at 1000 in order to hide the number of issued invoices from a customer.

Leading zeros are preserved. Other special values which can be embedded using `<?lsm ?>` tags include:

**DATE** expands to the current date

**YYMMDD** expands to a six-digit version of the date. The components of this date can be re-arranged in any order, so MMDDYY, DDMMYY, or even just MMY are all options.

**NAME** expands to the name of the customer or vendor

**BUSINESS** expands to the type of business assigned to the customer or vendor.

**DESCRIPTION** expands to the description of the part. Valid only for parts.

**ITEM** expands to the item field. Valid only for parts.

**PERISCOPE** expands to the partsgroup. Valid only for parts.

**PHONE** expands to the telephone number for customers and vendors.

## 4.2 Audit Control

Auditability is a core concern of the architects of any accounting system. Such ensures that any modification to the accounting information leaves a trail which can be followed to determine the nature of the change. Audits can help ensure that the data in the accounting system is meaningful and accurate, and that no foul play (such as embezzlement) is occurring.

### 4.2.1 Explaining transaction reversal

In paper accounting systems, it was necessary to have a means to authoritatively track corrections of mistakes. The means by which this was done was known as "transaction reversal."

When a mistake would be made, one would then reverse the transaction and then enter it in correctly. For example, let us say that an office was renting space for \$300 per month. Let us say that they inadvertently entered it in as a \$200 expense.

The original transaction would be:

Account	Debit	Credit
5760 Rent	\$200	
2100 Accounts Payable		\$200

The reversal would be:

Account	Debit	Credit
5760 Rent		\$200
2100 Accounts Payable	\$200	

This would be followed by re-entering the rent data with the correct numbers. This was meant to ensure that one did not erase data from the accounting books (and as such that erasing data would be a sign of foul play).

Ledger-SMB has a capability to require such reversals if the business deems this to be necessary. When this option is enabled, existing transactions cannot be modified and one will need to post reversing transactions to void existing transactions before posting corrected ones.

Most accountants prefer this means to other audit trails because it is well proven and understood by them.

### 4.2.2 Close books option

You cannot alter a transaction that was entered before the closing date.

### 4.2.3 Audit Trails

This option stores additional information in the database to help auditors trace individual transactions. The information stored, however, is limited and it is intended to be supplemental to other auditing facilities.

The information added includes which table stored the record, which employee entered the information, which form was used, and what the action was. No direct financial information is included.

### **4.3 Departments**

Departments are logical divisions of a business. They allow for budgets to be prepared for the individual department as well as the business as a whole. This allows larger businesses to use Ledger-SMB to meet their needs.

#### **4.3.1 Cost v Profit Centers.**

In general business units are divided into cost and profit centers. Cost centers are generally regarded as business units where the business expects to lose money and profit centers are where they expect to gain money. For example, the legal department in most companies is a cost center.

One of the serious misunderstandings people run up against is that Ledger-SMB tends to more narrowly define cost and profit centers than most businesses do. In Ledger-SMB a cost center is any department of the business that does not issue AR transactions. Although many businesses may have cost centers (like technical support) where customer fees may subsidize the cost of providing the service, in Ledger-SMB, these are profit centers.

Ledger-SMB will not allow cost centers to be associated with AR transactions. So if you want this functionality, you must create the department as a profit center.

### **4.4 Warehouses**

Ledger-SMB has the ability to track inventory by warehouse. Inventory items can be moved between warehouses, and shipped from any warehouse where the item is in stock. We will explore this concept more later.

### **4.5 Languages**

Languages allow for goods and services to be translated so that one can maintain offices in different countries and allow for different goods and service descriptions to be translated to different languages for localization purposes.

### **4.6 Types of Businesses**

One can create types of businesses and then give them discounts across the board. For example, one might give a firm that uses one's services as a subcontractor a 10% discount or more.

### **4.7 Misc.**

#### **4.7.1 GIFI**

GIFI is a requirement for Canadian customers. This feature allows one to link accounts with Canadian tax codes to simplify the reporting process. Some European countries now use a similar system.

People that don't otherwise have a use for GIFI can use it to create reports which aggregate accounts together.

#### **4.7.2 SIC**

Standard Industrial Classification is a way of tracking the type of business that a vendor or customer is in. For example, an accountant would have an SIC of 8721 while a graphic design firm would have an SIC of 7336. The classification is hierarchical so one could use this field for custom reporting and marketing purposes.



### 4.7.3 Overview of Template Editing

The templates for invoices, orders, and the like can be edited from within Ledger-SMB. The submenus within the System submenu such as HTML Templates, Text Templates and L<sup>A</sup>T<sub>E</sub>X templates provide access to this functionality.

### 4.7.4 Year-end

Although the Year-end functionality in Ledger-SMB is very useful, it does not entirely make the process simple and painless. One must still manually enter adjustments prior to closing the books. The extent to which these adjustments are necessary for any given business is a matter best discussed with an accountant.

The standard way books are normally closed at the end of the year is by moving all adjusted<sup>1</sup> income and expenses to an equity account usually called "Retained Earnings." Assets and liabilities are not moved. Equity drawing/dividend accounts are also moved, but the investment accounts are not. The reasoning behind this process is that one wants a permanent record of the amount invested in a business, but any dividends ought not to count against their recipients when new investors are brought on board.

Ledger-SMB automatically moves all income and expense into the specified year-end/retained earnings account. It does not move the drawing account, and this must be done manually, nor does it automate the process of making adjustments.

Contrary to its name, this function can close the books at any time, though this would likely be of limited use.

## 4.8 Options in the ledger-smb.conf

The ledger-smb.conf configures the software by assigning site-wide variables. Most of these should be left alone unless one knows what one is doing. However, on some systems some options might need to be changed, so all options are presented here for reference:

**templates** is the directory where the templates are stored.

**sendmail** is the command to use to send a message. It must read the email from standard input.

**language** allows one to set the language for the login screen and admin page.

**latex** tells Ledger-SMB whether L<sup>A</sup>T<sub>E</sub>X is installed. L<sup>A</sup>T<sub>E</sub>X is required for generating Postscript and PDF invoices and the like.

**Environmental variables** can be set here too. One can add paths for searching for L<sup>A</sup>T<sub>E</sub>X, etc.

**Printers** section can be used to set a hash table of printers for the software. The primary example is

[printers]

Default = lpr

Color = lpr -PEpson

However, this can use any program that can accept print documents (in Postscript) from standard input, so there are many more possibilities.

## 5 Goods and Services

The Goods and Services module will focus on the definition of goods and services and the related accounting concepts.

---

<sup>1</sup>Adjustments would be entered via the General Ledger. The exact process is beyond the scope of this document, however.

## 5.1 Basic Terms

**COGS** is Cost of Goods Sold. When an item is sold, then the expense of its purchase is accrued as attached to the income of the sale.

**List** Price is the recommended retail price.

**Markup** is the percentage increase that is applied to the last cost to get the sell price.

**ROP** is re-order point. Items with fewer in stock than this will show up on short reports.

**Sell** Price is the price at which the item is sold.

## 5.2 The Price Matrix

It is possible to set different prices for different groups of customers, or for different customers individually. Similarly, one can track different prices from different vendors along with the required lead time for an order.

## 5.3 Pricegroups

Pricegroups are used to help determine the discount a given customer may have.

## 5.4 Groups

Groups represent a way of categorizing POS items for a touchscreen environment. It is not fully functional yet, but is sufficient that with some stylesheet changes, it could be made to work.

## 5.5 Labor/Overhead

Labor/overhead is usually used for tracking manufacturing expenses. It is not directly billed to a customer. It is associated with an expense/Cost of Goods Sold (COGS) account.

## 5.6 Services

Services include any labor that is billed directly to the customer. It is associated with an expense/COGS account and an income account. Services can be associated with sales tax.

### 5.6.1 Shipping and Handling as a Service

One approach to dealing with shipping and handling is to add it as a service. Create a service called "Shipping and Handling", with a sell price \$1 per unit, and a 0% markup. Bill it as \$1 per unit. This allows one to add the exact amount of shipping and handling as necessary.

## 5.7 Parts

A part is any single item you might purchase and either might resell or use in manufacturing an assembly. It is linked to an expense/COGS account, an income account, and an inventory account. Parts can be associated with sales tax.

## 5.8 Assemblies and Manufacturing

Manufacturers order parts but they sell the products of their efforts. Ledger-SMB supports manufacturing using the concept of assemblies. An assembly is any product which is manufactured on site. It consists of a selection of parts, services, and/or labor and overhead. Assemblies are treated as parts in most other regards.

However, one cannot order assemblies from vendors. One must instead order the components and stock them once they are manufactured.

### 5.8.1 Stocking Assemblies

One stocks assemblies in the Stock Assembly entry on the Goods and Services submenu. When an assembly is stocked the inventory is adjusted properly.

The Check Inventory option will cause Ledger-SMB to refuse to stock an assembly if the inventory required to produce the assembly would drop the part below the reorder point.

## 5.9 Reporting

### 5.9.1 All Items and Parts Reports

The All Items report provides a unified view of assemblies, parts, services, and labor for the company, while the Parts report confines it to parts.

Types of reports are:

**Active** lists all items not marked as obsolete.

**On Hand** lists current inventory .

**Short** Lists all items which are stocked below their ROP.

**Obsolete** Lists all items which are marked as obsolete.

**Orphaned** Lists all items which have never had a transaction associated with them.

One can also list these goods by invoice, order, or quotation.

For best results, it is a good idea to enter some AR and AP data before running these reports.

### 5.9.2 Requirements

This report is designed to assist managers determine the quantities of goods to order and/or stock. It compares the quantity on hand with the activity in a given time frame and provides a list of goods which need to be ordered and the relevant quantity.

### 5.9.3 Services and Labor

This is similar to the Parts and All Items menu but only supports Active, Obsolete, and Orphaned reports.

### 5.9.4 Assemblies

This is similar to the Parts and All Items reports but it also provides an ability to list individual items in the assemblies as well.

AP Invoices, Purchase Orders, and RFQ's are not available on this report.

### 5.9.5 Groups and Pricegroups

These reports provide a simple interface for locating groups and pricegroups. The report types are similar to what they are for services.

### 5.10 Translations

One can add translations so that they show up in the customer's native language in the issued invoice.

To issue translations, one must have languages defined. One can then add translations to descriptions and part groups.

### 5.11 How Cost of Goods Sold is tracked

Cost of Goods Sold is tracked on a First-In, First-out (FIFO) basis. When a part is purchased, its cost is recorded in the database. The cost of the item is then added to the inventory asset account. When the good is sold, the cost of the item is moved to the cost of goods sold account.

This means that one must actually provide invoices for all goods entered at their actual cost. If one enters in \$0 for the cost, the cost of goods sold will also be \$0 when the item is sold. We will cover this entire process in more depth after we cover the AP and AR units below.

## 6 AP

### 6.1 Basic AP Concepts

The Accounts Payable module tracks all financial commitments that the company makes to other businesses. This includes rent, utilities, etc. as well as orders of goods and services.

### 6.2 Vendors

A vendor is any business that the company agrees to pay money to.

One can enter vendor information under AP->Vendors->Add Vendor. The vendor list can be searched under AP->Vendors->Reports->Search.

Enter start and end-dates for each vendor. This will make searches and drop-down boxes faster when some vendors are no longer actively supporting your company.

A few fields that need explanation are:

**BIC** Bank Identifier Code is often the same as the S.W.I.F.T. code. This is a code for the bank a customer uses for automated money transfers.

**IBAN** International Bank Account Number is related to the BIC and is used for cross-border automated money transfers.

**Terms** is the number of days one has to pay the invoice.

**Vendor** Number is automatically generated.

### 6.3 AP Transactions

AP Transactions are generally used for items other than goods and services. Utilities, rent, travel expenses, etc. could be entered in as an AP transaction.

If the item is paid partially or in full when the transaction is entered, one can add payments to the payment section.

All other payments can and should be entered under cash payment (below).

The PO Number and Order Number fields are generally used to track associations with purchase orders sent to vendors, etc. These fields can be helpful for adding misc. expenses to orders for reporting purposes.

The department drop-down box appears when one has created one or more departments. A transaction is not required to be associated with a department, but one can use this feature for budget tracking.

