



Running your business with LedgerSMB 1.13

DRAFT / WORK IN PROGRESS
See the [Preface](#) for current status.

Erik Hülsmann Neil Tiffin
Chris Travers

<https://book.ledgersmb.org>

August 21, 2026

Contents

List of Figures	15
List of Tables	18
Preface	21
Organization of the book	21
Comments and questions	22
Other information sources	22
Acknowledgments	22
I Overview	23
1 What is LedgerSMB	25
1.1 Introduction	25
1.2 Supported functionality	26
1.3 Feature comparison with alternatives	27
1.4 System requirements	27
1.5 Application architecture	27
1.6 License	28
2 Reasons to use LedgerSMB	31
2.1 Internal control	32
2.2 Accounting principles	32
2.3 Impact of tight integration	32
3 Introduction to accounting	35
3.1 Double-entry accounting	36
3.2 Cash versus accrual basis	37

3.3	Valuation of inventory	38
II	Getting started	41
4	Overview	43
4.1	Introduction	43
5	Creating a company database	45
5.1	Using setup.pl	45
5.1.1	Step 1: setup.pl login	45
5.1.2	Step 2: Company creation	46
5.1.3	Step 3: Selection of a Chart of Accounts	47
5.1.4	Step 4: Load template	48
5.1.5	Step 5: Initial user	49
6	The first login	53
6.1	Introduction	53
6.2	Steps to the first login	53
6.2.1	Login screen	54
6.2.2	Selecting a password	54
6.2.3	Setting user preferences	57
6.2.4	Setting system defaults	58
6.3	Setting up a bank account or credit card	59
6.4	Creating initial bank balance	62
6.5	Checking and adjusting the chart of accounts	65
6.5.1	Accounts list	65
6.6	Checking sales tax rates	65
7	Building up stock	67
7.1	Overview	67
7.2	Defining parts	68
7.2.1	Single items	68
7.2.2	Single-item & “multi-item pre-packaged”	69
7.3	Defining part groups	69
7.4	Defining vendors	70

CONTENTS	5
7.5 Requesting quotations	71
7.6 Following up on a quotation	72
7.7 Receiving ordered items	73
7.8 Receiving an invoice	74
7.9 Paying an invoice	74
8 Ramping up to the first sale	77
8.1 Creating a customer	77
8.2 Sending out a quote	81
8.3 Sending out a sales order	81
9 Shipping sales	83
10 Invoicing	85
10.1 Handling sales taxes	85
10.1.1 Invoices with taxes included	85
10.1.2 Invoices with explicit tax amounts	85
10.2 Invoice Editing	85
11 Collecting sales invoice payments	87
11.1 Customer payments	87
11.2 Customer payment mismatch	87
11.2.1 Pardoning and underpayment	87
11.2.2 Handling large mismatches	87
11.2.3 Pardoning small mismatches	87
12 Paying vendor invoices	89
12.1 Handling vendors who match amounts to exact invoices	89
12.2 Handling vendors with running balances	89
12.3 Handling bounced checks	89
12.3.1 Voiding checks	89
13 Monitoring arrears	91
13.1 Handling interest on arrears	91
14 Branching out: services	93

14.1	Creation / assignment to different accounts	93
14.2	Recording service hours	93
14.3	Customer approval on service hours	93
14.4	Invoicing services	93
15	Branching out II: service subscriptions	95
III	Configuration	97
16	Overview	99
16.1	Introduction	99
17	Global configuration	101
17.1	Apache	101
17.2	PostgreSQL	101
17.3	LedgerSMB version numbers	101
17.4	LedgerSMB Configuration	102
17.4.1	ledgersmb.yaml	102
18	Per company configuration	107
18.1	Matching your business processes	107
18.2	Administrative user	108
18.3	Chart of Accounts	108
18.3.1	Predefined Chart of Accounts	108
18.3.2	Account structure	109
18.3.3	Account setup	110
18.3.4	Special accounts	114
18.3.5	Importing chart of accounts	115
18.4	System menu settings	115
18.4.1	Audit control	115
18.4.2	Taxes	116
18.4.3	Defaults	117
18.4.4	Year end	121
18.4.5	Admin users	121
18.4.6	Chart of accounts	121

CONTENTS	7
18.4.7 Warehouses	121
18.4.8 Reporting units	121
18.4.9 Type of business	123
18.4.10 Languages	123
18.4.11 Standard Industry Code (SIC)	123
18.4.12 Templates	124
 IV Administration	 131
19 Overview	133
19.1 Introduction	133
 20 First Use	 135
 21 Within-application tasks	 137
 22 DBA tasks from setup.pl	 139
 23 Outside-application tasks	 141
 24 User management	 143
24.1 Introduction	143
24.2 User creation	143
24.3 User authorization	144
24.4 Maintaining users	145
24.4.1 Editing user information and authorizations	145
24.4.2 Changing user preferences	145
24.4.3 Deleting users	146
24.5 User imports	146
24.6 setup.pl users	147
 25 Definition of goods and services	 149
25.1 Definition of goods	149
25.1.1 Definition of parts	149
25.1.2 Definition of part groups	152
25.1.3 Definition of assemblies	152

25.1.4	Definition of overhead or labor	152
25.1.5	Parts versus assemblies for multi-item-package sales . . .	152
25.2	Definition of services	153
25.2.1	Taxation of services	153
25.3	Alternative accounts (GIFI)	153
25.3.1	Maintaining GIFI	153
25.4	Reporting	154
26	Taxes	155
26.1	Overview	155
26.2	Tax account configuration	155
26.3	Tax calculation examples	157
26.4	Tax calculation plug-ins	157
26.5	Tax forms	157
26.5.1	Overview	157
26.5.2	Creating tax forms	158
26.5.3	Using tax forms	158
27	Pricing	159
27.1	Introduction	159
27.2	Definition of types of business	160
27.3	Definition of price groups	160
27.4	Using the price matrix	160
27.4.1	Price matrix example: simple pricing	161
27.4.2	Price matrix example: Quantity pricing	161
27.4.3	Price matrix example: Time-based (Sale) pricing	162
28	Auditing	163
29	Contingency planning	165
29.1	Backup and restore	165
29.1.1	Backup using setup.pl	165
29.1.2	Backup using PostgreSQL administration tools	165
29.1.3	Restore	165
29.2	Advanced PostgreSQL: replication	165

<i>CONTENTS</i>	9
30 Software updates	167
30.1 Introduction	167
30.2 Backups	167
30.3 Software upgrade	168
30.4 Database upgrade	168
31 Optional Features	169
31.1 PDF and Postscript Documents	169
31.2 Attaching Uploaded Files to PDF Invoices	170
31.3 OpenDocument Spreadsheet Output	170
31.4 Microsoft Excel Output	170
V Business processes	171
32 Overview	173
33 Customers and vendors	175
33.1 Introduction	175
33.2 Contacts	175
33.3 Contact Search	177
33.4 Creating customers and vendors	177
33.4.1 Creating a Company	177
33.4.2 Creating a customer account	178
33.4.3 Creating a vendor account	179
33.5 Multiple customers within one company	179
33.6 Creating vendors from customers	179
33.7 Maintaining contact information	179
34 Quotations from Vendors and for Customers	181
34.1 Creating Quotations and RFQs	181
34.2 Expediting quotations	183
34.2.1 Printing	183
34.2.2 Sending by e-mail	183
34.3 Attaching files to quotations	183

35 Sales and vendor orders	185
35.1 Creating new orders	185
35.2 Creating orders from quotations	185
35.3 Creating orders from projects	185
35.4 Creating purchase orders from sales orders	185
35.5 Combining orders	185
35.6 Recurring orders	185
35.7 Incomplete orders	185
36 Inventory management	187
36.1 Shipping	187
36.1.1 Pick lists	187
36.1.2 Packing list	187
36.2 Receiving	187
36.3 Partial shipments or receipts	188
36.4 Handling returns	189
36.5 Transferring between warehouses	189
36.6 Inventory reporting	189
36.6.1 Inventory status report	189
36.6.2 Inventory activity report	189
37 Sales and vendor invoice handling	191
37.1 Creating invoices from orders	191
37.2 Creating new invoices	191
37.2.1 Invoices	192
37.2.2 Transactions	192
37.2.3 Email Invoices	192
37.3 Recurring invoices	192
37.4 Invalidating invoices	192
37.5 Correcting and deleting invoices	193
37.6 Handling invoice disputes	194
37.6.1 Credit and debit invoices	194
37.6.2 Credit and debit notes	194
37.6.3 AR and AP Vouchers	194

CONTENTS	11
37.7 Remarks	195
38 Shop sales	197
38.1 Opening and closing the cash register	197
38.2 Shop sales invoices	197
39 Manufacturing management	199
39.1 Producing sales orders	199
39.1.1 Work orders	199
40 Managing accounts receivable and payable	201
40.1 Creating generic AR/AP items	201
40.2 Handling refunds, over-payments and advances	201
40.3 Handling returns	201
41 Credit risk management	203
41.1 Introduction	203
41.2 Limit management	203
41.3 Granting payment terms	203
41.4 Managing arrears	203
41.4.1 Monitoring AR aging	204
41.4.2 Tracking invoice history	204
41.4.3 Special treatment of invoices	204
41.5 Interest on arrears	205
41.6 Allowance for doubtful accounts	205
41.7 Writing off bad debt	205
41.7.1 Direct write-off	205
41.7.2 Allowed-for write-off	205
42 Receipts and payment processing	207
42.1 Introduction	207
42.2 Single receipts or payments	207
42.3 Batch receipts and payments	208
42.4 Check payments	208
42.5 Using over payments	208

42.6 Receipt and payment reversal	209
42.7 Over payment reversal	209
42.8 Receipts and payments in foreign currencies	210
42.8.1 Exchange rates	210
42.8.2 Exchange rate classes	211
42.8.3 User workflow	211
43 Accounting	213
43.1 Separation of duties: Transaction approval	213
43.2 Entering non sales transactions	213
43.2.1 Corrections	213
43.2.2 Transfer of money between bank accounts	213
43.2.3 Wages	213
43.2.4 Entering other general accounting documents	213
43.3 Bank reconciliation	214
43.3.1 Importing bank transactions - OFX	214
43.3.2 Importing bank transactions - CAMT.053	214
43.3.3 Importing bank transactions - CSV	214
43.4 Period closing	214
43.5 Year-end processing	215
43.6 Fixed asset accounting	216
43.7 Reporting	216
43.7.1 Tax reporting	216
43.7.2 Tax reporting using tax forms	216
43.7.3 Income statement	216
43.7.4 Balance sheet	216
43.7.5 Trial balance	216
44 Budgets	217
 VI General Operation	 219
45 User Interface	221
45.1 Reports	221

CONTENTS	13
VII Customization	223
46 Overview	225
46.1 Introduction	225
46.2 Workflow configuration	226
46.2.1 Workflow observers	228
46.3 Document formatters	228
46.4 Bank statement importers	229
46.5 HTTP request interceptors	229
46.6 Outgoing mail transport	230
46.7 Sales/VAT tax calculation	230
47 Batch data import methods	231
47.1 Custom bank statement import	231
47.2 Chart of Accounts import	231
48 Add-ons and plug-ins	233
49 Web service APIs	235
VIII Workflows	237
50 Overview	239
IX Appendices	241
A Differences between version 1.2 and 1.3	243
A.1 Users	243
B Why deleting invoices is not supported	245
B.1 Invoices are debt claims	245
B.2 Cost Of Goods Sold become ambiguous	245
C Installation	247
C.1 Quick preview using Docker	247
C.2 Production using Docker	247

C.3	Native installation	248
C.4	Manual installation	248
C.5	Development using Docker	248
C.6	Linux distribution	249
D	Migration	251
D.1	Introduction	251
D.2	From older LedgerSMB versions	252
D.3	From SQL-Ledger 2.8	252
D.4	From older SQL-Ledger versions	252
D.5	From other accounting packages	252
D.5.1	Overview	252
D.5.2	Migrating customers	252
D.5.3	Migrating parts, services,	253
D.5.4	Migrating the balance	253
D.6	Migrating between PostgreSQL versions	253
D.6.1	Migration from PostgreSQL 8.3 (or earlier) to 8.4	255
E	Listing of application roles	257
F	Deprecated Content	267
F.1	ledgersmb.conf	267
G	Open source explained	271
G.1	An open source application	271
G.2	An open source book	271
H	Project history	273
I	Copyright and license	275
I.1	License summary	275
I.2	Legal full text	276
	Acronyms	284
	Acronyms	285

<i>CONTENTS</i>	15
Definitions of terms	287
Definition of terms	287
Index	290

List of Figures

5.1	setup.pl login screen	46
5.2	Chart of accounts - country selection	47
5.3	Chart of accounts - selection	48
5.4	Template selection	49
5.5	setup.pl initial user creation screen	50
5.6	setup.pl successful completion screen	51
6.1	login.pl opening screen	54
6.2	login.pl welcome screen	55
6.3	Password change screen	56
6.4	User preferences	57
6.5	System defaults	58
6.6	Bank account setup - menu items	60
6.7	Bank account setup - account setup screen	61
6.8	Bank account setup - initial bank balance	63
6.9	Bank account setup - initial bank balance after update	63
6.10	Bank account setup - initial bank balance after save	63
6.11	Bank account setup - balance sheet confirmation	64
6.12	Tax rate adjustment screen	66
7.1	Add first part	68
7.2	Company entry screen	70
7.3	Vendor account screen	71
7.4	Vendor account screen - with purchase discount account	72
7.5	RFQ entry screen	73

7.6	RFQ e-mail screen	74
7.7	RFQ entry screen	75
7.8	Order receipt entry screen	75
8.1	Add contact screen	78
8.2	Credit account screen	79
8.3	Credit account 2nd screen	80
8.4	Addresses screen	80
18.1	Configure Account Hierarchy	109
18.2	Account Setup	110
18.3	Report Using Hierarchy	110
18.4	System Default Accounts	114
18.5	System templates screen	124
18.6	Templates edit invoice screen	125
24.1	Screen for user creation - step 1	144
24.2	Screen for user creation - step 2	145
24.3	User preferences screen	146
26.1	Tax account configuration screen	156
34.1	RFQ entry screen	181
36.1	Saved purchase order	188
36.2	Search screen for purchase order item receipts	189
36.3	Search results screen for purchase order item receipts	190
36.4	Purchase order receipt screen	190
43.1	Period and Year End Closing	215
46.1	Workflow triggered from another workflow	226
46.2	Conditional workflow actions	227

List of Tables

5.1	Create first user entry data	51
6.2	First Login - Change initial password data	57
6.3	First Login - Change user defaults data	59
8.1	Company data	78
8.2	Credit Accounts data	79
8.3	First company address data	81
18.1	Reporting Unit Class Fields	122
18.2	Reporting Unit Fields	122
27.1	Customer specific price matrix	161
27.2	Price group specific price matrix	161
27.3	Price matrix configuration	161
27.4	Price matrix configuration	162

Preface

The current development goal for the book is to get the content current and accurate, not necessarily perfect structure and grammar.

You will find many incomplete sections which should be regarded as an outline for the finished book. However, some sections are nearly complete.

Because LedgerSMB is undergoing active infrastructure and API changes in 2026, some portions of this book will not be written until those changes are nearing completion.

Intended audience

The book has been split up into several parts intended for starting entrepreneurs, (potential) users, application integration developers and administrators. It offers something for everybody who works with or considers working with LedgerSMB.

Organization of the book

The first part ('**Overview**') is intended for everybody who tries to get an impression of LedgerSMB, with chapters on technology, features, licensing and a bit of project history.

The second part ('**Getting started**') leads new users and especially new entrepreneurs step by step through the application using a tutorial format. This part also contains additional explanations on good business and accounting processes.

The third part ('**Configuration**') describes system wide configuration settings such as the LedgerSMB configuration file and those required for dependencies like PostgreSQL. Additionally it describes the company specific settings within the application, such as audit control settings and required special accounts. This part is required material for anybody who wants to set up a LedgerSMB instance.

The fourth part ('**Administration**') discusses the different topics regarding application administration such as user management, product definition, taxes and etc. Anybody responsible for maintaining application instances in good health should read this part.

The fifth part ('**Business processes**') discusses LedgerSMB's best practices. This part is highly advised reading for everybody who is a LedgerSMB user and should be considered required reading for 'business architects': those in the company who decide about process design and execution.

The sixth part ('**Customization**') discusses how to customize the application. This part is intended for developers of custom extensions (such as company

specific integrations), add-ons and plug-ins.

The last part ('[Appendices](#)') contain supplemental reference information on various subjects, such as a listing and description of the authorization roles in the application as well as information on migration to or upgrading of LedgerSMB versions.

Comments and questions

The content of this book is being developed in the ledgersmb-book Github project at <https://github.com/ledgersmb/ledgersmb-book>.

Comments and enhancement ideas can be filed in the Github issue tracker. Additionally, discussion about the content can be organized on the mail list or chat channels listed at <https://ledgersmb.org/topics/support>.

We desperately need to finish adding content. Feel free to suggest new or changes to the content using the Github issue tracker. Don't worry about knowing details of L^AT_EX or how the content should be structured. We'll take care of that.

Up-to-date versions of this book in both PDF and HTML formats can be found on the book website at <https://book.ledgersmb.org>.

General information about the community around LedgerSMB can be found in the source repository contributing document [CONTRIBUTING.md](#)

Other information sources

The following sources of information, while not maintained by the LedgerSMB project, might be helpful in understanding the project.

<https://deepwiki.com/ledgersmb/LedgerSMB> Contributed by jgart.

<https://gropedia.com/page/ledgersmb>

<https://en.wikipedia.org/wiki/LedgerSMB>

Acknowledgments

Håvard Sørli for his efforts to help develop the original book outline.

Members of the <https://app.element.io/#/room/#ledgersmb:matrix.org> chat room.

Part I

Overview

Chapter 1

What is LedgerSMB

1.1 Introduction

LedgerSMB is an open source, web based **Enterprise Resource Planning (ERP)** system. Which is a system which supports business processes of all disciplines throughout the organization, automating as much of those processes as possible.

To illustrate this, consider the process of selling goods to a customer in a trade company. Typically, a customer will request a quote, which Sales will provide. When satisfied he or she will convert the quote to an order. Sales will register the order, leading to order-pickers to collect one or more shipments. Upon completion, Finance sends out an invoice and records the customer's payment. LedgerSMB supports this process by automating the conversion from quotation to order, order to shipment and shipment to invoice, as well as providing pack lists and other production related documents.

LedgerSMB includes a powerful framework which supports building your own extensions and integrations with other applications. Through this philosophy, it aspires to be *the* (open source) integrated administration system.

The software is being developed by the LedgerSMB project with its homepage at <https://ledgersmb.org/>.

However, the actual project activity can be witnessed at the Github project site hosted at <https://github.com/ledgersmb/LedgerSMB> and on the mailing lists or chat channel listed at <https://ledgersmb.org/topics/support>.

Its open source nature allows you to download LedgerSMB and use it with any infrastructure you like. So there's no vendor lock-in: you can always take your data and set up your system with another hardware vendor or set up your own hardware.

1.2 Supported functionality

As most **ERP** systems, LedgerSMB's functionalities are grouped into modules. Many modules are integrated parts of the base application. New features can be implemented in separate modules at first to allow evaluation of the feature set. When a it has become sufficiently stable, the new module may be integrated in the base application. As of that time, the existing features will be frozen, meaning that the utmost will be done to prevent it from being changed.

These separate modules - which are called **add-ons** - have to be installed separately. After installation they become seamless parts of LedgerSMB with no visible difference from the base application. An additional benefit of having **add-ons** is that they can follow different release schedules and separate maturity levels than the main application.

LedgerSMB 1.13 features the following integrated modules:

- General ledger
- Payment and Accounts payable
- Invoicing and Accounts receivable
- Fixed asset accounting
- Time registration and invoicing
- Point of Sale
- Quotation and Order management
- Manufacturing
- Inventory (warehousing) and shipping management
- VAT reporting
- Controlling
 - Department accounting
 - Project accounting
 - Job accounting
 - Fund accounting
 - Customer accounting
 - Vendor accounting
 - Lot accounting
- Application administration

There are no known **add-ons** available at this time.

With this list of modules and **add-ons** LedgerSMB has been successfully implemented in a wide range of companies of varying types and sizes: shops, manufacturing companies and service oriented businesses up to as big as @@TODO million transactions, growing by 50,000 transactions (spread across **AR**, **AP** and **GL**) *per month*.

1.3 Feature comparison with alternatives

@@@ TODO

Packages to compare to:

- GNUcash
- OSfinancials
- ERP5
- Odoo
- xTuple

1.4 System requirements

The [README file](#) which comes with every LedgerSMB software release should be considered the canonical source for system requirements. In summary, the following technical components are required:

- A web server (known working are Nginx, Apache, lighttpd and Varnish)
- Modern Perl 5, with [additional modules](#)
- PostgreSQL 13 or higher
- (optional) $\LaTeX$ or $X_{\LaTeX}$ from the Te $\TeX$ or TeXLive TeX distributions

More information about system requirements for clients and servers is available at <https://ledgersmb.org/content/system-requirements>.

System requirements such as required RAM and number of CPUs and their speed largely depend on the expected system activity. However, any modern [VPS](#) should provide enough memory and storage to satisfy a reasonable number of users.

1.5 Application architecture

Due to its heritage from [SQL Ledger](#) and the on-going process of rewriting the inherited code, the architecture differs between parts of the application: the old parts and the ones which have already been rewritten.

Overall, the application consists of five layers:

- The web browser (as the user interface)
- The web server (as a network traffic handler)

- The Perl (web)server process (as an application [Perl Web Server Gateway Interface \(PSGI\)](#))
- The database (as an application - [PL/SQL](#))
- The database (as storage)

In version 1.10 the user interface is a [Single Page Application](#) using a mixture of new (e.g. [Vue 3](#), [webpack 5](#) and [Web Components](#)) and old (e.g. [Dojo Toolkit](#)) technology.

The web server is an optional component that's highly recommended for TLS termination and protecting the back end Perl HTTP server from malicious traffic¹. It may also be used for serving static page components such as images, style sheets and JavaScript component as an efficiency measure.

Recent releases add separation between business logic and process state with the latter being stored in the database and managed at the Perl layer. Other than that, it's the Perl layer's responsibility to forward web requests to the database and presenting the resulting data in response to the user interface. The original goal to reduce the Perl layer to be a "glue" layer between the web server and the database has been abandoned with the introduction of this additional (business process) layer.

At the "database as an application" layer, [PL/SQL](#) functions implement business functionality, such as creating [cost of goods sold \(COGS\)](#) accounting entries.

The "database as storage" layer is responsible for storing data with consistency and integrity; to that extent (and much more than in [SL](#)) constraints and triggers have been implemented. An additional role for the storage layer is to enforce data access rules; i.e. to protect data from being accessed by unauthorized users.

The main difference between old and new coding paradigms is at the Perl application layer: older code generates HTML fragments while newer code delivers data through web services. As part of the on-going code restructuring, there's a major effort to create a [Representational State Transfer \(REST\)](#) web service based [Application Programmer Interface \(API\)](#). More [API](#) information is available in Chapter [49](#) starting at page [235](#). The intent here is to facilitate integration with other applications used by businesses (both in the business itself or as provided by customers and vendors).

1.6 License

LedgerSMB is available under the terms of the *GNU Public License version 2*, in short: *GPLv2*².

¹This setup is an industry-wide accepted best practice for deploying web applications; web servers like Apache or Nginx have been and continue to be extensively security-reviewed. No other implementations can achieve the same level of scrutiny.

²<https://opensource.org/licenses/gpl-2-0/>

The project attaches the following meaning to this license: The copyright holders grant you the right to copy and redistribute the software. In case you make any modifications to the software you're obligated to publish the changes *if you distribute the software*. You are always free to provide third party access to the **API** from your modified software without being required to disclose your changes.

The project considers the **API** to include:

- Database tables
- URLs with their input and output
- Webservices of any kind
- Function and object calls

The effect of this interpretation is that changes directly to the code base as well as inheritance of classes defined in the software constitute "making modifications".

Chapter 2

Reasons to use LedgerSMB

Jack finds several tools which suit his requirements to some extent or another. After evaluation of his options he decides to use LedgerSMB for the following reasons:

- Centralized data storage
- Actively developed
- Development team with security focus
- Access to the application requires only a web browser
- Integrated sales, shipping, invoicing, purchasing and accounting
- Open source solution, so no vendor lock in
- The roadmap appeals to him, because it has web services payrolling on it
- There are multiple vendors offering commercial support, including hosted options
- The developers envision building a platform out of it: creating the building blocks required to build a company on
- Own your own data
- Freedom to change
 - Support organization
 - Developer (organization)
 - Data storage provider
 - Application service provider

2.1 Internal control

Internal control¹ helps organizations to prevent and detect fraud by introducing checks and balances to assess effectiveness and validity of transactions in the organization and thereby in its ERP and accounting system(s).

2.2 Accounting principles

The International Accounting Standards (IAS) and International Financial Reporting Standards (IFRS) describe requirements for financial statements (reports) and the underlying accounting process². Said requirements include qualitative characteristics:

1. Relevance
2. Faithful representation
3. Comparability
4. Verifiability
5. Timeliness
6. Understandability

While some - if not most - of these characteristics relate to the process of accounting, the “Verifiability” item clearly has an impact on the underlying accounting systems: In order to be verifiable, there must be a clear audit trail to show the origin of the figures. To make sure users leave behind the required audit trail, some actions can’t be performed in LedgerSMB, even though it would seem to be a logical requirement to be able to do so - from the perspective of a non-accountant.

2.3 Impact of tight integration

While both the qualitative characteristics from IFRS and the checks and balances from the internal controls are pose restrictions on the accounting process, sometimes these restrictions require support from the underlying accounting software.

One example is the support for creating reliable audit trails by protecting accounting data from deletion. It’s important to realize the scope of accounting data in this respect: because invoices are being registered in the accounts receivable administration - which is summarized in the general ledger - they are part of the data for which the audit trail needs to be recorded.

¹See also https://en.wikipedia.org/wiki/Internal_control

²See also https://en.wikipedia.org/wiki/International_Financial_Reporting_Standards

Another example is separation of duties (also known as the “four eye principle”), where one accountant enters financial transactions and another is responsible for posting them. This procedure protects the company from an accountant single-handedly faking transactions and possibly masking fraud.

The requirements for good accounting processes and internal control have impact on the work flows supported by LedgerSMB. As a consequence some of the work flows described in part V may seem unwieldy; an example being the lack of functionality to delete or correct incorrect invoices (See Section 37.5 on page 193 for more details).

Chapter 3

Introduction to accounting

The purpose of an accounting system is to keep track of the company's financial status. The following reports are used to present it:

Balance Sheet

Provides a snapshot of the financial position of a company, listing its possessions (assets), debts (liabilities) and residual value (equity).

Income Statement

Also known as the "Profit & Loss Statement", summarizes the income generated over a specific period and the expenses associated with it, resulting in "Net Result".

Cashflow Statement¹

Summarizes the incoming and outgoing cash flows over a specific period, resulting in "Free Cash Flow".

Statement of Owner's Equity¹

Summarizes the changes in equity over a specific period.

Trial Balance

An intermediate report used for preparing the Balance Sheet and Income Statement, tallies transactions over the reporting period; used to assert that all accounts in the General Journal are balanced.

The accounting system is composed of these parts:

Journals

Contain the company's transactions with extensive additional data; they are the first point of entry for transactions, ordered by order of entry.

Ledgers

Contain aggregated data from the journals, ordered by date.

¹Not currently implemented in LedgerSMB

Chart of Accounts

The categories by which financial data are classified.

All accounting data ends up summarized in the **GL**. Transactions may be entered in a different (sub)ledger before ending up in the **GL**; e.g. the Sales Ledger. With the advent of computerized accounting, the use of journals is in decline: transactions are classified while directly being entered into ledgers.

In addition to the parts mentioned above the following terms are used throughout this document:

Assets

Money and anything that can be converted into money without reducing the net equity of the business. Assets include money owed, money held, goods available for sale, property, etc.

Liabilities

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

Equity

What would be left for the owner if all the assets were converted to money and all the liabilities paid off.

Revenue

Income from business activities.

Expense

Money paid to operate the business.

Cost of Goods

Money that was spent to acquire material and services to build a product being sold.

Operating Expenses

Expenses that are consumed to administer the business.

Accounts Receivable

An asset on the books as a claim for a future payment from a customer.

Accounts Payable

A liability on the books for a future payment to a supplier or vendor.

3.1 Double-entry accounting

The basic concept in double-entry accounting is that every transaction is an exchange. For example, when a business sells goods to a customer, it issues an invoice. The exchange is to provide the goods and receive the right for (future) payment. When the customer later pays the invoice, the right for payment is exchanged for cash.

Each exchange is to have the same value on both sides of the exchange, making sure no transaction is incomplete. This is where the requirement comes from

that double-entry transactions need to be balanced. It can occur that a customer's payment is a bit short of the owed amount. Should the company decide not to pursue payment, the transaction would end up being unbalanced. After all the exchange is not equal-value between the providing and receiving sides. Double-entry accounting accommodates this scenario by explicitly recording the unpaid amount as an expense.

Because every transaction is balanced, so is the balance sheet. Through this approach, double-entry accounting provides strong control over correctness of the numbers: as soon as a transaction is unbalanced (i.e. contains an error), the balance sheet becomes unbalanced – a relatively simple check.

3.2 Cash versus accrual basis

Financial statements, such as the Income Statement and Balance Sheet can be prepared using either a cash or accrual basis.

In **cash basis** accounting, the income is deemed earned when the business physically receives the customer payment, and the expenses are deemed incurred when the business physically pays them. Cash basis accounting does not require the use of purchase orders, invoices, or long term liabilities.

In **accrual basis** accounting, income is considered earned when a valid asset is received for services or product provided. The asset is the claim on the customer, by way of the invoice, to collect an amount at a later date. This asset is called Accounts Receivable. An expense is considered earned when a liability is created, by way of a purchase order, to the supplier or vendor. The liability is the commitment to pay the supplier or vendor at a later date. This liability is called Accounts Payable. Accrual basis accounting requires the use of invoices and purchase orders.

There can be a number of problems with **cash basis** accounting:

- No visibility in the accounting system regarding your cash commitments leading you to think you have more or less money to spend than you actually have.
- No visibility in the accounting system about unpaid customer debts.
- Your taxing jurisdiction may limit businesses that can use cash basis accounting. For example in the United States, if you sell products or services on credit, have gross receipts higher than allowed, or need inventory to account for income.
- Because cash basis accounting isn't part of any accounting standards, there are varying expectations of what a cash balance or income statement looks like.

3.3 Valuation of inventory

@@@ TODO Need why is this important paragraph?

The cost of inventory, in theory, includes all costs incurred to acquire the goods and make them ready for sale. This theoretical cost may include shipping costs, discounts, insurance, receiving costs, handling costs, storage costs, etc.

In practice, the cost used is often limited to the total invoice price for the goods. This formula may or may not include shipping and handling. Other costs are often ignored if they are immaterial to the overall cost of the inventory, if there is no easy way to allocate the costs to the inventory, or they are relatively constant period to period.

Inventory valuation must consider in the following situations:

- When the valuation of inventory varies from purchase to purchase. The costing method determines the valuation. This situation is automatically handled in LedgerSMB using one of the inventory valuation methods discussed below.
- When the valuation of inventory is less than what was paid for it. The inventory must be written down. This is usually a manual calculation that is the result of damage, obsolescence, or decline in the vendor or suppliers selling price.
- When estimates are required, such as when inventory is stolen, destroyed, or when a physical inventory cannot be performed. In this case, a reasonable and consistent manual method must be used to estimate the value.

There are several general automated inventory valuation methods including the following:

- Specific Identification – Each specific inventory item has its cost tracked individually. This method is usually used for large and expensive items that can be tracked by serial number or identification tag. Typically, the specific identification inventory valuation method results in an inventory valuation close to value of using the **First in, first out (FIFO)** method.
- **FIFO** – This method is similar to selling the oldest product first merchandising policy. In the **FIFO** method the latest costs are included in inventory cost and the older costs are charged back to **COGS**.

FIFO is the preferred method for maintaining accurate historical costs and is less susceptible to income manipulation by the timing of new purchases. Typically **FIFO** results in the highest inventory value of these methods.

In LedgerSMB the inventory cost is tracked on a **FIFO** basis. When a part is purchased, its cost is added to the inventory asset account. When the part is sold, the cost of the item is moved to the cost of goods sold account.

@@@TODO Is this paragraph redundant?

@@@TODO What sold cost is used; oldest, newest, average?

@@@TODO How is the inventory asset account debited?

- **Last in, first out (LIFO)** – Under **LIFO** the costs of the last goods purchased are charged against revenues as the **COGS** and the inventory is composed of the costs of the oldest goods acquired.

LIFO is preferred when prices are rising as it typically results in reducing net income and thereby reducing taxes. However, **LIFO** allows manipulation of income by simply changing the time at which additional purchases are made and does not represent accurate historical costs. Typically **LIFO** results in the lowest inventory value.

- **Weighted Average** – Under the weighted average method, the total number of units purchased plus those on-hand at the beginning of the period is divided by the total costs of purchases plus the cost of the beginning inventory.

Typically the weighted average inventory valuation method results in an inventory value between **FIFO**, which is the highest, and **LIFO** which is the lowest.

Part II

Getting started

Chapter 4

Overview

This part of the book will run the reader through the LedgerSMB using an example startup company run by Jack: Example Inc, which starts its life as a computer parts store for the business-to-business market.

4.1 Introduction

Jack just completed incorporation of Example Inc and is ready to start doing business. Before starting his operation Jack was looking for tooling to run his operation efficiently. To that extent he'll be running LedgerSMB using the domain he acquired for his business: <https://example.com/>.

The other chapters in this part of the book show you what steps Jack has to go through to get LedgerSMB up and running for Example Inc., as well as the steps he has to take to keep LedgerSMB in good health.

