Asterisell Documentation

Release asterisell-free-stable-3.12.00

Massimo Zaniboni

November 24, 2011

Contents

Asterisell is a web application for rating, showing to customers, and billing VoIP calls.

Usage scenario:

- you are a vendor of Voice over IP Telephony (VoIP) services;
- your *customers* pay you for this service;
- your customers can call users residing on telephone networks that are not directly managed/owned from you;
- for routing calls to external networks, you use the services of other *telephone vendors*, and you pay them for this service;

Every call has:

- an income: what your *customer* pays to you;
- a cost: what you pay to other telephone service vendors (*VoIP vendor*) in order to route the call;
- an earn: the difference between the income and the cost;

You can use Asterisell for:

- rating calls;
- showing to your customers details about their calls;
- billing your customers;
- calculating what you owe to other telephone service vendors (*VoIP vendor*)

You can have different *partners*, and different *resellers*.

CHAPTER 1

Installation

1.1 Requirements

- HTTP WEB Server;
- PHP 5.0 or greater (PHP 4.0 is not supported);
- GD support enabled in PHP;
- MySQL DBMS;

1.2 Software Infrastructure

VoIP calls are managed usually from an Asterisk Server

- it routes the calls to the proper dial peer;
- it associates to every customer a unique "accountcode";
- it writes Call Detail Records (CDRs) on a specific MySQL database table shared with Asterisell.

VoIP calls can be managed from other servers, or call detail records can be imported from CSV files.

Asterisell is a PHP5 web application that:

- reads calls information from the CDR table shared with Asterisk Server;
- associate to every call a cost (what the service provider pays to other vendors for routing the customer's call);
- associate to every call an income (what the customer pays to the service provider);
- displays calls info;

1.3 Download

The suggested way to download this release is using Git version control system . This allows to:

- download application updates in an efficient way;

- be advised of conflicts between your customizations and application updates;
- send back to me bug-fixes and improvements;
- receive improvements from other in an efficient way;
- upgrade to commercial release easily without loosing your customizations;

There is an Asterisell repository at http://github.com/massimo-zaniboni/Asterisell.

Choose a directory where installing Asterisell. For example */var/www/asterisell3*.

For initial download of the free version:

```
git clone http://github.com/massimo-zaniboni/Asterisell.git
```

For the commercial version:

```
git clone http://your-license:your-license@support.asterisell.com/asterisell/git/asterisell-commercia
```

replacing *your-license* with your license code, obtained after purchasing the commercial version. NOTE: the download operation is rather slow (1-8 minutes).

It is always possible upgrading a free version to a commercial version. For updating/upgrading to new releases see *asterisell_upgrading*.

1.3.1 File Permissions

In Linux Debian make sure that the "www-data" user can read the content of the installation directory:

```
chown -R :www-data asterisell3
chmod -R g+rwx asterisell3
```

1.3.2 Packages

The typical installation requires:

- Apache2 HTTP server
- mod_php5
- mod_ssl
- MySQL database server
- awk

Asterisell use some PHP5 libraries. On Fedora/Centos for example you must install:

```
yum install awk
yum install httpd
yum install php php-cli php-common
yum install mysql mysql-server php-mysql
yum install php-bcmath php-xml php-mbstring
```

and then execute:

```
/etc/init.d/httpd restart
```

1.4 MySQL Database Configuration

Up to date Asterisell is tested only with MySQL DBMS , but the Symfony framework support many other popular open source DBMS.

1.4.1 Database Access Parameters

Update:

```
config/databases.yml
config/propel.ini
scripts/initdb.sh
```

according your needs using the correct host, username and password for MySQL access.

Initially you must use the root/admin account of MySQL, because configuration scripts will create a new database. Later you can create a specific user for the Asterisell database, in order to limit the capability of the Asterisell user.

The suggested database name is *asterisell3* in order to avoid conflicts with previous or new version of Asterisell needing different database schemas.

1.4.2 How to create a Database User with Limited Access to Asterisell Database

Enter into MySQL shell:

```
mysql -u root -p mysql
```

Create a user that can invoke MySQL only from localhost (supposing that Asterisell webserver hosts also the MySQL DBMS):

```
CREATE USER 'asterisell3user'@'localhost' IDENTIFIED BY 'some-password-here';
GRANT ALL ON asterisell3.* TO 'asterisell3user'@'localhost';
```

This example, supposes that *asterisell3* is the name of created Asterisell database, and *asterisell3user* the name of database user.