With AP Transactions, there is no option for internal notes. All notes will appear on any printed version of the transaction.

Note: Printing a transaction does not post it. No data is committed until the invoice is posted.

## **6.4 AP Invoices**

AP Invoices are used to enter in the receipt of goods and services. Goods and services are deemed entered into the inventory when they are invoiced.

This screen is reasonably similar to the AP Transaction Screen, though the part entry section is a bit different.

The AP Invoice section has a capacity to separate internal notes from notes printed on the invoice. Note, however, that since these are received invoices, it is rare that one needs this ability.

Note that Ledger-SMB can search for partial part numbers or descriptions.

Also if you have a group you can use this to select the part.

To remove a line item from an invoice or order, delete the partnumber and click update.

### **6.4.1 Correcting an AP Invoice**

If an invoice is entered improperly, the methods used to correct it will vary depending on whether transaction reversal is enforced or not. If transaction reversal is not enforced, one can simply correct the invoice or transaction and repost. Note, however, that this violates generally accepted accounting principles.

If transaction reversal is in effect, one needs to create a duplicate invoice with exactly opposite values entered. If one part was listed as received, then one should enter a negative one for the quantity. Then one can enter the invoice number as the same as the old one. Add an R to the end to show that it is a reversing transaction. Once this is posted, one can enter the invoice correctly.

## **6.5 Cash payment And Check Printing**

It is a bad idea to repost invoices/transactions just to enter a payment. The Cash->Payment window allows one to enter payments against AP invoices or transactions.

The printing capability can be used to print checks. The default template is NEBS 9085, though you can use 9082 as well (as Quickbooks does).

The source field is used to store an identifying number of the source document, such as the check number. One must select the item to have it paid, and then enter the amount. One can then print a check.

### **6.5.1 Rapid Payment Entry Screen**

One can also use the rapid payment entry screen to print multiple checks. However, this does not allow you to print the multiple checks to the screen as a separate document is created for each check. In this event, one must print directly to a printer as postscript.

## 6.6 Transaction/Invoice Reporting

### 6.6.1 Transactions Report

This report is designed to help you locate AP transactions based on various criteria. One can search by vendor, invoice number, department, and the like. One can even search by the shipping method.

The summary button will show what was placed where, while the details button will show all debits and credits associated with the transaction.

To view the invoice, click on the invoice number. In the detail view, to view the account transactions as a whole, click on the account number.

Open invoices are ones not fully paid off, while closed invoices are those that have been paid.

### 6.6.2 Outstanding Report

The outstanding report is designed to help you locate AP transactions that are not paid yet. The ID field is mostly useful for locating the specific database record if a duplicate invoice number exists.

### 6.6.3 AP Aging Report

This report can tell you how many invoices are past due and by how much.

A summary report just shows vendors while a detail report shows individual invoices.

### 6.6.4 Tax Paid and Non-taxable Report

These reports have known issues. It is better to use the GL reports and filter accordingly.

## 6.7 Vendor Reporting

### 6.7.1 Vendor Search

The Vendor Search screen can be used to locate vendors or AP transactions associated with those vendors.

The basic types of reports are:

**All** Lists all vendors

**Active** Lists those vendors currently active

**Inactive** Lists those vendors who are currently inactive. time frame.

**Orphaned** Lists those vendors who do not have transactions associated with them. These vendors can be deleted.

One can include purchase orders, Requests for Quotations, AP invoices, and AP transactions on this report as well if they occur between the from and to dates.

### 6.7.2 Vendor History

This report can be used to obtain information about the past goods and services ordered or received from vendors. One can find quantities, partnumber, and sell prices on this report. This facility can be used to search RFQ's, Purchase Orders, and AP Invoices.

## **7 AR**

### **7.1 Customers**

Customers are entered in using the AR->Customers->Add Customer menu.

The salesperson is autopopulated with the current user who is logged in. Otherwise, it looks fairly similar to the Vendor input screen. Customers, like vendors can be assigned languages, but it is more important to do so because invoices will be printed and sent to them.

The credit limit field can be used to assign an amount that one is willing to do for a customer on credit.

#### **7.1.1 Customer Price Matrix**

The price list button can be used to enter specific discounts to the customer, and groups of customers can be assigned a pricegroup for the purpose of offering specific discounts on specific parts to the customer. Such discounts can be temporary or permanent.

### **7.2 AR Transactions**

AR Transactions are where one can add moneys owed the business by customers. One can associate these transactions with income accounts, and add payments if the item is paid when the invoice is issued.

The PO number field is used to track the PO that the customer sent. This makes it easier to find items when a customer is asking for clarification on a bill, for example.

### **7.3 AR Invoices**

AR Invoices are designed to provide for the delivery of goods and services to customers. One would normally issue these invoices at the time when the everything has been done that is necessary to get paid by the customer.

As with AP invoices, one can search for matches to partial part numbers and descriptions, and enter initial payments at this screen.

### **7.4 Cash Receipt**

The Cash->Receipt screen allows you to accept prepayments from customers or pay single or multiple invoices after they have been posted. One can print a receipt, however the current templates seem to be based on check printing templates and so are unsuitable for this purpose. This presents a great opportunity for improvement.

#### **7.4.1 Cash Receipts for multiple customers**

The cash->receipts screen allows you to accept payments on all open customer invoices of all customers at once. One could print (directly to a printer only) all receipts to be sent out if this was desired.

### **7.5 AR Transaction Reporting**

The AR Outstanding report is almost identical to the AP Outstanding report and is not covered in any detail in this document.

### **7.5.1 AR Transactions Report**

This is almost identical to the AP Transactions Report.

If a customer's PO has been associated with this transaction, one can search under this field as well.

### **7.5.2 AR Aging Report**

This report is almost identical to the AP Aging report, with the exception that one can print up statements for customer accounts that are overdue. One more application is to calculate interest based on balance owed so that these can be entered as AR transactions associated with the customer.

## **7.6 Customer Reporting**

These reports are almost identical to the AP Vendor reports and are not discussed in these notes.

## **8 Projects**

### **8.1 Project Basics**

A project is a logical collection of AR and AP transactions, orders, and the like that allow one to better manage specific service or product offerings. Ledger-SMB does not offer comprehensive project management capabilities, and projects are only used here as they relate to accounting.

One can also add translated descriptions to the project names as well.

### **8.2 Timecards**

Timecards allow one to track time entered on specific services. These can then be used to generate invoices for the time entered.

The non-chargeable is the number of hours that are not billed on the invoice.

One can then generate invoices based on this information.

The project field is not optional.

### **8.3 Projects and Invoices**

One can select the project id for line items of both AR and AP invoices. These will then be tracked against the project itself.

## **8.4 Reporting**

### **8.4.1 Timecard Reporting**

The Timecard Report allows one to search for timecards associated with one or more projects. One can then use the total time in issuing invoices (this is not automated yet).

### **8.4.2 Project Transaction Reporting**

The Standard or GIFL options can be used to create different reports (for example, for Canadian Tax reporting purposes).

This report brings up a summary that looks sort of like a chart of accounts. Of one clicks on the account numbers, one can see the transactions associated with the project.



### **8.4.3 List of Projects**

This provides a simple way of searching for projects to edit or modify.

## **8.5 Possibilities for Using Projects**

- One can use them similar to departments for tracking work done for a variety of customers.
- One can use them for customer-specific projects, such as this training.

# **9 Quotations and Order Management**

This unit will introduce the business processes that Ledger-SMB allows. These processes are designed to allow various types of businesses to manage their orders and allow for rudimentary customer relationship management processes to be built around this software. In this section, we will introduce the work flow options that many businesses may use in their day-to-day use of the software.

## **9.1 Sales Orders**

Sales orders represent orders from customers that have not been delivered or shipped yet. These orders can be for work in the future, for back ordered products, or work in progress. A sales order can be generated from an AR invoice or from a quotation automatically.

## **9.2 Quotations**

Quotations are offers made to a customer but to which the customer has not committed to the work. Quotations can be created from Sales orders or AR Invoice automatically.

## **9.3 Shipping**

The Shipping module (Shipping->Shipping) allows one to ship portions or entireties of existing sales orders, printing pick lists and packing slips.

One can then generate invoices for those parts that were shipped.

In general, one will be more likely to use these features if they have multiple warehouses that they ship from. More likely most customers will just generate invoices from orders.

## **9.4 AR Work Flow**

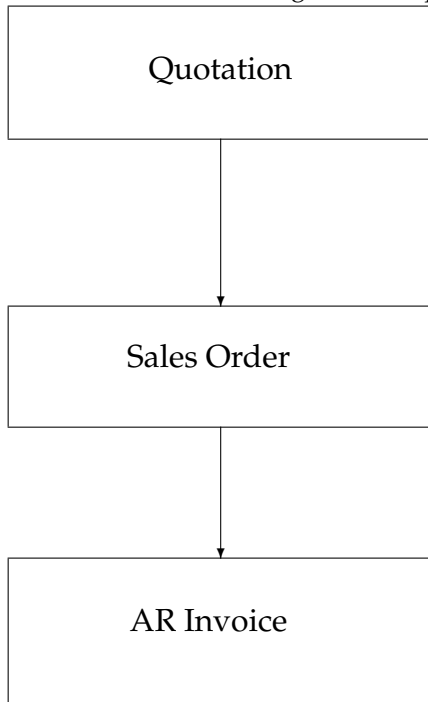
### **9.4.1 Service Example**

A customer contacts your firm and asks for a quote on some services. Your company would create a quotation for the job and email it to the customer or print it and mail it. Once the customer agrees to pay, one creates a sales order from the quotation.

When the work is completed, the sales order is converted into a sales invoice and this is presented to the customer as a bill.

Note that in some cases, this procedure may be shortened. If the customer places an order without asking for a quotation and is offered a verbal quote, then one might merely prepare the sales order.

Figure 1: Simple AR Service Invoice Workflow Example



#### 9.4.2 Single Warehouse Example

A customer contacts your firm and asks for a quotation for shipping a part. You would create the quotation and when you get confirmation, convert it to an order. Once the parts are in place you could go to shipping and ship the part.

The billing department can then generate the invoice from the sales order based on what merchandise has been shipped and mail it to the customer.

Note that this requires that you have the part in your inventory.

#### 9.4.3 Multiple Warehouse Example

A customer contacts your firm and asks for a quotation for a number of different parts. You would create a quotation and when you get confirmation, convert it to a sales order. When you go to ship the item, you would select the warehouse in the drop-down menu, and select the parts to ship. One would repeat with other warehouses until the entire order is shipped.

Then the billing department would go to the sales order and generate the invoice. It would then be mailed to the customer.

### 9.5 Requests for Quotation (RFQ)

A request for quotation would be a formal document one might submit to a vendor to ask for a quote on a product or service they might offer. These can be generated from Purchase Orders or AP Invoices.