Due to its success Example Inc. will grow, posing new challenges to LedgerSMB and we'll show you how Jack can change the configuration to adapt to his growing business's needs.

Jack chooses to use a hosted LedgerSMB, so he doesn't need to concern himself with the technical details of getting the application up and running. Instead he can start by setting up the company database immediately.

In Chapter 5 starting at page 45 and Chapter 6 starting at page 53 Jack goes through the steps of setting up a basic company. The chapters after that may not apply to every business. Chapter 7 starting at page 67, Chapter 8 starting at page 77 and Chapter 9 starting at page 83 apply to businesses dealing with physical goods: buying, selling and shipping. Chapter 37 starting at page 191 discusses how to handle invoicing from LedgerSMB. Chapter 11 starting at page 87, Chapter 12 starting at page 89 and Chapter 13 starting at page 91 discuss how to manage accounts receivable and payable including arrears monitoring.

Not all chapters may be relevant to the reader, e.g. when he or she is starting up

or running a services company in which case the chapter “Building up stock” doesn’t apply. Chapters can be skipped based on relevance both to the type of business and its growth phase.

@@@ more chapters??

Chapter 5

Creating a company database

5.1 Using setup.pl

LedgerSMB comes with a tool called `setup.pl`. It's the beginning of a web-based ¹ database administration interface to LedgerSMB. This tool can be used to create company databases as well as backups of existing ones.

Please note that while executing the steps described in this section, it may take a while for the next screen to appear after clicking each button: Some buttons involve a large amount of server processing before the next screen can be presented.

5.1.1 Step 1: setup.pl login

Jack installed LedgerSMB using the default installation instructions, which means the url for his `setup.pl` is <https://example.com/setup.pl>. Figure 5.1 on page 46 shows the login screen for the tool.

Please note that in version 1.7 and earlier you can't be logged into the administration tool (`setup.pl`) and the webapp through the same browser at the same time due to limitations of the implementation.

The login screen shows three fields: (a) a user, (b) a password and (c) a database name.

The user name you use with `setup.pl` needs to be a PostgreSQL user. The user needs at least the right to create databases and other users; the `postgres` super user has these rights and more, so it's generally convenient (but not very secure) to use it.

¹There's also a command line based setup tool called `ledgersmb-admin`; it's covered as part of the section '[Administration](#)'

You might also see the user name `lsmb_dbadmin`. This user is hard-coded as an alternate to `postgres` because `setup.pl` can't connect to the database before it receives a valid user name and password. To use this user name it has to be setup during the installation process. This is not done by default, but is mentioned in some installation instructions as a more secure alternative and should be used if available. Also see Section 24.6 on page 147.

The password must be the same as that used to log in from the command line using the `psql` tool or the password you assigned on user creation using the `createuser` tool². Both `psql` and `createuser` are PostgreSQL tools (not LedgerSMB tools).

The third field is the name of the company database to be created. For practical reasons it's best to use only lower case letters, numbers and the underscore (`_`).

The "Login" button is used when the company database already exists. The user can then perform some administrative tasks on the existing database.

The "Create" button is used to create a new company.

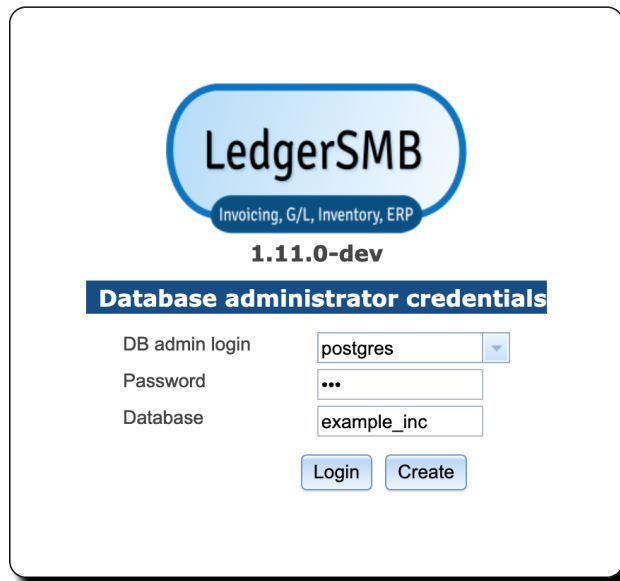


Figure 5.1: setup.pl login screen

5.1.2 Step 2: Company creation

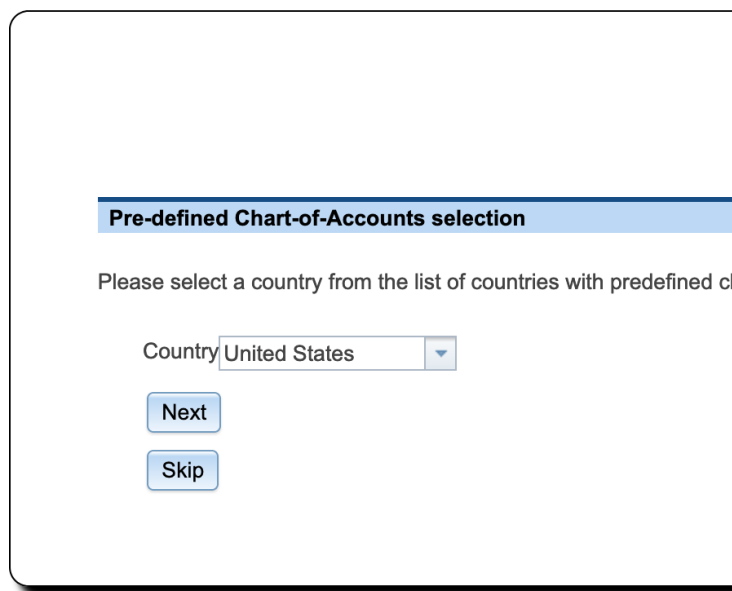
When creating a company database, there are a few things that are of importance:

²Most PostgreSQL deployments need configuration changes for this to work, if you're installing LedgerSMB yourself. See <https://ledgersmb.org> for installation instructions.

- The name of the company database will be used at login time and hence will be used by all users - a choice of recognizable value is important
- The value entered (and hence the company name) is case-sensitive
- The name can't be more than 63 characters long

After choosing "example_inc" as his company name, Jack clicks "Create" to create the database and load it with LedgerSMB's database schema, authorization rules and stored procedures³. It may take a while (30 seconds or more) for the next screen to appear⁴.

After the delay the screen from Figure 5.2 on page 47 shows up.



Pre-defined Chart-of-Accounts selection

Please select a country from the list of countries with predefined ch

Country

Figure 5.2: Chart of accounts - country selection

5.1.3 Step 3: Selection of a Chart of Accounts

LedgerSMB comes with numerous predefined charts of accounts. These have been grouped per country making the selection of a chart a two-step procedure. `setup.pl` allows for users wanting to define their own charts by offering a "Skip" button. This button skips the process of loading a chart.

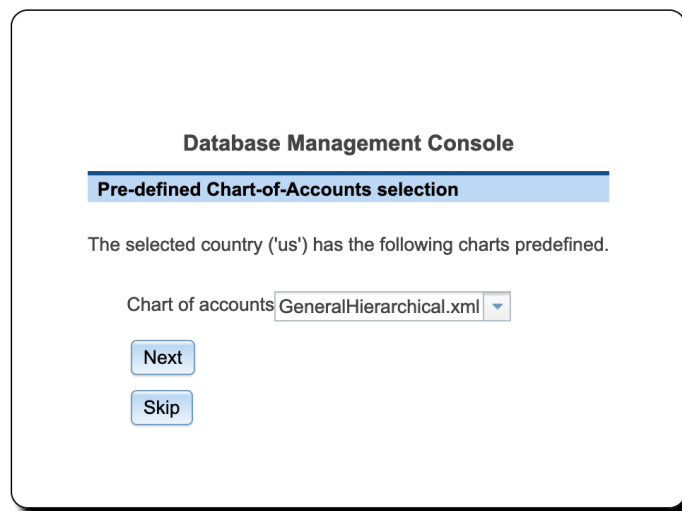
³Parts of the program inside the database.

⁴Note that during the creation of the database, logs are kept so that in case of errors these can be reviewed either by the person running the installation or by support personnel. On Linux/Unix systems these are stored, by default, in the `/tmp/ledgersmb/` directory and named `"dblog"`, `"dblog-stderr"` and `"dblog-stdout"`.

Note that you need to define a chart of accounts before you can meaningfully do anything inside LedgerSMB. If you don't load a pre-defined one you'll need to create or upload your own from inside the application once setup has completed.

Figure 5.3 on page 48 shows the first screen in the CoA selection procedure. Here you select the country for which you want to use the CoA. Note that charts of accounts are highly country dependent and you may want to consult an accountant if no default chart of accounts is included for your country.

As Jack runs his company in the United States, that's what he selects.



The screenshot shows a web interface titled "Database Management Console". Below the title is a blue header bar with the text "Pre-defined Chart-of-Accounts selection". The main content area states: "The selected country ('us') has the following charts predefined." Below this text is a label "Chart of accounts" followed by a dropdown menu showing "GeneralHierarchical.xml". At the bottom of the form are two buttons: "Next" and "Skip".

Figure 5.3: Chart of accounts - selection

Figure 5.4 on page 49 shows the second screen in the chart of accounts selection procedure. The drop down contains a list of all charts of accounts defined for the selected country.

Jack selects the `GeneralHierarchical.xml` chart of accounts: that will leave him enough room to specialize the setup later if he has to, but for the time being offers a broadly usable setup.

5.1.4 Step 4: Load template

LedgerSMB comes with a set of default reporting templates. These control the formatting of documents like invoices, checks, balance sheet, purchase orders, income statement, etc. Jack is now presented with the screen to load templates. LedgerSMB 1.13 comes with a single choice⁵:

⁵Administrators may define extra sets for users to be selected upon company creation; see ??

- demo – Example template set for various output formats including PDF, HTML and Excel

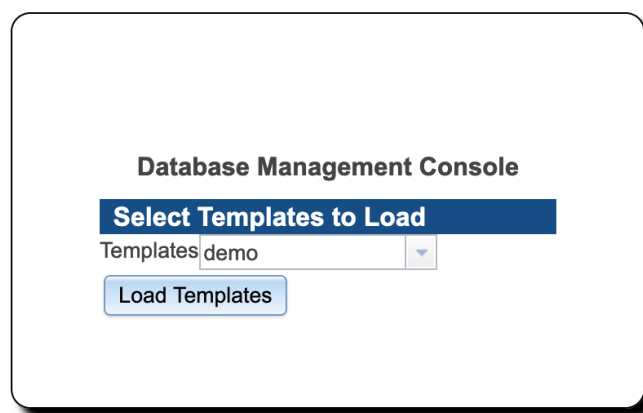


Figure 5.4: Template selection

5.1.5 Step 5: Initial user

In the previous step, the technical part of the company creation procedure was completed. However, it's not possible to log in to the company yet. Figure 5.5 on page 50 shows the next step in the setup process: the user creation screen. The fields shown have the same meaning as those discussed as part of user management in Section 24.2 on page 143.

Database Management Console

Enter User

Username

admin

Password

User creation

☒ Create new user

☐ Import existing user

Salutation

Mr.

First Name

Ad

Last name

Min

Employee Number

1

Date of Birth

1/1/2000

Tax ID/SSN

0

Country

United States

Assign Permissions

Full Permissions

Create User

Figure 5.5: setup.pl initial user creation screen

Jack chooses to create an administrative user called `admin` who will be authorized to do everything in the application. Later on he will also create a user `jack` who will be authorized to do everything but changing the configuration and doing application administration. He'll use the latter user to log in for day to day operations. This will help him prevent changing settings by accident.

Note that none of the fields in this screen are optional. If the name of the user being created isn't already used with other companies, leave the `Import` option set to `No`, otherwise please read the chapter on user creation mentioned above.

Note: The password you enter here is a temporary one which will remain in effect **for 24 hours only**. Be sure to execute the steps in Section 6.2 on page 53 before these 24 hours elapse, because the user will be disabled after that.

Jack proceeds to enter the values as follows:

Field	Value	Description
User Name	admin	The login user name
Password	asdfg	The password to use for your first login
Create New User	Checked	
Import Existing User	Unchecked	Only used when the user exists in another database

Field	Value	Description
Salutation	Mr .	
First Name	Ad	Used in combination with Last Name to identify the user
Last Name	Min	
Employee Number	1	
Date of Birth	1/1/2000	not used by the application
Tax ID/SSN	0	Tax or SSN; not used by the application
Country	United States	
Assign Permissions	Full Permissions	

Table 5.1: Create first user entry data

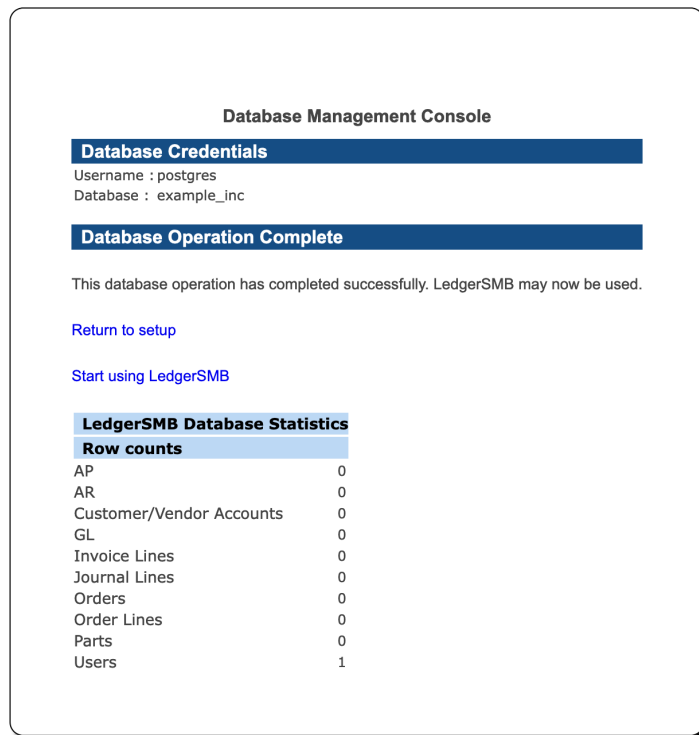


Figure 5.6: setup.pl successful completion screen

Jack has created his company database and now sees the “Database Management Console” screen as shown in Figure 5.6 on page 51.

By selection `Start using LedgerSMB` in the “Database Operation Complete” section of the screen his story continues in the next chapter “The first login”.

Chapter 6

The first login

6.1 Introduction

After the company database has been created by executing the procedure described in the last chapter it is still an empty shell which needs to be populated. The correct data needs to be entered for things like bank accounts and company contact data to be used on invoices.

These steps have to be completed before LedgerSMB can be used meaningfully: these settings have to be present for many workflows. A major reason is that with LedgerSMB - as most **Enterprise Resource Planning system (ERP system)** - financial consequences of events in many workflows are directly reflected in the company's books. Some accounting related settings have to be completed before LedgerSMB can do so.

This chapter documents the steps to create a basic usable configuration for Example Inc. It assumes that one of the default chart of accounts was selected as shown in Chapter 5 starting at page 45 or that a custom chart of accounts was imported using the instructions in Section 18.3.5 on page 115. Whatever method was chosen, this chapter assumes that a chart of accounts is available.

6.2 Steps to the first login

Jack is ready for his first login and should see the screen as shown in Figure 6.1 on page 54.

If this screen is not shown then Jack navigates to the login screen at <https://example.com/ledgersmb/login.pl>.

6.2.1 Login screen



Figure 6.1: login.pl opening screen

The login screen shows three fields which Jack proceeds to fill in as follows:

Field	Value	Description
Name	admin	The login user name
Password	asdfg	The password to use for your first login
Company	example_inc	The name of the company database

After entering all of the information and tapping the `Login` button Jack will see an alert letting him know that his password will expire today. Jack clicks OK to dismiss the alert.

6.2.2 Selecting a password

After successful login, the system shows the Welcome to LedgerSMB screen as depicted in Figure 6.2 on page 55.

The initial password has a 24-hour validity limit to prevent unused user accounts from posing a security risk.

To set a new password Jack navigates to `Preferences` → `Password` and sees the screen as depicted in Figure 6.3 on page 56.

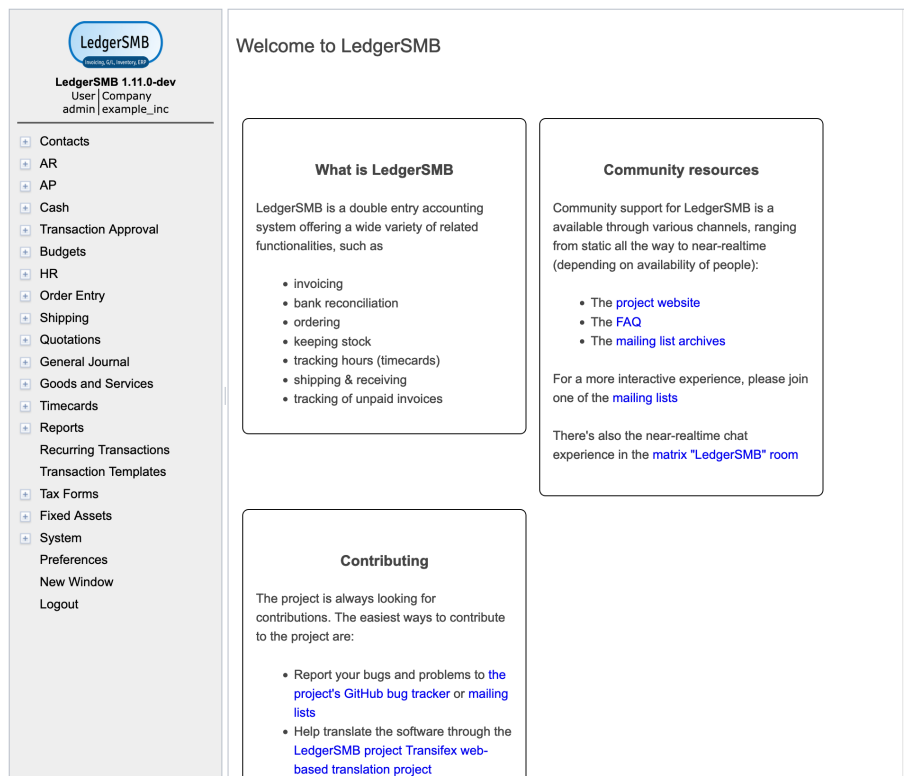


Figure 6.2: login.pl welcome screen

The new password that Jack chooses will be different than any password used before and different than the temporary password set by the administrator. Not clicking the `Save` button means the password remains unchanged and the 24-hour limit remains in effect.

Preferences Password

Change Password

Old Password
.....

New password
.....

Strength 25

Verify
.....

Change Password

Figure 6.3: Password change screen

Jack enters the information in the Change Password screen as follows:

Field	Value	Description
Old Password	asdfg	The old password set by the administrator using setup.pl
New Password	lkjhg	The new password that Jack wants to use for admin
Verify	lkjhg	Repeats the new password that Jack wants to use for admin

Table 6.2: First Login - Change initial password data

Jack then clicks the `Change Password` button.

The new password has a validity of determined by the `Password Duration` setting from the `System → Defaults` screen. User management is discussed in detail in Chapter 24 starting at page 143.

Login will be denied to users with expired passwords; they can request password resets through user admins.

6.2.3 Setting user preferences

Jack clicks on the tab `Preferences` and the system shows the `Preferences` screen as depicted in Figure 24.3 on page 146

Figure 6.4: User preferences

Jack selects his language, in this case American English and clicks Save.

6.2.4 Setting system defaults

Out of the box, LedgerSMB contains reasonable system defaults, but Jack needs to add some specific company information. In order to do so, Jack navigates to `System → Defaults` and sees the screen depicted in Figure 6.5 on page 58.

System Defaults	
Company Information	
Company Name	example_inc
Company Address	215 Example St Any City, CA
Company Phone	555 836 2255
Company Fax	
Business Number	12345
Default Email To	
Default Email CC	
Default Email BCC	
Default Email From	info@example.com
Company Sales Tax ID	
Company License Number	
Base Currency	USD
Weight Unit	lbs
Default Format	PDF
Default Paper Size	Letter
Security Settings	
Disable Back Button	Yes ☐ No ☒
Password Duration (days)	180
Session Timeout (e.g. "90 minutes")	1 day
Separate Duties	Yes ☒ No ☐
Lock Item Description	Yes ☐ No ☒
Gapless AR	Yes ☐ No ☒
Default Accounts	
Current earnings	3999 -- NET MARGIN
Inventory	1510 -- Inventory
Income	4010 -- Sales
Cost of Goods Sold	5010 -- Purchases
Foreign Exchange Gain	4450 -- Foreign Exchange Gain
Foreign Exchange Loss	5810 -- Foreign Exchange Loss

Figure 6.5: System defaults

Jack sets the following defaults:

Field	Value	Description
Company Name	Example Inc.	
Company Address	215 Example St Any City, CA	Note the use of new line character
Company Phone	555 836 2255	
Business Number	12345	e.g. Chamber of commerce number
Default Email From	info@example.com	
Password Duration	180	Days

Table 6.3: First Login - Change user defaults data

Then clicks the Save button at the bottom of the screen.

A more elaborate description of the parameters in this screen is provided in subsection 18.4.3.

Next Jack navigates to `System → Country` and sets the default country to `United States` by clicking on the radio button.

And finally Jack needs to set the default language by navigating to `System → Language` and clicking the radio button for `English (US)`.

6.3 Setting up a bank account or credit card

As part of the start up activities of his company, Jack comes to an agreement with the bank for three products:

- A current account with number “C54769”
- Deposit account with number “D54990”
- Credit card with a number ending with “.7734”

Most accounting systems - LedgerSMB included - use separate GL accounts to represent each bank account. This allows easy reconciliation of the ending balance on the bank account with the balance in the books.

Knowing this, Jack looks up the example bank account from his previously configured US chart of accounts using the `General Journal → Chart of Accounts` menu as shown in Figure 6.6 on page 60.

Jack adds the new bank accounts by doing the following:

1. Click on “1060”
2. The screen appears as shown in Figure 6.7 on page 61 appears
3. Change the Description “Checking Account” to “Checking Account C54769”

Report Name: Chart of Accounts
Company: Example Inc.

Account Number	Description	Category	Contra	Recon	Ta
1000	ALL ASSETS				
1050	CURRENT ASSETS				
1060	Checking Account	Asset		✓	
1065	Petty Cash	Asset		✓	
1200	Accounts Receivables	Asset			
1205	Allowance for doubtful accounts	Asset			
1500	INVENTORY ASSETS				
1510	Inventory	Asset			
1800	CAPITAL ASSETS				
1820	Office Furniture & Equipment	Asset			
1825	Accum. Amort. -Furn. & Equip.	Asset	✓		
1840	Vehicle	Asset			
1845	Accum. Amort. -Vehicle	Asset	✓		
2000	ALL LIABILITIES				
2050	CURRENT LIABILITIES				
2100	Accounts Payable	Liability			

Figure 6.6: Bank account setup - menu items

4. Click "Save"
5. In the same screen change the Account Number to "1061"
6. Change the Description to "Cash Deposit Account D54990"
7. Click "Save as new"
8. In the same screen change the Account Number to "1062"
9. Change the description to "Credit Card xxxx.xxxx.7734"
10. Click "Save as new"

Section 18.3.3 on page 110 discusses the options in detail - for now using the settings as configured for the sample checking account will do.

Account

Translations

Account Number

1060

Description

Checking Account

Account Type

Asset

Heading

1050--CURRENT ASSETS

Negative balance heading

Configuration

Summary account for

Options

Contra

Recon

Tax

Open items

Obsolete

Include in drop-down menus

Receivables

Income

Payment

Tax

Overpayment

Discount

Payables

Expense/Asset

Payment

Tax

Overpayment

Discount

Tracking items

Income

COGS

Tax

Returns

Non-tracking items

Income

Expense

Tax

Fixed Assets

Fixed Asset

Depreciation

Expense

Gain

Loss

Save

Save as new

Figure 6.7: Bank account setup - account setup screen

6.4 Creating initial bank balance

When a company is formed there is an initial investment. This varies by the type of company.

Jack's company was formed as a corporation. The initial purchase of the company's common shares needs to be entered into the bank account to establish the initial balance.

To do this Jack will create a manual Journal Entry which will debit the Checking Account (1060) and credit the Common Shares account (3350). Jack's initial investment was \$5000.00.

To do this Jack will:

1. Navigate to General Journal → Journal Entry
2. Enter "0001" into the Reference field
3. Enter "Initial Investment" into the Description field
4. Select account 1060 in the first row of the Account column
5. Select account 3360 in the second row of the Account column
6. Enter "5000" under in Debit column for account 1060
7. Enter "5000" under the Credit column for account 3360
8. In the Source column for each row enter the source. This could be a check number, but for our purposes we are going to enter "EFT" for **Electronic Funds Transfer (EFT)**
9. The screen should look like the screen show in Figure 6.8 on page 63
10. Click "Update"
11. The screen should look like the screen shown in Figure 6.9 on page 63
12. Click "Save" to save the transaction
13. The screen should look like the screen shown in Figure 6.10 on page 63
14. Click the "Post" button to approve the transaction
15. To confirm the initial balance navigate to General Journal → Reports → Balance Sheet
16. Observe the new \$5000.00 balance in the Current Assets section and the \$5000.00 balance in the Share Capital section of the balance sheet report as shown in Figure 6.11 on page 64

Add General Ledger Transaction

Reference
0001
Date
2026-03-30

Description
Initial Investment

Notes

FX

Status: INITIAL

Account	FX	Debit	Credit	Source	Memo
1060--Checking Account	USD	5000		EFT	
3350--Common Shares	USD		5000	EFT	
	USD				
	USD				
	USD				
	USD				
	USD				
	USD				
	USD				
	USD				
	USD				
	USD				
	USD				
		0.00	0.00		

Update

Figure 6.8: Bank account setup - initial bank balance

Add General Ledger Transaction

Reference

0001

Date

2026-03-30

Description

Initial Investment

Notes

FX

Status: INITIAL

Account	FX	Debit	Credit	Source	Memo
1060--Checking Account	USD	5000.00		EFT	
3350--Common Shares	USD		5000.00	EFT	
	USD				
		5000.00	5000.00		

Update

Save

Figure 6.9: Bank account setup - initial bank balance after update

Edit General Ledger Transaction					
Reference	0001	Date	2026-03-30		
Description	Initial Investment				
Notes					
FX					Status: SAVED
Account	FX	Debit	Credit	Source	Memo
1060--Checking Account	USD	5000.00		EFT	
3350--Common Shares			5000.00	EFT	
	USD				
		5000.00	5000.00		

Attached and Linked Files

[Attach]

Figure 6.10: Bank account setup - initial bank balance after save

Balance Sheet

Example Inc.
123 Main St. Anytown USA 55555

ALL ASSETS

CURRENT ASSETS

1060 - Checking Account 5000.00
1050 - CURRENT ASSETS 5000.00

1000 - ALL ASSETS 5000.00

TOTAL EQUITY

SHARE CAPITAL

3350 - Common Shares 5000.00
3300 - SHARE CAPITAL 5000.00

3299 - TOTAL EQUITY 5000.00

[\[PDF\]](#) [\[HTML\]](#) [\[CSV\]](#)

Figure 6.11: Bank account setup - balance sheet confirmation

6.5 Checking and adjusting the chart of accounts

First and foremost the chart of accounts serves to register income, expenses, assets and liabilities in categories which support financial decision making or regulatory requirements. When checking his chart of accounts, this is the first thing Jack checks for.

Many business events in LedgerSMB trigger the creation of financial transactions. If the configuration required for these transactions to be created isn't in place, users won't be able to complete their workflows.

6.5.1 Accounts list

Jack wants to make sure his chart of accounts fits his purposes. To perform these checks Jack goes into the `General Journal` → `Chart of Accounts` page. For now, he finds the ledger to be in order. Although the single Sales account stands out a bit against the numerous expense accounts, it turns out that there is also a single Purchases account on which all the expenses for parts purchases are going to be booked.

He decides that if this isn't enough, he can add accounts later. ¹

6.6 Checking sales tax rates

First off, Jack asserts that a sales tax² account has been provisioned. He finds it in the Current Liabilities section of his `Chart of Accounts (CoA)`. In his jurisdiction there is only one sales tax rate applicable at any one time, which means this single account will suit his needs just fine. If he had been in a jurisdiction with multiple tax rates applicable, e.g. different rates for different types of goods, he would have been required to create more accounts.

The procedure to create more sales tax accounts is the same as the one used in Section 6.3 on page 59, with the notable difference that this time the base account to be used is the sales tax account.

With the accounts in place, the tax rates have to be checked and possibly adjusted. To do so, Jack navigates to the `System` → `Taxes` page as shown in Figure 6.12 on page 66.

The rate shown (5%) is exactly what Jack needs. The procedure to set rates is a bit long to describe and hence has its own chapter. Please refer to Chapter 26 starting at page 155 for details on how to change taxes.

¹Note that Jack will discover in Section 7.4 on page 70 that he does indeed need to create additional accounts to support sales and purchase discounts.

²Sales tax may be called `Value Added Tax (VAT)` in some jurisdictions.

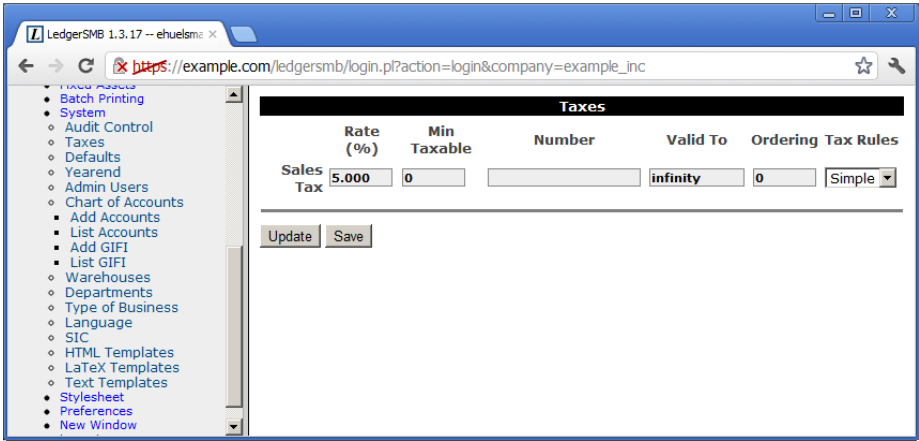


Figure 6.12: Tax rate adjustment screen

Chapter 7

Building up stock

7.1 Overview

In this chapter Jack goes through the process of setting up LedgerSMB for his trade activities in computer parts, which includes deciding which parts he wants to resell. From there on he goes to contact a vendor to request a quotation, convert that to an order and receive goods into inventory and invoices into accounts payable.

To prepare LedgerSMB for his parts sales and purchases, Jack needs to configure Parts. The system records inventory for parts and assemblies. Jack won't use them for his business. There's more on assemblies in Section 25.1.3 on page 152.

Once set up, Jack is ready to execute the ordering process. Even though the process is described here from a purchasing perspective, sales work the same way with the roles reversed (Jack will act as a vendor in the sales process).

To start his purchase, Jack creates a [Request for quotation \(RFQ\)](#) document which he sends one or multiple vendors to let them know he's interested in their products.

The vendor responds to Jack's request by issuing a Quotation. From a legal perspective a quotation is a document which promises to deliver the requested goods or services at a certain rate - subject to conditions specifically mentioned. If Jack accepts the quotation and meets the conditions, the vendor is obligated to deliver.

In response to the quotation, Jack will place an order with the vendor to indicate acceptance of the quotation (or he can let it expire). When he places the order, that legally means he agrees to the terms and conditions in the quotation. If the vendor delivers the goods or services as per the order, Jack has accepted the legal obligation to pay.

The vendor responds to the order by shipping the goods and services as well as sending an invoice. The invoice legally means the vendor considers to have a

claim on the assets of Jack’s company. Jack creates a vendor invoice in his system to record the claim on his company by the vendor and the vendor creates a sales invoice in their system to do the same.

As a result of the above it’s considered bad practice to delete or change invoices once created. The accepted process to adjust invoices is to generate a debit invoice (for purchases) or a credit invoice (for sales) to “undo” the effects of the invoice and letting the other party know about it. Then a new invoice can be generated with the appropriate content. However, when the order process is correctly followed from order to invoice chances of sending the wrong invoice are greatly reduced.

7.2 Defining parts

Jack needs to enter a large number of items he’ll be offering in his new shop. He starts out with the easy ones: the ones which will be sold as single items.

7.2.1 Single items

Jack chooses a 5TB hard drive by Samsung to be entered into the system as the first item. To do so, he goes to the Goods and Services → Add part page from the menu as shown in Figure 7.1 on page 68.

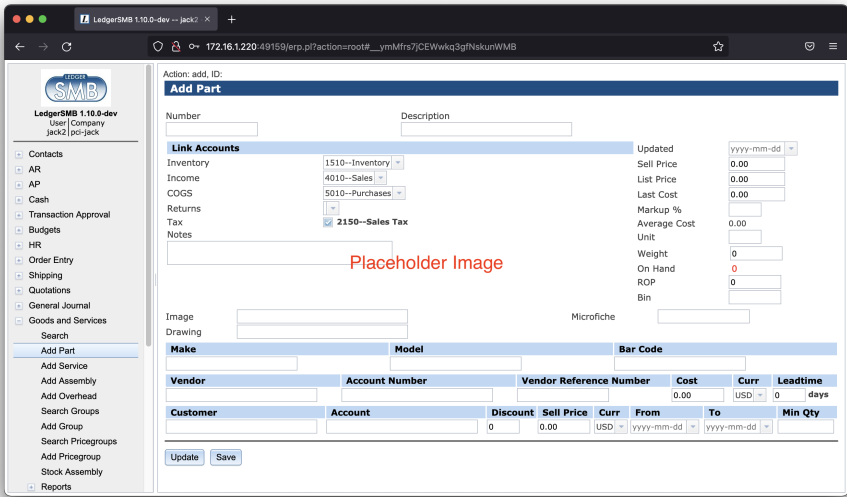


Figure 7.1: Add first part

Based on his reading from Section 25.1.1 on page 149, Jack decides to enter the hard drive with the following data:

Field	Value
Number	SAM1TB
Description	SAMSUNG 980 PRO 1TB PCIe NVMe Gen4 SSD
Inventory account	1510 - Inventory
Income account	4010 - Sales
COGS account	5010 - Purchases
Sell price	175.00

After entering the data Jack clicks *Save*.