NOTE: *GRANT ALL* is needed in order to support the database upgrade script.

1.4.3 Installation Script

The installation script will:

- create a new *asterisell3* database;
- load the tables;
- initializing the Asterisell application with demo data;
- fix directory permissions;
- reset the Symfony cache;

It must not be used if it is an update of Asterisell, because otherwise all data will be lost.

In any case the script will warn you about this.

Execute:

```
cd scripts
sh initdb.sh
```

and answer to questions.

The very last versions of MySQL requires *Engine=InnoDB;* instead of *Type=InnoDB;* inside file *data/sql/lib.model.schema.sql*. In case of problems during database creation (initial installation of Asterisell), you must update manually this file.

1.4.4 Change Demo Data Later

The installation script will load also some demo data in order to test Asterisell.

If you want later, start with an empty database, you can execute:

```
cd scripts
php reset_db_and_init_data.php init <some-password-for-root-user-access>
```

for creating an empty database, with only minimal data, and a "root" user with the chosen password for initial login.

The scripts supports other type of data initialization. This command will display all its options:

```
cd scripts
php reset_db_and_init_data.php
```

You can also inspect the content of *scripts/initdb.sh* in order to see the details of performed actions.

1.4.5 Optional Configuration for Developers

If you are a developer, and you are changing Asterisell source code, maybe you want to change also its database schema, contained in *config/schema.yml*.

In this case, you must also instruct *config/propel.ini* about how to access the MySQL database. These are the same parameters of *config/databases.yml*, but they are used from a different tool, and so they must be repeated here.

Update *config/propel.ini* according your needs.

Adapt it, using the same parameters of the *databases.yml*. This file contains a lot of parameters, but you must only change these lines:

```
propel.database.createUrl  = mysql://localhost/
propel.database.url        = mysql://localhost/asterisell3
propel.output.dir          = /var/www/asterisell3
```

If you have changed the database structure in the file *config/schema.yml*, you must also update the SQL instructions creating the MySQL database. You can use the script *scripts/makedb.sh*. It will recreate the SQL commands and then it call *initdb.sh* in order to load the new database.

1.5 Web Server

1.5.1 Website Configuration

Create the file *asterisell.conf* inside the directory:

- */etc/httpd/conf.d/* or

- */etc/apache2/conf.d/*

The content is something like:

```
Alias /your-voip-service /var/www/asterisell3/web/

<Location /your-voip-service>
  Order allow,deny
  Allow from all

  # If you have mod_ssl installed
  # then with these lines you can force the usage of https connections
  # for all Asterisell access.
  # If you omit them, then the passwords are sent in plain text and
  # they can be intercepted from hackers...
  #
  AllowOverride All
  <IfModule mod_rewrite.c>
    RewriteEngine On
    RewriteCond %{HTTPS} off
    RewriteRule (.*) https://%{HTTP_HOST}%{REQUEST_URI}
  </IfModule>

</Location>
```

NOTE: remember to add */web* to your Asterisell installation directory. This because it is only the *web* subdirectory of Asterisell project that must be part of the web-space.

Restart the apache2 httpd server in order to render active the configurations using a command like:

```
/etc/init.d/httpd restart
```

or:

```
/etc/init.d/apache2 restart
```

If it is all correct then *http://your-voip-service* will display the login form.

1.5.2 PHP Resources

The administrator can invoke heavy jobs, for example graph and stats about calls. In this case it is usefull to increase the resources of PHP scripts inside on-line sessions.

In CENTOS the settings are in */etc/php.ini*, in Debian are in */etc/php5/apache2/php.ini*. Change somethiing like:

```
;;;;;;;;;;;;;;;;;;;
; Resource Limits ;
;;;;;;;;;;;;;;;;;;;

max_execution_time = 30     ; Maximum execution time of each script, in seconds
max_input_time = 60 ; Maximum amount of time each script may spend parsing request data
;max_input_nesting_level = 64 ; Maximum input variable nesting level
memory_limit = 128M      ; Maximum amount of memory a script may consume (128MB)
```

into something like:

```
;;;;;;;;;;;;;;;;;;;
; Resource Limits ;
;;;;;;;;;;;;;;;;;;;
```

```
max_execution_time = 180     ; Maximum execution time of each script, in seconds
max_input_time = 60 ; Maximum amount of time each script may spend parsing request data
;max_input_nesting_level = 64 ; Maximum input variable nesting level
memory_limit = 128M      ; Maximum amount of memory a script may consume (128MB)
```

increasing the execution time from 30 seconds to 180 seconds.