Figure 2: AR Workflow with Shipping

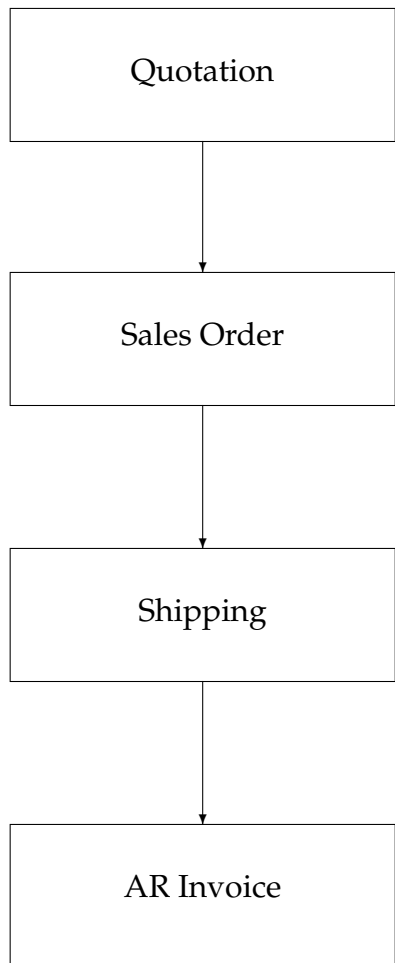
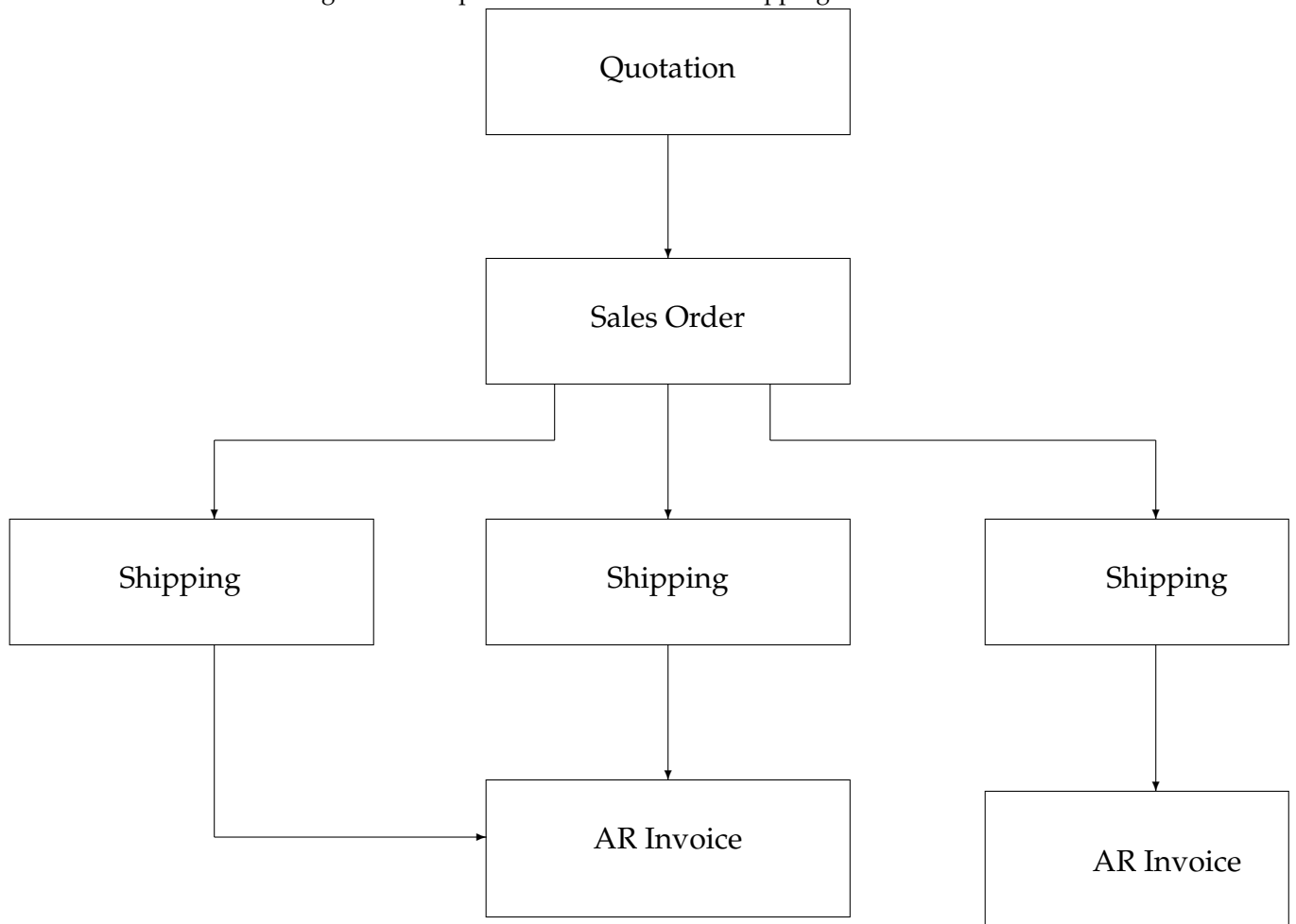


Figure 3: Complex AR Workflow with Shipping



## 9.6 Purchase Orders

A purchase order is a confirmation that is issued to the vendor to order the product or service. Many businesses will require a purchase order with certain terms in order to begin work on a product. These can be generated from RFQ's or AP Invoices.

## 9.7 Receiving

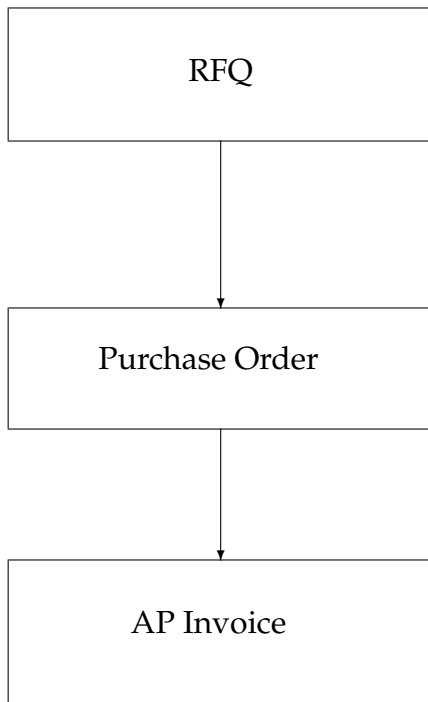
The Shipping->Receiving screen allows you to track the parts received from an existing purchase order. Like shipping, it does not post an invoice but tracks the received parts in the order.

## 9.8 AP Work Flow

### 9.8.1 Bookkeeper entering the received items, order completed in full

Your company inquires about the price of a given good or service from another firm. You submit an RFQ to the vendor, and finding that the price is reasonable, you convert it to an order, adjust the price to what they have quoted, and save it. When the goods are delivered you convert the order into an AP invoice and post it.

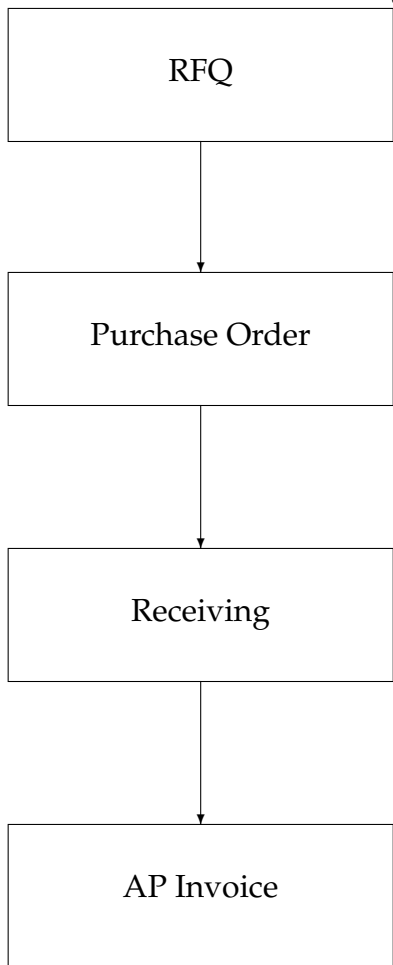
Figure 4: Simple AP Workflow



### 9.8.2 Bookkeeper entering received items, order completed in part

Your company inquires about the price of a given good or service from another firm, You submit an RFQ to the vendor, and finding that the price is acceptable, you convert it into an order, adjusting the price to what they have quoted, and save it. When some of the goods are received, you open up the purchase order, enter the number of parts received, convert that order into an invoice, and post it. Repeat until all parts are received.

Figure 5: AP Workflow with Receiving

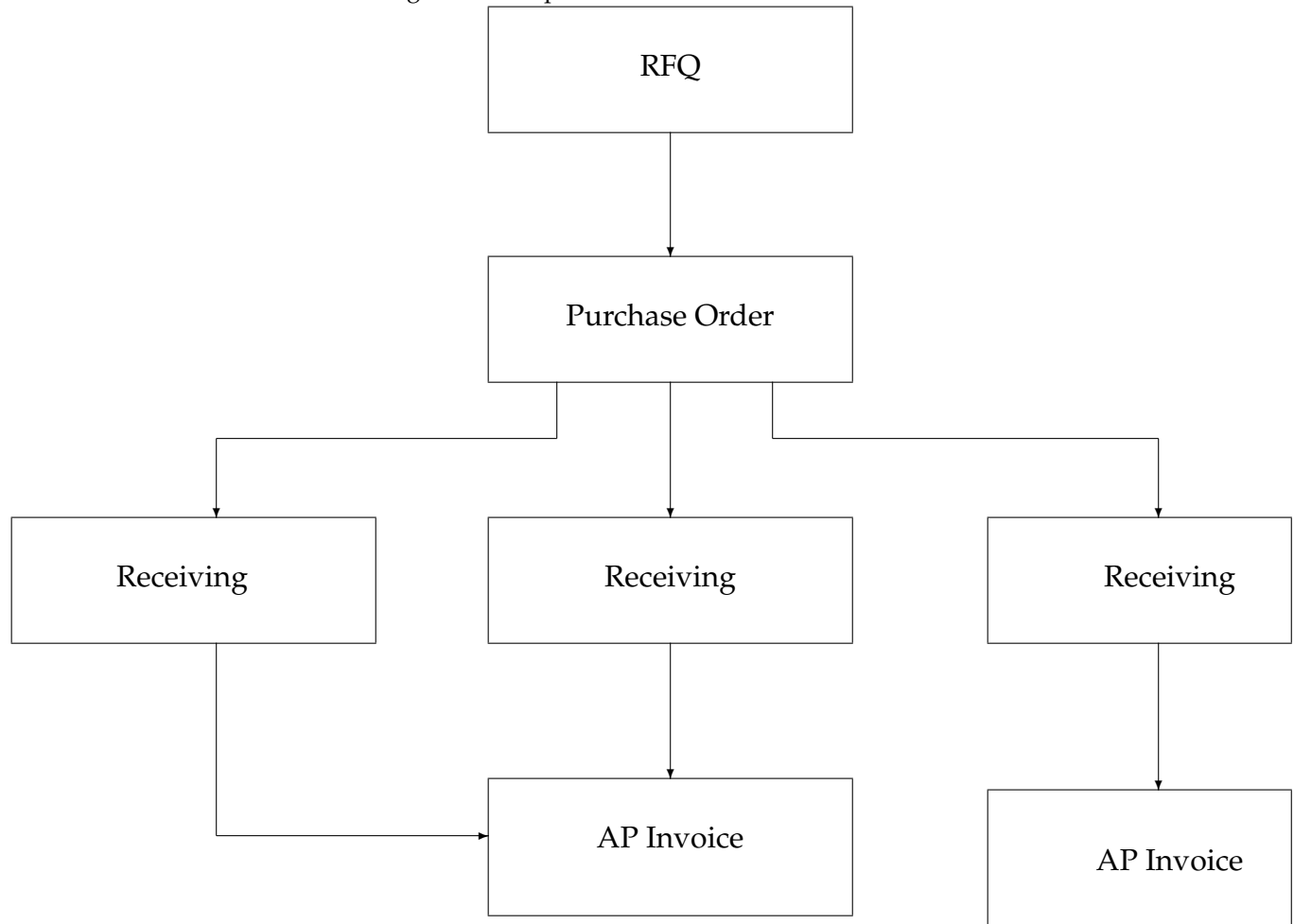


### 9.8.3 Receiving staff entering items

Your company inquires about the price of a given good or service from another firm, You submit an RFQ to the vendor, and finding that the price is acceptable, you convert it into an order, adjusting the price to what they have quoted, and save it. When some or all of the goods are received, the receiving staff goes to Shipping-Receiving, locates the purchase order, and fills in the number of items received.

The bookkeeper can then determine when all items have been received and post the invoice at that time.

Figure 6: Complex AP Workflow



## 9.9 Generation and Consolidation

### 9.9.1 Generation

The Generation screen allows you to generate Purchase Orders based on sales orders. One selects the sales orders one wants to use, and clicks "Generate Purchase Orders." Then one selects clicks on the parts to order, adjusts the quantity if necessary, and clicks "Select Vendor." This process is repeated for every vendor required. Then the Generate Orders button is clicked.

### **9.9.2 Consolidation**

One can consolidate sales and/or purchase orders using this screen. For the consolidation to work you must have more than one order associated with the relevant customer or vendor.

### **9.10 Reporting**

The reporting functionality in the order management is largely limited to the ability to locate purchase orders, sales orders, RFQ's, and quotations.