He decides not to include make/model information, drawings or images yet and since he hasn't entered vendors or customers in his system yet, he decides to leave those sections blank as well.

7.2.2 Single-item & “multi-item pre-packaged”

After having finished setting up the solid state drive, Jack now wants to enter the memory modules he's going to sell. The problem is that they usually go in pairs, since that's what the systems consuming them need. However, he expects them to be sold as single items as well and he wants to be able to set a separate price for those occasions.

From his reading of Section 25.1.5 on page 152, it should be possible to support this scenario with a small work around¹. From the two solutions available, he chooses option (b): to create a part and an assembly and regularly restock the assembly to 0 (zero) in order to remove the stock from the single item.

7.3 Defining part groups

As Jack continues to enter more parts into the system, he wonders how he's going to look up the parts efficiently later on. Returning his reading to Section 25.1.1 on page 149, he understands that 'part groups' are the solution to that problem.

He decides to create the part groups by navigating to:

Goods and Services → Part Groups.

For each group in the list below Jack enters the group name and clicks the green check mark after each one:

Field	Value
Group	Storage
Group	Monitors
Group	Input devices
Group	Printers

¹It's planned to directly support this use-case in some version higher than 1.3

After creating these part groups, the “Group” drop down appears on the parts entry screen, allowing him to assign each of his parts to one of these groups.

Since he doesn’t expect to be running more than one or two types of CPUs, he decides not to create a separate group for those and leaves these two parts unassigned.

7.4 Defining vendors

Jack selects “ABC Parts” to purchase the inventory he needs to run his company. In order to start buying inventory, ABC Parts needs to be entered as a Vendor to LedgerSMB.

Using the work flow detailed in Section 33.4 on page 177 Jack starts to do so by going through the menu `Contacts → Add Entity → Company`. He fills out the Company creation form by clicking the `Generate control code` button and adding the data as shown in Figure 7.2 on page 70.

The screenshot shows the LedgerSMB 1.3.19 web interface. The browser address bar shows the URL: `example.com/smb-1.3/login.pl?action=login&company=example_inc`. The page title is "LedgerSMB 1.3.19 -- ehuelm". The main content area is titled "Company" and contains the "Add Vendor" form. The form has the following fields and values:

- Control Code: `A-00005`
- Class: `Vendor` (dropdown menu)
- Name: `ABC Parts`
- Country: `United States` (dropdown menu)
- Tax Number/SSN: (empty field)
- SIC: (empty field)

There are three buttons: `Generate Control Code`, `Retrieve`, and `Save`. The `Save` button is highlighted. On the left side, there is a sidebar menu titled "Logged into example_inc" with the following items:

- AR
- Point of Sale
- AP
- Add Transaction
- AP Voucher
- Vendor Invoice
- Debit Note
- Debit Invoice
- Reports
- Vendors
 - Add Vendor
 - Reports
- Cash
- Transaction Approval
- HR
- Order Entry
- Shipping
- Quotations
- General Journal
- Goods and Services
- Projects
- Reports
- Recurring Transactions
- Tax Forms
- Fixed Assets
- Batch Printing
- System
- Stylesheet
- Preferences
- Logout

Figure 7.2: Company entry screen

After saving the company data, Jack is presented the account data screen which he fills out as shown in Figure 7.3 on page 71.

When he’s done filling out and saving the form, he notices the empty “Discount” drop down. Reading more about account configuration check marks for AR and AP as described in Section 18.3.3 on page 112 and going back to the checks on his chart of accounts (Section 6.5 on page 65), he finds he’s missing the purchase and sales discount accounts. He adds two accounts as follows:

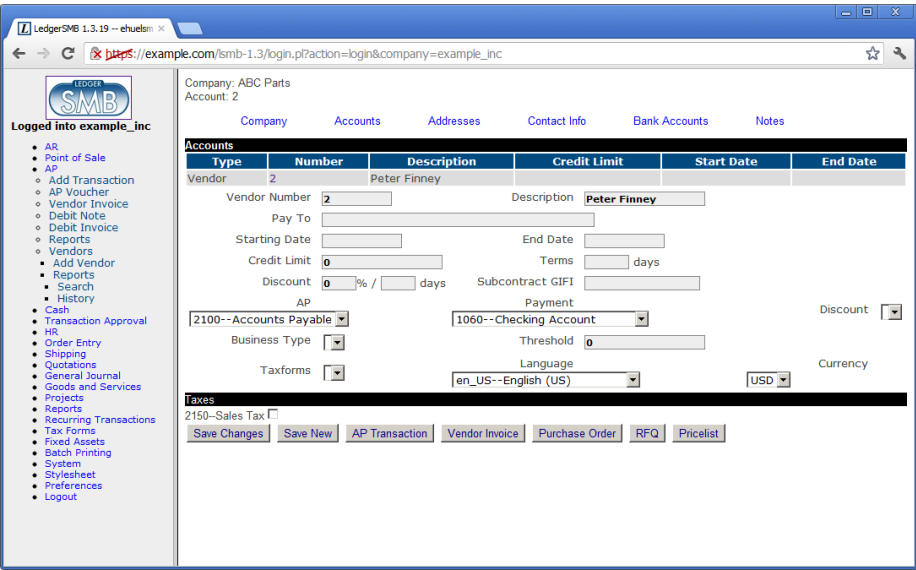


Figure 7.3: Vendor account screen

4020 Sales discount

5020 Purchase discount

After which the page looks like Figure 7.4 on page 72.

Note the top-left corner stating “Company: ABC Parts” and “Account: 2”. The information entered on the “Addresses”, “Contact Info” and “Bank Accounts” tabs will be attached to the account listed, i.e. account number 2 in this case.

The reader is referred to Section 33.4 on page 177 for a more in-depth description of the vendor data screens.

7.5 Requesting quotations

After Jack finishes setting up the parts and vendor information, he decides to use LedgerSMB to draw up a list of items he wants to order from this company. To do so he follows the menu path *Quotations* → *RFQ* which opens up a screen (shown in Figure 34.1 on page 181) for entering a new *RFQ*.

Remark Note that the *RFQ* entry screen contains prices; this is misleading at least: the printed output to be sent to the vendor does not. The fact that this screen allows entry of prices could be considered a bug.

After filling out the form in accordance with the description in Section 34.1 on page 181, Jack expedites his *RFQ* to his vendor through e-mail by clicking the “E-mail” button. He finds himself in the screen shown in Figure 7.6 on page 74.

Figure 7.4: Vendor account screen - with purchase discount account

The From field of the e-mail to be sent out will be filled using the “Default From” setting documented in Section 18.4.3 on page 117. The other address fields can be entered by the user and may be readily populated if the customer account has the right contact info items attached: if there are Email, Cc and/or Bcc contact items set up, those will be used to fill these fields.

At the bottom there are three selection lists. The first allows selection of the format used to send the RFQ. Available options are HTML, CSV, Postscript (PS) and PDF. The last two require Postscript and PDF support to be correctly set up. The last selection list selects the language to be used for the RFQ. If no value is selected the system default language is used.

For more detail, the reader is referred to Section 34.2.2 on page 183.

7.6 Following up on a quotation

Jack’s vendor (ABC Parts) sends him a quotation in response to his RFQ. Jack and his vendor can go back and forth a few times until Jack likes the offer he’s getting, but for the sake of argument let’s assume this is the final quotation.

Since Jack likes the offer, he wants to place an order with his vendor. To do so he looks up the RFQ he sent to the vendor using the menu path Quotations → Reports → Quotation. The screen shows additional buttons now that it shows a saved RFQ.

Jack clicks the “Purchase Order” button which creates a new order from the data in the RFQ. He completes it by entering the prices his vendor has quoted and by modifying it to be in accordance with the quotation. See Section 35.2

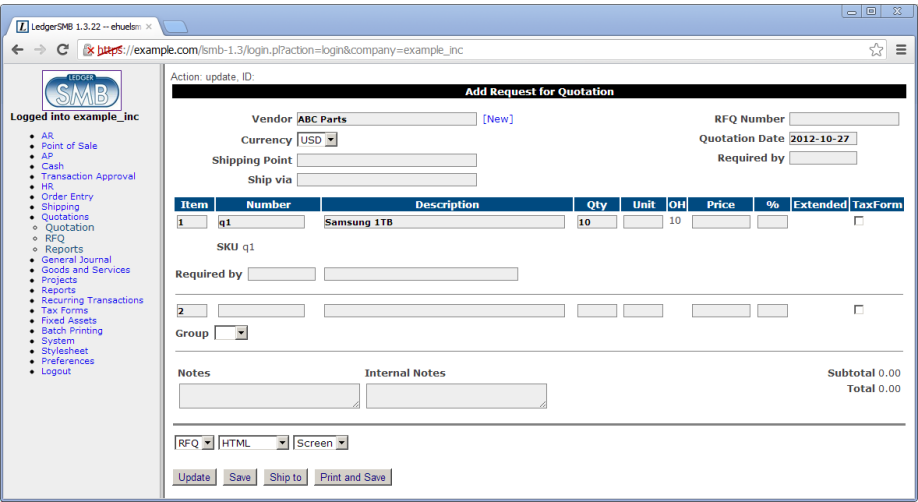


Figure 7.5: RFQ entry screen

on page 185 for more detail on the order entry screen. When finished he saves the order and mails it to ABC Parts just like he mailed the RFQ in the previous section.

Figure 36.1 on page 188 shows the screen of a saved purchase order.

7.7 Receiving ordered items

Having ordered his inventory, the vendor starts shipping. There’s too much to ship at once so the vendor ships the goods in batches: every week he ships what’s available at the end of that week - he needed to order some of the products with the manufacturer.

LedgerSMB helps Jack keep track to see if he has received everything he has ordered and that he’s not receiving too much. Jack goes through the menus Shipping → Receive. In the search screen, he fills the vendor name (ABC Parts) and clicks “Continue” to be listed all open orders from ABC Parts. By clicking on the order number, the “Receive Merchandise” screen opens as presented in Figure 7.8 on page 75. This allows Jack to handle the incoming shipment. LedgerSMB will automatically update inventory based on the amounts entered as received ².

²To resolve problems in the inventory tracking parts of LedgerSMB (inherited from before the fork), a significant change has been implemented in 1.3.31: inventory changes won’t be recorded until invoices have been posted.

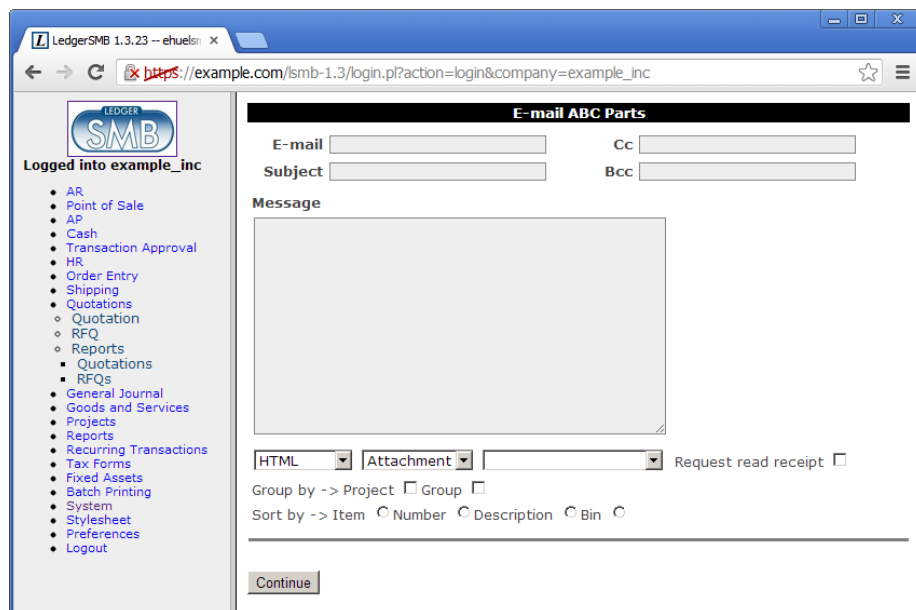


Figure 7.6: RFQ e-mail screen

7.8 Receiving an invoice

7.9 Paying an invoice

LedgerSMB 1.3.22 -- ehuelm

https://hix.nu/smb-1.3/login.pl?action=login&company=example_inc

Logged into example_inc

- AR
- Point of Sale
- AP
- Cash
- Transaction Approval
- HR
- Order Entry
- Shipping
- Quotations
 - Quotation
 - RFQ
 - Reports
 - Quotations
 - RFQs
- General Journal
- Goods and Services
- Projects
- Reports
- Recurring Transactions
- Tax Forms
- Fixed Assets
- Batch Printing
- System
- Stylesheet
- Preferences
- Logout

Action: edit, ID: 8

Edit Request for Quotation

Vendor **ABC Parts** [New] Open Closed

Currency **USD** RFQ Number **3**

Shipping Point Quotation Date **2012-10-27**

Ship via Required by

Item	Number	Description	Qty	Unit	OH	Price	%	Extended	TaxForm
1	q1	Samsung 1TB	10		9				☐
SKU q1									
Required by									
2									☐
Group									
Notes Internal Notes									Subtotal 0.00
									Total 0.00

RFQ HTML Screen

Update Print Save Ship to E-mail Print and Save Save as new Print and Save as new Vendor Invoice Purchase Order Delete

File name	File type	Attached at	Attached by
File name	File type	Attached to type	Attached to
File name	File type	Attached to type	Attached to

[Attach]

Figure 7.7: RFQ entry screen

LedgerSMB 1.3.23 -- ehuelm

view-source:https://example.com/smb-1.3/login.pl?action=login&company=example_inc

https://example.com/smb-1.3/login.pl?action=login&company=example_inc

Logged into example_inc

- AR
- Point of Sale
- AP
- Cash
- Transaction Approval
- HR
- Order Entry
- Shipping
- Ship
- Receive
- Transfer
- Quotations
- General Journal
- Goods and Services
- Projects
- Reports
- Recurring Transactions
- Tax Forms
- Fixed Assets
- Batch Printing
- System
- Stylesheet
- Preferences
- Logout

Receive Merchandise

Vendor **ABC Parts** Contact

Shipping Point Order Number **4**

Ship via Order Date **2012-10-27**

Warehouse PO Number

Date Received **2012-11-21**

Number	SKU	Description	Qty	Recd	Unit	Bin	Serial No.
q1	q1	Samsung 1TB	2	1			

Bin List HTML Screen

Update Print Done

Figure 7.8: Order receipt entry screen

Chapter 8

Ramping up to the first sale

8.1 Creating a customer

A customer is considered type of **entity** or **counterparty**¹. There are two types of **counterparties**, "company" and "person". A "company" entity is legalized by the state and typically incorporated, a sole proprietorship, or an LLC. A "person" is considered an unincorporated entity.

The first step in creating a customer is to create the **counterparty**, then annotate the **counterparty** as a "customer".²

Jack is going to setup a company as a customer and starts by navigating to add contact screen using the menu path `Contacts → Add Entity → Company`.

¹The terms "counterparty" and "entity" mean the same thing in LSMB.

²A single counterparty can also be annotated as a vendor allowing both sales to and purchasing from a single entity.

Company

Person

Add Customer

Control Code

Class

Vendor

Name

Country

United States

Tax Number/SSN

Sales Tax ID

License Number

Generate Control Code

Retrieve

Save

Figure 8.1: Add contact screen

Referencing the workflow defined at Section 33.4 on page 177 Jack creates the company using the screen shown in Figure 8.1 on page 78.

Jack does the minimum to set up the counterparty by making these changes to the screen:

Field Name	Field Value	Notes
Name	ABC, Inc.	
Class	Customer	
Country	United States	

Table 8.1: Company data

When Jack is done entering the data he clicks the `Generate Control Code`, then clicks the `Save` button, which brings up the screen shown in Figure 8.2 on page 79.

Company

Credit Accounts

Addresses

Contact Info

Bank Accounts

Notes

Files

Accounts

Type	Number	Description	Credit Limit	Start Date	End Date
Class	Customer	Number		Description	
Pay To					
Starting Date		End Date		yyyy-mm-dd	
Threshold		Credit Limit			
Payment Terms		Discount	% / days	Account	
AR	1200--Accounts Receivables	Payment	1060--Checking Account	Currency	USD
		Language	English		

Taxes

2150--Sales Tax

Tax Included

Save New

Figure 8.2: Credit account screen

This brings up the "Credit Accounts" screen Figure 8.2 on page 79 where Jack can make this company **counterparty** a customer.

Jack does the minimum to set up the customer by making these changes to the screen:

Field Name	Field Value	Notes
Number	1	There can be multiple credit accounts associated with this counterparty . This is the first of those accounts. The number must be unique to the counterparty .
'Pay To	ABC, Inc.	If printed checks are used, then this line is the Pay To line on the check. It also shows up when picking a customer with multiple credit accounts

Table 8.2: Credit Accounts data

When complete jack clicks the Save New button.

Company

Credit Accounts

Addresses

Contact Info

Bank Accounts

Notes

Files

Accounts

Type	Number	Description	Credit Limit	Start Date	End Date
Customer 1	0				

Class

Customer

Number

1

Description

Pay To

ABC, Inc.

Starting Date

yyyy-mm-dd

End Date

yyyy-mm-dd

Threshold

0

Credit Limit

0

Payment Terms

days

Discount

0 % / days

Account

AR

1200--Accounts Receivables

Payment

1060--Checking Account

Currency

USD

Language

English

Taxes

☐ 2150--Sales Tax

☐ Tax Included

Save Changes

Save New

AR Transaction

Customer Invoice

Sales Order

Quotation

Pricelist

Profit and Loss

Period selection

Select using

☒ Dates

☐ Periods

From

yyyy-mm-dd

To

yyyy-mm-dd

Report options

Account numbers

☒

Hierarchy type

☐ Full

☒ Account category

Not set up for hierarchy reporting, please see linked instructions

Continue

Figure 8.3: Credit account 2nd screen

Now that the customer credit account is setup, Jack wants to add the customers address. He clicks the **Addresses** tab at the top of the screen and sees the screen shown in Figure 8.4 on page 80.

Company

Credit Accounts

Addresses

Contact Info

Bank Accounts

Notes

Files

Locations

This Account	Type	Address	City	State/Province	Zip/Post Code	Country
--------------	------	---------	------	----------------	---------------	---------

Attach To

Entity

Type

Billing

Address

City

State/Province

Zip/Post Code

Country

United States

Save New Location

Figure 8.4: Addresses screen

Jack has to decide where to attach his customer’s address. An address on the “entity”, which is also called a **counterparty**, is an address that applies to all accounts associated with that entity. An address on a “credit account” applies to that specific customer/vendor instance. One entity may have multiple customers, multiple locations, multiple reporting units, etc., each with their own address details

See Section 18.4.8 on page 121 for more details on reporting units.

Jack enters the address by making these changes to the screen:

Field Name	Field Value	Notes
Attached To	Entity	This company has only 1 address
Type	Billing	
Address	123 Main St	In the first address field
City	Augusta	
State/Province	Mariana Islands	
Zip/Post Code	00000	

Table 8.3: First company address data

When Jack is done entering the data he clicks the `Save New Location` button.

8.2 Sending out a quote

8.3 Sending out a sales order

Chapter 9

Shipping sales

Chapter 10

Invoicing

10.1 Handling sales taxes

10.1.1 Invoices with taxes included

10.1.2 Invoices with explicit tax amounts

10.2 Invoice Editing

Posted invoices only have 2 fields that can be edited:

1. Internal notes field
2. Sales person - The Sales Person drop-down is only shown if there are sales people. Sales People are created through Employees and checking the Sales check-mark.

After editing, click the `Save` `Info` button to save the changes.

Chapter 11

Collecting sales invoice payments

11.1 Customer payments

11.2 Customer payment mismatch

11.2.1 Pardoning and underpayment

11.2.2 Handling large mismatches

As in partial payments or largish under/over payments

11.2.3 Pardoning small mismatches

Chapter 12

Paying vendor invoices

12.1 Handling vendors who match amounts to exact invoices

12.2 Handling vendors with running balances

12.3 Handling bounced checks

12.3.1 Voiding checks

Voiding checks to undo payments of vendor invoices relating to bounced checks

Chapter 13

Monitoring arrears

13.1 Handling interest on arrears

Chapter 14

Branching out: services

14.1 Creation / assignment to different accounts

14.2 Recording service hours

14.3 Customer approval on service hours

14.4 Invoicing services

Chapter 15

Branching out II: service subscriptions

Part III

Configuration

Chapter 16

Overview

16.1 Introduction

This section of the book describes how to set up LedgerSMB and its components. Configuration is assumed to be mostly one-off and rather technical in nature. To find out which tasks might need to be performed in order to keep the application in good health the reader is referred to the 'Administration Introduction', Section [19.1](#) on page [133](#).

Chapter 17

Global configuration

17.1 Apache

Section about installing on Apache 2+ items to be discussed:

Forwarding of authentication
@@@TODO

PSGI configuration
@@@TODO

17.2 PostgreSQL

pg_hba.conf
authentication @@@TODO

security
local vs IP connections @@@TODO

17.3 LedgerSMB version numbers

LedgerSMB version numbers are in the form of Major.Minor.Patch-Optional Tag. For example, if the production release is 1.9.27, then the 1.9 development release is 1.9.28-dev.

Major
An increment in the major release number can mean significant architectural, [API](#), functional, or usage changes. It is expected that an upgrade

to a new Major version is going to be planned and tested by the user organization before upgrading.

Minor

An increment in the minor release number may indicate changes to **APIs**, interfaces, user instructions, etc. But these changes are not expected to have major impact and should require minor planning and testing. Often small impact enhancements are back-patched to previous Minor versions as patches.

Patch

Increments in the Patch number represent bug fixes, security improvements, or enhancements that do not break existing functionality, usage, or **APIs**. These changes are expected to be applied to an installation as soon as possible.

Optional Tag

Common values include dev, beta, or alpha. Typically, these are only used internally by the LedgerSMB developers.

17.4 LedgerSMB Configuration

LedgerSMB configuration using `ledgersmb.conf` is deprecated as of 1 Jan 2023. New functionality may only available when using `ledgersmb.yaml` configuration file.

For the time being there is a conversion step that converts the old '`ledgersmb.conf`' to '`ledgersmb.yaml`', but the old conf file does not support new functionality.

17.4.1 `ledgersmb.yaml`

For an example of the default, non debug `ledgersmb.yaml` see the file in the [Github Repository](#).

cookie

@@@TODO

db

@@@TODO

default_locale

@@@TODO

environment.variables

@@@TODO

extra.middleware

@@@TODO

logging

@@@TODO

login.settings

@@@TODO

mail

Email can be configured by selecting which of the three available transports to use. The default `ledgersmb.yaml` file contains examples for the first two.

`Email::Sender::Transport::Sendmail` – Emails are sent using the local server's `sendmail` binary. The configuration parameters are:

```
transport:$class - Email::Sender::Transport::Sendmail
transport:path - optionally provide a path to the directory that
                  contains the 'sendmail' binary.
```

`LedgerSMB::Mailer::TransportSMTP` - Emails are sent using a remote SMTP server. The configuration parameters are:

```
transport:$class - LedgerSMB::Mailer::TransportSMTP
transport:host - The required host name of the smtp server.
transport:port - The required port number of the smtp server.
                  Note this might vary depending on whether TLS or SSL is used.
saslname:$class - The required smtp server authentication
                  method. The values can be 'Authen::SASL' or 'Authen::SASL::SCRAM'.
saslname:mechanism - The available mechanism are defined at
https://metacpan.org/dist/Authen-SASL or
https://metacpan.org/dist/Authen-SASL-SCRAM depending
                  on the selected '$class'.
saslname:callback:user - The required SMTP user name.
                  'the-user' in the default file is a place holder and must be replaced.
```

`sasl_username:callback:pass` – The required SMTP password. ‘SECURITY-FIRST’ in the default file is a place holder and must be replaced.

`Email::Sender::Transport::DevNull` - Emails are sent to `/dev/null`, in other words emails are not sent anywhere. This prevents errors in the user interface, but throws away any mail. There is only one configuration parameter. This is not a recommended production configuration. It is usually used for testing.

`transport:$class` – `Email::Sender::Transport::DevNull`

miscellaneous

@@@TODO

output_formatter

@@@TODO I really don’t have a good way to format lists explanations. See below at `paths:config:workflows` for an example. Guidance welcome.

paths

This section configures the various paths used by LedgerSMB. The configuration parameters are:

`config:locale` – Path to locale files. Defaults to `locale/po`.

`config:sql` – Path to the SQL schema definition files.

`config:sql_data` – Path to the reference and initial SQL database load files. For example, the names of the countries.

`config:templates` – Path to the templates base directory. Typically set to `templates`.

`config:UI` – Path to the UI HTML files. Defaults to `./UI/`

`config:UI_cache` – Path to the location of the UI template cache. These are cached after they have been parsed and translated. This improves performance.

`config:workflows` – A list of the Directories where workflow files are stored. Contains the default and custom workflows. Custom workflows are used to override behavior of the default workflows by providing actions, conditions, etc. with the same name and type or by providing workflows of the same type with additional states and actions. Default workflows defaults to `workflows`. Custom workflows defaults to `custom_workflows`. Default workflow path must precede all the custom workflow paths. @@@TODO is the last statement correct? Can there be more than 2 paths?

printers

This section contains a list of printers and their definition.

The default `ledgersmb.yaml` file shows two printers named 'Laser' and 'Epson' with the printers defined using the linux `lpr` command and its arguments.

This default definition will provide for the selection of the printers named 'Laser' and 'Epson' in the LedgerSMB user interface.

For more information search the internet for 'linux `lpr` command' or use `man lpr` at the linux command line.

reconciliation_importer

@@@TODO

setup_settings

@@@TODO

ui

@@@TODO

workflows

@@@TODO

Chapter 18

Per company configuration

18.1 Matching your business processes

By default, LedgerSMB operates such that all optional functionality is available and the user decides to use it or not based on which menu items they select, what fields they enter data into, and what buttons they click.

Removing application roles (see Appendix E starting at page 257) can limit the visibility of menu items, data fields, and buttons. This will simplify the users view of the system, reduce training, and better configure LedgerSMB to your business requirements.

Outside of application roles, the only other enforced business process configuration is whether the same person can both create and post transactions. It is called 'separation of duties' and is defined in Section 18.4.3 on page 117.

LedgerSMB is configured to adjust inventory when a Sales or Purchase Invoice is posted. This closely matches retail business processes where the customer walks out of the establishment with the product and an invoice (or receipt). This functionality can also be used for wholesale shipping applications because LedgerSMB can also produce picking and shipping documents, just remember that the inventory transaction happens when an invoice is posted.

LedgerSMB uses FIFO for COGS and inventory calculations. There are provisions for alternatives, but the code is not yet complete. See Section 3.3 on page 38 for a detailed explanation of FIFO calculations.

In addition to the above, LedgerSMB has configurable Workflows . These are not user configurable but your technical support staff should be able to create and edit them. See Chapter VIII starting at page 239.

Contact the development team for other business process customizations. In most cases the need for more flexible matching of LedgerSMB to your business processes is understood, but the project has not yet had any customer requests or development volunteers.

18.2 Administrative user

@@@ TODO Need to add content explaining the administrative user.

18.3 Chart of Accounts

The Chart of Accounts or **CoA** defines how financial reports are organized and summarized. This section describes the various configuration options available when creating or editing the **CoA**.

18.3.1 Predefined Chart of Accounts

LedgerSMB provides predefined Chart of Accounts that can be used out of the box for many businesses. Most new LedgerSMB setups start with one of the predefined Charts of Accounts and add or delete as necessary to customize it for their business.

There is also an option to import a Chart of Accounts. Importing is described in Section 18.3.5 on page 115.

The predefined **CoA** files are **Extensible Markup Language (XML)** format files.

LedgerSMB provides an **XML Schema Definition (XSD)** file, which might help in developing a custom CoA XML file, located in the [Github Repository doc/company-setup/configuration.xsd](#).

To use a custom **CoA XML** file, it will need to be placed in the correct directory prior to running `setup.pl`. The correct directory will be one of the locale sub-directories in `locale/coa`.

The predefined Chart of Accounts are organized by country and localized. For the United States the following predefined Chart of Accounts are available:

- General.xml
- GeneralHierarchical.xml
- Manufacturing.xml
- Service.xml
- UCOA-Form990.xml
- UCOA-Form990EZ.xml

Similar predefined Charts of Accounts are available for other countries and languages.

See Section 5.1.3 on page 47 for instructions on using these predefined accounts.

The following examples use the `GeneralHierarchical.xml` account structure. This file can be found at [GitHub Repository locale/coa/us/GeneralHierarchical.xml](#).

18.3.2 Account structure

The system allows ordering accounts into groups by assigning accounts to headers. Headers can themselves be assigned to other headers resulting in hierarchies of account groups.¹

There are multiple reasons for wanting to use hierarchies. One of those reasons is that expense accounts in prior versions were listed as one long list under Expenses while income accounts are listed as a -usually shorter- list under Income. Personnel expenses are usually spread across multiple expense accounts. With lots of other expense accounts also listed, what’s the total expense on personnel?

Another use-case for using hierarchies might be because you’re running a “margin business”. That is, you’re only buying what you’re selling and the cost of goods sold is the largest part of your expenses. In cases like this it can be a good idea to use a different setup of the income statement: one where cost of goods sold is immediately subtracted from income, leaving gross margin to cover all other expenses in the company.

In order to set a heading’s parent, go to the General Journal → Chart of Accounts. Click on the account number. If the account is a heading account that has a parent heading, the resulting screen shown in Figure 18.1 on page 109.

Heading		Translations	
Account Number	2050		
Description	CURRENT LIABILITIES		
Heading	2000--ALL LIABILITIES		
Save Save as new			

Figure 18.1: Configure Account Hierarchy

If the account is a not a heading account, but has a parent heading, then the result is shown in Figure 18.2 on page 110.

¹Although the database structure supports this type of account hierarchy from 1.3, it was not fully in use in reports until 1.4.17. In 1.3 accounts can be assigned a header, but headers can’t be assigned to headers themselves.

Account Setup Form:

- Account Information:**
 - Account Number: 2100
 - Description: Accounts Payable
 - Account Type: Liability
 - Heading: 2050-CURRENT LIABILITIES
 - Negative balance heading: []
- Configuration:**
 - Summary account for: AP
 - Options: Contra [], Recon [], Tax [], Obsolete []
- Include in drop-down menus:**
 - Receivables: Income [], Payment [], Tax [], Overpayment [], Discount []
 - Payables: Expense/Asset [], Payment [], Tax [], Overpayment [], Discount []
 - Tracking Items: Income [], COGS [], Tax [], Returns []
 - Non-tracking Items: Income [], Expense [], Tax [], Gain [], Loss []
 - Fixed Assets: Fixed Asset [], Depreciation [], Expense [], Gain [], Loss []
- Buttons:** Save, Save as new

Figure 18.2: Account Setup

If the "Current Earnings" account has been set in *System* → *Defaults*, see Figure 18.4 on page 114, it is possible to select either the full or non hierarchical (labeled "Account category") account layouts when generating the "Balance Sheet" or "Income Statement" reports.

For example, *General Journal* → *Reports* → *Income Statement* as shown in Figure 18.3 on page 110.

Report options:

- Account numbers: ☒
- Hierarchy type:
 - ☒ Full
 - ☐ Account category
- Language: English
- Continue**

Figure 18.3: Report Using Hierarchy

18.3.3 Account setup

Headers don't have any configuration, other than their number and description.

Accounts also have a number and description, but require additional configuration for the application to work correctly.

The settings, which are shown in Figure 18.2 on page 110 are described in the following paragraphs.

The main setup consists of the following:

Account Number

Income statements and balance sheets reports are ordered by account number. Traditionally, account numbers are 4 digits with accounts being grouped by the first digit, but this type of grouping is not enforced by LedgerSMB. Grouping is handled using headings as explained in Section 18.3.2 on page 109.

Description

Should be a relative short, easy to understand description of what the account is used for.

Account Type

Asset, Liability, Equity, Equity temporary, Income or Expense.

Heading

Optional entry. Used if this account is a child of a heading account.

Negative balance heading

Balance sheet account balances should - generally - not be reported as negative amounts²: a negative asset is in fact a liability. This functionality reports the given account under the primary header as long as the asset/liability is positive and reports it under the "negative balance heading" as soon as the asset/liability turns negative (the expectation is that the alternative header is a liability/asset header instead of an asset/liability header).

Account configuration

The items in the configuration box as shown in Figure 18.2 on page 110 are used to configure the account for use in LedgerSMB.

There are currently three types of optional summary accounts:

AR Marking an account as a summary account for AR means that all outstanding receivable amounts will be posted to this account. The Accounts Receivable administration will contain the details of which amount is owed by which customer.

AP Same as the AR account, except for amounts owed to vendors.

Inventory

This account holds the monetary value equal to the items on stock.

Each account may have zero, one or more of the following options selected.

²Notable exception here are "cumulative depreciation" accounts in fixed asset accounting, but these are usually marked as "Contra" – causing them to be reported as positive numbers..

Contra

This checkmark identifies the account as a **contra** account, which means that the account is going to hold the opposite of an account it's associated with. A good example of this kind would be the depreciation account associated with a fixed asset account where the depreciation account contains the credit amount to be added to the original asset (debit) value to get the current asset value.