NOTE: heavy jobs like email sending, are done from the cron-job, in off-line mode, and they are not affected from these settings / constraints.

1.5.3 PHP Speedup

APC is a free, open source framework that caches data and compiled code from the PHP bytecode compiler in shared memory.

I suggest installing it, because it is well tested, and it speeds-up a lot the execution of Asterisell and other PHP applications.

For installing it in Debian:

```
aptitude install php-apc
```

or in CENTOS follow the instruction of this webpage (up to date the APC package is not part of CENTOS).

IMPORTANT: In order to activate it, the Apache web server must be restarted.

IMPORTANT: Up to date there can be conflicts between APC (version 3.0.19) and SwiftMailer. If after the sending of the first mail, an error message appears in the problem table (something like *Fatal error: Cannot inherit previously-inherited or override constant LEVEL_TOP from interface Swift_Mime_Message*), then try to upgrade APC or disactivate APC.

1.5.4 MySQL

Rates based on CSV files can be rather big, because they contain a lot of data. Asterisell save them in a record, and the SQL command can be rather big.

So you must use a reasonable *max_allowed_packet* size, for communication between PHP and MySQL.

Check that inside */etc/mysql/my.cnf* there is something like:

```
[mysqld]
max_allowed_packet = 16M
```

In case of problems, increase this limit, and then restart MySQL (NOTE: during restart new inserted CDR records will be lost):

```
/etc/init.d/mysql restart
```

Last versions of MySQL execute a fsync (flush) on disk for every transaction. This is a very big slow down for Asterisell, because it does not group batch work on CDRs on a single transaction. If MySQL is slow, it can be configured for flushing on disk the transactions every seconds. Add inside */etc/mysql/my.cnf*:

```
innodb_flush_log_at_trx_commit = 0
```

Consult MySQL related documentation for more details on side effects of this setting.

1.5.5 Additional File Access Permissions

If you have not used the default installation script (*scripts/initdb.sh*), then make sure to execute:

```
./symfony fix-perms
```

or:

```
chmod -R a+rw log
chmod -R a+rw cache
```

in order to enable the write rights on (only) these two directories.

1.6 Cron Job

1.6.1 Calls Rating and Jobs Processing

Asterisell rates the calls and execute other jobs off-line, using a cron-job process. This allows to use all the resources of the machine, without the constraints of online sessions.

The list of jobs to execute and the various parameters are inside *apps/asterisell/config/app.yml*.

In case of problems during job execution, a mail is sent to the administrator.

1.6.2 User Permissions

The process must be executed from the same user that is associated to the http connection.

On Fedora/Centos it is typically *apache*, on Debian *www-data*. In order to retrieve it, you can enter into Asterisell web interface, select Help menu and PHP-INFO option. Then you can search the element *User/Group*. The first value is the name of the PHP/webserver user.

1.6.3 Cron Job Setting

Supposing the http user is "apache" you can associate to it a cron job using the command:

```
crontab -u apache -e
```

This command will open an editor. If the defalt editor is *vi*, then digit *:i* (and press enter) in order to start editing, and *:wq* (and press enter) at the end of editing, in order to *write* and *quit/exit* from the editor

Add this line:

```
5,10,15,20,25,30,35,40,45,50,55 * * * * sh -c "cd /your-asterisell-dir/scripts/ ; nice php process_j
```

The meaning of the line is to execute at minutes 5, 10, 15, 20, ... of every hour the *php rate_all_and_test.php* command inside the Asterisell directory.

1.6.4 Log Rotate

In order to reduce log file size put *asterisell-logrotate* with execution privileges in */etc/cron.monthly* directory:

```
#!/bin/sh
cd /your-asterisell-install-dir
./symfony log-rotate asterisell prod
```

and set it as an executable file

```
chmod u+x asterisell-logrotate
```

Note that *asterisell* after *symfony* command is the name of the application and it is not depends from the directory installation.

1.7 Asterisell Customizations

1.7.1 Jobs/Workflow Customizations

Asterisell uses a blackboard approach to process information:

- events are put inside the database/blackboard;
- customizable/configurable jobs react to events;
- jobs can produce new events;

In this way it is possible to add new jobs, changing the behaviour of the application.