### **9.11 Shipping Module: Transferring Inventory between Warehouses**

One can transfer inventory between warehouses if necessary by using the Shipping->Transfer Inventory screen.

## **10 HR**

The HR module is currently limited to tracking employees for and their start and end dates. It has very little other functionality. One could build payroll systems that could integrate with it however.

## **11 POS**

LedgerSMB 1.2 includes a number of components merged from Metatron Technology Consulting's SL-POS. Although it is still not a perfect solution, it is greatly improved in both workflow and hardware support. It is suitable for retail establishments at the moment.

### **11.1 Sales Screen**

The sales screen looks very much like a normal invoice entry screen with a few differences.

- The discount text field is not available, nor is the unit field..
- The next part number is automatically focused when the data loads for rapid data entry.
- Hot keys for the buttons are Alt-U for update, Alt-P for print, Alt-O for post, and Alt-R for print and post.
- Part Groups appear at the bottom of the screen.
- Alt-N moves the cursor to the next free payment line.

### **11.2 Possibilities for Data Entry**

- Barcode scanners can be used to scan items in as they are being rung in.
- One could use touch screens, though this would ideally require some custom stylesheets to make it efficient.



## 11.3 Hardware Support

As LedgerSMB is a web-based application, the web browser usually does not allow the page to write to arbitrary files. Therefore hardware support for pole displays, etc. is not readily possible from the application itself. LedgerSMB gets around this limitation by using an additional set of network sockets from the server to the client to control its hardware. This naturally requires that other software is also running on the client.

Notes for specific types of hardware are as follows:

**Touch** screens: The default stylesheet is not really usable from a touchscreen as the items are often too small. One would need to modify the stylesheets to ensure that the relevant items would be reasonable. Setting down the resolution would also help.

**Receipt** Printers: ESC/POS printers generally work in text mode. Control sequences can be embedded in the template as necessary.

**Pole** Displays: Generally supported. Only the Logic Controls PD3000 is supported out of the box, but making this work for other models ought to be trivial.

**Cash** Drawers: These should be attached to the printer. The control codes is then specified in the `pos.conf.pl` so that the command is sent to the printer when the open till button is pushed.

**Barcode** Scanners: Most customers use decoded barcode scanners through a keyboard wedge interface. This allows them to scan items as if they were typing them on the keyboard.

## 11.4 Reports

### 11.4.1 Open Invoices

The POS->Open screen allows one to find any POS receipts that are not entirely paid off.

### 11.4.2 Receipts

The POS->Receipts screen allows one to bring up a basic record of the POS terminals. It is not sufficient for closing the till, however, though it may help for reconciliation.

The till column is the last component or octet of the terminal's IP address. Therefore it is a good idea to try to avoid having IP addresses where the last octet is the same.

All entries are grouped by date and source in this report.

## 12 General Ledger

### 12.1 GL Basics

The General Ledger is the heart of Ledger-SMB. Indeed, Ledger-SMB is designed to be as close as possible to a software equivalent of a paper-based accounting program (but with no difference between the General Ledger and General Journal).

#### 12.1.1 Paper-based accounting systems and the GL

In order to understand the principle of the General Ledger, one must have a basic understanding of the general process of bookkeeping using double-entry paper-based accounting systems.

Normally when a transaction would be recorded, it would first be recorded in the "General Journal" which would contain detailed information about the transaction, notes, etc. Then the entries

from the General Journal would be transcribed to the General Ledger, where one could keep closer tabs on what was going on in each account.

In the general journal, all transactions are listed chronologically with whatever commentary is deemed necessary, while in the general ledger each account has its own page and transactions are recorded in a simple and terse manner. The General Journal is the first place the transaction is recorded and the General Ledger is the last.

At the end of the accounting period, the GL transactions would be summarized into a trial balance and this would be used for creating financial statements and closing the books at the end of the year.

### 12.1.2 Double Entry Examples on Paper

Let us say that John starts his business with an initial investment of \$10,000.

This is recorded in the General Journal as follows (in this example, suppose it is page 1):

Date	Accounts and Explanation	Ref	DEBIT	CREDIT
March 1	Checking Account	1060	10000.00	
	John Doe Capital	3011		10000.00
	John Doe began a business with an investment of \$10000			

This would then be transcribed into two pages of the General Ledger. The first page might be the Checking Account page:

DATE	EXPLANATION	REF.	DEBITS	DATE	EXPLANATION	REF.	CREDITS
March 1		J1	10000.00				

On the John Doe Capital page, we would add a similar entry:

DATE	EXPLANATION	REF.	DEBITS	DATE	EXPLANATION	REF.	CREDITS
				March 1		J1	10000.00

### 12.1.3 The GL in Ledger-SMB

The paper-based accounting procedure works well when one is stuck with paper recording requirements but it has one serious deficiency— all of this transcribing creates an opportunity for errors.

Relational databases relieve the need for such transcription as it is possible to store everything physically in a way similar to the way a General Journal is used in the paper-based systems and then present the same information in ways which are more closely related to the General Ledger book.

This is the exact way that the General Ledger is used in Ledger-SMB. The actual data is entered and stored as if it was a general journal, and then the data can be presented in any number of different ways.

All modules of Ledger-SMB that involve COA accounts store their data in the General Ledger (it is a little more complex than this but this is very close to the actual mechanism).

## 12.2 Cash Transfer

The simplest form of GL entry in Ledger-SMB is the Cash->Transfer screen. This screen shows two transaction lines, and fields for reference, department, description, and notes.

The field descriptions are as follows:

**Reference** refers to the source document for the transfer. One can use transfer sheets, bank receipt numbers, etc for this field.

**Description** is optional but really should be filled in. It ought to be a description of the transaction.

**Notes** provide supplemental information for the transaction.

**FX** indicates whether foreign exchange is a factor in this transaction.

**Debit** indicates money going **into** the asset account.

**Credit** indicates money coming **out** of the asset account.

**Source** is the source document for that portion of the transaction.

**Memo** lists additional information as necessary.

**Project** allows you to assign this line to a project.

The credit and debit options seem to be the opposite of what one would think of concerning one's bank account. The reason is that your bank statement is done from the bank's point of view. Your bank account balance is an asset to you and therefore you show it as having a debit balance, but to the bank it is money they owe you and so they show it as having a credit balance.

Note that in this screen, when an item is updated, it will reduce the number of lines to those already filled in plus an extra line for the new line in the data entry.

### 12.3 GL Transactions

The GL Transaction screen (General Ledger->Add Transaction) is identical to the Cash Transfer screen with the exception that it starts with nine lines instead of two. Otherwise, they are identical.

Again, one must be careful with debits and credits. Often it is easy to get confused. It is generally worth while to go back to the principle that one tracks them with regard to their impact on the equity accounts. So expenses are credits because they debit the equity accounts, and income is a debit because it credits the retained earning equity account.

### 12.4 Payroll as a GL transaction

Currently payroll must be done as a GL transaction. The attempts to create a payroll system that would ship with LSMB have largely stalled.

Most customers running their businesses will have an idea of how to do this.

Figure 7: Payroll as a GL Transaction (Purely fictitious numbers)

Account	Debit	Credit
5101 Wages and Salaries	500	
2032 Accrued Wages		450
2033 Fed. Income Tax withd		30
2034 State Inc. Tax. withd		15
2035 Social Security withd		3
2036 Medicare withd		2
2032 Accrued Wages	450	
1060 Checking Acct		450

### 12.5 Reconciliation

To reconcile an account (say, when one would get a checking account statement), one would go to cash/reconciliation, and check off the items that have cleared. One can then attempt to determine where any errors lie by comparing the total on the statement with the total that LSMB generates.

This can be done for other accounts too, such as petty cash.<sup>2</sup>

## 12.6 Reports

The most flexible report in Ledger-SMB is the GL report because it has access to the entire set of financial transactions of a business. Every invoice posted, payment made or received, etc. can be located here.

The search criteria include:

**Reference** is the invoice number, or other reference number associated with the transaction.

**Source** is the field related to the source document number in a payment or other transaction.<sup>3</sup>

**Memo** relates to the memo field on a payment.

**Department** can be used to filter results by department.

**Account Type** can be used to filter results by type of account (Asset, Liability, etc.)

**Description** can be used to filter by GL description or by customer/vendor name.

The actual format of the report looks more like what one would expect in a paper accounting system's general journal than a general ledger per se. A presentation of the data that is more like the paper general ledger is found in the Chart of Accounts report.

### 12.6.1 GL as access to almost everything else

The GL reports can be used to do all manner of things. One can determine, for example, which AP invoice or transaction was paid with a certain check number or which invoice by a specific customer was paid by a specific check number.

## 13 Recurring Transactions

Any transaction or invoice may be repeated a number of times in regular intervals. To schedule any GL, AR, or AP transaction or invoice, click the schedule button.

In general the reference number should be left blank as this will force Ledger-SMB to create a new invoice or transaction number for each iteration. The rest of the options are self-explanatory. Note that a blank number if iterations will result in no recurrences of the transaction.

To process the recurring transactions, click on the Recurring Transactions option on the main menu select the ones you want to process and click "Process Transactions."

## 14 Financial Statements and Reports

Financial statements and reports are a very important part of any accounting system. Accountants and business people rely on these reports to determine the financial soundness of the business and its prospects for the next accounting period.

---

<sup>2</sup>Petty cash denotes a drawer of cash that is used to pay small expenses. When an expense is paid, it is recorded on a slip of paper that is stored for reconciliation purposes.

<sup>3</sup>Source documents are things like receipts, canceled checks, etc. that can be used to verify the existence and nature of a transaction.

## **14.1 Cash v. Accrual Basis**

Financial statements, such as the Income Statement and Balance Sheet can be prepared either on a cash or accrual basis. In cash-basis accounting, the income is deemed earned when the customer pays it, and the expenses are deemed incurred when the business pays them.

There are a number of problems with cash-basis accounting from a business point of view. The most serious is that one can misrepresent the wellbeing of a business by paying a large expense after a deadline. Thus cash-basis accounting does not allow one to accurately pair the income with the related expense as these are recorded at different times. If one cannot accurately pair the income with the related expense, then financial statements cannot be guaranteed to tell one much of anything about the well-being of the business.

In accrual basis accounting, income is considered earned when the invoice is posted, and expenses are considered incurred at the time when the goods or services are delivered to the business. This way, one can pair the income made from the sale of a product with the expense incurred in bringing that product to sale. This pairing allows for greater confidence in business reporting.

## **14.2 Viewing the Chart of Accounts and Transactions**

The Reports->Chart of Accounts will provide the chart of accounts along with current totals in each account.

If you click on an account number, you will get a screen that allows you to filter out transactions in that account by various criteria. One can also include AR/AP, and Subtotal in the report.

The report format is similar to that of a paper-based general ledger.

## **14.3 Trial Balance**

### **14.3.1 The Paper-based function of a Trial Balance**

In paper-based accounting systems, the accountant at the end of the year would total up the debits and credits in every account and transfer them onto another sheet called the trial balance. The accountant would check to determine that the total debits and credits were equal and would then transfer this information onto the financial statements. It was called a trial balance because it was the main step at which the error-detection capabilities of double-entry accounting systems were used.

### **14.3.2 Running the Trial Balance Report**

This report is located under Reports -> Trial Balance. One can filter out items by date, accounting period, or department. One can run the report by accounts or using GIFI classifications to group accounts together.

From this report, you can click on the account number and see all transactions on the trial balance as well as whether or not they have been reconciled.

### **14.3.3 What if the Trial Balance doesn't Balance?**

If the trial balance does not balance, get technical support immediately. This usually means that transactions were not entered properly. Some may have been out of balance, or some may have gone into non-existent accounts (believe it or not, Ledger-SMB does not check this latter issue).

### **14.3.4 Trial Balance as a Summary of Account Activity**

The trial balance offers a glance at the total activity in every account. It can provide a useful look at financial activity at a glance for the entire business.

#### 14.3.5 Trial Balance as a Budget Planning Tool

By filtering out departments, one can determine what a department earned and spent during a given financial interval. This can be used in preparing budgets for the next accounting period.

### 14.4 Income Statement

The Income Statement is another tool that can be used to assist with budgetary planning as well as provide information on the financial health of a business.