Recon

This checkmark identifies the account as one which needs reconciliation as described in Section 43.3 on page 214.

Tax This checkmark identifies the account as a Tax (VAT/Sales) account. Tax accounts need

to be further configured. See Chapter 26 starting at page 155 for further discussion of the subject.

Obsolete

This checkmark identifies the account as not being usable going forward. The account cannot be deleted because it may contain data.

Include in drop-down menus

In order to enhance the user interface, non-summary accounts may be added to drop down menus throughout LedgerSMB. Adding an account is done by checking the box. Each row in the UI is explained below.

Receivables checkmarks**Income (AR_amount)**

This check mark adds the account to the list of accounts in the transaction screens which are used to post income on.

Payment (AR_paid)

This check mark adds the account to the list of accounts to choose from in the Receipts (AR) and Payments (AP) screens as well as the payments lines on invoices and transactions. Additionally, it

adds the account to the part entry screen as described in Section 25.1.1 on page 149.

Tax (AR_tax)

This check mark makes the account show up as a check mark on the **customer** (AR) or vendor (AP) entry screen. See Chapter 26 starting at page 155 for further discussion.

Overpayment (AR_overpayment)

Adds the account to the receipts screen as discussed in Section 42.5 on page 208.

Discount (AR_discount)

Adds the account to the customer entry screen's selection list for accounts to post payment term discounts on.

Payables checkmarks

The payables UI works the same way as the receivables described in the previous list. The difference is that the technical names of the configuration identifiers are prefixed by AP_ instead of AR_.

Tracking Items

The items on this line relate to stocked items, i.e. those tracked for inventory: parts and assemblies.

Income (IC_sale)

Adds the account to the selection list of income accounts on the part and assembly definition screens.

COGS (IC_cogs)

Adds the account to the selection list of COGS @@@ accounts on the part, assembly and overhead definition screen.

Tax (IC_taxpart)

Adds a check mark to the part and assembly definition screen for the applicable account. See Chapter 26 starting at page 155 for more details on how taxes work in LedgerSMB.

Returns (@@@ Need help here)

@@@

@@@ TODO Labor/Overhead, when part of an assembly, are capitalized in inventory and accounted for in COGS when the resulting assembly is sold. We should add an explanation to that extent somewhere in the book and refer to it from here.

@@@ TODO Labor sold isn't "inventory tracked", so it is a service. Labor/Overhead used to manufacture assemblies is "inventory tracked" and capitalized. We should explain somewhere that Labor sold immediately is a service instead.

Non-tracking items

The items on this line relate to untracked (non stocked) items, i.e. services.

Income (IC_income)

Adds the account to the income account selection list in the service definition screen.

Expense (IC_expense)

Adds the account to the expense account selection list in the service definition screen.

Tax (IC_taxservice)

Adds a check mark to the service definition screen for the applicable account. See Chapter 26 starting at page 155 for more details on how taxes work in LedgerSMB.

Fixed assets**Fixed asset (Fixed_Asset)**

Marks the account as holding the original asset value for the fixed assets module, for some classes of fixed assets.

Depreciation (Asset_Dep)

Marks the account as holding the cumulative depreciation amount for the fixed assets module, for some classes of fixed assets.

Expense (asset_expense)

Adds the expense account to the selection list of the fixed assets accounting module. See Section 43.6 on page 216 for more details.

Gain (asset_gain)

Account to hold book value gain upon disposal of a fixed asset.

Loss (asset_loss)

Account to hold book value loss upon disposal of a fixed asset.

18.3.4 Special accounts

Special accounts are set by going to `System` → `Defaults` and scrolling down to "Default Accounts".

As you can see, in Figure 18.4 on page 114 the heading "3999 -- NET MARGIN" has been selected and is the heading which is used as the highest level for the P&L and lowest level at which income and expenses are being reported in the balance sheet.

Default Accounts	
Current earnings	3999 -- NET MARGIN ▾
Inventory	1510 -- Inventory ▾
Income	4010 -- Sales ▾
Cost of Goods Sold	5010 -- Purchases ▾
Foreign Exchange Gain	4450 -- Foreign Exchange Gain ▾
Foreign Exchange Loss	5810 -- Foreign Exchange Loss ▾

Figure 18.4: System Default Accounts

The other special accounts are used as follows:

- Current earnings - Used to summarize the P&L result as a single line on the Balance Sheet,
- Inventory - Used to summarize the inventory value on the balance sheet.
- Income - Used to summarize income on the Income Statement.
- Cost of Goods Sold - Used to summarize the cost of goods on the Income Statement.
- Foreign Exchange Gain - Accounts for the gain in value differences in foreign currencies when recording payments to invoices in foreign currencies. In case an invoice is issued at 1 EUR == 1 USD, but payments arrive when 1 EUR == 0.80 USD, the amount of USD paid is still the invoiced amount, but the EUR amount differs. The EUR difference ends up in the Foreign Exchange Gain or Foreign Exchange Loss accounts. See Section 42.8 on page 210 for more information.
- Foreign Exchange Loss - Accounts for the loss in value differences in foreign currencies when recording payments to invoices in foreign currencies. See Section 42.8 on page 210 for more information.

18.3.5 Importing chart of accounts

A custom Chart of Accounts can be imported in CSV format by going to `General Journal` → `Import Chart` and selecting the import file.

The import screen list the required fields. Field content definitions are explained above.

18.4 System menu settings

This section enumerates the “System” menu’s immediate children. In some cases the functionality is too complex and is referred to a chapter of its own.

18.4.1 Audit control

Enforce transaction reversal for all dates

This is a Yes/No value which affects the actions which can be performed on posted financial transactions.

- No means transactions can be altered or deleted, even after posting them. Note that if a transaction has been posted before the latest closing date, it can never be altered, not even when this value is in effect.

- Yes means transactions can't be altered after posting. This setting is highly preferred and considered the only correct approach to accounting as it assures visible audit trails and thereby supports fraud detection.

Close books up to

@@@ This item isn't a system setting; shouldn't it move to "Transaction approval"?? That way system settings (config) and processes are separated.

@@@ My preference is to remove the setting entirely and rely on year-end workflow. We might add an account checkpoint interface as well at some point –Chris T

It's advisable to regularly close the books after review. This prevents user error changing reviewed numbers: after closing the books, it's no longer possible to post in the closed period.

There are also performance benefits to closing the books, because LedgerSMB uses the fact that the figures are known-stable as a performance optimization when calculating account balances.

Activate audit trail

This is a Yes/No value which - when Yes - causes the system to install triggers to register user actions (creation/adjustments/reversals/etc...) executed on financial transactions.

@@@ Once activated, where can we see it the audit trail??

@@@ This setting should go. In 1.3 the audit trails are always enforced via triggers so this setting does nothing. –CT

18.4.2 Taxes

This page lists all accounts which have the "Tax" account option enabled as discussed in Section 18.3.3 on page 110.

Each account is listed at least once, but can be listed many times, if it has had different settings applied over different time periods. E.g. if one of the current VAT rates is 19%, today but it used to be 17.5% until last month, there will be 2 rows for the applicable VAT account. See Chapter 26 starting at page 155 for further discussion of how taxes work in LedgerSMB and the choices involved when being required to handle changes in tax rates.

Each row lists the following fields:

Rate (%)

The tax rate to be applied when calculating VAT to be posted on this account.

Number

Account number

Valid To

The ending date of the settings in this row. This can apply to the rate as well as the ordering or the tax rules (but usually applies to the rate).

Ordering

This has to do with cumulative taxes. For example if two taxes exist and one has an ordering of 0 and one of 1, then the second tax will be calculated on a basis that includes the first. One place where this used to be used was in Quebec, where GST was taxable under PST.

Tax rules

LedgerSMB features a flexible structure to facilitate complex tax calculations (see Section 26.4 on page 157). By default the “Simple” module is the only one installed.

18.4.3 Defaults

Business number

This is used to store an arbitrary identification number for the business. It could be used to store a business license number or anything similar.

Weight unit

The unit of measurement for weights. @@@ why don't we have a unit of measurement for distance as well??? And maybe a unit of measurement for content?

Separation of duties

Separation of duties is a method to help reduce fraud where one employee can't modify the accounting ledger without another employee's approval.

Select “Yes” if you want to activate separation of duties or “No” if you don't want to activate it.

In order for separation of duties to be enforced, user roles have to be set differently for each user. This is done by removing the `draft_post` role from the users that cannot post and making sure that the users that can post have the role enabled. See Section 24.3 on page 144 for more details about changing and setting User Roles.

Default accounts

This setting will be used to preselect an account in the listings of the three categories listed below:

- Inventory
- Income
- Expense

Foreign exchange gain and loss accounts

When working with foreign currencies, the system needs two special purpose accounts. One to post the gains onto which are caused by foreign currencies increasing in value; the other to post the losses onto which are caused by foreign currencies decreasing in value.

Default country

This setting indicates which country needs to be pre-selected in country selection lists.

Default language

The language to be used when no other language has been selected. Several parts of the application require language selection, such as customer, vendor and employee entry screens.

Templates directory

This setting indicates which set of templates - stored in the `templates/` directory - should be used. In a standard installation, the drop down lists two items:

demo

which contains templates based on $\text{\LaTeX}$, which is more commonly installed but has issues dealing with accented characters

Base currency

Select the base currency to use in your company, identified by their 3-letter codes in the drop down.

If the desired currency is not listed then navigate to `System → Currency → Edit Currencies`.

All currencies listed in the Defined Currencies view will be shown for use in LedgerSMB.

Add the new currency and delete any unwanted currencies. Then return to `System → Defaults` and select the newly added currency.

Company data (name /address)

The fields “Company Name”, “Company Address”, “Company Phone” and “Company Fax” will be used on printed/e-mailed invoices.

Password duration

This is an integer value field measuring the validity period in days for passwords set through the user’s *Preferences* screen. If this field is empty, passwords set through that method won’t expire.

The user has to log into LedgerSMB after this field is changed and prior to the expiration of the previous setting in order for the new duration to take effect.

A user will receive password expiration reminders upon logging starting a week before password expiry. When not acted upon, starting two days before expiry an hourly popup will appear requesting the user to change the password.

The application behaves this way because users with expired passwords won’t be able to log in: their password will need to be reset by a user admin.

Note that passwords set by admins for other users expire within 24 hours after setting them. This value is hard coded and can’t be overruled. This is a security measure taken to make sure as few unused accounts as possible exist: Existence of such accounts could open up security holes.

Default E-mail addresses

These addresses will be used to send e-mails from the system. Note that the “Default Email From” address should be configured in order to make sure e-mail doesn’t look like it’s coming from your webserver. The format to be used is ‘Name’ <e-mail address> where the e-mail address should be inserted between the “(” and “)”.

Max per dropdown

Some elements in the screens may present a drop down. However, drop downs are relatively unwieldy to work with when used to present a large number of values to choose from.

This configuration option sets an upper limit on the number of records to be presented as drop down. When the number is exceeded, no drop down is used. Instead, a multi-step selection procedure will be used.

Item numbering

Many items in the system have sequence numbers: invoices, parts, etc. These can be just a number (i.e. 1 or 37) or they can also be both prefixed and suffixed. For example, INV0001 for invoices and EMP001 for employees or YOU-0001TOO, in which case the next item will be YOU-0002TOO.

You can only issue every number in the sequence once, but you can issue Y21-001 and Y22-001 by changing the sequence number format at the beginning of the year.

The numbers shown in the input boxes will be used to generate the next number in the numbering sequence.

GL Reference number

The default reference number for the next GL transaction.

Sales invoice/ AR Transaction number

This number is used to generate an invoice number when none is being filled out by the user.

Sales order number

Same as Sales invoice number, except that it's used for sales orders @@@ layout issue: the label is too big to fit on the page

Vendor invoice/ AP Transaction number

Same as Sales invoice, except that the number is used for accounts payable transactions. @@@ layout issue: the label is too big to fit on the page

Sales quotation number

Same as sales order number, except that it's used for quotations.

RFQ number

Request for quotation number is like the sales quotation number, except that it is used to track which vendors have been asked for quotes.

Part number

All parts, services and assemblies are identified by a unique number. When an item is created and no number is entered by the user, a number is generated from this sequence.

Employee number

Same as the sales invoice number, used by new employee entry.

Customer number

@@@ is this the control code number? or is this meta_number?? – Meta-number (CT)

Vendor number

@@@ same question as customer number

Check prefix

The prefix to use when printing checks. There's no check sequence number. That sequence number is requested from the check printing interface, because checks can be created outside the application as well, meaning the numbers can get out of sync.

18.4.4 Year end

@@@ Rename "Yearend" in menu interface to "Year end".

@@@ IMO this section doesn't belong here, because it's a process, not config, but does it belong in this menu then? IMO it doesn't...

18.4.5 Admin users

@@@ Same as Year end; doesn't belong here...

18.4.6 Chart of accounts

@@@ Chart of accounts isn't exactly a "process", but it doesn't feel like being pure config either. At any rate it's a fact that the CoA discussion is a full chapter in and of itself - so discussion here isn't necessary anymore.

18.4.7 Warehouses

Warehouses are stocking locations. They don't have any properties (in the system) other than that they have a name. Warehouses can be added, modified and deleted from the `Goods & Services → Warehouses` menu item.

18.4.8 Reporting units

There are a number of ways to divide customer and vendor data into smaller pieces for reporting and analysis. LedgerSMB calls these divisions **reporting units**. Every reporting unit belongs to a reporting unit class. For example, in the reporting unit class called "project", there might be reporting units for "Project 1", "Project 2", etc.

Reporting units can be structured hierarchically. For example, "Project 1" could be the parent of "Assessment", "Planning", "Execution", and "Finalize". This allows tracking of project phases or sub-components.

Reporting unit classes can be created (added), modified or deleted through the `System → Reporting Units` menu item. Reporting units can be created and modified using the same menu item. Reporting units have start and end dates, but may not be deleted to ensure financial integrity.

The default install includes reporting unit classes for department, project, job, fund, customer, vendor, and lot. However, these are all marked inactive in the default install and they have no defined reporting units. The result is that reporting unit class selections do not show anywhere in the user interface by default.

The reporting unit class configuration has the following fields.

Field	Description
ID	Generated by the system
Label	The label shown on screen
Active	Determines if the drop-down appears
Ordering	The order in which multiple Reporting Unit Classes appear and should be a unique integer
AR	Shows in AR Transaction, Sales Invoice, Credit Note, Credit Invoice, Customer Return, Aging Report screens
AP	Shows in AP Transaction, Vendor Invoice, Debit Note, Debit Invoice, Aging Report screens
GL	Shows in General Ledger Transaction, Trial Balance, Add budget, Income Statement screens
Entity	Not currently available
Manufacturing	Not currently available
Fixed Assets	Add Assets screen, Is hard-coded to show "Department" reporting unit class only
Timecards	Time and Materials Entry screen, is hard-coded to show "Department" reporting unit class only for Report In

Table 18.1: Reporting Unit Class Fields

Once the reporting unit class is defined, marked active and saved, the reporting unit is defined by clicking on the "Add Unit" button. This will allow the entry of the following data.

Field	Description
Control Code	The data entry short code, must be unique for all reporting units
Parent	A previously entered reporting unit
Start Date	The optional active transaction start date
End Date	The optional active transaction end date
Description	The name of the department, job, etc.
Customer	Not used, will be removed

Table 18.2: Reporting Unit Fields

The start and end dates determine if the reporting unit is shown in the user interface drop-downs based on the transaction date. If the start date is omitted then the start date is assumed to be the far past. If the end date is omitted then the end date is assumed to be the far future.

Clicking “Save” makes the specific department, job, etc. available to LedgerSMB.

Once the Reporting Unit Classes and Reporting Units are configured they are shown in the screens listed in Figure 18.1 on page 122 based on the check-boxes.

18.4.9 Type of business

Types of business are used in sales operations where customers can be assigned a type of business. Based on the type of business assignment, quotations, sales orders and invoices will automatically apply discount rates. For each type of business you enter a description and a discount rate to be applied.

18.4.10 Languages

The language table is the table users can select languages from, both to present the UI of the application as well as the setting for customers to be used to generate documents.

This listing should correspond to the actual translations of the application being available in the program installation directory.

Languages can be added, modified or deleted through the System → Language menu item.

18.4.11 Standard Industry Code (SIC)

SIC codes feature these three fields:

Code

Heading

Description

When creating a company you can assign that it an **Standard Industrial Classification (SIC)** code, irrespective of its role (i.e. customer, vendor, lead or anything else). An example of an SI code system is the US’s NAICS³ code. Other countries have their own coding systems such as ANZSIC⁴ for Australia and New Zealand and NACE⁵ for Europe

The SIC field currently doesn’t support a specific function in the application and is there merely for informational purposes. However in the future its role could be extended to include impact on reports, taxes or other functionalities where type of industry could matter.

³<https://www.census.gov/naics/>

⁴<https://www.abs.gov.au/statistics/classifications/australian-and-new-zealand-standard-industrial-cla>

⁵https://ec.europa.eu/competition/mergers/cases/index/nace_all.html

18.4.12 Templates

Templates are available to control the output format of many LedgerSMB outputs including Balance Sheet, Sales Orders, etc.

All downloaded documents are rendered based on templates stored in the database. This is to allow different templates with different output for each company, even when run on top of the same LedgerSMB installation.

In `setup.pl` there is also a button to upload all Demo templates into the database. These templates will replace all templates in the database.

There are 3 types of templates: $\text{\LaTeX}$ (tex), HTML (html), and CSV (csv). Templates are accessed by navigating to `System` $\rightarrow$ `Templates`. You should see the view shown in Figure 18.5 on page 124.

The template you want to change is selected in the "Template" drop-down. The template format you want to change is selected in the "Format" drop-down.

There are also two buttons to use after selecting the Template and Format:

Edit allows editing and saving the database version of the template.

Upload

copies a local version of a template into the database.

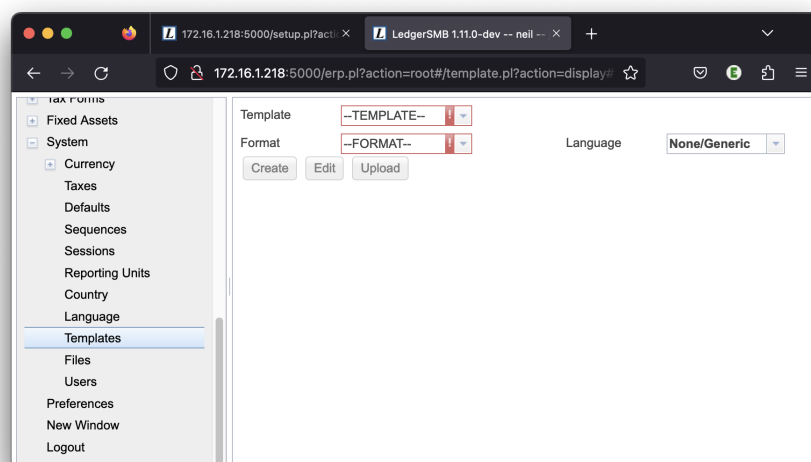


Figure 18.5: System templates screen

$\text{\LaTeX}$ templates

To change a $\text{\LaTeX}$ template navigate to `System` $\rightarrow$ `Templates`. Select, for example, Template "Invoice" and Format "tex", you should see the view shown in Figure 18.6 on page 125

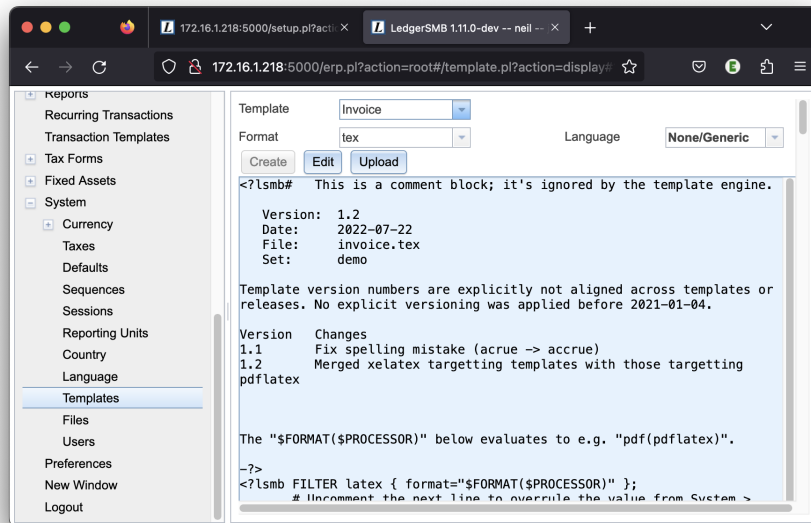


Figure 18.6: Templates edit invoice screen

To add a file to the latex template, first upload the image file to the database. This can be accomplished by navigating to *System* → *Files*.

To include this graphic file in your $\text{\LaTeX}$ document, it needs to be retrieved from the database and temporarily stored in a location accessible to the PDF generator. Once the file is in the database, then the function `dbfile_path` handles that.

For example, If the graphic file is named “`FL_Logo_icon_250x250.png`”, then add something like the following to the $\text{\LaTeX}$ template using the `Edit` button.

```
\parbox[b]{.1\textwidth}{%
  \includegraphics[scale=0.7]{%
    <?lsmb dbfile_path("FL_Logo_icon_250x250.png") ?>}
}
```

After editing the template must be uploaded to the database using the `Upload` button.

HTML templates

Editing an HTML template is exactly the same as editing a $\text{\LaTeX}$ document except that the language is HTML instead of $\text{\LaTeX}$.

CSV templates

Editing an CSV template is exactly the same as editing a $\text{\LaTeX}$ document except that the language is basically all template vars instead of $\text{\LaTeX}$.

Template Syntax

Templating directives in are expressed with the base form of `<?lsmb foo ?>`. That, and basic variable substitution, is about the only part of templating that has remained unchanged. Starting with LedgerSMB1.3, the primary rendering engine is Template Toolkit, with `START_TAG` changed to `"<?lsmb"` and `END_TAG` changed to `"?>"`.

Variable Substitution

For simple output, the name of the variable is placed on its own, as in `<?lsmb foo ?>`. Hash elements are accessed using the form `<?lsmb hash.key ?>`. Arrays are treated as a form of hash where the key is the index of the element, indexed from zero. In more complex cases, such as where the desired hash key is the value of another variable, values can be extracted by prefixing the name of the other variable with `$`. This results in the form `<?lsmb row.$column ?>`.

Through Template Toolkit, LedgerSMB::Template has the concept of private variables. If any component of the variable name starts with an underscore, it is private and will not be output to the rendered document.

Something to watch for when dealing with variable substitution for hashes is that in some cases, the key name conflicts with one of Template Toolkit's `VMethods` for the type. When that occurs and the value is not defined, the `VMethod` is executed instead of retrieving the value. An example of where this could happen is when retrieving the value of `form.sort`. As it is often undefined and included in "hidden" HTML elements, it can cause application errors when a templater attempts to retrieve it using `form.sort` instead of `form.item('sort')`.

Internationalization

In addition to the regular Template Toolkit functions, LedgerSMB has the additional functions `text` and `gettext`. Both of these functions are interfaces to the underlying `gettext` infrastructure and both can handle simple `gettext` substitutions. The difference between these functions is that `text` makes use of the locale of the current user, while `gettext` must be passed a string that identifies one of the `gettext` translations.

These `gettext` lookups can take placeholders in the same syntax that the underlying Perl does, `"[.n]"`. The output of the `i18n` functions is not escaped and the input values are not unescaped.

```
<?lsmb gettext('fr.CA', 'Item #[.1]', value) ?>
```

Formatting

The only formatting function that is added is *escape*. The escape function escapes a value using the rules for the template format.

Control Structures

The base IF, ELSIF, ELSE FOREACH

Debugging

Two template-side tools that can help with debugging are the interface to Data::Dumper and setting a debug message format.

Variable Dumping

The Data::Dumper interface is loaded using `<?lsmb USE dumper() ?>` and used with either `<?lsmb dumper.dump(var) ?>` or `<?lsmb dumper.dump_html(var) ?>`. Both `dumper.dump` and `dumper.dump_html` perform the same basic dumping, however `dump_html` also escapes the values and applies basic formatting. See Template::Plugin::Dumper for details.

Logging

Template Toolkit has a DEBUG directive that can cause line-based logging messages to appear in the rendered output when the debug flag to LedgerSMB::Template is true. A sample tag to start a debugging region with L^AT_EX templates is `<?lsmb DEBUG format '% $file:$line $text' ?>`. The debugging messages will start with the line that has the directive and a debugging region can be optionally terminated with `<?lsmb DEBUG format " ?>`, which tells TT to use an empty message.

Templating API

A new templating API was added in LedgerSMB 1.3. This API is accessed through LedgerSMB::Template, and has several support modules. Typical usage of this API is fairly simple in that most places that use it instantiate a LedgerSMB::Template object in auto-output mode then use the resulting object's render method. The constructor options are detailed in the POD for LedgerSMB::Template with the most interesting one for designing and testing user interface templates probably being "debug", which enables the display of debugging messages. The render method handles escaping the passed in template data for the selected output format and the actual rendering with the appropriate support module.

Options

@@@TODO

Input

@@@TODO

Output

@@@TODO

Support Modules

LedgerSMB::Template::LaTeX

This module uses Template::Latex to generate DVI, PDF, or Postscript output. It looks for template files that end in “.tex” and take marked-up L^AT_EX. Rendering of templates by these modules happens in two phases: the TT phase and the L^AT_EX phase. All templates that are directly called by LedgerSMB::Template that use these modules must be enclosed by the tag pair `<?lsmb FILTER latex ?>` and `<?lsmb END ?>` for the second rendering phase to occur. The final output file format is determined by the `format_option` filetype.

LedgerSMB::Template::HTML

This is the most frequently used LedgerSMB::Template module as the migration of the user interface to templates has started. It looks for template files that end in “.html”.

LedgerSMB::Template::CSV

A new form of output with LedgerSMB1.3, this module is used by some reports to give output in a standard form. It looks for template files that end in “.csv”.

LedgerSMB::Template::ODS, LedgerSMB::Template::XLS This pair of module outputs spreadsheets. Their input is specially formatted XML files that are processed by TT before the final conversion from XML to their respective formats. Details about the XML format are available in the documentation for Excel::Template.

The templates used by these modules have the extensions “.odst” and “.xlst”.

LedgerSMB::Template::TXT

A format infrequently used by LedgerSMB, this is used for Point Of Sale invoices. It looks for templates that end in “.txt”.

LedgerSMB::Template::TTI18N

This module defines some internationalization functions that can be called in templates that use Template Toolkit.

LedgerSMB::Mailer

This module provides the functions used by LedgerSMB::Template and the backup routines in the administration module to email files and documents.

Quick Reference

Variable Access	
varname	The value of the scalar varname
listname. <i>i</i>	Element <i>i</i> of the list, lists are indexed from 0
hashname.key	The value of the hash with the key <i>key</i>
hashname.\$key	The value of the hash with the key whose name is stored in \$key
Flow Control	
IF <i>condition</i> ; ELSIF; ELSE	A basic if/else block
FOREACH <i>var</i> IN <i>list</i>	Iterate over the members of <i>list</i>
END	Conclude an open block
Special Variables	
loop	Refers to the immediately containing loop
Built-in Methods	
any.defined	Returns true if <i>any</i> is defined
scalar.split(delimiter)	Returns a list of elements from the split string
list.size	Returns the number of elements in the list
list.join(delimiter)	Returns the list's elements joined together
list.defined(<i>i</i>)	Returns true if the specified element is defined
hash.item(name)	Returns the value of the element name from hash
hash.keys	Returns a list containing the hash's keys
hash.pairs	Returns a list of key/value pairs, sorted by key
hash.size	Returns the number of key/value pairs in the hash
loop.count	Returns the iteration number
loop.last	Returns true if it is the final iteration
loop.first	Returns true if it is the first iteration
Internationalisation	
text(string, @args)	Returns the LSMB locale's gettext value for string
gettext(locale, string, @args)	Returns the locale's gettext value for string
Debugging	
DEBUG format '\$file:\$line'	Starts and specifies the format of a debug region
DEBUG format ''	Ends a region of debug output
USE dumper()	Load the dumper plugin
dumper.dump_html(var)	Dump variables

Templating Changes Between Versions**LedgerSMB1.3 to LedgerSMB1.2**

- A new templating API was added to replace Form::parse_template
- The main rendering engine was changed to Template Toolkit

- Migration of the user interface to using LedgerSMB::Template was started
- DVI, Excel, OpenDocument Spreadsheet, and CSV output support was added
- The LedgerSMB::Mailer API was changed and documented

LedgerSMB1.2 to LedgerSMB1.1

- Tags were changed from the form `<% foo %>` to `<?lsmb foo ?>`
- The ability to compare two variables in templates was removed
- LedgerSMB::Mailer was rewritten to use MIME::Lite while retaining the old API
- SMTP support was added to LedgerSMB::Mailer

Part IV

Administration

Chapter 19

Overview

19.1 Introduction

This part of the book describes the tasks and processes that may need to be carried out on a regular basis in order to keep the application in good health and in line with requirements from end users.

Maintenance may require different types of system access for different types of tasks:

1. Within application tasks, such as user management, require an appropriately authorized application login
2. Database administration tasks, such as creation of backups and application upgrades, require a database level login to be used with (amongst others) 'setup.pl'
3. Other system-level maintenance tasks, such as updating PostgreSQL or Apache, which require user accounts on the server hosting LedgerSMB

Each of the above tasks requires its own type of user account. Creating and managing accounts for the first type of task is the subject of the next chapter.

Chapter 20

First Use

This chapter covers the general steps to use when installing LedgerSMB for the first time. If you elected to skip Part II Getting Started of this document then this summary will help you with the main steps normally required after installation to get LedgerSMB up and running.

It is assumed that:

- you have followed the installation instructions and have a working install. if not, then see the install instructions at @@@TODO.
- you have setup the company database using `setup.pl`. if not, then see the instructions at @@@TODO.
- you have created an admin user with all privileges. if not, then see the instructions at @@@TODO.

The basic accounting system setup steps are:

1. Create a new password for the user using `Preferences → Password`.
2. Setup the company defaults in `System → Defaults`.
3. Edit the Chart of Accounts as necessary using `General Journal → Chart of Accounts`. This includes setting up your bank account.
4. Enter or import customers.
5. Enter or import vendors.
6. Enter or import products or services.
7. Customize Sales Order format.
8. Customize Sales Invoice format.
9. Customize Purchase Quotation format.
10. Customize Purchase Order format.
11. Testing your setup.

Chapter 21

Within-application tasks

@@@TODO

Chapter 22

DBA tasks from setup.pl

@@@TODO

Chapter 23

Outside-application tasks

@@@TODO

Chapter 24

User management

@@@TODO

24.1 Introduction

This chapter deals with management of application level user accounts. This is the first of the three types of system access required to do LedgerSMB administration described in the previous chapter.

24.2 User creation

Users experienced with LedgerSMB 1.2 or before or SQL Ledger (any version) are referred to appendix [A.1](#) to read about the differences with version 1.3.

In order to create users, the current user must be sufficiently authorized. The user created at application set up time is such a user.

Go to the `System → Users`. You'll be presented the page as shown in figure [Figure 24.2](#) on page [143](#).

The value entered in the 'Username' field will cause a database user by that name to be created. Database users are a global resource, meaning that a collision will occur if multiple people try to define the same user in multiple companies. Section [24.5](#) on page [146](#) describes how to use the same user across multiple companies.

Enter the password to be used for this user into the "Password" field. If you're importing a user, please leave the field empty – that will prevent the password from being changed. Note that initial passwords (and password resets) are only valid for one day unless the user logs in and changes his/her password.

The "Import" field is discussed in Section [24.5](#) on page [146](#). To create a new user, leave the setting at "No".



The screenshot shows a web form titled "New User" with a black header bar. The form contains the following fields and controls:

- Username**: A text input field.
- Password**: A text input field.
- Import**: Radio buttons for "Yes" and "No", with "No" selected.
- Dr.**: A dropdown menu currently showing "Dr.".
- First Name**: A text input field.
- Last Name**: A text input field.
- Employee No.**: A text input field.
- Country**: A dropdown menu currently showing "Netherlands".
- Save User**: A button at the bottom left.

Figure 24.1: Screen for user creation - step 1

All of the "First Name", "Last Name" and "Employee No." fields are required. However, when no employee number is specified, the system will generate one using the sequence specified in the Defaults screen as documented in Section 18.4.3 on page 120.

The "Country" field speaks for itself and is required as well.

24.3 User authorization

After filling out all the fields as described in the previous section and clicking "Save user", you'll be presented a second screen in the user creation process: the user authorization screen. See Figure 24.2 on page 145 for a screenshot of the top of that screen.

The process of assigning user authorizations is the process by which the granted access to specific parts of the application. One can imagine that - in a moderately sized company - sales should not be editing accounting data and accountants should not be editing sales data. Yet, in order to cooperate, both parties need to be given access to the same application. This is where authorizations come in.

In aforementioned screen, which equals the "Edit user" screen, you have to assign the newly created user his application rights. By default, the user doesn't have any rights. Checking all check marks makes the user an application "super user", i.e. gives the user all available application rights.

For a description of the roles a user can be assigned and their effects, the reader is referred to Appendix E starting at page 257.

Editing User

Username admin

Password

First Name Erik

Last Name Hulsmann

Employee No.

Set Password [\[Edit\]](#)

- ☐ account all
- ☐ account create
- ☐ account edit
- ☐ ap all
- ☐ ap all transactions
- ☐ ap all vouchers
- ☐ ap invoice create
- ☐ ap invoice create voucher
- ☐ ap transaction create
- ☐ ap transaction create voucher
- ☐ ap transaction list
- ☐ ar all
- ☐ ar invoice create
- ☐ ar transaction all
- ☐ ar transaction create
- ☐ ar transaction create voucher
- ☐ ar transaction list
- ☐ assembly stock
- ☐ assets administer

Figure 24.2: Screen for user creation - step 2

24.4 Maintaining users

24.4.1 Editing user information and authorizations

When the role of a user in the company changes, it may be necessary to assign that user new roles and possibly revoke some other roles. This can be done through by navigating to `System → Users` which brings you to the same screen as presented in Figure 24.2 on page 145.

Similarly, there may be reasons to change the user information, such as a last name (e.g. upon marriage).

Note that if you reset a password, the new password is valid for one day unless changed. The preferred business process is for the individual with permissions to reset the password, give the new password to the user, they then log in and immediately change it.

24.4.2 Changing user preferences

Each user can change his preferences and password through the `Preferences` top level menu. See Figure 24.3 on page 146.

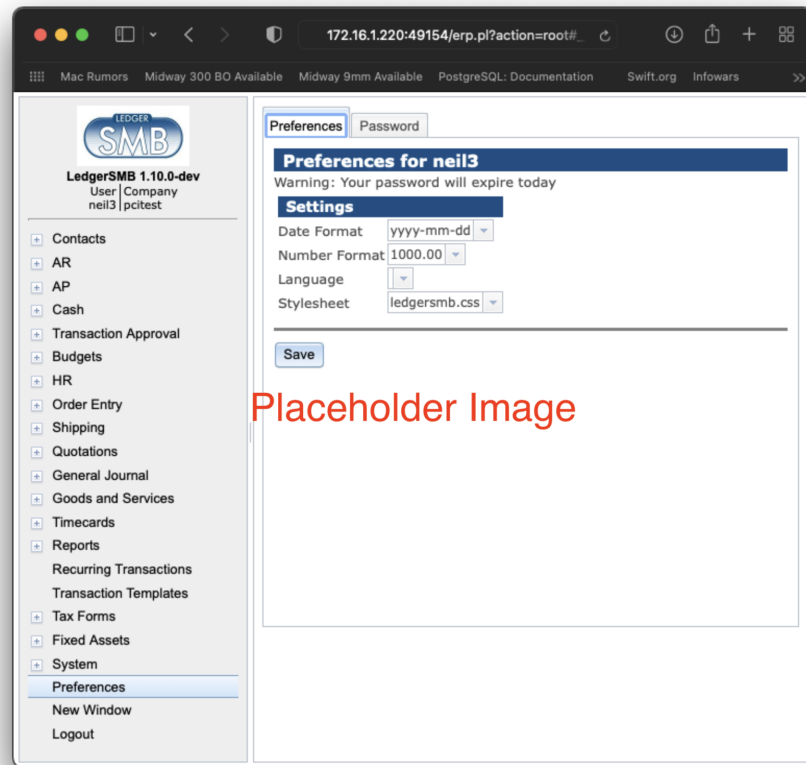


Figure 24.3: User preferences screen

24.4.3 Deleting users

From the “User search” result screen, users can be “deleted” from the company: they have their access to the current company revoked.

Note that the user is only revoked access to the current company; the login remains a valid login for the database cluster. Administrators wanting to remove user accounts at the database level need to take additional action.

24.5 User imports

If a database user already exists, e.g. because this user was created to be used with another LedgerSMB company, it can’t be created a second time. In order to be able to use that user with the current company, it needs to be “imported” instead.

The difference between creating a new user and importing one is that the “Import” radio button should be set to “Yes” and that you should not fill out a password. If you do, the password of that user will be overwritten for all companies.

All other fields are still applicable: the data entered for other companies isn’t copied to the current company.

24.6 **setup.pl** users

As mentioned in the introduction, users created through the process documented above don’t have rights to execute work with the setup.pl database administration tool. Note that this is on purpose. You will need access to the server to create such users, or request one from your application service provider (ASP) if you use a hosted solution.

During the set up process such a user is normally being created. This user can later be used to manage the database from the database administration tool setup.pl.

See also Section [5.1.1](#) on page [45](#)

Chapter 25

Definition of goods and services

25.1 Definition of goods

Structure of products in the system.

25.1.1 Definition of parts

The part entry screen consists of four parts:

1. Part information
2. Vendor information
3. Customer information
4. File attachments

The paragraphs below discuss each of the four sections. The part definition section contains both required and optional fields. The information in the remaining three sections is entirely optional.

Part information

Every part requires the following fields to be entered:

Number

The (alphanumeric) code the company uses to identify the item

Description

The (native language) description of the item, used as the default description on sales invoices

Inventory account

The asset account used to maintain the monetary equivalent of the inventory amount

Income account

The P&L (income) account to post the sales revenues on

COGS account

The P&L (expense) Cost of Goods Sold (COGS) account to use to post cost of sold items on

Sell price

The default selling price used on sales invoices

The other fields in this section of the screen are optional:

List price

Informational; can be used for any (monetary) purpose

Last cost

Last buying price, updated when a vendor invoice listing the current part is posted

Markup percentage

Markup on Last cost to calculate the Sell price

Image**Drawing****Microfiche****Unit**

A five-character field shown on the invoice

Weight

Informational; can be used by **add-ons** or when customizing

ROP

Reorder point - when the inventory drops below this number, the part will show up in "Short parts" reports

Bin The storage location in the warehouse

Apart from these fields, there are also the Make and Model paired fields. Every part can have as many Make/Model lines as required. They are informational, but can be used when customizing the software.

The Average Cost and On Hand fields are output-only calculated fields. Average Cost is calculated from the historic buying prices. On Hand is the current inventory, which is updated when posting a vendor invoice (increased) or sales invoice (decreased).

Taxation of parts

Right below the accounts selection section, there is the Tax section, which lists all tax accounts with a check mark. Each account corresponds with a certain tax type and rate. E.g. in the Netherlands, there's one VAT tax type (BTW) which has two rates, one of which applies to every product. This setup requires two accounts. There's more on the subject of sales taxes in Chapter 26 starting at page 155.

By checking the check mark on an account, the system is signaled to calculate that kind of tax for the part *if the customer (or vendor) has the appropriate setup* as described in section

Vendor information

This section of the screen lists one or more vendors from which the part can be purchased, with purchasing information for the given vendor:

Vendor code

Code used by the vendor to identify the good, to be used by and future customization and enhancements (currently informational only)

Lead time

Lead time of the part from the vendor in days

Last cost

Last price at which the good was purchased from the vendor

Currency

The currency belonging to the Last cost field

Customer information

The customer information section specifies sell prices per customer or price group where those are required to deviate from the default sell price. This mechanism exists to support the marketing principle of categorizing customers.

Sell price

Price for this part to be used for this customer

From

Start of the applicability window of the price (inclusive)

To End of the applicability window of the price (inclusive)

File attachments

The process of attaching files to parts is very similar to that of attaching files to quotations, which is covered in Section 34.3 on page 183.

25.1.2 Definition of part groups

When adding parts, services, or overhead in Goods and Services they can be categorized in 'part groups'. When searching, the part group can help to limit the number of matches. Currently, this is the only use of 'part groups'.

Parts groups have a name and an – optional – parent. When selecting a parent, this creates a hierarchy of parts groups which can be used when searching for parts: selecting a parent parts group as a filter on parts search will include all parts in the groups for which the selected group is the parent.

When no part groups have been defined, the part group assignment field doesn't show up on any of the goods entry screens. Part groups never show up on the add assembly screen.

There's no requirement that a good be assigned to any specific part group.

25.1.3 Definition of assemblies

25.1.4 Definition of overhead or labor

25.1.5 Parts versus assemblies for multi-item-package sales

Often times, one may want to sell pre-packaged multiples of a single item, such as Jack in Chapter 7 starting at page 67 who wants to sell memory modules in pairs as well as single items, with the price for the pair set separately from the single-item price.

There are basically two use-cases

1. Pre-packaged sales which are separately stocked
2. Single item sales which are achieved by unwrapping the multi-item packages

The former case would be that of a supermarket selling packages of coffee in singles and shrink-wrapped in pairs. These items get stocked, sold and produced separately. Handling these is straight forward: as they are basically separate products from the point of administration and inventory management, they're handled as separate parts.

The latter case would be Jack's case where single-item sales is achieved by unwrapping a multi-item package. Basically, there's a single inventory for both types of packaging. This situation can be handled (a) by creating separate parts or (b) by creating a part and an assembly. To reiterate: The fact that one wants to be able to separately set a pricing strategy for the bundled items is the driver to go look for other solutions than just sell a multiple of the original item.

There is no option which matches the actual practice: one inventory for two parts. The solution always will keep two separate stocks for the two items, but one may work better in practice than the other.

Option a (separate parts) keeps the inventory of both items strictly separate, meaning there's no way to convert between the two, other than selling and buying.

Option b (a part and an assembly) mismatches reality in so far that it will require one to stock the single items and update the assembly stock regularly while in practice the multi-item packages are stocked and unwrapped upon single-item sales. This procedure works to have the assembly stock go negative on sales and restock regularly (e.g. daily) to update assembly stock back to 0 (zero). This removes inventory from the single-item stock, allowing for a semi-automated way to convert stock from one type of inventory to another.

25.2 Definition of services

25.2.1 Taxation of services

25.3 Alternative accounts (GIFI)

Next to the regular LedgerSMB G/L account numbering scheme, LedgerSMB supports a second numbering scheme: **GIFI** numbering. **GIFI** is a standardized account coding system used in Canada by the Canada Revenue Agency for processing corporate Tax Returns forms.

Other countries (e.g. France) may have required numbering schemes for corporate accounting as well and the GIFI account functionality can be used for them also.

If you use GIFI account numbers, each LedgerSMB G/L account is associated with a GIFI account. Multiple G/L accounts may map to a single GIFI account. Note that the hierarchy for G/L accounts is completely separate from GIFI accounts.

General Ledger reports may exist in two variants: a variant using the normal G/L accounts and one with the GIFI numbering. In the GIFI variant, when a single GIFI account is mapped to multiple LedgerSMB accounts, the total reported under GIFI is the sum of the mapped G/L accounts.

The GIFI reporting variant only shows if GIFI accounts are defined.

25.3.1 Maintaining GIFI

GIFI accounts must be created before they can be assigned to the LedgerSMB G/L account. Some predefined Chart of Accounts define GIFI accounts and map the association of GIFI to G/L accounts.

GIFI accounts can be reviewed, lightly edited, entered or imported by navigating to `General Journal` → `GIFI`.

Once GIFI accounts are defined the account number cannot be changed. How-

ever, the GIF I description can be changed any time. Be aware that changing this description will effect reports generated on historical data prior to this change.

LedgerSMB G/L accounts can be assigned to GIF I accounts by navigating to `General Journal → Chart of Accounts` and selecting a G/L account. In the resulting view, the Configuration section shows the GIF I dropdown.

G/L account headings cannot be assigned to GIF I accounts.

25.4 Reporting

As for the general journal being presented in the "ordered by account" and "ordered by date": these are both in the `General Journal → Search` menu.

Once you run the report, you can sort the lines by date (journal) or by account (which I believe is sorted by date within the account). My accountant either lists "all transactions for one specific account" or "all transactions for 202x; all accounts" so usually, sorting by date isn't much of an issue.

I believe that in the `General Journal → Search` search options, you may need to check the "Account" indicator to add the account to the output. This comes in handy when you decide not to filter for a specific account.

@@@ Needs Editing

Chapter 26

Taxes

26.1 Overview

When an account has been marked as a Tax account (see Section 18.3.3 on page 110, item 18.3.3) several things happen:

- The account will be shown in the customer and vendor account screens with a check mark to mark it “relevant for the customer”
- The account will be shown in the part and service configuration screens with a check mark to mark it “relevant for the part/service item”
- The account will be shown in the tax configuration screen in order to set a tax percentage on the account as discussed in the next section

By marking an account relevant for a customer, taxes will be calculated when creating a sales invoice for the given customer which includes parts which also have the specific account marked as relevant.

26.2 Tax account configuration

The screen (shown in Figure 26.2 on page 155) presents a table with the full history of the percentage rates, their dates of applicability and the tax accounts they apply to. An account can show up any number of times, with different Valid To dates.

Account name

The first column; not an input field

Rate (%)

The rate to be applied to the sales amount

	Rate (%)	Min Taxable	Number	Valid To	Ordering Tax Rules
Sales Tax	5.000	0		infinity	0

Figure 26.1: Tax account configuration screen

Min Taxable

The minimum amount for which the tax is applicable

Number

The tax reporting number to be used for this tax

Valid To

Date until which the tax rate is applicable

Ordering

The meaning of this field is described below

Tax Rules

The selection list presents a list of available 'tax rules engines'. By default the list consists of the 'Simple' module only, which is the engine which comes with LedgerSMB; the reader is referred to Section 26.4 on page 157 for a more detailed description of this subject

To determine which line is applicable at any time, order the lines by their validity date and select the first one for which the validity date is later than the date being checked for. E.g. if you have two lines - one with a Valid To date of 2010-01-01 and one with a Valid To date of "infinity" - the 2010 line would be selected when checking for a date in 2009 while the "infinity" line would be selected for any date later than Jan 1st, 2010. This facilitates advance entry of a new tax rate in case of a rate change if a tax rate changes: the new rate will automatically be selected beyond the validity date of the old rate.

The order field is used to layer taxes, e.g. Canada where the province of Quebec taxes the taxes collected by the national authority, which works like this numerically: National tax with order 0 and rate 10% and Quebec tax with order 1 and rate 5% applied to a 10\$ amount charges 1\$ of national tax and 0.55\$ of Quebec tax (5% of 11\$).

Note that - as pointed out in the Overview section of this chapter
- the **customer** related tax calculation on invoices is triggered by

the settings in this chapter as well as the customer/vendor settings described in Section 33.4 on page 177 and part settings and service settings described in Section 25.1.1 on page 151 and Section 25.2.1 on page 153 respectively.

26.3 Tax calculation examples

@@@ TODO

26.4 Tax calculation plug-ins

The tax calculation system has been designed to handle the most complex tax system thinkable. Because the tax calculation rules for most set ups aren't really all that complex, LedgerSMB comes with a single tax calculation plug in out of the box: the "Simple" tax calculation rule.

For more complex needs, more complex routines can be developed and plugged into LedgerSMB side-by-side with the simple rule.

26.5 Tax forms

The tax forms facility exists to help file sales tax forms. It's primarily modeled after the 1099 US tax reporting forms functionality, which means that it's cash based. The fact that it's cash based means that invoices will be included in the tax report as soon as cash for the invoice is received. Accrual based reporting means that the tax is reported based on the invoice date instead of the payment date.

This system can only be used in some EU countries and by some companies at that: the country needs to allow cash based reporting and the company needs to have filed for cash based reporting as well.

Before using this functionality be sure to check with an accountant or the tax authorities if cash based reporting is appropriate.

26.5.1 Overview

Tax forms are meant to support tax reporting requirements in the countries where the company has tax or other government reporting obligations. This means that tax forms are country specific: each country for which reporting obligations exist needs to have its own form associated.

Using tax forms requires that:

- A template has to exist in order to format the tax report. Two United States based templates are provided by default: 1099-MISC and 1099-

INT, provided they were loaded during the company setup. To view them navigate to `System → Templates` and select them in the “Template” drop down. For more information on templates see Section 18.4.12 on page 124

- A tax form has to be created for each type of tax reporting or jurisdiction. See Section 26.5.2 on page 158.
- Each customer and vendor that has government reporting requirements needs to be linked to a tax form. Section 33.4.2 on page 178 describes how to do this.

When associations have been set up between companies and tax forms, the “Tax form” check mark can be used in invoices to include (or exclude) invoice lines from the tax form. See Section 37.2.1 on page 192 for entry of invoices.

26.5.2 Creating tax forms

Tax forms can be created by navigating to `Tax Forms → Add Tax Form`.

Tax forms can be edited by navigating to `Tax Forms → List Tax Forms`.

Tax forms have the following fields:

Country

Country to which the tax form applies

Description

Name of the tax form to be displayed in drop down lists

Select by Default

Determines if the tax form check mark on invoices is checked or unchecked by default when the form is loaded

Accrual Basis

Use accrual based tax reporting for the report instead of cash based.

26.5.3 Using tax forms

To print the Tax Form navigate to `Tax Forms → Reports` where there are selections for which tax form, customer number, vendor number, date range etc.

Chapter 27

Pricing

27.1 Introduction

There are a number of mechanisms which interact to create a price for an item on an invoice, order or quote:

1. Through direct entry on the document being created (either the price, the discount or both)
2. Using the "Sellprice" (or "Last cost" @@@ really?) on the parts entry screen
3. Using a "Type of Business" associated discount
4. Using the price matrix functionality, which combines (for customers), from least to most specific:
 - (a) A price group
 - (b) A customer
 - (c) A minimum order quantity

The price matrix for vendors is simpler and only includes the vendor as a selection criterion.

The price matrix can be used to set a price, a discount or both. Do note that the **customer's** payment terms may cause another discount to be applied for early payment. This type of discount is not part of the topic at hand.

This chapter deals with business type and price matrix discounts.

@@@ types of business are 'old school'; price groups have been introduced to replace types of business with a more fine-grained structure.

27.2 Definition of types of business

Types of business are really straight forward: they feature a description which allows them to be identified in the **customer** account screen and a discount percentage which is applied across the board. I.e. all invoices to the customer will have that discount applied.

27.3 Definition of price groups

Price groups feature only a description and constitute a category. Customers can be assigned a price group (but this is optional). In case a customer is in a price group, this price group is taken into account when determining a price for an item through the price matrix.

27.4 Using the price matrix

At the lower half of the parts and services definition screen, there are two sections; one with the first heading being “Vendor” and the other with the first heading being “Customer”. These are the vendor and customer price matrix sections for the given item. As the customer price matrix is both more complex *and* much more interesting from a functional perspective, this section will only cover the customer side.

The customer price matrix lists the following fields:

- Customer (and customer account)
- Price group
- Discount (optional)
- Sell price (optional)
- Currency
- From (date)
- To (date)
- Minimum quantity

At most one of the “Customer” and “Price group” values should be set; in case a price group has been selected, the row applies to all customers which have been assigned that group in the customer “Account” page. If there is a row listing the customer explicitly as well as one for the customer’s price group, the most specific one – the customer – is taken. If neither a customer nor a price group has been entered, the row applies to all customers.

When applying the price matrix algorithm, the Currency, From, To and Minimum quantity values restrict applicability of the row: the invoice's currency needs to match, the invoice date must be between From (inclusive) and To (inclusive). The row with the highest value in Minimum quantity which is lower than the total number of items specified on the invoice is selected.

To visualize the algorithm, please consider the following examples.

27.4.1 Price matrix example: simple pricing

Consider customers A and B and a part with a generic "Sell price" of 110.00. The simplest price matrix for a part, could look like:

Table 27.1: Customer specific price matrix

Customer	Price group	Discount	Sell price	Currency	From	To	Min Qty
B			99.00				

When creating an invoice for customer A (not in the prices matrix), the part will be listed at generic sales price of 110.00 while at the same time an invoice for customer B lists it at 99.00.

If customer B had been in the "Frequenters" price group, and the matrix had looked like this:

Table 27.2: Price group specific price matrix

Customer	Price group	Discount	Sell price	Currency	From	To	Min Qty
	Frequenters		99.00				

The outcome would have been exactly the same. Discounts are determined exactly the same way as absolute prices, when the "Discount" field is set.

27.4.2 Price matrix example: Quantity pricing

Consider a part with a generic "Sell price" of 110.00 and the price matrix below, which includes "Minimum quantity" pricing:

Table 27.3: Price matrix configuration

Customer	Price group	Discount	Sell price	Currency	From	To	Min Qty
			99.00				200
			90.00				400

With this price matrix in place, any customer ordering up to 199 items will pay 110.00 per item, whereas orders of over 400 items will cost 90.00 per item and every order in between will be 99.00.

27.4.3 Price matrix example: Time-based (Sale) pricing

Consider a part with a generic “Sell price” of 100.00 and the price matrix below, which includes time-based pricing:

Table 27.4: Price matrix configuration

Customer	Price group	Discount	Sell price	Currency	From	To	Min Qty
		30%			Oct 15	Oct 31	

Any invoice outside the interval Oct 15th through Oct 31st will list the part at a price of 100.00. During the interval the listed price will be 70.00.

Chapter 28

Auditing

There's an audit-trail table which tracks (and this can't be disabled) who changed what in the transactions you post. This table isn't currently visible in the application; it's not hard to make it visible. However, when we look at what auditing usually means, it is that you're able to completely demonstrate everything that happened to your books while compiling your accounting data. That is: you can show that the invoices sent to your customers are the same as shown in the system. Also, auditors want to see completeness: that you've recorded all invoices sent out in the books and that all work performed has been invoiced.

For completeness, we have "gap-less numbering" for invoices. That means the system generates invoices with numbers that are always exactly 1 apart. so when a number misses, you know the books have been tampered with.

There's another thing with LedgerSMB that helps auditability: transactions can't be deleted or changed once they have been posted. Some people hate that: they find mistakes and it's a lot of work to correct those. Also, large vendors like Quickbooks, Sage and Xero allow changing transactions after they are recorded in the books (but they have a special "auditor mode" which shows the old versions of the same transaction).

@@@ Needs Editing

Chapter 29

Contingency planning

29.1 Backup and restore

29.1.1 Backup using setup.pl

29.1.2 Backup using PostgreSQL administration tools

29.1.3 Restore

Note that if the database being restored is a backup from a different (earlier) PostgreSQL version, additional steps described in Appendix [D.6](#) starting at page [253](#) will be required.

29.2 Advanced PostgreSQL: replication

Chapter 30

Software updates

30.1 Introduction

This chapter describes patch and minor version release updates from 1.4 or higher. Upgrades from older versions (e.g. 1.2 or 1.3) are covered in Appendix D starting at page 251. The same is true for upgrades from SQL Ledger 2.6 or 2.8.

The remainder of this chapter discusses the steps to be executed. Please note that during execution of the instructions below the application isn't to be used.

30.2 Backups

All upgrades should start by creating backups of the system as it runs. This means:

- Backup database roles (authorizations)
- Backup database content (structure and data)
- Backup software and settings files

The database content and roles backups can be created through the `setup.pl` web interface. The database content backup can also be created using the `ledgersmb-admin` administration command line tool.

A backup of the software itself is only relevant when using a source installation. When using the (recommended) Docker installation, this step can be skipped; it *is* important to record which image version the container is based on. For source installs the backup can simply be performed by recursively copying the containing directory: `cp -Rp /srv/ledgersmb /srv/ledgersmb.bak`

30.3 Software upgrade

When using Docker, the upgrade is as simple as spinning up a container with the new version. For source installation the procedure is simple as well: The tarball needs to be unpacked, replacing the existing sources. From the archive directory, the file `ledgersmb.yaml` (or `ledgersmb.conf`) must be copied. The same needs to be done for any custom CSS theme files in the `UI/css` directory.

If your setup is using the deprecated `ledgersmb.conf` INI-style configuration file, you should use the migration utility `utils/migration/convert-ini-to-di` to generate the new YAML-based configuration format `ledgersmb.yaml`.

30.4 Database upgrade

After upgrading the installed software, all LedgerSMB company databases need to be upgraded in order to become compatible with the new software. This applies to minor as well as to patch releases: next to tables the schema also contains a large number of functions. While it's policy for the project not to change the table structure in patch releases, these functions may need to be changed as part of a patch release.

The upgrade is performed by logging into the database through `setup.pl`. After successful login, the question "LedgerSMB X.YY db found. Rebuild/Upgrade?" with a `Yes` button right below. "X.YY" is the version you had running before starting the upgrade process.

After clicking the `Yes` button a series of database upgrade scripts will be executed. Prior to executing each script, checks will be performed to test whether the upgrade can be successful. If these checks find problems that need to be resolved before running the script, it will show a webpage listing offending data with instructions on how to resolve the situation.

Important This database upgrade process is one-directional. Once started there are no scripts to reverse the process. The only option to back out of the upgrade is by restoring the backup created at the start of the process.

The community is always interested in improving the upgrade experience, so when stuck it's highly advised to contact the developers on their mailing list (devel@lists.ledgersmb.org), log an issue in the issue tracker (<https://github.com/ledgersmb/LedgerSMB/issues>) or join the chat channel (<https://app.element.io/#/room/#ledgersmb:matrix.org>) to report the problem.

Chapter 31

Optional Features

As business requirements change sometimes it may be necessary to add or remove some of the optional features of LedgerSMB. This chapter describes how these optional features work, how to troubleshoot them if things go wrong, and how to enable or disable them.

31.1 PDF and Postscript Documents

LedgerSMB can create PDF and Postscript documents for orders, invoices, and more. This is an optional feature and requires some additional software not included with LedgerSMB, including a $\LaTeX$ distribution and extensions to Perl's TemplateToolkit framework.

The PDF and PS invoices are generated using a program called $\LaTeX$ which handles the layout and typesetting. The actual $\LaTeX$ files are created using Template Toolkit with extensions for $\LaTeX$. These extensions are in the Template::Latex package available from CPAN. The software then generates a $\LaTeX$ file which is then processed to create a PDF or PS.

Typically the first thing to do is to install a $\LaTeX$ distribution like TexLive (distributed with many Linux distributions and available for OSX and Windows). This provides $\LaTeX$ and many of the modules needed. In general I recommend that if your distro has a texlive-extras package that you install this too.

After this is installed, you must then install Template::Latex. This can be done by typing on the command line:

```
# cpan Template::Latex
```

This will also install a number of dependencies including LaTeX::Driver, which will need to know where your LaTeX binaries are. It is usually pretty good at finding them.

If things go wrong and you can't get it to work, the following commands may provide useful diagnostic information when requesting help:

From the LedgerSMB application directory: `$ perl -MLedgerSMB::Template::Latex -e print`

From the doc/manual directory in the LedgerSMB application directory: `# pdflatex LedgerSMB-manual`

31.2 Attaching Uploaded Files to PDF Invoices

31.3 OpenDocument Spreadsheet Output

31.4 Microsoft Excel Output

Part V

Business processes

Chapter 32

Overview

This part of the document describes how LedgerSMB handles specific business processes. For example, Order Entry, Shipping, etc.

Some of these processes may not apply to your specific use of the software and can be ignored.

Chapter 33

Customers and vendors

33.1 Introduction

This section describes how LedgerSMB implements **customers** and **vendors**.

When creating a customer or vendor in LedgerSMB, you have to first create a **contact**. This contact can be created as either a **company** or person. However, this contact can't itself be used directly as a **vendor** or **customer**. Instead, you have to create a "**credit account**" which is linked to the contact. A "**credit account**" can have the role of vendor, customer or both.

LedgerSMB works equally well using a person as a vendor, customer, or both.

A contact can have multiple **customer** and/or **vendor** type accounts. Each account has its own language settings, shipping address, contact data and payment conditions. Basically, you will create multiple accounts when you have to record different data for any of the items listed above.

Note that at least as of LedgerSMB 1.9, a contact can only be linked to one bank account. This means that if your contact as a vendor has separate bank accounts from your contact as a **customer**, you will need to set them up as different contacts.

The contact entity itself is used - albeit not as customer or vendor - in the context of credit risk management (See Chapter 41 starting at page 203).

33.2 Contacts

In LedgerSMB, a "Contact" is either a person, a company or a robot.

The person, company or robot has a uniquely corresponding entity, where it is assigned an `entity_class` such as Vendor, Customer, Employee, Robot. This class is a marker only. It has no bearing on what the Contact is able to do, so, for example, a Contact marked as "Employee" can also act as a vendor who we purchase from and a customer who we sell things to.

The entity may be further extended such that it becomes:

A `LedgerSMBUser`, by linking with the `users` table. The schema allows the `Contact` to link with multiple `LedgerSMBUsers` (so a person may have one username they use for general accounting and another they use for administrative functions), though the UI does not currently support this. A user belongs only to a single `Contact` entity.

An employee, by linking with `entity_employee`. This is a 1:1 relationship. Table `entity_employee` is used to extend the entity. Note that employee has start and end dates, but there is no provision for multiple employments, such as when an employee leaves and is later re-employed by the company.

The `Contact` entity may have multiple `entity_credit_accounts` associated, which can be classed as either "Vendor" or "Customer". These do affect what a `Contact` is able to do. For example, to act as a vendor who we purchase from, the `Contact` must have an `entity_credit_account` of class "Vendor". For us to be able to raise a sales invoice to the `Contact`, it must have an `entity_credit_account` of class "Customer".

A `Contact` may have multiple `entity_credit_accounts` of each class, for example it may have multiple "Customer" accounts corresponding with different divisions of a single company to be invoiced separately.

The `entity_credit_account` may optionally be assigned a business type.

A `Contact` may be known by multiple names through the `entity_other_name` table, though this is not yet supported in the UI.

Multiple notes, locations (being addresses) or contact elements (being a single telephone number, e-mail address, social media handle or similar) may be associated with either the overall `Contact`, or individual `entity_credit_accounts`.

An summary outline of the related database tables is as follows:

```
company -|
  robot -|--entity
  person -|  |
            |--users
            |--entity_employee
            |--entity_other_name
            |--entity_note
            |--entity_to_contact
            |--entity_to_location
            |-- location
            |--entity_credit_account
            |-- eca_to_location
            |   |-- location
            |--eca_to_contact
            |--eca_note
            |--person
            |--business
```


33.3 Contact Search

The LedgerSMBUI provides a Contact Search screen at `Contacts → Search`. Each filled-in field tightens the filter criteria. Clicking 'Search' without filling any fields in returns a list of all Contacts.

33.4 Creating customers and vendors

The typical procedure to create customers and vendors works exactly alike:

1. Create the **contact** as a company
2. Create **credit accounts**
3. Attach addresses, contact info and notes to the accounts

33.4.1 Creating a Company

To create a company navigate to `Contacts → Add Entity → Company Tab`.

The company tab has the following fields:

Control code

Code to uniquely identify this entity. This is usually generated by clicking on `Generate Control Code` button. It is also generated automatically when clicking the `Save` button.

Name

Legal name of the company. ****Required.****

Country

Country of incorporation. ****Required****, but defaults to a known value so it needs to be checked.

Tax number/SSN

Tax (VAT/Sales tax) number of the company.

Sales Tax ID

The sales tax reporting number.

License Number

The company operating license number.

The `Generate Control Code` button generates a new control code upon user request when the user is entering a company which isn't yet known in the system.

The `Retrieve` button is discussed in Section 33.6 on page 179.

The `Save` button creates the company.

33.4.2 Creating a customer account

When a company has been created then “Customer” or “Vendor” Credit Account can be added. The Credit Accounts entry screen lists the following fields¹:

Class

Either a **customer** or **vendor**

Number

Number to identify this account among all other accounts in the company; when left empty, the system will generate one when you click “Save New”

Description

Textual representation of the account, usually a name

Pay To

If printed checks are used, then this line is the Pay To line on the check. It also shows up when picking a customer with multiple credit accounts

Starting Date

Date from which the account is valid

End date

Date until which the account is valid, or empty if there’s no known end date

Threshold

Minimum amount for invoices to be sent out

Credit limit

Maximum amount of open invoices and orders allowed for the account, see Chapter 41 starting at page 203

Payment Terms

Number of days within which invoices have to be paid

Discount (conditions)

Percentage discount the account is entitled to when payment is within the given number of days

Account

Account used to post discounts as calculated based on discount conditions

AR or AP

For customers, the Receivables G/L Account (AR) used to post created sales invoices on. For vendors, the Payables G/L Account (AP) used as default to post received purchase order invoices on. Only one of AP or AR is shown.

¹To simplify the interface if they’re unused, some fields are not shown in case their selection lists are empty

Payment

Which bank account to use when receiving or paying funds

Currency

The default currency to be used with the **customer** or vendor

Language

The language parameter is used to select templates for communication with the customer

Taxes

The applicable tax form for this vendor; tax forms are discussed in more detail in Section 43.7.2 on page 216 Chapter 26 starting at page 155

33.4.3 Creating a vendor account

The only difference between creating a **customer** account and a **vendor** account is the Class and the selection of AP or AR G/L accounts. See Section 33.4.2 on page 178 for setting up a customer account.

33.5 Multiple customers within one company

33.6 Creating vendors from customers

33.7 Maintaining contact information

Chapter 34

Quotations from Vendors and for Customers

34.1 Creating Quotations and RFQs

The RFQ entry screen is located at Quotations → RFQ.

Logged into example_inc

- AR
- Point of Sale
- AP
- Cash
- Transaction Approval
- HS
- Order Entry
- Shipping
- Quotations
 - Quotation
 - RFQ
 - Reports
 - General Journal
 - Goods and Services
 - Projects
 - Reports
 - Recurring Transactions
 - Tax Forms
 - Fixed Assets
 - Batch Printing
 - System
 - Stylesheet
 - Preferences
 - Logout

Action: update, ID:

Add Request for Quotation

Vendor: ABC Parts [New] RFQ Number: []

Currency: USD Quotation Date: 2012-10-27

Shipping Point: [] Required by: []

Ship via: []

Item	Number	Description	Qty	Unit	OH	Price	%	Extended	TaxForm
1	q1	Samsung 1TB	10	10					☐

SKU: q1

Required by: []

2 [] [] [] [] [] [] [] ☐

Group: []

Notes: [] Internal Notes: []

Subtotal: 0.00
Total: 0.00

RFQ [] HTML [] Screen []

Update Save Ship to Print and Save

Figure 34.1: RFQ entry screen

Remark Note that the **RFQ** entry screen contains prices; this is misleading at least: the printed output to be sent to the vendor does not. The fact that this screen allows entry of prices could be considered a bug.

The listing below describes the meaning of the per **RFQ** fields presented in the screen.

Vendor (Customer)

Name of the company the quotation is requested from (issued to). **Required**

Currency

Currency for the intended transaction

Shipping point

The address the quoted material will be shipped from

Ship via

The expected shipping method to use in the quote

RFQ number

The number of the document (automatically generated when left empty)

Quotation date

Date the RFQ document is issued

Required by

Date of the intended delivery of the goods and services requested

Notes

Notes to be included on the printed RFQ document sent to the vendor

Internal notes

Notes to be kept internal to the company - not disclosed through the RFQ document

The following per item fields are listed.