The report is run from Reports->Income Statement. The report preparation screen shows the following fields:

**Department** allows you to run reports for individual departments. This is useful for budgetary purposes.

**Project** allows you to run reports on individual projects. This can show how profitable a given project was during a given time period.

**From** and **To** allow you to select arbitrary from and to dates.

**Period** allows you to specify a standard accounting period.

**Compare to** fields allow you to run a second report for comparison purposes for a separate range of dates or accounting period.

**Decimalplaces** allows you to display numbers to a given precision.

**Method** allows you to select between accrual and cash basis reports.

**Include** in Report provides various options for reporting.

**Accounts** allows you to run GIFI reports instead of the standard ones.

The report shows all income and expense accounts with activity during the period when the report is run, the balances accrued during the period, as well as the total income and expense at the bottom of each section. The total expense is subtracted from the total income to provide the net income during the period. If there is a loss, it appears in parentheses.

#### 14.4.1 Uses of an Income Statement

The income statement provides a basic snapshot of the overall ability of the business to make money. It is one of the basic accounting statements and is required, for example, on many SEC forms for publicly traded firms.

Additionally, businessmen use the income statement to look at overall trends in the ability of the business to make money. One can compare a given month, quarter, or year with a year prior to look for trends so that one can make adjustments in order to maximize profit.

Finally, these reports can be used to provide a look at each department's performance and their ability to work within their budget. One can compare a department or project's performance to a year prior and look for patterns that can indicate problems or opportunities that need to be addressed.

## 14.5 Balance Sheet

The balance sheet is the second major accounting statement supported by Ledger-SMB. The balance sheet provides a snapshot of the current financial health of the business by comparing assets, liabilities, and equity.

In essence the balance sheet is a statement of the current state of owner equity. Traditionally, it does not track changes in owner equity in the same way the Statement of Owner Equity does.

The Balance Sheet report preparation screen is much simpler than the Income Statement screen. Balance sheets don't apply to projects, but they do apply to departments. Also, unlike an income statement, a balance sheet is fixed for a specific date in time. Therefore one does not need to select a period.

The fields in creating a balance sheet are:

**Department** allows you to run separate balance sheets for each department.

**As at** specifies the date. If blank this will be the current date.

**Compare to** specifies the date to compare the balance sheet to.

**Decimalplaces** specifies the number of decimal places to use.

**Method** selects between cash and accrual basis.

**Include** in report allows you to select supplemental information on the report.

**Accounts** allows you to select between standard and GIF1 reports.

The balance sheet lists all asset, liability, and equity accounts with a balance. Each category has a total listed, and the total of the equity and liability accounts is also listed.

The total assets should be equal to the sum of the totals of the liability and equity accounts.

## 14.6 What if the Balance Sheet doesn't balance?

Get technical support immediately, This may indicate that out of balance transactions were entered or that transactions did not post properly.

## 14.7 No Statement of Owner Equity?

The Statement of Owner Equity is the one accounting statement that Ledger-SMB does not support. However, it can be simulated by running a balance sheet at the end of the time frame in question and comparing it to the beginning. One can check this against an income statement for the period in question to verify its accuracy. The statement of owner equity is not as commonly used now as it once was.

## 15 The Template System

Ledger-SMB allows most documents to be generated according to a template system. This allows financial statements, invoices, orders, and the like to be customized to meet the needs of most businesses. Company logos can be inserted, the format can be radically altered, one can print letters to be included with checks to vendors instead of the checks themselves, and the like. In the end, there is very little that cannot be accomplished regarding modification of these documents with the template system.

One can define different templates for different languages, so that a customer in Spain gets a different invoice than a customer in Canada.

## 15.1 Text Templates

The only template that uses a text-only format is the POS receipt. This example provides the simplest way to understand the template system.

The first two lines are:

```
<%company align=center width=40%>  
<%address align=center width=40%>
```

The first line tells Ledger-SMB to print the company name as passed to it via a variable, centered, with a page width of 40 characters. The second line does the same thing with the address.

These variables are usually passed to the invoice using form fields (hidden or otherwise) in the submitting web page. The printing script, however, can disable some of these fields or add others via database lookups and the like.

In all types of templates, variable substitution occurs between `<%` and `%>`. One can optionally specify an alignment or a width but these are really only useful in text templates.

## 15.2 HTML Templates

The following templates exist in HTML format:

- Income Statement
- Balance Sheet
- Invoice (AR)
- AR Transaction
- AP Transaction
- Packing List
- Pick List
- Sales Order
- Work Order
- Purchase Order
- Bin List
- Statement
- Quotation
- RFQ
- Time Card

These templates can be edited by an HTML editor. However, it is generally recommended that one back up templates first. The reason is that some HTML editors will fully re-parse the HTML and save it back without what they see as invalid tags. Most editors, however, will save the variable substitution tags because similar tags are also used by Microsoft's active server pages.

Finally, some editors are known to mangle formatting, so many problems can be avoided by ensuring that one has a backup of the templates, especially if they have already been customized.



## 15.3 L<sup>A</sup>T<sub>E</sub>X Templates

The following templates, by default, are available in L<sup>A</sup>T<sub>E</sub>X :

- Invoice
- AR Transaction
- AP Transaction
- Packing List
- Pick List
- Sales Order
- Work Order
- Purchase Order
- Bin List
- Statement
- Check
- Receipt
- Quotation
- RFQ
- Time Card

L<sup>A</sup>T<sub>E</sub>X templates allow one to generate PDF and postscript documents and print directly to a postscript-enabled printer or print software (like CUPS).

L<sup>A</sup>T<sub>E</sub>X templates can be edited using a standard text editor (like vim or emacs), or using a synchronous L<sup>A</sup>T<sub>E</sub>X implementation such as L<sup>y</sup>X.

### 15.3.1 What is L<sup>A</sup>T<sub>E</sub>X ?

L<sup>A</sup>T<sub>E</sub>X (pronounced LAY-tech) is an extension on the T<sub>E</sub>X typesetting system. It largely consists of a set of macros that allow one to focus on the structure of the document while letting the T<sub>E</sub>X engine do the heavy lifting in terms of determining the optimal formatting for the page. L<sup>A</sup>T<sub>E</sub>X is used in a large number of academic journals (including those of the American Mathematics Association). It is available at <http://www.tug.org> and is included in most Linux distributions.

Like HTML, L<sup>A</sup>T<sub>E</sub>X uses plain text documents to store the formatting information and then when the document is rendered, attempts to fit it onto a page. L<sup>A</sup>T<sub>E</sub>X supports the concept of stylesheets, allowing one to separate content from format, and this feature is used in many higher-end applications, like journal publication.

Unlike HTML, L<sup>A</sup>T<sub>E</sub>X is a complete though simple programming language that allows one to redefine internals of the system for formatting purposes.

This document is written in L<sup>A</sup>T<sub>E</sub>X.

### 15.3.2 Using L<sup>y</sup>X to Edit L<sup>A</sup>T<sub>E</sub>X Templates

L<sup>y</sup>X is a synchronous L<sup>A</sup>T<sub>E</sub>X editor that runs on Windows, UNIX/Linux, and Mac OS X. It requires an installed L<sup>A</sup>T<sub>E</sub>X-2e implementation and can be obtained at <http://www.lyx.org>. Like the most common L<sup>A</sup>T<sub>E</sub>X implementations, it is open source and is included with most Linux distributions.

## 15.4 Customizing Logos

L<sup>A</sup>T<sub>E</sub>X requires different formats of logos depending on whether the document is going to be generated as a PDF or as postscript. Postscript requires an embedded postscript graphic, while PDF requires any type of graphic other than embedded postscript. Usually one uses a PNG's for PDF's, though GIF's could be used as well. The logo for a L<sup>A</sup>T<sub>E</sub>X document resides in the users directory.

HTML documents can have logos in many different formats. PNG's are generally preferred for printing reasons. The image can be stored anywhere and merely referenced in the HTML.

Note: Always test the an invoice with images to ensure that the rest of the page format is not thrown off by it.

## 15.5 How are They Stored in the Filesystem?

The template directory ("templates" in the root Ledger-SMB install directory) contains all the root templates used by Ledger-SMB. These follow a naming convention of COAType-templatename.ext where COAType is the type of dataset that was created when the user was created, templatename is the name of the template, and ext is either txt, html, or tex (for text, html, and L<sup>A</sup>T<sub>E</sub>X respectively).

Inside this directory are one or more subdirectories where the relevant templates have been copied as default language templates for the user. Many users can use the same user directory (which bears the name of the Ledger-SMB username). Within this directory are more subdirectories for translated templates, one for each language created.

## 15.6 Upgrade Issues

When Ledger-SMB is upgraded, the templates are not replaced. This is designed to prevent the upgrade script from overwriting changes made during the course of customizing the templates.

Occasionally, however, the data model changes in a way which can cause the templates to stop printing certain information. When information that was showing up before an upgrade stops showing up, one can either upgrade the templates by copying the source template over the existing one, or one can edit the template to make the change.

# 16 An Introduction to the CLI

## 16.1 Conventions

The command-line API will be referred to as the API.

## 16.2 Preliminaries

Logging into Ledger-SMB (1.2+) updates a row in the users\_conf table containing your account configuration defaults and other information. The implication for API users of LSMB is that you must login as part of running API scripts. For security purposes, it is recommended that scripts prompt for a password rather than storing it.

All scripts included in the documentation can also be found in the doc/samples directory.

Consider a simple example:

```
cd /usr/local/ledger-smb ./ct.pl "login=name&path=bin&password=xxxxx&action=search&db=customer"
```

The cd command moves your terminal session's current working directory into the main Ledger-SMB directory. Then the Ledger-SMB perl script ct.pl is called with one long line as an argument. The argument is really several variable=value pairs separated by ampersands (&). The value for the login variable is the username that Ledger-SMB is to use, and the value for the password variable is the plaintext password.

To build our examples we will use a username of "clarkkent" who has a password of "lOis,lAn3".

```
cd /usr/local/ledger-smb ./ct.pl "login=clarkkent&path=bin&password=lOis,lAn3&action=search&db=customer"
```

If we execute these commands we will get the html for the search form for the customer database. This result isn't useful in itself, but it shows we are on the right track.

### 16.3 First Script: lsmb01-cli-example.sh

With a working example, we can start to build reproducible routines that we can grow to do some useful work.

This is a bash script which:

1. sets NOW to the current working directory
2. prompts for and reads your Ledger-SMB login
3. prompts for and reads (non-echoing) your Ledger-SMB password
4. changes directory to /usr/local/ledger-smb
5. constructs login and logout commands and a transaction command
6. logs into ledger-smb (in a real program, output would be checked for success or failure)
7. executes the transaction
8. logs out of ledger-smb (although this is not necessary)
9. returns to the original working directory
10. exits

Running lsmb01-cli-example.sh produces:

```
$ lsmb01-cli-example.sh
Ledger-SMB login: clarkkent
Ledger-SMB password:
```

```
<body>

<form method=post action=ct.pl>

<input type=hidden name=db value=customer>

<table width=100%>
  <tr>
    <th class=listtop>Search</th>
  .
  .
  .
```

A script like this would work well for simple batch transactions, but bash is not a very friendly language for application programming.

A nicer solution would be to use a language such as perl to drive the command line API.

#### 16.3.1 Script 1 (Bash)

```
#!/bin/bash
#####
#
# lsmb01-cli-example.sh
# Copyright (C) 2006. Louis B. Moore
#
# $Id: $
#
# This program is free software; you can redistribute it and/or modify
# it under the terms of the GNU General Public License as published by
# the Free Software Foundation; either version 2 of the License, or
# (at your option) any later version.
#
```

```

# This program is distributed in the hope that it will be useful,
# but WITHOUT ANY WARRANTY; without even the implied warranty of
# MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
# GNU General Public License for more details.
# You should have received a copy of the GNU General Public License
# along with this program; if not, write to the Free Software
# Foundation, Inc., 675 Mass Ave, Cambridge, MA 02139, USA.
#
#####

NOW=`pwd`

echo -n "Ledger-SMB login: "
read LSLOGIN
echo

echo -n "Ledger-SMB password: "
stty -echo
read LSPWD
stty echo
echo

ARG="login=${LSLOGIN}&password=${LSPWD}&path=bin&action=search&db=customer"

LGIN="login=${LSLOGIN}&password=${LSPWD}&path=bin&action=login"
LGOT="login=${LSLOGIN}&password=${LSPWD}&path=bin&action=logout"

cd /usr/local/ledger-smb

./login.pl $LGIN 2>&1 > /dev/null
./ct.pl $ARG
./login.pl $LGOT 2>&1 > /dev/null

cd $NOW

exit 0

```

## 16.4 Second Script: lsmb02-cli-example.pl

Our second script is written in perl and logs you in but it still uses the API in its simplest form, that is, it builds commands and then executes them. This type of script can be used for more complex solutions than the simple bash script above, though it is still fairly limited. If your needs require, rather than have the script build and then execute the commands it could be written to generate a shell script which is executed by hand.