Item

Number of the item in the RFQ

Number

Part number of the item to be ordered

Description

Description or name of the item to be ordered

Qty The number of items to be ordered

Unit

The unit in which the quantity is measured; e.g. 'each' or '6pack'

OH Number of items currently on hand (i.e. in stock)

Price

Item price

% Discount percentage rate

Extended

Item price after discount

TaxForm

SKU

Stocking unit - code used to track stock

Required by

Required date of delivery; can be used to specify a different date than the date in the header

Remarks

Item specific remarks

Group

(Not shown in the screen shot) Part group for the item

Note that the quotation entry screen offers a number of extra buttons after the data has been saved.

34.2 Expediting quotations

34.2.1 Printing

34.2.2 Sending by e-mail

34.3 Attaching files to quotations

Chapter 35

Sales and vendor orders

In this chapter the options for creating orders will be discussed. After successful order creation, there are two possible next steps. The first applies to cases where actual goods have to be handled and goes through shipping and receiving as discussed in Chapter 36 starting at page 187. The other skips handling of goods and directly proceeds to invoicing in Chapter 37 starting at page 191.

35.1 Creating new orders

35.2 Creating orders from quotations

35.3 Creating orders from projects

@@@ What about from time sheets??

35.4 Creating purchase orders from sales orders

35.5 Combining orders

35.6 Recurring orders

35.7 Incomplete orders

Orders fully shipped and invoiced are automatically closed. However, this isn't always an option.

In case an order ends up being partially shipped and parties agree not to ship

the remaining items in the order, the order stays open and incomplete. In situations like these, the order needs to be marked “Closed” in the order entry screen. A note stating why the order was manually closed can be put in the “Internal Notes” input box. Clicking the “Save” button stores the changes in the application database.

Chapter 36

Inventory management

36.1 Shipping

36.1.1 Pick lists

@@ Order picking is the work(flow), not the picklist!

36.1.2 Packing list

@@ Order packing is the work(flow), not the packlist!

36.2 Receiving

@@@TODO the role of shipping points in receipts??

Reception of goods into inventory assumes having at least one saved and incomplete purchase order such as shown in Figure 36.1 on page 188 stored in the system.

The reception process starts by going through the menu to the “Receive” order lookup screen (Shipping → Receive). This will show the order search screen which will help finding any orders which have items available for reception, see Figure 36.2 on page 189.

After filling order selection criteria and hitting the “Continue” button (or doing so immediately to see all orders with outstanding items in the system), the system will return a listing of orders with matching the selection criteria as in Figure 36.3 on page 190.

Upon selection of one of the orders by clicking the order number the next screen is loaded as shown in Figure 36.4 on page 190. In this screen you can enter the amounts received in the current lot. The entered data is to be confirmed

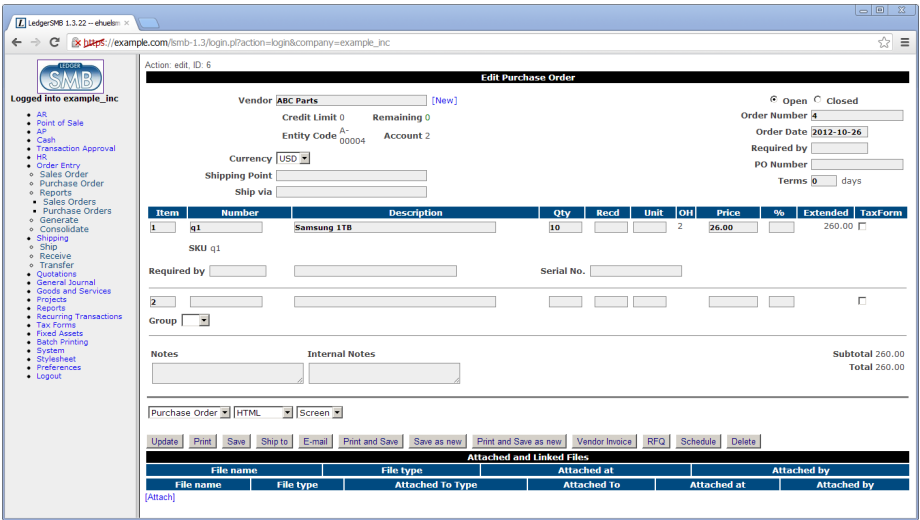


Figure 36.1: Saved purchase order

with the “Done” button. After confirmation of the receipt screen, items will be added to inventory.

LedgerSMB supports warehouse management on the Receipt screen by offering the ability to print a “Bin List”. This list contains the bin (storage) locations configured for each part.

36.3 Partial shipments or receipts

Both shipping and receiving support partial shipments and per-lot invoicing. If the vendor ships multiple lots to a single order (and charges a single invoice), you can simply call up the order and enter items into inventory on the same order. The outstanding items are automatically decremented by the amounts entered before.

To create an invoice for a received lot simply follow the process as detailed in Section 37.1 on page 191. If the lot is a partial shipment, the invoice will be for the current receipt by default.

The screenshot shows the LedgerSMB 1.3.22 web interface. The browser address bar displays `https://example.com/smb-1.3/login.pl?action=login&company=example_inc`. The sidebar on the left indicates the user is logged into `example_inc` and lists various navigation options such as AR, Point of Sale, AP, Cash, Transaction Approval, HR, Order Entry, Shipping, Ship, Receive, Transfer, Quotations, General Journal, Goods and Services, Projects, Reports, Recurring Transactions, Tax Forms, Fixed Assets, Batch Printing, System, Stylesheet, Preferences, and Logout.

The main content area is titled "Receive Merchandise" and contains the following fields and options:

- Vendor**: Text input field.
- Warehouse**: Dropdown menu.
- Order Number**: Text input field.
- PO Number**: Text input field.
- Ship via**: Text input field.
- Description**: Text input field.
- From**: Text input field.
- To**: Text input field.
- Period**: Two dropdown menus.
- Frequency**: Radio buttons for Current, Month, Quarter, and Year.
- Include in Report**: A section with checkboxes for:
 - No. ☐ ID ☐ Order Number ☒ Date ☐ PO Number
 - Required by ☒ Vendor ☒ Number ☐ Ship via ☐ Amount
 - Tax ☐ Total ☒ Currency
 - Subtotal ☐

A "Continue" button is located at the bottom left of the form.

Figure 36.2: Search screen for purchase order item receipts

36.4 Handling returns

36.5 Transferring between warehouses

36.6 Inventory reporting

36.6.1 Inventory status report

36.6.2 Inventory activity report

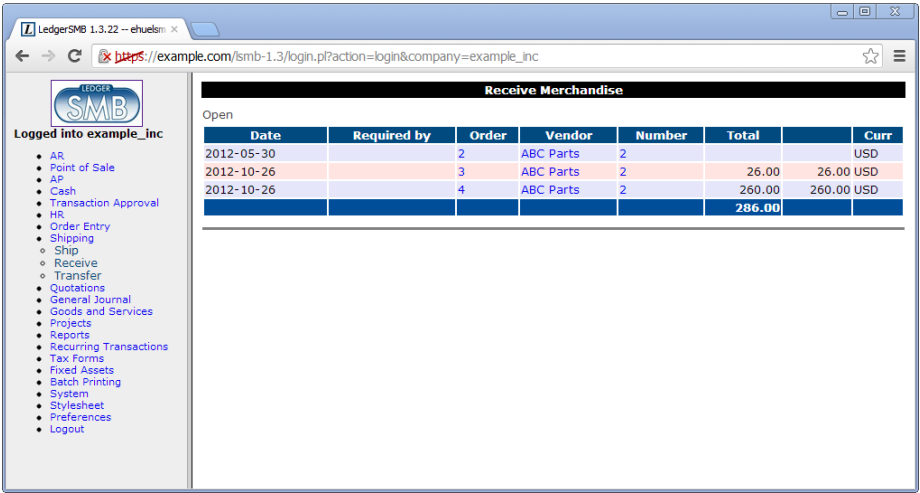


Figure 36.3: Search results screen for purchase order item receipts

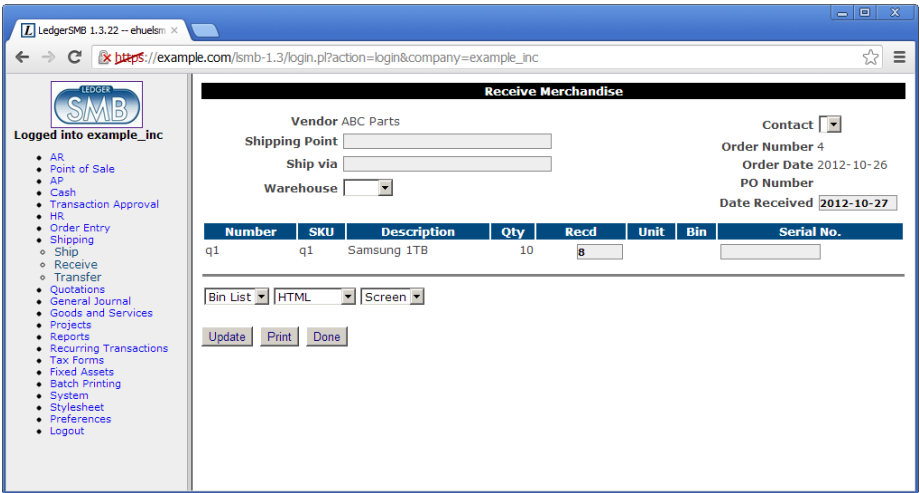


Figure 36.4: Purchase order receipt screen

Chapter 37

Sales and vendor invoice handling

Invoices can come from multiple sources. When the quotation and order management functionalities in LedgerSMB aren't used, they will usually be entered manually. This work flow is covered in Section 37.2. When order management *is* being used they mostly originate from orders which is covered in Section 37.1.

37.1 Creating invoices from orders

37.2 Creating new invoices

When a business decides not to use the order management as per the previous chapter it may find itself in need to manually enter invoices. But even if it does use order management, it may be necessary to enter an invoice directly.

When creating a transaction to record that the company owes another entity (a vendor invoice) or that it has outstanding receivables, LedgerSMB offers two options:

1. Invoices
2. Transactions

Transactions have very limited functionality: they allow a user to enter a debt owed or owned into the AR and AP subsystems. They also require the user to think how the other side of the transaction should be registered; i.e. which cost account the AP transaction should be posted against, or which income account the AR transaction should be posted against. If there are sales taxes applicable, the user is required to manually calculate and enter them.

Invoices offer a much more clever set of functionalities. First of all, it allows the user to create a document to be sent to the vendor or **customer**. Second, invoices take advantage of parts and services to automate calculation of sales taxes. Third, invoices update inventory for items held in stock (parts, assemblies). Transactions offer none of this.

37.2.1 Invoices

As mentioned in the previous paragraph, invoices can perform automatic sales tax calculations, maintain inventory and post income (or expense) to the correct GL accounts.

To be able to do so, they need the items on the invoice to be correctly configured. See Chapter 25 starting at page 149 how to set up products and services.

37.2.2 Transactions

Transactions serve an important purpose not handled by invoices: payroll calculations are often too difficult to fit in the simple “amount times price” model offered by invoices. In order to still be able to track which “vendor” was paid which amount such payment obligations can be recorded in the AP subsystem with a transaction.

Likewise it’s often more hassle than it’s worth to create the parts and services required to correctly calculate the utility bill. In such cases the transaction (possibly with a linked document as supporting evidence) offers good per-vendor traceable history records.

37.2.3 Email Invoices

Posted invoices can be emailed from the Edit or Add Sales Invoice screens.

When emailing an invoice, the ‘from’ email address will only auto fill when the email address is set in `System → Defaults → Default Email From`.

The ‘to’ email address will only auto fill if a billing email address is set in the customer. Select the company. Then go to `Contact → Search → Contact Info` tab. Set ‘Attach to’ to ‘Credit Account’ and ‘Type’ to ‘Billing Email’, add the email address, then click ‘Save Contact’.

37.3 Recurring invoices

37.4 Invalidating invoices

Sometimes, it’s necessary to invalidate an invoice. When an invoice has been posted, this also means derived administrations have been updated, such as

inventory for the items on the invoice.

To undo the effects of an invoice, i.e. to reduce the amount outstanding with a **customer**, use the VOID button on the invoice screen as shown in @@@figref . This creates a new invoice by the same number as the original, except that the new invoice has a suffix -VOID.

Unfortunately, in LedgerSMB 1.3 - the earlier versions - voiding an invoice did not automatically close the original and voiding invoices. To close both invoices from the open invoice overview, use the cash receipt process as described in Section 42.2 on page 207 to make a zero amount payment.

37.5 Correcting and deleting invoices

There's only one way to persist an invoice in LedgerSMB: posting it. This means the invoice becomes part of the accounting information. One of the primary properties of an accounting system is to record full audit trails and help enforce internal controls as detailed in Section 2.2 on page 32. Because of that fact there's no way to delete or edit invoices after they have been posted ¹.

The only way to "undo" an invoice is by voiding it. This is important for several reasons:

1. Invoices can't be deleted (because they're accounting data)
2. Invoices pose a claim on the assets of a **customer**
3. @@@ others?

Specifically item 2 is important: when you sent the invoice to your **customer**, you effectively sent them a claim. When you decide to refrain from pursuing that claim, you should notify them of that fact so they have the documentation to update their accounting system to reflect that fact: they need your documentation to void their vendor invoice, instead of paying it.

For the same reason it's ill-advised (and no longer supported) to edit invoices: when a **customer** has multiple invoices, each stating a different amount, all using the same invoice number; how is that **customer** supposed to document (verifiably) that the claim has been settled satisfactorily by paying the one he did?

See the Remarks section at the end of this chapter for details on how to handle the draft invoice requirement.

¹LedgerSMB currently does not support saving an invoice without posting it. This functionality is on the roadmap for addition when the AR/AP functionality is being rewritten - currently 1.5 or 1.6.

37.6 Handling invoice disputes

When an invoice has been sent to a **customer** it could happen that the customer disagrees with the invoice due to incorrect amounts, discounts, products, etc.

Depending on whether the customer has already paid the invoice there are several options to handle this situation:

@@@ AR Invoices have nothing to do with it! They parallel AR Transactions!!

Credit invoices

The **customer** has not paid the invoice yet

Credit notes

The customer has paid the invoice and allows offsetting against other (possibly future) invoices

AR vouchers

The customer has paid the invoice and demands repayment

Note that the naming in the listing above applies to AR items but could equally apply to AP items by replacing the word “Credit” by “Debit” and “AR” by “AP”.

37.6.1 Credit and debit invoices

@@@ produce an accounting document to exchange with the customer/vendor

@@@ restock credited parts

@@@ reverse taxes

37.6.2 Credit and debit notes

@@@ generic transaction, not related to orders, inventory or anything else

@@@ amount taken out of an income account, which is to be selected

@@@ i.e. based on the account on which the original income was posted.

37.6.3 AR and AP Vouchers

@@@ like a credit note or debit note, meant to initiate

@@@ a repayment to the customer

@@@ exists as a batch workflow only – to support separation of duties

37.7 Remarks

Why can't I send a draft invoice to a [customer](#) and edit it to match their expectations?

You can't edit invoices any more in LedgerSMB 1.3 because it breaks the audit trail in financial accounting. But in fact there's functionality available which is meant exactly for this purpose. It's called "Sales order" and its details are in Chapter [35](#) starting at page [185](#). Sales orders can be converted - upon [customer](#) approval - into an invoice with a click of a button.

Chapter 38

Shop sales

38.1 Opening and closing the cash register

38.2 Shop sales invoices

Chapter 39

Manufacturing management

39.1 Producing sales orders

39.1.1 Work orders

@@ section misnomer "Work orders" is not a workflow or workflow step

Chapter 40

Managing accounts receivable and payable

40.1 Creating generic AR/AP items

40.2 Handling refunds, over-payments and advances

- this bit is about credit notes and debit notes

40.3 Handling returns

- this bit is about credit (sales) and debit (vendor) invoices

Chapter 41

Credit risk management

41.1 Introduction

A company runs credit risk when it gives credit: it runs the risk of the creditor not paying off its debts. LedgerSMB features two ways to manage the risks involved:

1. Limit management
2. Arrears management

The former tries to limit the risk involved by making sure no **customer** receives more credit than a certain limit while the latter tries to make sure any over due payments get cashed.

41.2 Limit management

Limit management should prevent a company on one hand from delivering too much to its customers at once and on the other from taking (and delivering) new orders to customers with too high amount of unpaid invoices.

@@@ Where to set up limits

@@@ How to monitor limits

41.3 Granting payment terms

41.4 Managing arrears

As mentioned before, the process of managing arrears is directed toward detecting arrears positions with customers early and taking appropriate action.

41.4.1 Monitoring AR aging

In order to find out about over due invoices, a company should run the AR aging report available under Accounts Receivable → Reports → AR Aging. The initial screen presents parameters for the aging report to be generated.

@@@ discuss parameters

This report shows customers and their outstanding invoices categorized as:

Current

Invoices not over due

30 Invoice amounts over due by 30 days or less

60 Invoice amounts over due by 60 days or less (but more than 30)

90 Invoice amounts over due by 90 days or less (but more than 60)

@@@ Printing / mailing aging reports to customers

41.4.2 Tracking invoice history

After an invoice becomes over due a process will be started to remind the **customer** of the outstanding amount requiring payment.

In order to keep records of actions taken to chase customer payment, the invoice screen has an "Internal Notes" field which can be edited after the invoice has been posted.

In order to save any edits to that field, hit the "Save Info" button.

Note that the "Save Info" button also saves any changes to the "Tax-Form" column or rather, any information that's not accounting information (posted to the books and thereby fixed) nor information which appears on the invoice - which also should remain unedited in order to be able to generate an exact copy at a later date.

41.4.3 Special treatment of invoices

There may be good reasons to treat some over due invoices differently. E.g. in case payment arrangements have been made with the **customer** and further standard arrears management would not be appropriate any more.

In this case, you can put an invoice "On Hold". The opposite of being on hold is being active. The AR Aging report allows selection of all invoices, only active invoices (those not on hold) or only invoices on hold.

41.5 Interest on arrears

41.6 Allowance for doubtful accounts

41.7 Writing off bad debt

41.7.1 Direct write-off

41.7.2 Allowed-for write-off

Chapter 42

Receipts and payment processing

42.1 Introduction

Receipt (incoming payments for sales invoices) and payments (outgoing payments for purchase invoices) use the same process. This chapter describes the steps using receipts only for brevity but be equally applied to payments - except when explicitly stated.

42.2 Single receipts or payments

LedgerSMB provides two ways to process receipts (and payments). One for single transactions, the other for batches. The next section discusses the steps to do batch processing.

To record an amount received from a **customer** as an invoice payment, go through the menus `Cash & Banking` → `Receipt` and fill out the search criteria to find the customer from whom the payment has been received. After clicking “Continue” the application lists all matching customers.

Remark The customers listed may not have open invoices. The list only serves to select the **customer** the user is looking for. To find customers with outstanding balances, please refer to Section [41.4.1](#) on page [204](#).

After selecting a **customer**, the cash entry screen as depicted in [figure reference](#) is shown. This screen looks the same for single receipts or payments. This screen consists of three parts. The upper block lists customer details on the left and transaction details on the right. The middle block lists any (partially) unpaid invoices. The lower block handles over payments.

Note that the cash transaction amount isn't being entered explicitly in this screen: the total is shown on the right of last line above the buttons.

The receipt entry screen can be used to register payment of any outstanding invoices. This is the normal scenario. However, if there are no outstanding invoices, or the amount paid is too high, the transaction should be (partially) recorded as over payment. Over payments can be used to pay off invoices at a later date.

@@@ Discuss the single receipt screen

42.3 Batch receipts and payments

42.4 Check payments

42.5 Using over payments

Currently, it is not recommended to use the over payment functionality.

The **over payment** functionality in LedgerSMB is used for handling customer or vendor deposits, prepayments, advances, etc. Essentially any activity that results in paying too much or in advance.

If over payments are used, the "stored" amounts for later use are not tracked in the normal places. Once over payments have been applied to AR/AP transactions they will be visible, and the totals will show up in balance sheets under the correct asset/liability, but running AR/AP reports for individual vendors/customers will not flag that there are over payments outstanding. The only place to find them is in the specific "Use [AR] Over payments" screen.

To work around the current limitations use transactions and debit/credit notes as follows.

For AR:

1. Create an AR transaction for the overpaid amount
2. Create a credit note for the same overpaid amount
3. Receive the extra payment with the AR transaction, and the credit note shows up on the payment screen where you can apply it to another invoice

For AP:

1. Create an AP transaction for the overpaid amount
2. Create a debit note for the same overpaid amount
3. Receive the extra payment with the AP transaction, and the debit note shows up on the receipt screen where you can apply it to another invoice

Note that this approach triggers two movements on the tax account, in case taxes apply to the customer or vendor. One movement to charge taxes and one to nullify the previous charge. Additionally one should be careful about invoice numbering when applying this procedure: the two transactions are synthetic (not real invoices) and should probably have invoice numbers outside of the regular numbering range (e.g. with a specific prefix or suffix such as 'OVP-') to prevent the need to report on these invoices to the tax authorities – especially if your jurisdiction requires gapless invoice numbering.

42.6 Receipt and payment reversal

42.7 Over payment reversal

Currently, it is not recommended to use the over payment functionality.

Reversal of over payments is a weak spot in LedgerSMB 1.3: there's no way to reverse or "undo" an over payment incorrectly entered. By consequence this section describes the workaround that's required to achieve the same effect.

This workaround needs an account which can be used to temporarily book income on.

Please note that the income will immediately be reversed, so technically any account can be used. To be able to assert that the entire process has been executed correctly, it's advisable to create a separate account, however, since it can be checked to be zero at the end.

With the prerequisites in place, you should execute the following steps - assuming the amount of the over payment needs to be placed back into a cash account.

1. Create an AR transaction for the company the over payment has been entered on
2. Add a single line to the transaction, with the selected account
3. Put the over payment amount to be canceled out in the Amount field for the line
4. Save and post the transaction
5. Pay the transaction from the over payment
6. Create a "General Journal" transaction debiting the income account and crediting the cash account the over payment was entered from

Steps 1 through 4 prepare the Accounts Receivable module with a transaction which allows the over payment to be used. After step 5, the over payment has been cleared, but the amount is in the wrong place, since it sits in the income account instead of the cash account, which is what step 6 corrects.

The side-effect from this workaround is an AR transaction registered against a **customer** which can't be reversed: doing so, would result in the reversed amount ending up in the AR summary account. Using a dummy company isn't an option, because over payments are registered to a specific **customer**. An over payment can only be used to clear open items on that specific **customer**.

Note that the above procedure applies to an AR over payment. However, the same steps apply to AP over payments, replacing "customer" with "vendor", "AR" with "AP", "Income" with "Expense" and "debit" with "credit".

42.8 Receipts and payments in foreign currencies

LedgerSMB supports foreign currency in both Accounts Receivable (AR) and Accounts Payable (AP) by storing the transaction currency, the exchange rate, and the resulting base (functional) currency amounts used in the general ledger.

LedgerSMB assumes:

- A company has one functional base currency.
- Customers and vendors may transact in different currencies.
- The foreign currency amount remains associated with the document so open invoices can be settled in their original currency.
- Each AR or AP document records:
 - the transaction currency (EUR, GBP, CAD, etc.)
 - the exchange rate used
 - base-currency values posted to the GL

42.8.1 Exchange rates

Exchange rates are maintained separately from invoices.

Current releases support:

- multiple currencies
- multiple exchange-rate classes
- validity periods for rates
- multiple rates on the same day (introduced in 1.7)

Beginning with 1.7, each transaction carries its own exchange rate instead of relying on a single daily buy/sell rate. This makes it possible to record the actual bank or negotiated rate used on an individual invoice or payment.

42.8.2 Exchange rate classes

LedgerSMB also supports exchange-rate classes, allowing organizations to maintain different sets of rates for purposes such as:

- operational transaction entry
- revaluation
- translation
- other accounting requirements

The default installation includes a standard exchange-rate class, and additional classes can be defined if needed.

42.8.3 User workflow

For AR and AP, LedgerSMB's foreign currency workflow is:

1. Enter the invoice in the customer's or vendor's currency.
2. Record the exchange rate for that specific transaction.
3. Post the GL in the company's functional currency.
4. Retain the original foreign amount on the open document.
5. When payment is recorded, compare the original and settlement exchange rates.
6. Automatically recognize any realized foreign exchange gain or loss.
7. Support different exchange rates for different transactions on the same day, reflecting actual bank or negotiated rates.

This design closely follows standard accounting practice for foreign-currency AR and AP while giving users the flexibility to use transaction-specific exchange rates instead of a single daily rate.

Chapter 43

Accounting

43.1 Separation of duties: Transaction approval

See Section 18.4.3 on page 117.

43.2 Entering non sales transactions

43.2.1 Corrections

43.2.2 Transfer of money between bank accounts

43.2.3 Wages

43.2.4 Entering other general accounting documents

Even though the application handles many general ledger postings as consequences from work flows elsewhere in the system - thus not requiring separate postings - sometimes the need may occur to create manual postings not resulting from AR or AP transactions or till and inventory adjustments.

One example of a case like that is the calculation, and posting of corporate taxes presumably at the end of each accounting period but at least at the end of the book year.

43.3 Bank reconciliation

43.3.1 Importing bank transactions - OFX

@@@ TODO - Quick capture of chat info, needs to be revised and maybe moved to customization chapter.

43.3.2 Importing bank transactions - CAMT.053

@@@ TODO - Quick capture of chat info, needs to be revised and maybe moved to customization chapter.

43.3.3 Importing bank transactions - CSV

Cash & Banking → Reconciliation

@@@ TODO - This subsection needs to be revised and maybe moved to customization chapter as it is no longer accurate.

Custom formats can be handled by creating an import file.

The file should reside in `LedgerSMB/Reconciliation/CSV/Formats/`

A perl sub must exist which is named:

```
parse_<name-of-your-company-database>_<id-of-account-to-reconcile>
```

With a company database name of `testdb` and an account id of 255 the parse code might look like the following:

```
sub parse_testdb_255 {
    parse_n(@_);
}
```

default format is:

```
cleared_date,amount,scn,type
2021-09-01,25.25,224435,CHECK
2021-09-01,-30.31,44351,CHECK
2021-09-02,100,,
2021-09-02,-425,3321435,
```

scn = Source Coding Number

43.4 Period closing

Period closing is a concept used by accountants to ensure that accounting data is not allowed to change without proper authority. This helps protect against

errors when entering new transactions and fraud. The best practice is to close an accounting period, usually monthly, as soon as it is reconciled.

When a period is closed no transactions can be entered prior to the end date of the period, including previous periods.

To open or close a period the user must have be assigned the `yearend_run` role, then they can navigate to the screen "Period closing dates" using `General Journal → Year End`. For an example, see Figure 43.1 on page 215.

In the bottom part of the screen the user can add a date in the input field and hit the corresponding button `Close Period` to close all periods prior to that date. Note that periods are normally closed one-by-one in successive order.

Re-opening a period also opens all succeeding periods.

Period closing dates

Post year closing

Yearend

Reference

Description

Retained Earnings

yyyy-mm-dd

Yearend

3590--Retained Earnings - prior years

Post Yearend

Date	Is Year-end	
yyyy-mm-dd	no	Close Period
2023-12-31	yes	Re-open Period

Figure 43.1: Period and Year End Closing

Note that due to a design limitation prior to the release of LedgerSMB 1.7, invoices in foreign currencies¹ can't be reversed on dates other than their original posting date. That is: they can, but their reversal will result in P&L and balance sheet effects which are not desirable. Since period closing disables posting before a certain date this functionality may have negative side effects in versions prior to 1.7.

43.5 Year-end processing

Year end closing is a concept which prepares the accounting books for the next **accounting year**. There's no inherent requirement for this process to be run at least once a year. This procedure closes the books up to the given date. If the first book year of the company spans more than a year, then the closing will span more than a given year. If it has been more than a year since the last year end close then this closing will span all data since the last year end close.

This procedure freezes the accounting data as described in Section 43.4 on page 214. Additionally it clears out all of the profit and loss accounts by setting

¹This pertains exclusively to invoices and transactions in foreign currencies and has no effect in case of invoices and transactions in the default currency.

the balances to zero and posting their balance to the retained earnings account.

Some businesses prefer to create a retained earnings account for each book year they close. LedgerSMB supports that use-case by allowing the user to select which retained earnings account the balance should be posted to.

In order to run year end closing the user must have the role `yearend_run`.

In order to re-open a year end closing the user must have the role `yearend_reopen`.

Typically the role to re-open an accounting period is reserved for upper management as any changes to previous periods may have tax, legal, or other government reporting impacts.

The year procedure is run by filling in the data in the top part of the "Period closing dates" view, then clicking the button `Post Yearend`. For example, see Figure 43.1 on page 215.

Be careful here, because the UI default value is the first G/L account available for this procedure, not necessary the correct retained earnings account.

Some companies want may to include additional transactions in the year end close. Direct support for this use case isn't currently available. Use a manual G/L transactions to accomplish this.

43.6 Fixed asset accounting

43.7 Reporting

43.7.1 Tax reporting

- 1099 (cash based)
- EU VAT (accrual based)

43.7.2 Tax reporting using tax forms

Collecting taxes in tax forms

Linking companies to tax forms

Running tax form reports

43.7.3 Income statement

43.7.4 Balance sheet

43.7.5 Trial balance

Chapter 44

Budgets

Currently, it is not recommended to use the budget functionality at Budgets
→ Add Budget or Budgets → Search.

This code has not been updating recently and needs it.

Part VI

General Operation

Chapter 45

User Interface

Description of the general operation of LedgerSMB that appears across multiple views in the application and that are not described elsewhere.

45.1 Reports

At the bottom of most report views there is a '[permalink]'. The idea of a [permalink](#) is that you get a link which you can share with your colleagues and when inserted into the browser, returns exactly the same report (or at least a search with exactly the same parameters as you used). To copy a [permalink](#) the user can right-click and select "Copy link" from the drop down menu and pass it to a colleague by pasting into an email or other messaging App.

Part VII

Customization

Chapter 46

Overview

46.1 Introduction

Customization is about adding or changing behavior of an application in ways that were not foreseen when it was written. This differs from configuration, because the latter can be used to select between planned behaviors. Since the functionality or behavior to be added wasn't foreseen, this means that customization implies application code being added or changed. Especially changing existing behavior has proven problematic (in the IT industry at large) in terms of being able to later upgrade the underlying software to newer releases. This problem can be mitigated by creation of specific "extension points": places in the software which are designed to expand or replace existing behaviors.

LedgerSMB has several such "extension points":

- Workflow configuration
- Dependency-injection based configuration:
 - Document formatters
 - Bank statement importers
 - HTTP request handlers/interceptors
 - Outgoing mail transport
- Sales/VAT tax calculation

Each bank has its own format for bank statement exports, making it impossible to develop import functionality that's generically usable. Some default importers come with LedgerSMB - mostly as example code for those developing their own. The dependency injection-based configuration supports adding these user-specific developments without changing any of the files that come in the standard distribution.

Another example is changing “Workflow configuration”, by which the life cycle of invoices and GL transactions can be changed. This way, additional steps can be introduced into the life cycle, or steps can be combined (from a user perspective) by automatically executing steps after another has been triggered manually.

Customization usually means additional or modified Perl code needs to be loaded.

@@@TODO: describe where to store this new or overriding code...

46.2 Workflow configuration

LedgerSMB uses the concept of workflows to manage object life cycle. An invoice, for example, may be *saved*, *posted* or *reversed*. These are states in the life cycle of the invoice. Each state may have associated actions. E.g. in case of the invoice there may be a *Post* action when it's *saved* or to *E-mail* it when it's *posted*. Actions may cause the document to change state: if the *Post* action is executed on a *saved* invoice, its new state will be *posted*. E-mailing an invoice does not change its state: it remains *posted*. Instead, a new workflow is created which manages state of the e-mail.

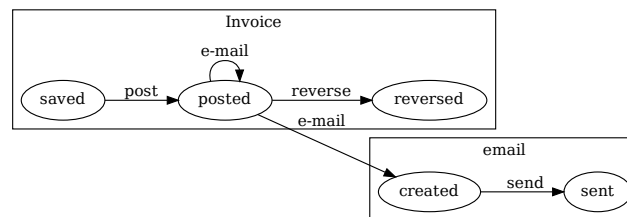


Figure 46.1: Workflow triggered from another workflow

Actions belonging to a state may (or may not) be available. This is determined by one or more conditions. Examples are “is the posting date of the transaction in a closed period” and “is the current user allowed to post transactions”. When the condition evaluates to *false*, the action is not available.

Workflows are configurable. Configurations list the following aspects:

1. A list of states
2. A list of allowable actions per state
3. A list of conditions per action
4. A mapping of actions to Perl code

5. A mapping of conditions to Perl code

With these ingredients, the software's behavior can be changed. E.g. by adding a new state and action in the invoice, an extra reviewing pairs of eyes can be added to the posting process. By adding a condition on this action, it can be made required for invoices over 100.000 USD (but not for other invoices).

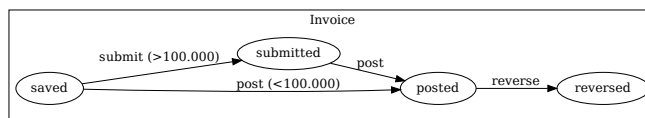


Figure 46.2: Conditional workflow actions

By changing association of the `save` action with its code, the application will act differently when saving the invoice. Customizations may associate new behaviors with existing actions or create new ones entirely. The behavior does not need to be known by the application developers, as the association may point to code added to the local installation. This way, a `remind` action can be added to posted invoices, triggering a text (SMS) message to the customer.