This script begins by prompting for your Ledger-SMB login and password. Using the supplied values a login command is constructed and passed into the runLScmd subroutine. runLScmd changes directory to /usr/local/ledger-smb/ for the length of the subroutine. It formats the command and executes it and returns both the output and error information to the caller in a scalar.

The script checks to see if there was an error in the login, exiting if there was.

Next, the script reads some records which are stored in the program following the \_\_END\_\_ token. It takes each record in turn, formats it then feeds each transaction through runLScmd and looks at the results for a string that signals success.

Once all the transactions are processed, runLScmd is called one last time to logout and the script exits.

#### 16.4.1 Script 2 (Perl)

```
#!/usr/bin/perl -w
#
# File:          lsmb02-cli-example.pl
# Environment:   Ledger-SMB 1.2.0+
# Author:       Louis B. Moore
#
# Copyright (C) 2006 Louis B. Moore
#
# This program is free software; you can redistribute it and/or
# modify it under the terms of the GNU General Public License
# as published by the Free Software Foundation; either version 2
# of the License, or (at your option) any later version.
#
# This program is distributed in the hope that it will be useful,
# but WITHOUT ANY WARRANTY; without even the implied warranty of
# MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
# GNU General Public License for more details.
#
# You should have received a copy of the GNU General Public License
# along with this program; if not, write to the Free Software
# Foundation, Inc., 59 Temple Place - Suite 330, Boston, MA 02111-1307, USA.
#
# Revision:
#     $Id$
#
#
use File::chdir;
use HTML::Entities;

print "\n\nLedger-SMB login: ";
my $login = <STDIN>;
chomp($login);

print "\nLedger-SMB password: ";
system("stty -echo");
my $pwd = <STDIN>;
system("stty echo");
chomp($pwd);
print "\n\n";

$cmd = "login=" . $login . '&password=' . $pwd . '&path=bin&action=login';

$signin = runLScmd("./login.pl",$cmd);
```

```

if ( $signin =~ m/Error:/ ) {

print "\nLogin error\n";
exit;

}

while (<main::DATA>) {

chomp;
@rec = split(/\|/);

$args = 'path=bin/mozilla&login=' . $login . '&password=' . $pwd .
'&action=' . escape(substr($rec[0],0,35)) .
'&db=' . $rec[1] .
'&name=' . escape(substr($rec[2],0,35)) .
'&vendornumber=' . $rec[3] .
'&address1=' . escape(substr($rec[4],0,35)) .
'&address2=' . escape(substr($rec[5],0,35)) .
'&city=' . escape(substr($rec[6],0,35)) .
'&state=' . escape(substr($rec[7],0,35)) .
'&zipcode=' . escape(substr($rec[8],0,35)) .
'&country=' . escape(substr($rec[9],0,35)) .
'&phone=' . escape(substr($rec[10],0,20)) .
'&tax_2150=1' .
'&taxaccounts=2150' .
'&taxincluded=0' .
'&terms=0';

$rc=runLScmd("./ct.pl",$args);

if ($rc =~ m/Vendor saved!/) {

print "$rec[2] SAVED\n";

} else {

print "$rec[2] ERROR\n";

}

}

$cmd = "login=" . $login . '&password=' . $pwd . '&path=bin&action=logout';

$signin = runLScmd("./login.pl",$cmd);

if ( $signin =~ m/Error:/ ) {

    print "\nLogout error\n";

```

```

}

exit;

#*****
# Subroutines
#*****

sub runLScmd {

    my $cmd = shift;
    my $args = shift;
    my $i = 0;
    my $results;

    local $CWD = "/usr/local/ledger-smb/";

    $cmd = $cmd . " \"" . $args . "\"";

    $results = `$cmd 2>&1`;

    return $results;

}

sub escape {

    my $str = shift;

    if ($str) {

decode_entities($str);
$str =~ s/([a-zA-Z0-9_.-])/sprintf("%%02x", ord($1))/ge;
    }

    return $str;

}

#*****
# Record Format
#*****
#
# action | db | name | vendornumber | address1 | address2 | city | state | zipcode | cou
#
__END__
save|vendor|Parts are Us|1377|238 Riverview|Suite 11|Cheese Head|WI|56743|USA|555-123-33

```

save|vendor|Widget Heaven|1378|41 S. Riparian Way||Show Me|MO|39793|USA|555-231-3309|  
save|vendor|Consolidated Spackle|1379|1010 Binary Lane|Dept 1101|Beverly Hills|CA|90210|



## Part II

# Technical Overview

### 17 Basic Architecture

Ledger-SMB is a web-based Perl program that interfaces with PostgreSQL using the relevant Perl modules. The code is well partitioned, and the main operation modules are written in an object oriented way.

#### 17.1 The Software Stack

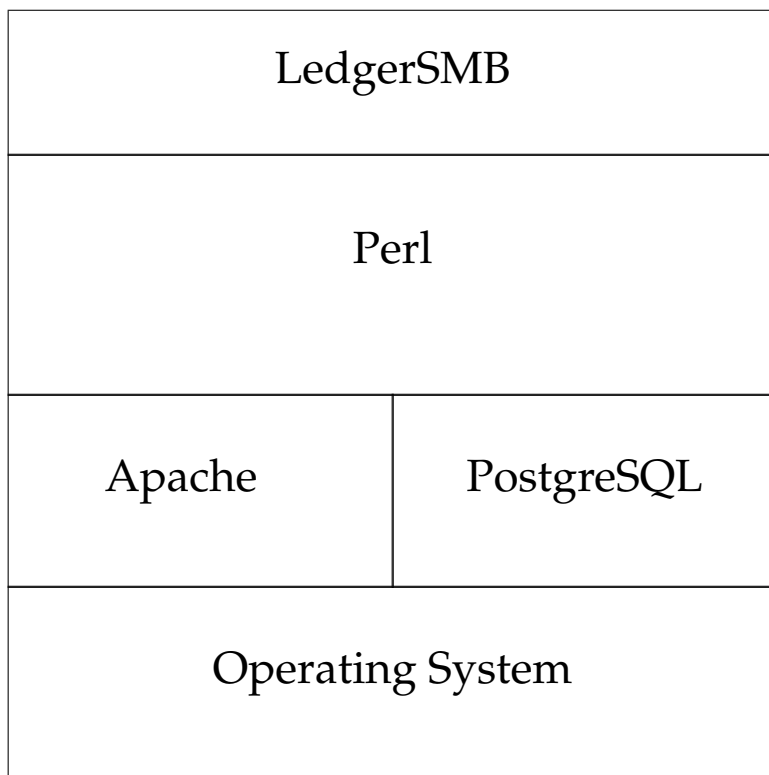


Figure 8: The Ledger-SMB software stack in a Typical Implementation

Ledger-SMB runs in a Perl interpreter. I do not currently know if it is possible to run it with Perl2C or other language converters to run in other environments. However, except for high-capacity environments, Perl is a good language choice for this sort of program.

Ledger-SMB used to support DB2 and Oracle as well as PostgreSQL. However, currently some of the functionality is implemented using PostgreSQL user-defined functions. These would need to be ported to other database managers in order to make the software work on these. It should not be too hard, but the fact that it has not been done yet may mean that there is no real demand for running the software under other RDBMS's.

One can substitute other web servers for Apache. Normally Ledger-SMB is run as a CGI program but it may be possible to run it in the web server process (note that this may not be entirely thread-safe).

The operating system can be any that supports a web server and Perl (since PostgreSQL need not run on the same system). However, there are a few issues running Ledger-SMB on Windows (most notably in trying to get Postscript documents to print properly).

On the client side, any web-browser will work. Currently, the layout is different for Lynx (which doesn't support frames), and the layout is not really useful under eLinks (the replacement for Lynx which does support frames). Some functionality requires Javascript to work properly, though the application is usable without these features.

## 17.2 Capacity Planning

Some companies may ask how scalable Ledger-SMB is. In general, it is assumed that few companies are going to have a need for a high-concurrency accounting system. However, with all the features available in Ledger-SMB, the staff that may have access to some of the application may be senior enough to make the question worthwhile.

This question also becomes more important when companies might look at integrating Ledger-SMB with a CRM solution, online store, or other environment. This section looks at a number of the known issues and their solutions.

### 17.2.1 Scalability Strategies

Ledger-SMB is a fairly standard web-based application. However, sometimes the database schema changes during upgrades. In these cases, it becomes impossible to use different versions of the software against the same database version safely. Ledger-SMB checks the version of the database and if the version is higher than the version of the software that is running, will refuse to run.

Therefore although one strategy might be to run several front-end web servers with Ledger-SMB, in reality this can be a bit of a problem. One solution is to take half of the front-end servers off-line while doing the initial upgrade, and then take the other offline to upgrade when these are brought back online.

The database manager is less scalable in the sense that one cannot just add more database servers and expect to carry on as normal. However, aside from the known issues listed below, there are few performance issues with it. If complex reports are necessary, these can be moved to a replica database (perhaps using Slony-I).

If this solution is insufficient for database scalability, one might be able to move staff who do not need real-time access to new entries onto a PG-Pool/Slony-I cluster where new transactions are entered on the master and other data is looked up on the replica. In certain circumstances, one can also offload a number of other queries from the master database in order to minimize the load. Ledger-SMB has very few issues in the scalability of the application.

### 17.2.2 Database Maintenance

PostgreSQL uses a technique called Multi-version Concurrency Control (MVCC) to provide a snapshot of the database at the beginning of a statement or transaction (depending on the transaction isolation level). When a row is updated, PostgreSQL leaves the old row in the database, and inserts a new version of that row into the table. Over time, unless those old rows are removed, performance can degrade as PostgreSQL has to search through all the old versions of the row in order to determine which one ought to be the current one.

Due to the way the SQL statements are executed in Ledger-SMB, most inserts will also create a dead row.

A second problem occurs in that each transaction is given a transaction id. These id's are numbered using 32-bit integers. If the transaction id wraps around (prior to 8.1), data from transactions that appear (due to the wraparound) to be in the future suddenly becomes inaccessible. This problem was corrected in PostgreSQL 8.1, where the database will refuse to accept new transactions if the

transaction ID gets too close to a wraparound. So while the problem is not as serious in 8.1, the application merely becomes inaccessible rather than displaying apparent data loss. Wraparound would occur after about a billion transactions between all databases running on that instance of PostgreSQL.

Prior to 8.1, the main way to prevent both these problems was to run a periodic `vacuumdb` command from cron (UNIX/Linux) or the task scheduler (Windows). In 8.1 or later, autovacuum capabilities are part of the back-end and can be configured with the database manager. See the PostgreSQL documentation for treatment of these subjects.

In general, if performance appears to be slowly degrading, one should try to run `vacuumdb -z` from the shell in order to attempt to reclaim space and provide the planner with accurate information about the size and composition of the tables. If this fails, then one can go to other methods of determining the bottleneck and what to do about it.

### 17.2.3 Known issues

The PostgreSQL planner assumes a minimum page size of ten pages for a physically empty table. The reasoning behind this choice is that a table could grow rapidly and one could end up with bad database performance if the planner assumes a very small table.

However, if you end up with joins between a very large table with millions of rows and a physically empty table, one can end up with a very bad query plan. In this case, the planner will choose a nested loop join and run through this loop for every row in the large table. As a result, performance will suddenly drop once the large table becomes too large to effectively do index scans of the join criteria on both tables. This problem most often occurs when people have no warehouses, departments, or projects defined and are running systems with a large number of transactions (such as a point of sale environment).

Last time I saw this problem, the server would wait for thirty seconds to display a new point of sale screen while the server CPU activity would spike to 100%.

One solution is to define one warehouse, department, and project, and then run `vacuumdb -z` from the shell to force the planner to acknowledge these tables as single-row tables. The other option is to go into the source code and edit the database queries to omit unused tables.

## 18 Customization Possibilities

Ledger-SMB is designed to be customized relatively easily and rapidly. In general, the source code is well written and compartmentalized. This section covers the basic possibilities involving customization.

### 18.1 Brief Guide to the Source Code

Ledger-SMB is an application with over 34000 lines of code. While it is not possible to cover the entire application here, a brief overview of the source code is in order.

In the root of the install directory, one will find a `setup.pl` program, a number of other `.pl` programs, and a number of directories. The `setup.pl` program is used to update or install Ledger-SMB. The other `.pl` programs provide a basic set of services for the framework (including authentication) and then pass the work on to the data entry screen file in the `bin` directory.

The `bin` directory contains another directory for each terminal type. The main two offered are `lynx` and `mozilla`. `Lynx` would be used for web browsers that do not support frames and is ideal for a text-mode VGA terminal. `Mozilla` is the terminal type used for most other web browsers. The perl files within these directories provides the user interface of the software.

The `css` directory in the root install directory contains CSS documents to provide various stylesheets one can select for changing various aspects of the look and feel of the application.

The locale directory contains translation files that Ledger-SMB uses to translate between different languages. One could add translations to these files if necessary.

The LSMB directory is where the Perl modules reside that provide the core business logic in Ledger-SMB. These modules provide functionality such as form handling, email capabilities, and access to the database through its at least partially object oriented API.

Finally, the sql directory provides the database schemas and upgrade scripts.

## **18.2 Data Entry Screens**

One can customize the data entry screens to optimize work flow, display additional information, etc.

### **18.2.1 Examples**

We set up hot keys for payment lines, automatically focused the keyboard on the last partnumber field, removed separate print and post buttons to ensure that invoices were always printed and posted together, and removed the ability to print to the screen, and even the ability to scan items in when an invoice was received (using a portable data terminal) and import this data into Ledger-SMB. Finally we added the ability to reconcile the till online in a paperless manner.

For another customer, we added the ability to print AR invoices in plain text format and added templates (based on the POS sales template) to do this.

## **18.3 Extensions**

One can add functionality to the Perl modules in the LSMB directory and often add missing functions easily.

### **18.3.1 Examples**

For one customer, we added a module to take data from a portable data terminal collected when inventory items were taken and use that to add shrinkage and loss adjustments. We also extended the parts model to add a check id flag (for alcohol sales) and added this flag to the user interface.

For another customer, we added a complex invoice/packing slip tracking system that allowed one to track all the printed documents associated with an order or invoice.

## **18.4 Templates**

As noted before templates can be modified or extended, though sometimes this involves extending the user interface scripts. Most templates are easy enough to modify.

### **18.4.1 Examples**

For one customer we added text-only invoices for AR and AP transactions/Invoices and an ability to use Javascript in them to automatically print them on load.

## **18.5 Reports**

The fact that all the data is available within the database manager is a huge advantage of Ledger-SMB over Quickbooks and the like. The rapid development of reports allows for one to easily develop reports of any sort within Ledger-SMB.

### 18.5.1 Examples

For one customer, we developed a report of parts sold and received during arbitrary time frames. The report allows one to go back and look up the invoices involved.

## 19 Integration Possibilities

An open database system and programming API allows for many types of integration. There are some challenges, but in the end, one can integrate a large number of tools.

### 19.1 Reporting Tools

Any reporting tool which can access the PostgreSQL database can be used with Ledger-SMB for custom reporting. These can include programs like Microsoft Access and Excel (using the ODBC drivers), PgAccess (A PostgreSQL front-end written in TCL/Tk with a similar feel to Access), Rekall, Crystal Reports, OpenOffice and more.

#### 19.1.1 Examples

We have created spreadsheets of the summaries of activity by day and used the ODBC driver to import these into Excel. Excel can also read HTML tables, so one can use PostgreSQL to create an HTML table of the result and save it with a .xls extension so that Windows opens it with Excel. These could then be served via the same web server that serves Ledger-SMB.

### 19.2 Line of Business Tools on PostgreSQL

Various line of business tools have been written using PostgreSQL in large part due to the fact that it is far more mature than MySQL in areas relating to data integrity enforcement, transactional processing, and the like. These tools can be integrated with Ledger-SMB in various ways. One could integrate this program with the HERMES CRM framework, for example.

#### 19.2.1 Known Issues

Ledger-SMB uses a single 'id' sequence across many tables. At the same time it is expected that these tables do not have identical id values in their records as they are used as a sort of pseudo-foreign key by the acc\_trans table which stores the financial transaction information.

If the integration solution does not keep this in mind, it is possible to create a situation where the account transactions are ambiguously associated with a number of different types of financial transactions. This would lead to a large number of problems.

#### 19.2.2 Strategies

In general, it is advisable to run all such programs that benefit from integration in the same database but under different schemas. This allows PostgreSQL to become the main method of synchronizing the data in real time. However, sometimes this can require dumping the database, recreating the tables etc. in a different schema and importing the data back into Ledger-SMB.

One possibility for this sort of integration is to use database triggers to replicate the data between the applications in real-time. This can avoid the main issue of duplicate id's. One issue that can occur however relates to updates. If one updates a customer record in HERMES, for example, how do we know which record to update in Ledger-SMB? There are solutions to this problem but they do require some forethought.

A second possibility is to use views to allow one application to present the data from the other as its own. This can be cleaner regarding update issues, but it can also pose issues regarding duplicate id fields.

### **19.2.3 Examples**

Others have integrated Lane POS and Ledger-SMB in order to make it work better with touch screen devices. Still others have successfully integrated Ledger-SMB and Interchange. In both cases, I believe that triggers were used to perform the actual integration.

## **19.3 Line of Business Tools on other RDBMS's**

Often there are requests to integrate Ledger-SMB with applications like SugarCRM, OSCommerce, and other applications running on MySQL or other database managers. This is a far more complex field and it requires a great deal more effort than integrating applications within the same database.

### **19.3.1 Strategies**

Real-time integration is not always possible. MySQL does not support the SQL extension SQL/MED (Management of External Data) that supports non-MySQL data sources so it is not possible to replicate the data in real-time. Therefore one generally resorts to integrating the system using time-based updates. Replication may be somewhat error-prone unless the database manager supports triggers (first added to MySQL in 5.0) or other mechanisms to ensure that all changed records can be detected and replicated. In general, it is usually advisable to add two fields to the record— one that shows the insert time and one that shows the last update.

Additionally, I would suggest adding additional information to the Ledger-SMB tables so that you can track the source record from the other application in the case of an update.

In general, one must write replication scripts that dump the information from one and add it to the other. This must go both ways.

### **19.3.2 Integration Products and Open Source Projects**

While many people write Perl scripts to accomplish the replication, an open source project exists called DBI-Link. This package requires PL/Perl to be installed in PostgreSQL, and it allows PostgreSQL to present any data accessible via Perl's DBI framework as PostgreSQL tables. DBI-Link can be used to allow PostgreSQL to pull the data from MySQL or other database managers.

DBI-Link can simplify the replication process by reducing the operation to a set of SQL queries.

## **20 Customization Guide**

This section is meant to provide a programmer with an understanding of the technologies enough information to get up to speed quickly and minimize the time spent familiarizing themselves with the software. Topics in this section are listed in order of complexity. As it appeals to a narrower audience than previous discussions of this topic, it is listed separately.

### **20.1 General Information**

The main framework scripts (the ar.pl, ap.pl, etc. scripts found in the root of the installation directory) handle such basic features as instantiating the form object, ensuring that the user is logged in, and the like. They then pass the execution off to the user interface script (usually in the bin/mozilla directory).

Ledger-SMB in many ways may look sort of object oriented in its design, but in reality, it is far more data-driven than object oriented. The Form object is used largely as a global symbol table and also as a collection of fundamental routines for things like database access. It also breaks down the query string into sets of variables which are stored in its attribute hash table.

In essence one can and often will store all sorts of data structures in the primary Form object. These can include almost anything. It is not uncommon to see lists of hashes stored as attributes to a Form object.

## 20.2 Customizing Templates

Templates are used to generate printed checks, invoices, receipts, and more in Ledger-SMB. Often the format of these items does not fit a specific set of requirements and needs to be changed. This document will not include L<sup>A</sup>T<sub>E</sub>X or HTML instruction, but will include a general introduction to editing templates. Also, this is not intended to function as a complete reference.

Template instructions are contained in tags `<?lsmb` and `?>`. The actual parsing is done by the `parse_template` function in `LSMB/Form.pm`.

### 20.2.1 Page Breaks in L<sup>A</sup>T<sub>E</sub>X

The first tag one will see with L<sup>A</sup>T<sub>E</sub>X templates is `<?lsmb pagebreak num1 num2 num3 ?>`

- `num1` represents characters per line
- `num2` represents lines on first page
- `num3` represents lines on second page.

The pagebreak block is terminated by `<?lsmb end pagebreak ?>`. Any text within the pagebreak block is ignored by the template.

### 20.2.2 Conditionals

- `<?lsmb if not varname ?>` tells the parser to include the next block only if `varname` was posted by the submitting form (or set via the form hash elsewhere in the scripts). The block ends with `<?lsmb end varname ?>`
- `<?lsmb if varname ?>` tells the parser to include the block if `varname` was posted in the submitting form (or set via the form hash elsewhere in the scripts). The block ends with `<?lsmb end varname ?>`
- Lines conditionals are otherwise ignored by the parser.
- Conditionals cannot be nested, but IF's can be nested inside loops.

### 20.2.3 Loops

`<?lsmb foreach varname ?>` is used to iterate through a list of vars set by the user interface system (usually one of the files under `bin/mozilla` (or otherwise)). The block is repeated for each `varname` in a list. Block ends with `<?lsmb end varname ?>`

### 20.2.4 File Inclusion

- Files may be included with the syntax `<?lsmb include template_name ?>` where `templatename` is the name of the template within the current template directory (usually `templates/$username/`). Note that for  $\text{\LaTeX}$  templates, the `input` or `include` functionalities might be better suited for many many things.
- Cannot be used with conditionals
- Filenames cannot use slashes (`/`) or `..` due to directory transversal considerations.
- Files can force other files to be included, but the same file cannot be included more than once.

### 20.2.5 Cross-referencing and multiple passes of $\text{\LaTeX}$

In  $\text{\LaTeX}$  cross-references require two passes with `latex` to resolve. This is because the type is set page by page and the program really doesn't know on which page a given reference will fall. This becomes an even larger issue where floats are concerned as they can move between pages for formatting reasons.

In rare cases, cross-references may point at incorrect pages even with two passes (if the inclusion of the cross-reference data moves the object to another page). In this case you will need to use three passes of  $\text{\LaTeX}$  in order to have accurate references.

Ledger-SMB as of the time of this writing (2.6.8) only makes one pass at the  $\text{\LaTeX}$  file. To force it to make more than one pass, open `Form.pm` with your favorite text editor. Look for the line:

```
system("latex -interaction=nonstopmode $self->{ tmpfile} > $ self->{ tmpfile} .err");
```

Duplicate this line for two passes, or add two copies if you need three passes.

### 20.2.6 Variable Substitution

The following format is used for variable substitution:

- `<?lsmb varname options ?>`Options are one or more (whitespace separated) of:
  - `align=left/right/center`
  - `width=chars` where `chars` is the width in characters before wrapping
  - `offset=chars` where `chars` is the number of spaces to (depending on alignment).

## 20.3 Customizing Forms

Data entry forms and other user interface pieces are in the `bin` directory. In Ledger-SMB 1.0.0 and later, symlinks are not generally used.

Each module is identified with a two letter combination: `ar`, `ap`, `cp`, etc. These combinations are generally explained in the comment headers on each file.

Execution in these files begins with the function designated by the `form->{action}` variable. This variable is usually derived from configuration parameters in the `menu.ini` or the name of the button that was clicked on to submit the previous page. Due to localization requirements, the following process is used to determine the appropriate action taken:

The `$locale->getsub` routine is called. This routine checks the locale package to determine if the value needs to be translated back into an appropriate LSMB function. If not, the variable is lower-cased, and all spaces are converted into underscores.