Workflows are stored in the `./workflows/` directory. Each workflow consists of several files, prefixed with the name of the workflow (e.g. `ar-ap.` for 'AR/AP' workflows):

ar-ap.actions.xml

(optional) names the list of available transitions in the workflow naming Perl code to be executed

ar-ap.conditions.xml

(optional) names the list of conditions to be used in the workflow naming Perl code to evaluate

ar-ap.persisters.xml

(optional) defines the Perl mechanism to load and store workflow instances

ar-ap.workflow.xml

(required) defines the life cycle by combining states with the actions and conditions listed in the respective configuration files. Each state lists the allowable actions with the conditions to enable the action and a resulting state.

Customized versions of workflow definition files must be stored in `./customized.workflows/`¹. Files stored in this directory will be used to override those in the standard location: the file

¹See Section 17.4.1 on page 102 on how to configure these paths

`./custom_workflows/ar-ap.workflow.xml` will replace the built-in workflow from `./workflows/ar-ap.workflow.xml` used for invoices and AR/AP transactions. No changes should be made directly to the standard files because of the risk that those files will be overwritten on upgrade – losing the changes.

Next to the per-workflow files, there is a generic definition file for actions, conditions and persisters. Elements defined in these files can be used in all workflows; those elements defined in the per-workflow files can only be used in the specific workflow.

The engine driving workflow state management is the Workflow² library. See its documentation on MetaCPAN as well as the built-in workflows as examples of how to define one's own.

46.2.1 Workflow observers

Particularly interesting is the concept of observers: this functionality allows the application to be extended based a specific action occurring without changing the life cycle. An example could be to create a PayPal payment request when an invoice is posted: an observer could be “listening” to “AR/AP” workflow events which at some point will flag the state of the invoice becoming `posted`.

Notes on Quote, Order, AR, AP and GL workflow customization

The actions in the workflows `order-quote`, `ar-ap` and `gl` are hard coded³. This means that the list of actions is restricted to those defined in the files `order-quote.actions.xml`, `ar-ap.actions.xml` and `gl.actions.xml`. It's not necessary to use all actions in the workflow, but additional ones cannot be created – contrary to what is in the Workflow documentation.

The `class` attribute for most actions in these workflows is set to the value `LedgerSMB::Workflow::Action::Null`. This means it does nothing. Rather, it doesn't do anything *in addition to* what the action already does in its hard coded implementation. Setting the action's class attribute to some other value allows the developer to add behavior on top of what's hard coded.

46.3 Document formatters

Document formatters transform templates into output documents; e.g. an HTML invoice template can be transformed into an HTML file with the details of the invoice filled out. LedgerSMB comes with a number of built-in formatters. Additional ones could be implemented to transform HTML documents to PDF (replacing the traditional $\text{\LaTeX}$ to PDF) or the creation of eInvoices in the format of the local jurisdiction.

²<https://metacpan.org/pod/Workflow>

³This is not a permanent situation, but rather a consequence of these workflows being used in old code

The generation of output entails these steps (preceded by the process to find a suitable document formatter):

- Loading the input template from the database
- Preprocessing the input variables using the formatter's `encode` function
- Initializing the template expansion using the formatter's `setup` function
- Expanding the template using `Template::Toolkit`
- Post-processing the expanded output using the formatter's `postprocess`

A custom formatter needs to implement the protocol as documented at <https://docs.ledgersmb.org/perl-api/1.11.0/LedgerSMB/Template.pm.html#FORMATS>.

46.4 Bank statement importers

LedgerSMB comes with three bank statement importers built-in:

CAMT.053 The **ISO-20022** “Cash Management” message 053 (Bank-to-Customer account statement)

OFX A standard to exchange data describing financial transactions

CSV File based using **Comma Separated Values (CSV)**

The bank statement importer is a Perl class implementing the `LedgerSMB::Reconciliation::Format` role; it parses a file with bank transactions, extracting the fields expected by the bank reconciliation procedure:

amount The amount involved in the transaction; negative if the bank reports a credit amount, positive when it's a debit

date The date on which the amount contributed to the bank account balance

source An identifier by which the bank identifies the transaction; to be used for matching with the “Source” field in the bank ledger

type Information clarifying the transaction; e.g. Cash, ACH, etc.

The CSV parser has many options to accommodate the tremendous variance that can be observed in CSV files. When using a custom bank reconciliation procedure *or* when the bank makes a more exotic format available, customization of the bank statement importer may be necessary.

46.5 HTTP request interceptors

@@@TODO: describe how to add `Plack::Middleware` instances in the request processing

46.6 Outgoing mail transport

The library used for sending e-mail (`Email::Sender`) includes a wide variety of transports that can be used to configure the library to (pretend to) send e-mail messages. For a list of available transports, see <https://metacpan.org/dist/Email-Sender>.

Should a different delivery method be required even considering the list of available transports, this can be achieved by implementing a custom `Email::Sender::Transport`.

46.7 Sales/VAT tax calculation

Tax calculations performed for VAT and/or sales taxes have been abstracted into so-called “tax modules”. An example of a custom tax module can be found in the directory `utils/wa-tax-service` in the source repository.

Before tax modules can be used, they need to be registered in the `taxmodule` table.

Important: Please note that there is a tight integration between the database “pointing” to the tax module and the web server which expects to be able to *load* this Perl code to perform the prescribed calculations. If for some reason the application loses the ability to load the Perl module, this will break the ability to create or view invoices.

Chapter 47

Batch data import methods

47.1 Custom bank statement import

47.2 Chart of Accounts import

This function allows bulk import of an entire chart of accounts using a single [CSV](#) file.

The first line of the file contains the headers of the columns. The import routine expects the columns as presented in (table name).

accno

Number of the account or header

desc Description for the account or header

charttype

"H" for heading record

"A" for account record

category

"I" for Income

"E" for Expense

"A" for Asset

"L" for Liability

contra

"0" not a [contra](#) account

"1" is a [contra](#) account

tax

"0" not a tax account

"1" is a tax account.

link A colon (':') separated list of account check-mark values (links) as described in Section [18.3.3](#) on page [112](#), or "AR", "AP" or "IC" for the specified summary accounts

heading

“accno” of the heading to group the account (or heading) under

gifi **GIFI** account number

All lines after the first one are considered to be data lines.

Chapter 48

Add-ons and plug-ins

Chapter 49

Web service APIs

LedgerSMB comes with a web service **API**. The version number assigned follows the rules of **semantic versioning**.

The current major version is 0 (zero), meaning that it's not considered to have stabilized yet. The main reason is that not all functions have been ironed out yet: filtering, sorting and pagination are to be specified and implemented.

The API is hosted on `erp/api/v0`, on the same root as `login.pl`.

That is to say that if LedgerSMB's login screen is hosted at

`https://example.org/login.pl`

then the API can be found at

`https://example.org/erp/api/v0`.

The definitive API definition is in the source code located at `doc/openapi/API.yaml`.

There is also a web reference version at

<https://docs.ledgersmb.org/openapi/1.13.0/LedgerSMB-api.html>

Note that the above URL is LedgerSMB version dependent. There may be newer versions available and you should refer to the version that matches your installation.

Part VIII

Workflows

Chapter 50

Overview

Workflows control certain LedgerSMB internal processes and can be created and edited by technical staff using the XML files in the directory `LedgerSMB/workflows`.

@@@TODO Need more info.

Part IX

Appendices

Appendix A

Differences between version 1.2 and 1.3

A.1 Users

The way users are defined and used differs greatly between LedgerSMB 1.3 and older versions. In version 1.3 user access to the database is enforced by the database itself. This means that users logging in to the LedgerSMB web application are in reality logging into the PostgreSQL database. In older versions, the web app would verify the user's credentials (using a common database connection used for all users).

The difference between these approaches is that security is no longer (solely) maintained by the web application - with all inherent risks. Instead, the database now plays an important role as well. The effect is that the LedgerSMB team now leverages the experience of the PostgreSQL community - a highly respected community well known for its security focus - to make sure your data stays secure.

This structure also enables LedgerSMB 1.3 to offer separation of duties and authorizations throughout the application without being required to do a full rewrite of the application.

It's this shift in paradigm that makes it impossible to meaningfully migrate users from older LedgerSMB and SQL-Ledger versions to LedgerSMB 1.3.

Appendix B

Why deleting invoices is not supported

B.1 Invoices are debt claims

B.2 Cost Of Goods Sold become ambiguous

From a mail by Chris Travers on the LedgerSMB mailing list:

Suppose I buy 10 widgets for \$1 each and then 1 more widget for \$10 each. My FIFO cost queue looks like this:

\$1 \$1 \$1 \$1 \$1 \$1 \$1 \$1 \$1 \$1 \$10

My inventory account shows \$20 in debits and I have credited my AP account to compensate.

I then sell the 11 parts to 11 different people.

The first 10 invoice show a \$1 credit against inventory and a \$1 debit against COGS

The 11th invoice shows a \$10 credit against inventory and a \$10 credit against COGS

and now inventory is down to 0.

Now the 5th customer invoice turns out to be in error. We never shipped this one. The customer never ordered it. it was a data entry error. Translation, we now have one in stock.

If we void the invoice properly, we will reverse the last sale, and put \$10 back in inventory.

If we delete the invoice, we will just remove the \$1 removed. But we don't really know which one was sold and so we de-allocate the \$10 sale.

So now our books are \$9 off of what they should be. They show a balance of \$1 in inventory, and \$19 in cogs. They should show \$10 in each. The worse is still to come however.

Now we sell the item we had in stock. This brings our (empty!) inventory value to -\$9 and our COGS value to \$29. Our books are still \$9 off and we now have impossible, nonsensical values. Delete and re-enter a few more invoices and you can inflate the COGS as far as you'd like. This doesn't work.

Worse, this can't be fixed. You can't make a deletion behave just like a reversal and still keep your books transparent auditability-wise. Even if it could be fixed mathematically (which it can't), there isn't any agreement as to what the proper behavior should be (except 'don't do that'). So it isn't possible to support the workflow "properly" because "properly" can't be defined.

So unless someone can show that the above issues are incorrect, I don't see how we can support deleting invoices after they are posted to the books.

The alternative is the draft/voucher system which is supported in 1.3 for all non-inventory transactions and will be supported for inventory transactions in 1.4. In this system, in the paper world, the clerk fills out a piece of paper with the information that will be entered as an invoice, and this is eventually gets entered and checked by someone else. Both papers are kept in paper systems for reconciliation purposes but typically we tend to assume they are the same (this may be changing and we may be keeping both copies if they are changed in the future). In this model, the voucher is not an invoice. It is simply a piece of paper that represents what will be on the invoice. It is subject to review, and may ultimately be denied.

So in this system we do **not** calculate extrinsic financial movements for documents until they post. This includes COGS. If a draft invoice or invoice voucher is deleted before it is approved it has none of the problems above. Once approved though, it is a part of the permanent record. This guards against data entry errors because a second person can review the data before it is posted (either in bulk or individually). Additionally this guards against theft by ensuring that a single individual cannot individually enter everything necessary to cover for theft, etc.

Appendix C

Installation

LedgerSMB can be installed on almost any Linux host including Raspberry Pi. The project uses Debian for its docker images and [Continuous Integration \(CI\)](#) testing.

The following installation methods are available.

C.1 Quick preview using Docker

Quick preview using a Docker container is the recommended way to quickly run and evaluate LedgerSMB. This method is not suitable for internet facing installations.

May require an understanding of Docker container images.

<https://www.docker.com>

The authoritative installation source are located at

<https://github.com/ledgersmb/LedgerSMB/blob/master/README.md>

C.2 Production using Docker

Using a Docker production container is a more secure installation suitable as a base for internet facing use.

May require an understanding of Docker container images which is available at

<https://www.docker.com>.

The authoritative installation instructions are located in the

[Github ledgersmb-docker Repository README](#).

The user should review all security settings before deploying.

C.3 Native installation

A native installation is suitable for previewing, production and testing, but without the complexity of containers.

With the recent creation of the installation script, this installation method is very easy to use. The native install is tested on Debian, Fedora and OpenSUSE.

The user should review all security settings before deploying in an internet facing installation.

The authoritative install instructions are located at

<https://get.ledgersmb.org>.

Any problems found during installation should have Github issues created at

<https://github.com/ledgersmb/ledgersmb-installer>.

To verify which Linux systems are tested by the project see

[Github ledgersmb-installer Repository](#) [.github/workflows/ci.yaml](#).

C.4 Manual installation

Manual installation is complex but offers the most flexibility for the experienced installer. This installation method is not recommended for anyone that does not have Linux web server, database and security experience.

See the following for the recommended manual installation instructions based on LedgerSMB versions. For example, the URL ending in "113" applies to LedgerSMB version 1.13.

<https://ledgersmb.org/content/installing-ledgersmb-111>

<https://ledgersmb.org/content/installing-ledgersmb-112>

<https://ledgersmb.org/content/installing-ledgersmb-113>

C.5 Development using Docker

Most of the developers use a Docker development container. This installation method installs extra libraries and code specifically for development.

This installation method is not recommended unless developing code for LedgerSMB.

This installation method should be considered not secure and not to be used for internet facing installations.

The authoritative installation instructions are located at

[Github ledgersmb-dev-docker Repository](#) [README](#).

C.6 Linux distribution

Installing from a Linux distribution is not recommended in any case.

Some distributions have very old versions of LedgerSMB. These old distributions only exist to support existing users that have not yet upgraded to a more recent version of LedgerSMB.

With recent LedgerSMB improvements, the LedgerSMB code structure no longer adheres to the distribution's guidelines. The consequence is that LedgerSMB distributions are no longer updated or maintained.

Appendix D

Migration

D.1 Introduction

There are plenty of reasons to want to migrate to LedgerSMB 1.3:

1. Separation of duties
2. Security better integrated into the application
3. Better, more strict data model
4. Some important sources of user error eliminated
5. Better workflows for cash reconciliation
6. @@@ others?

Yet, while item 3 is a good reason to want to switch, it's also a reason why migration from older versions to 1.3 can be harder than earlier migrations: when the data in your older version is not consistent, it won't fit into the new data model and will need to be fixed first.

Especially if your database has a very long line of history, being migrated through lots of SQL-Ledger and LedgerSMB versions, you may want to consider asking for help from a professional party. It could save you a lot of time.

However, don't be discouraged and have a go yourself first. Just be sure to run your upgrade on a backup database: the migration process is non-destructive, but in case accounting data is involved: better safe than sorry!

Also it is worth noting that a number of automatic checks are performed on your data prior to migration, and to the extent possible, you are given an opportunity to fix those issues identified. Because these checks are pre-migration checks, they are written to your old data and will persist after backing out of a migration to 1.3.

D.2 From older LedgerSMB versions

D.3 From SQL-Ledger 2.8

D.4 From older SQL-Ledger versions

D.5 From other accounting packages

D.5.1 Overview

While accounting and ERP solution have wildly differing structures to record their data, this sections uses data with a relatively simple structure as a show case of how this problem may be dealt with.

Note that the encoding you use to transfer to the database depends on the settings used to create the PostgreSQL database with. A migration is a good moment to think about encodings an solve older encoding issues. Now would be a good moment to anticipate the requirement for accented characters and non-western alphabets: set up a UTF-8 encoded database and recode your data accordingly.

D.5.2 Migrating customers

The source system for this section uses a structure where every company has one contact person, one address, one phone number and e-mail.

In order to understand how to migrate this data structure to LedgerSMB, it's important to understand that:

1. The company from the source maps to the *Company* and *Entity* entities
2. The contact person maps to the *Entity Credit Account* entity
3. The address maps to the *Location* entity - and requires a location class: Sales, Billing or Shipping
4. The phone number, fax number and e-mail map to *Contact items*

The reason behind the separation between the *Company* and *Entity* entities is that every customer is an *Entity*, but not all entities are companies, since some entities are *Persons* - natural persons.

@@@ How to

D.5.3 Migrating parts, services, ...

D.5.4 Migrating the balance

The strategy that I used when migrating to LedgerSMB is the following:

- create customers and vendors as needed
- create at least the parts and services for which there are open AR or AP items in the closing balance
- create the open AR/AP items in the closing balance by posting them on the original opening date (this allows you to do your aging management in LSMB from day 1)
- create in addition the parts for which you have stock in the closing balance
- import your stock by posting invoices against the inventory entity
- make sure you “pay” the invoices - e.g. by paying them from equity
- draw up the incomplete balance you have so far
- calculate the delta of the partially imported balance against the closing balance
- import the delta balance on top of what you already had to make a full balance

What I did is post the transaction from the last step on the day before I wanted to start my books; say you want to start your books on 2014-01-01, then you’d post it on 2013-12-31. That way, the opening balance of your books on the starting date is exactly what you had on the closing balance of the books you left behind.

Migrating open AR and AP items

Migrating your ledger

D.6 Migrating between PostgreSQL versions

There are generally 2 ways to upgrade PostgreSQL that apply to normal LSMB installations. Both are described in the current PostgreSQL documentation at <https://www.postgresql.org/docs/current/index.html>.

1. Dump and restore - Described in more detail below. Note that the roles must be restored before database creation and the data restore.

2. Upgrade in place using `pg_upgrade` - When doing an upgrade in place, there are no specific LedgerSMB requirements. Beware, upgrade in place requires both versions of the PostgreSQL to be installed and working. It is therefore more complex.

LedgerSMB versions prior to version 1.9.16 did not support PostgreSQL 14 or higher due to a backward incompatible change in PostgreSQL 14. Best practice in this situation is to update to at least LSMB 1.10. prior to upgrading to any version of PostgreSQL 14 or higher.

The following upgrade procedure uses `lsmb_dbadmin` for the database administrative user.¹ But if the original database installation did not follow the recommended installation practice, then the `postgres` superuser may have to be used. Using the `postgres` superuser is not considered best practice.

The following upgrade procedure should not be used when upgrading a cloud provided PostgreSQL installation. In this case, the cloud provider's instructions have to be consulted to determine the proper upgrade procedure.

The upgrade procedure when using dump and restore with a local LSMB native or docker installation is:

1. Stop users from accessing the current LSMB installation and make sure all users are logged off. The instructions for this step vary depending on how LSMB was installed. Note that access to `setup.pl` will be needed to create backups.
2. Make sure the current LedgerSMB installation is using the latest version. If not, see the Section D.2 on page 252 for instructions on upgrading LSMB.
3. Make a dump of both the roles and data using `setup.pl`.
4. Inspect the backup to make sure it is complete. Ideally, this inspection would include restoring the backup to temporary LedgerSMB installation with the same versions as the existing installation. Then make sure it is complete by logging into the temporary LedgerSMB installation and inspecting reports, accounts, etc. Do not proceed unless you are sure your backups can be restored completely and accurately.
5. Create a new LedgerSMB installation with the new version PostgreSQL. The instructions for a new install are at <https://ledgersmb.org/installation>.
6. The next 3 restore steps, using `psql`, came from <https://ledgersmb.org/faq/how-do-i-restore-my-data>, which these instructions supersede. Other restore tools might work, but they have not been tested.

¹Default LedgerSMB installations do not automatically create an `lsmb_dbadmin` user. This user is created manually during the initial database installation. For an example see, <https://ledgersmb.org/index.php/content/installing-ledgersmb-111>.

7. Restore the "roles" backup. The `psql` syntax is:

```
$ psql -h [database host] -U [database admin user] < [roles
backup file]
```

For example:

```
$ psql -h localhost -U lsmb_dbadmin < lsmb-roles.sqlc
```

8. Create a new database. The database name is usually very similar to the company name. Do not create the database using `setup.pl` as that creates the company structure inside the database, which will interfere with the restore. The `psql` syntax is:

```
$ psql -h [database host] -U [database admin user] -c 'CREATE
DATABASE [new company name]'
```

For example:

```
$ psql -h localhost -U lsmb_dbadmin -c 'CREATE DATABASE
newcompany'
```

9. Restore the "data" backup. The `psql` syntax is:

```
$ psql -h [database host] -U [database admin user] [new
company name] < [company backup file]
```

For example:

```
$ psql -h localhost -U lsmb_dbadmin newcompany < lsmb-db.sqlc
```

10. Read the beginning section of the PostgreSQL release notes for each PostgreSQL version from the old to the new and determine if there are any additional steps that must be performed on the database after the restore. If there are, do them now.
11. Verify all data restored and that LedgerSMB is working correctly.
12. Replace the old LSMB installation with the new installation. The exact steps to accomplish this varies based on how LSMB was originally installed.
13. Allow users to use LSMB.

D.6.1 Migration from PostgreSQL 8.3 (or earlier) to 8.4

The content of this sub-section is **DEPRECATED**, unedited, and only remains for historical reference purposes.

Performance benefit due to built in support for recursive queries only available after the next `setup.pl` run Also, after next `setup.pl` run, one should uninstall the `tablefunc` extension

In 9.1+ may need to `DROP EXTENSION tablefunc;`.

In 8.4, to 9.0 may need to `uninstall_tablefunc.sql` from the PostgreSQL contrib directory.

LedgerSMB 1.3 uses some extension modules for versions PostgreSQL 8.3 and before for functionality that has been built into PostgreSQL 8.4 and later. To make use of the (faster) built in version of that functionality, the following restore procedure should be used.

1. Migrate the database to the new function as described in section
2. If you're using 9.1 and up, issue the command "CREATE EXTENSION tablefunc FROM UNPACKAGED" from a psql prompt when connected to the company database
3. Run 'setup.pl' from your browser to upgrade the database's routines; this command will install routines optimized for your version of PostgreSQL
4. To clean up functions and procedures in the database which are no longer used you may need to run one of the commands below:

8.4 or 9.0

```
psql ... -f uninstall_tablefunc.sql in a shell
```

9.1 to 9.5

```
DROP EXTENSION tablefunc; from a psql session connected to  
the company database
```


Appendix E

Listing of application roles

Application roles specify the right (privilege) to execute one or more tasks in the application. LedgerSMB enforces these roles by allowing a user to select (list, read) data from or to insert (create), update (edit) or delete (delete) data in the tables holding the data related to the execution of these tasks.

These roles definitions are generated directly from a running database using `gather-db-info.pl`.

Auto generated using LedgerSMB version 1.13.0-dev on November 23, 08:01:18 2024 CST.

account.all

This role combines all **General Ledger (GL)** account and GIFI code rights.

account.create

This role allows creation of new **GL** accounts.

account.delete

This role allows deletion of **GL** accounts. Please note that there are only very few circumstances where a **GL** account can be deleted. Instead of deleting the account, the user is advised to mark the account as 'obsolete'.

account.edit

This role allows modification of **GL** accounts.

account.link_description.create

This role allows creating new use-cases for **GL** accounts (so called "account link descriptions").

ap.all

This role combines all (batches of) purchase transaction and invoices permissions.

ap_invoice_create

This role allows creation of new purchase invoices (not purchase transactions).

ap_invoice_create_voucher

This role allows creation of batches of new purchase invoices (not purchase transactions).

ap_transaction_all

This role allows creating and viewing purchase transactions and accounts as well as creating attachments.

ap_transaction_create

This role allows creation of purchase transactions (not invoices).

ap_transaction_create_voucher

This role allows creation of batches of new purchase transactions (not invoices).

ap_transaction_list

This role allows viewing of purchase transactions and invoices.

ap_voucher_all

This role allows creation of batches of both purchase transactions and invoices.

ar_all

This role combines all (batches of) sales transaction and invoices permissions.

ar_invoice_create

This role allows creation of new sales invoices (not sales transactions).

ar_invoice_create_voucher

This role allows creation of batches of new sales invoices (not sales transactions).

ar_transaction_all

This role allows creating and viewing sales transactions and accounts as well as creating attachments.

ar_transaction_create

This role allows creation of new sales transactions (not invoices).

ar_transaction_create_voucher

This role allows creation of batches of new sales transactions (not invoices).

ar_transaction_list

This role allows viewing of sales transactions and invoices.

ar_voucher_all

This role allows creation of batches of both sales transactions and invoices.

assembly_stock

This role allows triggering a stocking action on assemblies.

Stocking assemblies means converting labor and parts to stocked assemblies.

assets_administer

This role combines all assets rights.

assets_approve

This role allows approving the output of the depreciation procedure.

assets_depreciate

This role allows running the asset depreciation procedure.

assets_enter

This role allows creation of new assets.

audit_trail_maintenance

This role grants delete access to the audit trail table.

auditor

This role grants read access to the audit trail table.

base_user

Users need to be given this role in order to be granted access to the database schema which holds all LedgerSMB objects.

This role only allows access to menu items Preferences, Logout, and New Window. The user basically cannot do anything without added additional roles.

batch_create

This role allows creation of new batches and vouchers.

batch_list

This role allows listing existing batches.

batch_post

This role allows posting batches of e.g. transactions, payments and invoices.

budget_approve

This role allows searching, viewing and approving of budgets.

budget_enter

This role allows creation and updating of budgets.

budget_obsolete

This role allows searching and viewing budgets as well as marking them obsolete (=no longer applicable).

budget_view

This role allows searching and viewing of budgets.

business_type_all

This role combines the create and edit rights for 'type of business' classes.

business_type_create

This role allows creation of new 'type of business' classes.

business_type_edit

This role allows modification of 'type of business' classes.

business_units_manage

This role allows searching, viewing, creation and editing of business (reporting) classes and their members.

cash_all

This role combines the all reconciliation rights with the rights to enter payments and receipts.

contact_all_rights

This role combines all 'contact_class_*' and 'contact_*' roles and grants all access rights to all contact classes.

contact_class_cold_lead

This role allows access to cold sales lead contact data. Combine with 'contact_read', 'contact_create', 'contact_edit' and/or 'contact_delete' to determine the type of access granted.

contact_class_contact

This role allows access to contact data (e-mail, phone, etc) of all kinds of contacts (customer/vendor/...). Combine with 'contact_read', 'contact_create', 'contact_edit' and/or 'contact_delete' to determine the type of access granted.

contact_class_customer

This role allows access to customer contact data. Combine with 'contact_read', 'contact_create', 'contact_edit' and/or 'contact_delete' to determine the type of access granted.

contact_class_employee

This role allows access to employee contact data. Combine with 'contact_read', 'contact_create', 'contact_edit' and/or 'contact_delete' to determine the type of access granted.

contact_class_hot_lead

This role allows access to hot sales lead contact data. Combine with 'contact_read', 'contact_create', 'contact_edit' and/or 'contact_delete' to determine the type of access granted.

contact_class_lead

This role allows access to sales lead contact data. Combine with 'contact_read', 'contact_create', 'contact_edit' and/or 'contact_delete' to determine the type of access granted.

contact_class_referral

This role allows access to referral contact data. Combine with 'contact_read', 'contact_create', 'contact_edit' and/or 'contact_delete' to determine the type of access granted.

contact_class_robot

This role allows access to robot (automated process, acting on behalf of...) contact data. Combine with 'contact_read', 'contact_create', 'contact_edit' and/or 'contact_delete' to determine the type of access granted.

contact_class_sub_contractor

This role allows access to subcontractor contact data. Combine with 'contact_read', 'contact_create', 'contact_edit' and/or 'contact_delete' to determine the type of access granted.

contact_class_vendor

This role allows access to vendor contact data. Combine with 'contact_read', 'contact_create', 'contact_edit' and/or 'contact_delete' to determine the type of access granted.

contact_create

When paired with one or more 'contact_class.*' role/-s, this role allows creation of new entities, persons and companies (contacts).

Each contact_class.<resource> role, when paired with contact_read, enables this access for the specific <resource>. On it's own, the contact_create-role does not provide any rights.

contact_delete

When paired with one or more 'contact_class.*' role/-s, this role allows removal of existing entities, persons and companies (contacts).

Note that in order to be able to search for contacts to be deleted, the user needs to be assigned the 'contact_read' role.

Each contact_class.<resource> role, when paired with contact_read, enables this access for the specific <resource>. On it's own, the contact_delete-role does not provide any rights.

contact_edit

When paired with one or more 'contact_class.*' role/-s, this role allows editing of existing entities, persons and companies (contacts).

Each contact_class.<resource> role, when paired with contact_read, enables this access for the specific <resource>. On it's own, the contact_edit-role does not provide any rights.

contact_read

When paired with one or more 'contact_class.*' role/-s, this role allows searching and viewing entities, persons and companies (contacts).

Each contact_class.<resource> role, when paired with contact_read, enables this access for the specific <resource>. On it's own, the contact_read-role does not provide any rights.

country_all

This role combines all rights for countries.

country_create

This role allows creation of new countries.

country_edit

This role allows modification of countries.

draft_modify

This role allows modification of existing draft (= saved) transactions.

draft_post

This role allows posting of saved transactions to the ledger.

employees_manage

This role allows creation, updating and searching of employees.

exchangerate_edit

This role allows searching, viewing and editing of currencies, exchange rates and exchange rate types.

file.attach_eca

This role allows attaching files to entity credit accounts (customers/vendors).

file.attach_entity

This role allows attaching files to entities (contacts).

file.attach_order

This role allows attaching files to orders and quotes.

file.attach_part

This role allows attaching files to goods and services.

file.attach_tx

This role allows attaching files to transactions and invoices.

file.read

This role allows reading of files attachments and files uploaded through the system menu.

file.upload

This role allows uploading of files through the system menu.

financial_reports

This role allows running of financial reports: Income Statement, Balance Sheet, Trial Balance and Inventory & COGS.

gifi.create

This role allows creation of new GIFI codes.

gifi.edit

This role allows modification of GIFI codes.

gl_all

This role combines GL transaction and batch creation with GL reporting and year-end processing.

gl_reports

This role allows searching transactions in the general ledger.

gl_transaction_create

This role allows creation of new and updating of saved GL transactions.

gl_voucher_create

This role allows creation of batches of GL transactions.

inventory_adjust

This role allows adjusting inventory by creating inventory adjustment reports.

inventory_all

This role grants all rights to manage warehouse configuration, stock receipt, shipping and transfer.

inventory_approve

This role allows confirmation of inventory adjustments by approval of inventory adjustment reports.

inventory_receive

This role allows receiving of parts into stock.

inventory_reports

This role allows searching for and reading existing inventory adjustment reports.

inventory_ship

This role allows shipping of stocked parts.

inventory_transfer

This role allows moving stock between warehouses.

language_create

This role allows creation of new languages.

language_edit

This role allows modification of languages.

orders_generate

This role combines the rights to generate orders from time cards, purchase orders from sales orders and consolidate (purchase and sales) orders.

orders_manage

This role combines all order generation and consolidation rights.

orders_purchase_consolidate

This role allows generating consolidated purchase orders from multiple outstanding purchase orders.

orders_sales Consolidate

This role allows generating consolidated sales orders from multiple outstanding sales orders.

orders_sales_to_purchase

This role allows generating purchase orders from sales orders.

part_create

This role allows creation of new parts.

So as to let the user of this role see/manage pricing per customer, this role includes the ability to read contacts.

part_delete

This role allows deletion of existing parts.

part_edit

This role allows changing existing parts.

payment_process

This role allows entry of payments to vendors.

pricegroup_create

This role allows creation of new price groups.

pricegroup_edit

This role allows changing existing price groups.

purchase_order_create

This role allows creating purchase orders.

purchase_order_delete

This role allows (searching for and) deleting existing purchase orders.

purchase_order_edit

This role allows (searching for and) modifying existing purchase orders.

purchase_order_list

This role allows searching and viewing sales orders.

receipt_process

This role allows entry of receipts from customers.

reconciliation_all

This role combines creation, updating and approval rights for reconciliation reports.

reconciliation_approve

This role allows approval of reconciliation reports.

reconciliation_enter

This role allows creation and updating of reconciliation reports.

recurring

This role allows access to the Recurring Transactions menu; it does not grant rights to list or create transactions.

rfq_create

This role allows creating (purchase) requests for quotation.

rfq_delete

This role allows (searching for and) deleting existing requests for quotation.

rfq_list

This role allows searching and viewing (purchase) requests for quotation.

sales_order_create

This role allows creating sales orders.

sales_order_delete

This role allows (searching for and) deleting existing sales orders.

sales_order_edit

This role allows (searching for and) modifying existing sales orders.

sales_order_list

This role allows searching and viewing sales orders.

sales_quotation_create

This role allows creating sales quotations.

sales_quotation_delete

This role allows (searching for and) deleting existing sales quotations.

sales_quotation_list

This role allows searching and viewing sales quotations.

sic.all

This role combines all rights for Standardized Industry Codes (SIC).

sic.create

This role allows creation of new Standardized Industry Codes (SIC).

sic.edit

This role allows modification of Standardized Industry Codes (SIC).

system_admin

This role combines the rights to manage settings, **GL** accounts, types of business, SIC, users and tax forms.

system_settings.change

This role allow changing items in the System > Defaults menu.

system_settings.list

This role allows viewing items in the System > Defaults menu.

tax_form.save

This role allows modification of tax forms.

taxes.set

This role allows changing tax rates on tax accounts.

template_edit

This role allows modification of document (e.g. invoice) templates.

timecard_add

This role allows adding time cards for which it needs read access to customers.

timecard_list

This role allows viewing the list of time cards; for which it needs read access to customers.

timecard_order_generate

This role allows generating orders from time cards.

transaction_template_delete

This role allows deletion of template (i.e. unposted) transactions.

translation_create

This role allows creation of translations for parts, parts groups and reporting units.

users_manage

This role allows addition and removal of users to the current company.

voucher_delete

This role allows deletion of vouchers (i.e. groups of e.g. payments).

warehouse_create

This role allows creation of (configuration of) new warehouses.

warehouse_edit

This role allows updating of (configuration of) existing warehouses.

yearend_reopen

This role allows undoing a prior year-end run by reversing the year-end transaction.

yearend_run

This role allows running the year-end process, i.e. clearing the P&L.

Appendix F

Deprecated Content

F.1 ledgersmb.conf

The use of `ledgersmb.conf` is deprecated as of 1 Jan 2023 in favor of `ledgersmb.yaml` and only remains here for historical reasons.

The environment variable for controlling the configuration is `LSMB_CONFIG_FILE`.

By default `LSMB_CONFIG_FILE` points to `/etc/ledgersmb/ledgersmb.conf` and can be set to an alternate location.