In general there is no substitute for reading the code to understand how this can be customized and how one might go about doing this.



## 20.4 Customizing Modules

The Perl Modules (.pm files) in the LedgerSMB directory contain the main business logic of the application including all database access. Most of the modules are relatively easy to follow, but the code quality leaves something to be desired. The API calls themselves are likely to be replaced in the future, so work on documenting API calls is now focused solely on those calls that are considered to be stable. At the moment, the best place to request information on the API's is on the Developers' Email List ([ledger-smb-devel@lists.sourceforge.net](mailto:ledger-smb-devel@lists.sourceforge.net)).

Many of these modules have a fair bit of dormant code in them which was written for forthcoming features, such as payroll and bills of materials.

One can add a new module through the normal means and connect it to other existing modules.

### 20.4.1 Database Access

The `$form` object provides two methods for accessing the database. The `$form->dbconnect(%myconfig)` method commits each individual statement as its own transaction. The `$form->dbconnect_noauto(%myconfig)` method requires a manual commit. Both these functions are thin wrappers around the standard Perl DBI operations.

## 20.5 CLI Examples

Louis Moore contributed some SQL-Ledger CLI examples that still work for LedgerSMB. You can find his page at <http://www.sql-ledger.org/cgi-bin/nav.pl?page=contrib/moore.html&title=Louis%20B.%20Moore>.

## Part III

# Appendix

## A Where to Go for More Information

There are a couple of relevant sources of information on Ledger-SMB in particular.

The most important resources are the LedgerSMB web site (<http://www.ledgersmb.org>) and the email lists found at our Sourceforge site.

In addition, it is generally recommended that the main bookkeeper of a company using Ledger-SMB work through at least one accounting textbook. Which textbook is not as important as the fact that a textbook is used.

## B Quick Tips

### B.1 Understanding Shipping Addresses and Carriers

Each customer can have a default shipping address. This address is displayed prominently in the add new customer screen. To change the shipping address for a single order, one can use the ship to button at the bottom of the quote, order, or invoice screen.

The carrier can be noted in the Ship Via field. However, this is a freeform field and is largely used as commentary (or instructions for the shipping crew).

### B.2 Handling bad debts

In the event that a customer's check bounces or a collection requirement is made, one can flag the customer's account by setting the credit limit to a negative number.

If a debt needs to be written off, one can either use the allowance method (by writing it against the contra asset account of "Allowance for Bad Debts" or using the direct writeoff method where it is posted as an expense.

## C Step by Steps for Vertical Markets

### C.1 Common Installation Errors

- Ledger-SMB is generally best installed in its own directory outside of the wwwroot directory. While it is possible to install it inside the wwwroot directory, the instructions and the faq don't cover the common problems here.
- When the chart of accounts (COA) is altered such that it is no longer set up with appropriate items, you can make it impossible to define goods and services properly. In general, until you are familiar with the software, it is best to rename and add accounts rather than deleting them.

### C.2 Retail With Light Manufacturing

For purposes of this example we will use a business that assembles computers and sells them on a retail store.

1. Install Ledger-SMB.
2. Set preferences, and customize chart of accounts.

- (a) Before customizing the COA it is suggested that you consult an accountant.
3. Define Goods, Labor, and Services as raw parts ordered from the vendors.
    - These are located under the goods and services menu node.
  4. Define assemblies.
    - These are also located under goods and services.
    - Component goods and services must be defined prior to creating assembly.
  5. Enter an AP Invoice to populate inventory with proper raw materials.
    - One must generally add a generic vendor first. The vendor is added under AP->Vendors->Add Vendor.
  6. To pay an AP invoice like a check, go to cash->payment. Fill out appropriate fields and click print.
    - Note that one should select an invoice and enter in the payment amount in the appropriate line of the invoice list. If you add amounts to the master amount list, you will find that they are added to the amount paid on the invoice as a prepayment.
    - The source field is the check number.
  7. Stock assemblies
  8. One can use AR Invoices or the POS interface to sell goods and services.
    - Sales Invoice
      - Can be generated from orders or quotations
      - Cannot include labor/overhead except as part of an assembly
      - One can make the mistake of printing the invoice and forgetting to post it. In this event, the invoice does not officially exist in the accounting system.
      - For new customers, you must add the customer first (under AR->Customers->Add Customer).
    - POS Interface
      - Cannot include labor/overhead except as part of an assembly
      - Printing without posting is often even easier in the POS because of the rapid workflow. Yet it is just as severe a problem.
    - Ecommerce and Mail Order Operations
      - See the shipping workflow documentation above.
    - Customers are set up by going to AR->Customers->Add Customer (or the equivalent localized translation). The appropriate fields are filled out and the buttons are used at the bottom to save the record and optionally use it to create an invoice, etc.
      - Saving a customer returns to the customer screen. After the appropriate invoice, transaction, etc. is entered and posted, Ledger-SMB will return to the add customer screen.
  9. One can use the requirements report to help determine what parts need to be ordered though one cannot generate PO's directly from this report.

Note, the needs of Ledger-SMB are mostly useful for light manufacturing operations (assembling computers, for example). More manufacturing capabilities are expected to be released in the next version.

A custom assembly is a bit difficult to make. One must add the assembly prior to invoice (this is not true of goods and services). If the assembly is based on a different assembly but may cost more (due to non-standard parts) you can load the old assembly using Goods and Services->Reports->Assemblies and then make necessary changes (including to the SKU/Partnumber) and save it as new.

Then one can add it to the invoice.

## D Glossary

**BIC** Bank Identifier Code is often the same as the S.W.I.F.T. code. This is a code for the bank a customer uses for automated money transfers.

**COGS** is Cost of Goods Sold. When an item is sold, then the expense of its purchase is accrued as attached to the income of the sale. It is tracked as COGS.

**Credit** : A logical transactional unit in double entry accounting. It is the opposite of a debit. Credits affect different account types as follows:

**Equity** : Credits are added to the account when money is invested in the business.

**Asset** : Credits are added when money is deducted from an asset account.

**Liability** : Credits are added when money is owed to the business account.

**Income** : Credits are added when income is earned.

**Expense** : Credits are used to apply adjustments at the end of accounting periods to indicate that not all the expense for an AP transaction has been fully accrued.

**Debit** : A logical transactional unit in double entry accounting systems. It is the opposite of a credit. Debits affect different account types as follows:

**Equity** : Debits are added when money is paid to business owners.

**Asset** : Debits are added when money is added to an account.

**Liability** : Debits are added when money that is owed is paid off.

**Income** : Debits are used to temporarily adjust income to defer unearned income to the next accounting period.

**Expense** : Debits are added as expenses are incurred.

**IBAN** International Bank Account Number is related to the BIC and is used for cross-border automated money transfers.

**List** Price is the recommended retail price.

**Markup** is the percentage increase that is applied to the last cost to get the sell price.

**ROP** Re-order point. Items with fewer in stock than this will show up on short reports.

**Sell** Price is the price at which the item is sold.

**SKU** Stock Keeping Unit: a number designating a specific product.

**Source** Document : a paper document that can be used as evidence that a transaction occurred. Source documents can include canceled checks, receipts, credit card statements and the like.

**Terms** is the number of days one has to pay the invoice. Most businesses abbreviate the terms as Net n where n is the number of days. For example, Net 30 means the customer has 30 days to pay the net due on an invoice before it is late and incurs late fees. Sometimes you will see 2 10 net 30 or similar. This means 2% discount if paid within 10 days but due within 30 days in any case.

## E GNU Free Documentation License

Version 1.2, November 2002

Copyright ©2000,2001,2002 Free Software Foundation, Inc.

51 Franklin St, Fifth Floor, Boston, MA 02110-1301 USA

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

### Preamble

The purpose of this License is to make a manual, textbook, or other functional and useful 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, in any medium, that contains a notice placed by the copyright holder saying it can be distributed under the terms of this License. Such a notice grants a world-wide, royalty-free license, unlimited in duration, to use that work under the conditions stated herein. The "**Document**", below, refers to any such manual or work. Any member of the public is a licensee, and is addressed as "**you**". You accept the license if you copy, modify or distribute the work in a way requiring permission under copyright law.

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. (Thus, 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. If a section does not fit the above definition of Secondary then it is not allowed to be designated as

Invariant. The Document may contain zero Invariant Sections. If the Document does not identify any Invariant Sections then there are none.

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 Front-Cover Text may be at most 5 words, and a Back-Cover Text may be at most 25 words.

A **"Transparent"** copy of the Document means a machine-readable copy, represented in a format whose specification is available to the general public, that is suitable for revising the document 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, or absence of markup, has been arranged to thwart or discourage subsequent modification by readers is not Transparent. An image format is not Transparent if used for any substantial amount of text. 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, PostScript or PDF designed for human modification. Examples of transparent image formats include PNG, XCF and JPG. Opaque formats include 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, PostScript or PDF 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.

A section **"Entitled XYZ"** means a named subunit of the Document whose title either is precisely XYZ or contains XYZ in parentheses following text that translates XYZ in another language. (Here XYZ stands for a specific section name mentioned below, such as **"Acknowledgements"**, **"Dedications"**, **"Endorsements"**, or **"History"**.) To **"Preserve the Title"** of such a section when you modify the Document means that it remains a section "Entitled XYZ" according to this definition.

The Document may include Warranty Disclaimers next to the notice which states that this License applies to the Document. These Warranty Disclaimers are considered to be included by reference in this License, but only as regards disclaiming warranties: any other implication that these Warranty Disclaimers may have is void and has no effect on the meaning of this License.

## 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 (or copies in media that commonly have printed covers) 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 computer-network location from which the general network-using public has access to download using public-standard network protocols a complete Transparent copy of the Document, free of added material. 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 fewer than five), unless they release you from this requirement.
- 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", Preserve 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. For any section Entitled "Acknowledgements" or "Dedications", Preserve the Title of the section, 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 to be Entitled "Endorsements" or to conflict in title with any Invariant Section.
- O. Preserve any Warranty Disclaimers.

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, and that you preserve all their Warranty Disclaimers.

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, is called an "aggregate" if the copyright resulting from the compilation is not used to limit the legal rights of the compilation's users beyond what the individual works permit. When the Document is included in an aggregate, this License does not apply to the other works in the aggregate which 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 half of the entire aggregate, the Document's Cover Texts may be placed on covers that bracket the Document within the aggregate, or the electronic equivalent of covers if the Document is in electronic form. Otherwise they must appear on printed covers that bracket 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, and all the license notices in the Document, and any Warranty Disclaimers, provided that you also include the original English version of this License and the original versions of those notices and disclaimers. In case of a disagreement between the translation and the original version of this License or a notice or disclaimer, the original version will prevail.

If a section in the Document is Entitled "Acknowledgements", "Dedications", or "History", the requirement (section 4) to Preserve its Title (section 1) will typically require changing the actual title.

## 9. TERMINATION

You may not copy, modify, sublicense, or distribute the Document except as expressly provided for under this License. Any other attempt to copy, modify, sublicense or distribute the Document is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

## 10. FUTURE REVISIONS OF THIS LICENSE

The Free Software Foundation may publish new, revised versions of the GNU Free Documentation License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns. See <http://www.gnu.org/copyleft/>.

Each version of the License is given a distinguishing version number. If the Document specifies that a particular numbered version of this License "or any later version" applies to it, you have the option of following the terms and conditions either of that specified version or of any later version that has been published (not as a draft) by the Free Software Foundation. If the Document does not specify a version number of this License, you may choose any version ever published (not as a draft) by the Free Software Foundation.

## **ADDENDUM: How to use this License for your documents**

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 1.2 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. A copy of the license is included in the section entitled "GNU Free Documentation License".

If you have Invariant Sections, Front-Cover Texts and Back-Cover Texts, replace the "with...Texts." line with this:

with the Invariant Sections being LIST THEIR TITLES, with the Front-Cover Texts being LIST, and with the Back-Cover Texts being LIST.

If you have Invariant Sections without Cover Texts, or some other combination of the three, merge those two alternatives to suit the situation.

If your document contains nontrivial examples of program code, we recommend releasing these examples in parallel under your choice of free software license, such as the GNU General Public License, to permit their use in free software.