[main] Section

language

Set language for login and admin pages.

date_format

Set the date format for the admin pages. The default is `yyyy-mm-dd`.
Unset means detected from the browser)

log_level

Specifies the logging level to be used. This setting is ignored when the `log_config` setting is supplied and indicates an existing file. Available values are `OFF`, `FATAL`, `ERROR`, `WARN`, `INFO`, `DEBUG`, `TRACE`.

log_config

Set the name of the extended logging configuration file. This file uses the `log4j/Log::Log4perl` syntax described at
<https://metacpan.org/pod/Log::Log4perl#Configuration-files>

dojo_theme

This is the Dojo theme that is used by default when no other theme has been selected. The default if unset is `claro`. Other values can be `nihilo`, `soria`, and `tundra`.

auth

Indicates which authentication module is to be used; default is DB, referring to LedgerSMB::Auth::DB.

cache.templates

Determines if HTML templates used for the LedgerSMB UI will be stored in precompiled form for improved performance. The value 0 (zero) means no caching; the value 1 means caching in the directory `$(tmpdir)/lsmb_templates`; systems running multiple LedgerSMB instances with customized UIs should use separate values for `tmpdir` for that reason

[environment] Section

Sets shell environment variables when starting up.

PATH

If the server can't find applications, append to the path. For example:
`PATH=/bin:/usr/bin:/usr/local/bin:/usr/local/pgsql/bin`

PERL5LIB

On some operating systems like macOS the perl path may need to be set. For example, on macOS using Fink's Perl libs it might be set like:
`PERL5LIB=/sw/lib/perl5`

[paths] Section**templates**

Templates base directory. Typically set to `templates`.

images

Images base directory. Typically set to `UI/images`.

localepath

Path to locale files. Defaults to `locale/po`.

templates.cache

Location where compiled templates are stored. When relative, appended to the directory specified by `File::Spec->tmpdir()`.

Defaults to `lsmb_templates`.

workflows

Directory where workflow files are stored. Defaults to `workflows`

custom_workflows

Directory where custom workflow files are stored. Custom workflows are used to override behavior of the default workflows by providing actions, conditions, etc. with the same name and type or by providing workflows of the same type with additional states and actions. Defaults to `custom_workflows`.

[programs] Section

Setup the various helper programs that can be used with LedgerSMB.

[mail] Section**smtphost**

The host name of the smtp server. If provided, then do not use local sendmail. Note that one of sendmail or smtp host has to be provided for email to work.

sendmail

Path the local sendmail executable. If provided then do not use an smtp server. Note that one of sendmail or smtp host has to be provided for email to work.

smtpport

The port number of the smtp server. Note this might vary depending on whether TLS or SSL is used.

smtpuser

The smtp server account name

smtpauthmech

The smtp server authentication method

smtp pass

The smtp server password

backup_email_from

Set to enable e-mail delivery of backups. For example: backup_email_from = backups@lsmb_hosting.com

smtp timeout

The timeout value in seconds.

smtp tls

If unset or 'no' then do not use either TLS or SSL. If 'yes' then use TLS and if 'tls' use SSL for the smtp connection.

smtp sender_hostname

Override the default host name for helo.

[printers] Section

This section contains a list of printers and the commands to be executed in order for the output to be sent to the given printer with the document to be printed fed to the program through standard input.

The example below shows how to send standard input data to a printer called "laser" when selecting the item "Laser" in the LedgerSMB printer selection drop down.

```
Laser = lpr -Plaser
```

Or to an Epson Printer.

```
Epson = lpr -PEpson
```

[database] Section

host Name of the host to connect to. See the documentation of the `-h` command line option at <https://www.postgresql.org/docs/current/static/app-psql.html> for more information (documentation unchanged since before 8.3, so applies to older versions as well)

port Port number to be used to connect to. See the documentation of the `-p` option at the same URL as the previous item

default_db

Database to connect to when the “Company” entry field in the login screen is left blank

db_namespace

The name space the company resides in; expert setting – should not be used (default is “public”)

sslmode

Selects whether to use [Secure Sockets Layer \(SSL\)](#) over TCP connections or not; can be “require”, “allow”, “prefer” or “disable”. Default is “prefer”.

Appendix G

Open source explained

G.1 An open source application

G.2 An open source book

Appendix H

Project history

The project started out as a fork of [SQL-Ledger \(SL\)](#) - the open source [ERP system](#) developed by Dieter Simader - somewhere between SQL-Ledger versions 2.6 and 2.8. A fork happens when a group of developers can't - for whatever reason - continue to work as one group on a project. At that time, the project splits into two or more projects and the fork is in effect.

LedgerSMB split off from the SQL-Ledger project (i.e. forked) because there was disagreement between developers about how to go forward both with respect to handling of security vulnerability reports as well as the general state of the code base.

After the fork, between versions 1.0 and 1.2 most energy was spent on making LedgerSMB more secure (i.e. less vulnerable to security attacks). In technical terms, measures were taken to fend off (amongst other things):

- Cross site scripting attacks
- Replay attacks
- SQL injection attacks

Come version 1.3 the development directed toward improvement of the overall quality of the code base as the old [SL](#) code was in very poor state: looking very much like web programs as they were written in 1998, the code had grown largely outdated in style and was no longer maintainable by 2007.

The 1.3 effort focused on bringing relief with a new application structure. Modern and important features were realized: separation of duties (for the accounting part of the application) and authorizations to allow distinguishing different roles in a company.

Unfortunately, by the beginning of 2011 the project looked mostly dead from an outside perspective: the team had not brought forward any releases since 2007, there were no signs of development and the mailing lists (a measure of community activity) were completely silent. [Subversion Version Control \(SVN\)](#) com-

mits were continuing, but were being made by ever fewer committers and contributors.

Fortunately development activity was increasing in the first half year of 2011, leading to the release of version 1.3 by September. Between September and the year end in total 10 small bug fixes were released, showing active commitment of the developers to maintain the application.

New committers showed up, indicating revived community interest. Other signs of increased interest are the higher number of bug reports and the creation of the Linux package for Debian 7, which has been included in Ubuntu 12.04 as of October 2012.

Appendix I

Copyright and license

Copyright (c) 2011, 2012 Erik Hülsmann.

This work is licensed under the Creative Commons Attribution License. To view a copy of this license, visit

<https://creativecommons.org/licenses/by/3.0/> or send a letter to Creative Commons, 559 Nathan Abbott Way, Stanford, California 94305, USA.

A summary of the license is given below, followed by the full legal text.

I.1 License summary

You are free:

- * to share -- to copy, distribute and transmit the work
- * to remix -- to adapt the work

Under the following condition:

You must attribute the work in the manner specified by the author or licensor (but in a way that suggests that they endorse you or your use of the work).

With the understanding that:

Waiver -- Any of the above conditions can be waived if you get permission from the copyright holder.

Other rights -- In no way are any of the following rights affected by the license:

- * Your fair dealing or fair use rights, or other applicable copyright exceptions and limitations;
- * The author's moral rights;

- * Rights other persons may have either in the work itself or in how the work is used, such as publicity or privacy rights.

I.2 Legal full text

License

THE WORK (AS DEFINED BELOW) IS PROVIDED UNDER THE TERMS OF THIS CREATIVE COMMONS PUBLIC LICENSE ("CCPL" OR "LICENSE"). THE WORK IS PROTECTED BY COPYRIGHT AND/OR OTHER APPLICABLE LAW. ANY USE OF THE WORK OTHER THAN AS AUTHORIZED UNDER THIS LICENSE OR COPYRIGHT LAW IS PROHIBITED.

BY EXERCISING ANY RIGHTS TO THE WORK PROVIDED HERE, YOU ACCEPT AND AGREE TO BE BOUND BY THE TERMS OF THIS LICENSE. TO THE EXTENT THIS LICENSE MAY BE CONSIDERED TO BE A CONTRACT, THE LICENSOR GRANTS YOU THE RIGHTS CONTAINED HERE IN CONSIDERATION OF YOUR ACCEPTANCE OF SUCH TERMS AND CONDITIONS.

1. Definitions

"Adaptation" means a work based upon the Work, or upon the Work and other pre-existing works, such as a translation, adaptation, derivative work, arrangement of music or other alterations of a literary or artistic work, or phonogram or performance and includes cinematographic adaptations or any other form in which the Work may be recast, transformed, or adapted including in any form recognizably derived from the original, except that a work that constitutes a Collection will not be considered an Adaptation for the purpose of this License. For the avoidance of doubt, where the Work is a musical work, performance or phonogram, the synchronization of the Work in timed-relation with a moving image ("synching") will be considered an Adaptation for the purpose of this License.

"Collection" means a collection of literary or artistic works, such as encyclopedias and anthologies, or performances, phonograms or broadcasts, or other works or subject matter other than works listed in Section 1(f) below, which, by reason of the selection and arrangement of their contents, constitute intellectual creations, in which the Work is included in its entirety in unmodified

form along with one or more other contributions, each constituting separate and independent works in themselves, which together are assembled into a collective whole. A work that constitutes a Collection will not be considered an Adaptation (as defined above) for the purposes of this License.

"Distribute" means to make available to the public the original and copies of the Work or Adaptation, as appropriate, through sale or other transfer of ownership.

"Licensor" means the individual, individuals, entity or entities that offer(s) the Work under the terms of this License.

"Original Author" means, in the case of a literary or artistic work, the individual, individuals, entity or entities who created the Work or if no individual or entity can be identified, the publisher; and in addition (i) in the case of a performance the actors, singers, musicians, dancers, and other persons who act, sing, deliver, declaim, play in, interpret or otherwise perform literary or artistic works or expressions of folklore; (ii) in the case of a phonogram the producer being the person or legal entity who first fixes the sounds of a performance or other sounds; and, (iii) in the case of broadcasts, the organization that transmits the broadcast.

"Work" means the literary and/or artistic work offered under the terms of this License including without limitation any production in the literary, scientific and artistic domain, whatever may be the mode or form of its expression including digital form, such as a book, pamphlet and other writing; a lecture, address, sermon or other work of the same nature; a dramatic or dramatico-musical work; a choreographic work or entertainment in dumb show; a musical composition with or without words; a cinematographic work to which are assimilated works expressed by a process analogous to cinematography; a work of drawing, painting, architecture, sculpture, engraving or lithography; a photographic work to which are assimilated works expressed by a process analogous to photography; a work of applied art; an illustration, map, plan, sketch or three-dimensional work relative to geography, topography, architecture or science; a performance; a broadcast; a phonogram; a compilation of data to the extent it is protected as a copyrightable work; or a work performed by a variety or circus performer to the extent it is not

otherwise considered a literary or artistic work.

"You" means an individual or entity exercising rights under this License who has not previously violated the terms of this License with respect to the Work, or who has received express permission from the Licensor to exercise rights under this License despite a previous violation.

"Publicly Perform" means to perform public recitations of the Work and to communicate to the public those public recitations, by any means or process, including by wire or wireless means or public digital performances; to make available to the public Works in such a way that members of the public may access these Works from a place and at a place individually chosen by them; to perform the Work to the public by any means or process and the communication to the public of the performances of the Work, including by public digital performance; to broadcast and rebroadcast the Work by any means including signs, sounds or images.

"Reproduce" means to make copies of the Work by any means including without limitation by sound or visual recordings and the right of fixation and reproducing fixations of the Work, including storage of a protected performance or phonogram in digital form or other electronic medium.

2. Fair Dealing Rights. Nothing in this License is intended to reduce, limit, or restrict any uses free from copyright or rights arising from limitations or exceptions that are provided for in connection with the copyright protection under copyright law or other applicable laws.

3. License Grant. Subject to the terms and conditions of this License, Licensor hereby grants You a worldwide, royalty-free, non-exclusive, perpetual (for the duration of the applicable copyright) license to exercise the rights in the Work as stated below:

to Reproduce the Work, to incorporate the Work into one or more Collections, and to Reproduce the Work as incorporated in the Collections;
to create and Reproduce Adaptations provided that any such Adaptation, including any translation in any medium, takes reasonable steps to clearly label, demarcate or otherwise identify that changes were made to the original Work. For example, a translation could be marked "The

original work was translated from English to Spanish," or a modification could indicate "The original work has been modified.";

to Distribute and Publicly Perform the Work including as incorporated in Collections; and, to Distribute and Publicly Perform Adaptations.

For the avoidance of doubt:

Non-waivable Compulsory License Schemes. In those jurisdictions in which the right to collect royalties through any statutory or compulsory licensing scheme cannot be waived, the Licensor reserves the exclusive right to collect such royalties for any exercise by You of the rights granted under this License; Waivable Compulsory License Schemes. In those jurisdictions in which the right to collect royalties through any statutory or compulsory licensing scheme can be waived, the Licensor waives the exclusive right to collect such royalties for any exercise by You of the rights granted under this License; and, Voluntary License Schemes. The Licensor waives the right to collect royalties, whether individually or, in the event that the Licensor is a member of a collecting society that administers voluntary licensing schemes, via that society, from any exercise by You of the rights granted under this License. The above rights may be exercised in all media and formats whether now known or hereafter devised. The above rights include the right to make such modifications as are technically necessary to exercise the rights in other media and formats. Subject to Section 8(f), all rights not expressly granted by Licensor are hereby reserved.

4. Restrictions. The license granted in Section 3 above is expressly made subject to and limited by the following restrictions:

You may Distribute or Publicly Perform the Work only under the terms of this License. You must include a copy of, or the Uniform Resource Identifier (URI) for, this License with every copy of the Work You Distribute or Publicly Perform. You may not offer or impose any terms on the Work that restrict the terms of this License or the ability of the recipient of the Work to exercise the rights granted to that recipient under the terms of the License. You may not sublicense the Work. You must keep intact all notices that refer to this License and to the disclaimer of warranties with every copy of the Work You Distribute or Publicly Perform. When You Distribute or

Publicly Perform the Work, You may not impose any effective technological measures on the Work that restrict the ability of a recipient of the Work from You to exercise the rights granted to that recipient under the terms of the License. This Section 4(a) applies to the Work as incorporated in a Collection, but this does not require the Collection apart from the Work itself to be made subject to the terms of this License. If You create a Collection, upon notice from any Licensor You must, to the extent practicable, remove from the Collection any credit as required by Section 4(b), as requested. If You create an Adaptation, upon notice from any Licensor You must, to the extent practicable, remove from the Adaptation any credit as required by Section 4(b), as requested.

If You Distribute, or Publicly Perform the Work or any Adaptations or Collections, You must, unless a request has been made pursuant to Section 4(a), keep intact all copyright notices for the Work and provide, reasonable to the medium or means You are utilizing: (i) the name of the Original Author (or pseudonym, if applicable) if supplied, and/or if the Original Author and/or Licensor designate another party or parties (e.g., a sponsor institute, publishing entity, journal) for attribution ("Attribution Parties") in Licensor's copyright notice, terms of service or by other reasonable means, the name of such party or parties; (ii) the title of the Work if supplied; (iii) to the extent reasonably practicable, the URI, if any, that Licensor specifies to be associated with the Work, unless such URI does not refer to the copyright notice or licensing information for the Work; and (iv), consistent with Section 3(b), in the case of an Adaptation, a credit identifying the use of the Work in the Adaptation (e.g., "French translation of the Work by Original Author," or "Screenplay based on original Work by Original Author"). The credit required by this Section 4 (b) may be implemented in any reasonable manner; provided, however, that in the case of a Adaptation or Collection, at a minimum such credit will appear, if a credit for all contributing authors of the Adaptation or Collection appears, then as part of these credits and in a manner at least as prominent as the credits for the other contributing authors. For the avoidance of doubt, You may only use the credit required by this Section for the purpose of attribution in the manner set out above and, by exercising Your rights under this License, You may not implicitly or explicitly assert or imply any connection with, sponsorship or endorsement by the Original Author, Licensor and/or Attribution Parties, as appropriate, of You or Your use of the Work,

without the separate, express prior written permission of the Original Author, Licensor and/or Attribution Parties.

Except as otherwise agreed in writing by the Licensor or as may be otherwise permitted by applicable law, if You Reproduce, Distribute or Publicly Perform the Work either by itself or as part of any Adaptations or Collections, You must not distort, mutilate, modify or take other derogatory action in relation to the Work which would be prejudicial to the Original Author's honor or reputation. Licensor agrees that in those jurisdictions (e.g. Japan), in which any exercise of the right granted in Section 3(b) of this License (the right to make Adaptations) would be deemed to be a distortion, mutilation, modification or other derogatory action prejudicial to the Original Author's honor and reputation, the Licensor will waive or not assert, as appropriate, this Section, to the fullest extent permitted by the applicable national law, to enable You to reasonably exercise Your right under Section 3(b) of this License (right to make Adaptations) but not otherwise.

5. Representations, Warranties and Disclaimer

UNLESS OTHERWISE MUTUALLY AGREED TO BY THE PARTIES IN WRITING, LICENSOR OFFERS THE WORK AS-IS AND MAKES NO REPRESENTATIONS OR WARRANTIES OF ANY KIND CONCERNING THE WORK, EXPRESS, IMPLIED, STATUTORY OR OTHERWISE, INCLUDING, WITHOUT LIMITATION, WARRANTIES OF TITLE, MERCHANTIBILITY, FITNESS FOR A PARTICULAR PURPOSE, NON-INFRINGEMENT, OR THE ABSENCE OF LATENT OR OTHER DEFECTS, ACCURACY, OR THE PRESENCE OF ABSENCE OF ERRORS, WHETHER OR NOT DISCOVERABLE. SOME JURISDICTIONS DO NOT ALLOW THE EXCLUSION OF IMPLIED WARRANTIES, SO SUCH EXCLUSION MAY NOT APPLY TO YOU.

6. Limitation on Liability.

EXCEPT TO THE EXTENT REQUIRED BY APPLICABLE LAW, IN NO EVENT WILL LICENSOR BE LIABLE TO YOU ON ANY LEGAL THEORY FOR ANY SPECIAL, INCIDENTAL, CONSEQUENTIAL, PUNITIVE OR EXEMPLARY DAMAGES ARISING OUT OF THIS LICENSE OR THE USE OF THE WORK, EVEN IF LICENSOR HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

7. Termination

This License and the rights granted hereunder will terminate automatically upon any breach by You of the

terms of this License. Individuals or entities who have received Adaptations or Collections from You under this License, however, will not have their licenses terminated provided such individuals or entities remain in full compliance with those licenses.

Sections 1, 2, 5, 6, 7, and 8 will survive any termination of this License. Subject to the above terms and conditions, the license granted here is perpetual (for the duration of the applicable copyright in the Work). Notwithstanding the above, Licensor reserves the right to release the Work under different license terms or to stop distributing the Work at any time; provided, however that any such election will not serve to withdraw this License (or any other license that has been, or is required to be, granted under the terms of this License), and this License will continue in full force and effect unless terminated as stated above.

8. Miscellaneous

Each time You Distribute or Publicly Perform the Work or a Collection, the Licensor offers to the recipient a license to the Work on the same terms and conditions as the license granted to You under this License. Each time You Distribute or Publicly Perform an Adaptation, Licensor offers to the recipient a license to the original Work on the same terms and conditions as the license granted to You under this License.

If any provision of this License is invalid or unenforceable under applicable law, it shall not affect the validity or enforceability of the remainder of the terms of this License, and without further action by the parties to this agreement, such provision shall be reformed to the minimum extent necessary to make such provision valid and enforceable. No term or provision of this License shall be deemed waived and no breach consented to unless such waiver or consent shall be in writing and signed by the party to be charged with such waiver or consent.

This License constitutes the entire agreement between the parties with respect to the Work licensed here. There are no understandings, agreements or representations with respect to the Work not specified here. Licensor shall not be bound by any additional provisions that may appear in any communication from You. This License may not be modified without the mutual written agreement of the Licensor and You. The rights granted under, and the subject matter referenced, in this License were drafted utilizing the terminology of the Berne Convention for the Protection of Literary and Artistic Works (as amended

on September 28, 1979), the Rome Convention of 1961, the WIPO Copyright Treaty of 1996, the WIPO Performances and Phonograms Treaty of 1996 and the Universal Copyright Convention (as revised on July 24, 1971). These rights and subject matter take effect in the relevant jurisdiction in which the License terms are sought to be enforced according to the corresponding provisions of the implementation of those treaty provisions in the applicable national law. If the standard suite of rights granted under applicable copyright law includes additional rights not granted under this License, such additional rights are deemed to be included in the License; this License is not intended to restrict the license of any rights under applicable law.

Acronyms

- AP** Accounts payable. 26
- API** Application Programmer Interface. 28, 29, 101, 102, 235
- AR** Accounts receivable. 26
- CI** Continuous Integration. 247
- CoA**
Chart of Accounts. 48, 65, 108
- COGS**
cost of goods sold. 28, 38, 39, 107
- CSV**
Comma Separated Values. 229, 231
- EFT** Electronic Funds Transfer. 62
- ERP** Enterprise Resource Planning. 25, 26, 32
- ERP system**
Enterprise Resource Planning system. 53, 273
- FIFO**
First in, first out. 38, 39, 107
- GL** General Ledger. 26, 36, 257, 263, 265
- IAS** International Accounting Standards. 32
- IFRS**
International Financial Reporting Standards. 32
- LIFO**
Last in, first out. 39
- PSGI**
Perl Web Server Gateway Interface. 28

- REST** Representational State Transfer. [28](#)
- RFQ** Request for quotation. [67](#), [71–73](#), [181](#)
- SIC** Standard Industrial Classification. [123](#)
- SL** SQL-Ledger. [28](#), [273](#)
- SSL** Secure Sockets Layer. [270](#)
- SSN** Social Security Number. [51](#)
- SVN** Subversion Version Control. [273](#)
- VAT** Value Added Tax. [65](#), [216](#)
- VPS** Virtual Private Server. [27](#)
- XML** Extensible Markup Language. [108](#)
- XSD** XML Schema Definition. [108](#)

Definition of terms

accounting year

An accounting year, which may also be called fiscal year, financial year or budget year is a twelve month period used for annual financial reporting. An accounting year may be the same as a calendar year, but most jurisdictions allow the accounting year to be defined differently from the calendar year. [215](#)

accrual basis

Income is deemed earned when the customer invoice is prepared and sent, and the expenses are deemed incurred when the purchase order is prepared and sent. [37](#)

add-on

Extension to the base LedgerSMB system to add new functionality. [26](#), [150](#)

cash basis

Income is deemed earned when the customer pays it, and the expenses are deemed incurred when the business pays them. [37](#)

company

A company is a contact initially setup as a company and usually has further configuration as a vendor or customer. [175](#)

contact

A contact is primarily a company or person. The contact can be further identified as employee, vendor, customer, contact, lead, referral, hot lead, cold lead, sub-contractor, etc. The contact's setup determines how it is used in LedgerSMB. [175](#), [177](#)

contra

A contra account is used in a general ledger to reduce the value of a related account when the two are netted together. A contra account's natural balance is the opposite of the associated account. If a debit is the natural balance recorded in the related account, the contra account records a credit. [112](#), [231](#)

counterparty

A legal entity, unincorporated entity, or collection of entities to which an exposure of financial risk may exist. In LSMB, a counterparty, is the

same as an entity, and can be either a company entity (suppliers, vendors, customers), or a person entity (including an employee). 77–80

credit account

The link between contacts and the accounting system. Credit accounts can be setup to identify contacts as vendors, customers, or both. When setup as both there may be different identifying numbers for the customer and vendor. The term “credit account” is a generalization of the customer and vendor accounts. 175, 177

customer

Is a contact initially setup as a company or person and further configured with at least one customer credit account. 111, 112, 155, 156, 159, 160, 175, 178, 179, 192–195, 203, 204, 207, 210

entity

A legal entity, unincorporated entity, or collection of entities to which an exposure of financial risk may exist. In LSMB, an entity, is the same as a counterparty, and can be either a company entity (suppliers, vendors, customers) or a person entity (including an employee). 77

GIFI

Stands for Generalized Index for Financial Information, a system used by the Canada Revenue Agency for filing corporate tax returns (see <http://www.cra-arc.gc.ca>). 153, 232

ISO-20022

ISO 20022 is a multi part International Standard prepared by ISO Technical Committee TC68 Financial Services. It describes a common platform for the development of messages relating to the financial services industry¹. 229

OFX

Open Financial Exchange (OFX) is a data-stream format for exchanging financial information that evolved from Microsoft’s Open Financial Connectivity (OFC) and Intuit’s Open Exchange file formats². 229

old code

Stands for the code inherited from SQL Ledger at the time of the fork. This code stands to be replaced by “Modern Perl”. 228

over payment

Over payment is the term used to represent invoice over payments, deposits, pre-payments, advances, etc. Essentially any activity that results in paying too much or in advance. Over payments can apply to both vendors and customers. 208

¹From <https://iso20022.org/>

²From https://en.wikipedia.org/wiki/Open_Financial_Exchange

permalink

A permalink is a link which you can share with your colleagues and when inserted into the browser, returns exactly the same report (or at least a search with exactly the same parameters as you used). To access the link the user can use the browsers "Copy link" function to copy the link and pass it to a colleague. [221](#)

PL/SQL

Procedural Language for SQL, a programming language from Oracle for building database functionality. LedgerSMB uses a variant supported by PostgreSQL called PL/pgSQL. [28](#)

reporting units

Reporting units are primarily used to divide sales and purchase reporting into departments, jobs, projects, etc.. [121](#)

Single Page Application

Single Page (web) Applications are a type of web applications; they employ JavaScript and HTML5 techniques to retrieve data, code and layout fragments from the server as the UI needs them. They are called "single page" because they use JavaScript to update the current page instead of loading a completely new page. [28](#)

vendor

Is a contact initially setup as a company or person and further configured with at least one vendor credit account. [175](#), [178](#), [179](#)

Index

- 1099-INT, [158](#)
- 1099-MISC, [157](#)
- account number alternatives, [153](#)
- accounting year, [215](#)
- accrual basis, [37](#)
- add-on, [26](#), [150](#)
- address, [80](#)
- advances, [208](#)
- apache, [101](#)
- API, [28](#), [102](#), [235](#)
- application roles, [107](#)
- backup, [45](#), [133](#), [165](#), [167](#)
- bank account, [62](#)
- bank account, initial balance, [62](#)
- bank statements, [225](#)
- budget, [217](#)
- Canada Revenue Agency, [153](#)
- cash basis, [37](#)
- Chart of Accounts, [47](#), [108](#)
- chart of accounts, [48](#)
- Chart of Accounts Import, [115](#)
- Chart of Accounts predefined, [108](#)
- closing, [214](#)
- CoA, [108](#)
- CoA XSD, [108](#)
- COGS, [28](#), [107](#)
- company, [77](#), [175](#)
- configuration, [99](#), [225](#)
- contact, [175](#)
- contact search, [177](#)
- contra, [112](#), [231](#)
- control code, [177](#)
- counterparty, [77](#)
- createuser, [46](#)
- credit account, [79](#), [80](#), [175](#)
- csv, [229](#)
- currency, [118](#), [151](#), [160](#), [179](#), [215](#)
- customer, [69](#), [72](#), [77](#), [175](#), [178](#)
- customer reporting unit class, [122](#)
- customization, [225](#)
- database load path, [104](#)
- database restore, [254](#)
- department reporting unit class, [122](#)
- deposit, [208](#)
- discount, [159](#)
- dump, [253](#)
- eInvoice, [228](#)
- email, [59](#), [103](#), [119](#), [192](#)
- entity, [77](#)
- Exchange rate classes, [211](#)
- Exchange rates, [210](#)
- extension points, [225](#)
- faq database restore, [254](#)
- FIFO, [107](#)
- foreign currency, [210](#)
- foreign currency workflow, [211](#)
- foreign exchange, [115](#)
- France, [153](#)
- functional base currency, [210](#)
- fund reporting unit class, [122](#)
- General Ledger, [257](#), [263](#), [265](#)
- gifi, [153](#)
- GL, [257](#), [263](#), [265](#)
- HTML path, [104](#)
- import Chart of Accounts, [115](#)
- install Docker development, [248](#)
- install Docker preview, [247](#)
- install Docker production, [247](#)
- install from distribution, [249](#)
- install manual, [248](#)
- install native, [248](#)
- installation, [247](#), [254](#)
- inventory, [107](#)
- invoice, [107](#), [192](#)

ISO20022, 229

job reporting unit class, 122

ledgersmb-admin, 45

ledgersmb.conf, 102, 267

ledgersmb.yaml, 102, 267

locale path, 104

lot reporting unit class, 122

LSMB_CONFIG_FILE, 267

lsmb_dbadmin, 46, 254

mail, 103

OFX, 229

old code, 228

over payment, 208, 209

part group, 69, 152

part hierarchy, 152

password, 50, 143, 145, 147

password duration, 119

password expiration, 119

period closing, 214

permalink, 221

person, 77

pg_hba.conf, 101

pg_upgrade, 254

picking, 107

PL/SQL, 28

Postgres, 101

postgres, 45

Postgres Security, 101

PostgreSQL upgrade, 253

preferences, 145

prepayment, 208

price group, 160

price matrix, 159, 160

printer configuration, 105

printing, 105

project reporting unit class, 122

psql, 46

reconciliation, 112

release number, 101

reporting unit, 121

reporting unit class, 121

reporting units, 80

REST, 28

restore, 165, 253

role account_all, 257

role account_create, 257

role account_delete, 257

role account_edit, 257

role account_link_description_create, 257

role ap_all, 257

role ap_invoice_create, 258

role ap_invoice_create_voucher, 258

role ap_transaction_all, 258

role ap_transaction_create, 258

role ap_transaction_create_voucher, 258

role ap_transaction_list, 258

role ap_voucher_all, 258

role ar_all, 258

role ar_invoice_create, 258

role ar_invoice_create_voucher, 258

role ar_transaction_all, 258

role ar_transaction_create, 258

role ar_transaction_create_voucher, 258

role ar_transaction_list, 258

role ar_voucher_all, 258

role assembly_stock, 259

role assets_administer, 259

role assets_approve, 259

role assets_depreciate, 259

role assets_enter, 259

role audit_trail_maintenance, 259

role auditor, 259

role base_user, 259

role batch_create, 259

role batch_list, 259

role batch_post, 259

role budget_approve, 259

role budget_enter, 259

role budget_obsolete, 259

role budget_view, 259

role business_type_all, 260

role business_type_create, 260

role business_type_edit, 260

role business_units_manage, 260

role cash_all, 260

role contact_all_rights, 260

role contact_class_cold_lead, 260

role contact_class_contact, 260

role contact_class_customer, 260

role contact_class_employee, 260

role contact_class_hot_lead, 260

role contact_class_lead, 260

role contact_class_referral, 261

role contact_class_robot, 261

role contact_class_sub_contractor, 261

- role contact_class_vendor, 261
- role contact_create, 261
- role contact_delete, 261
- role contact_edit, 261
- role contact_read, 261
- role country_all, 262
- role country_create, 262
- role country_edit, 262
- role draft_modify, 262
- role draft_post, 262
- role employees_manage, 262
- role exchangerate_edit, 262
- role file_attach_eca, 262
- role file_attach_entity, 262
- role file_attach_order, 262
- role file_attach_part, 262
- role file_attach_tx, 262
- role file_read, 262
- role file_upload, 262
- role financial_reports, 262
- role gifi_create, 262
- role gifi_edit, 262
- role gl_all, 263
- role gl_reports, 263
- role gl_transaction_create, 263
- role gl_voucher_create, 263
- role inventory_adjust, 263
- role inventory_all, 263
- role inventory_approve, 263
- role inventory_receive, 263
- role inventory_reports, 263
- role inventory_ship, 263
- role inventory_transfer, 263
- role language_create, 263
- role language_edit, 263
- role orders_generate, 263
- role orders_manage, 263
- role orders_purchase Consolidate, 263
- role orders_sales Consolidate, 264
- role orders_sales_to_purchase, 264
- role part_create, 264
- role part_delete, 264
- role part_edit, 264
- role payment_process, 264
- role pricegroup_create, 264
- role pricegroup_edit, 264
- role purchase_order_create, 264
- role purchase_order_delete, 264
- role purchase_order_edit, 264
- role purchase_order_list, 264
- role receipt_process, 264
- role reconciliation_all, 264
- role reconciliation_approve, 264
- role reconciliation_enter, 264
- role recurring, 264
- role rfq_create, 265
- role rfq_delete, 265
- role rfq_list, 265
- role sales_order_create, 265
- role sales_order_delete, 265
- role sales_order_edit, 265
- role sales_order_list, 265
- role sales_quotation_create, 265
- role sales_quotation_delete, 265
- role sales_quotation_list, 265
- role sic_all, 265
- role sic_create, 265
- role sic_edit, 265
- role system_admin, 265
- role system_settings_change, 265
- role system_settings_list, 265
- role tax_form_save, 265
- role taxes_set, 265
- role template_edit, 266
- role timecard_add, 266
- role timecard_list, 266
- role timecard_order_generate, 266
- role transaction_template_delete, 266
- role translation_create, 266
- role users_manage, 266
- role voucher_delete, 266
- role warehouse_create, 266
- role warehouse_edit, 266
- role yearend_reopen, 266
- role yearend_run, 266
- roles, 257
- sendmail, 103
- separation of duties, 107, 117
- sequence numbers, 120
- setup.pl, 45
- shipping, 107
- SIC, 123
- SMTP, 103
- smtp, 269
- SPA, 28
- SQL schema path, 104
- tax, 112
- tax forms, 157, 158

- template $\text{\LaTeX}$, 124
- template api, 127
- template cache path, 104
- template csv, 126
- template debugging, 127
- template html, 125
- template logging, 127
- template path, 104
- template quick reference, 129
- template syntax, 126
- template var dumping, 127
- template var substitution, 126
- templates, 48, 124
- transaction reversal, 215

- user creation, 49
- user lsmb_dbadmin, 46
- user postgres, 45

- vendor, 175, 179
- vendor reporting unit class, 122
- version number, 101

- web service, 28
- web service API, 235
- workflow, 239
- workflow configuration, 226
- workflow observers, 228
- workflow path, 104
- workflows, 107, 226

- XSD, 108

- year end, 215
- yearend_reopen, 216
- yearend_run, 215, 216