If you add new *jobs* make sure to add them to the *apps/asterisell/lib/jobs/userjobs*, otherwise during subsequent update of Asterisell application, all new jobs could be lost or there can be a lot of Git merge conflicts.

Some Jobs like PDF invoicing, email composition and so on require some customization. You can create a custom Job modifying already existing jobs, put them in *apps/asterisell/lib/jobs/userjobs* directory. The more elegant way is to change also the name of the job, add it to the *apps/asterisell/config/app.yml* file, removing the old job.

1.7.2 Rates Customizations

New type of rate methods can be added:

- add a new rate inside *apps/asterisell/lib/rates*
- enable the rate inside *apps/asterisell/config/app.yml*
- refresh the cache *./configure.sh*

1.8 Troubleshooting

1.8.1 File Permissions

Log Directory

Often after upgrades there are problems related to the Asterisell log files inside log directory.

If this file was created from root then the *apache* user is not able to add info to it and the entire Asterisell application is blocked. In this case change the permissions of file with something like:

```
cd log
chmod a+rw asterisell_prod.log
chmod a+rw asterisell_dev.log
```

or better change the owner in something like:

```
cd log
chown apache asterisell_prod.log
chown apache asterisell_dev.log
```

or execute:

```
./symfony fix-perms
```

Try also to restart the web server.

Created Files

During initial Asterisell configuration new files can be created. Check if these files are readable from web-server process. For example on my debian machine I performs:

```
chown -R :www-data *
chmod -R g+rx *
sh configure.sh
```

where *www-data* is the group used from my webserver.

1.8.2 Additional Run-Time Debug Info

In case of problems you can enable the development/debug version of Asterisell that shows useful informations about its execution and related problems. Execute:

```
./symfony enable asterisell dev
```

and open the url *http://your-web-url/asterisell_dev.php/login*

When finished remember to execute:

```
./symfony disable asterisell dev
```

NOTE: on the contrary of Asterisell production version, in the development version if you change Asterisell source code, you must not execute *sh configure.sh*, because the development version recreate all files every time in order to speed up development at the expense of run-time executions.

NOTE: as in the Asterisell product version, in the development version if you change some configuration parameters you must execute *sh configure.sh* because certain Asterisell modules are regenerated according the configured values.

1.8.3 Run Time Configuration Problems

Every problem encountered from Asterisell during the call rating process is clearly reported to the administrator with hints of how to resolve it.

The presence of problems is signaled also via email to the administrator.

In case of dubious configurations Asterisell advise the administrator and suspend the rate of affected calls.

The rate process is rather robust and error-free regarding ill defined configurations because:

- re-rating of calls is always possible;
- problems are always signaled;

1.8.4 Feedback

Web servers, PHP environment, libraries, and os on... can interact in strange ways. If something is not working properly let me know!

1.9 Security

The end user interacts only with the *Call Report* form. The content of user input fields is checked by a very conservative function that removes every non proper character.

No particular care is put on other forms because they are accessible only from the administrator.

User session handling is managed directly from Symfony and PHP engine.

1.10 MySQL advanced optimizations

For obtaining a 10x-20x speedup of call-report and related queries, the CDRs of last 1-2 months must be saved in the MySQL database cache in RAM. This makes a big difference in performances. Make sure having a *my.cnf* file like this:

```
[mysqld]
################################
# ASTERISELL RELATED SETTINGS #
################################
# Optimize InnoDB performances for Asterisell usage.
# NOTE: it will use the majority of the RAM of the server
# for Asterisell and other MySQL databases.
#
# Set this to 60-80% the RAM of the server
innodb_buffer_pool_size = 1G

sort_buffer_size = 50M
thread_cache_size = 4
tmp_table_size = 50M
sync_binlog = 0

# not very important
innodb_additional_mem_pool_size = 10M

# Commits are flushed on the disk each second,
# and not immediately.
innodb_flush_log_at_trx_commit = 0

innodb_lock_wait_timeout = 50
table_cache = 92
innodb_flush_method=O_DIRECT

# speed-up rating of CDRs and also other performances - very important
innodb_log_buffer_size = 8M
```

Restart MySQL for enabling changes in configuration.

CHAPTER 2

Upgrading

2.1 Create deploy directory

In order to reduce the downtime of main Asterisell application, you can test the new version in a separate directory, and then deploy the new version, when it is all ok.

Copy the current Asterisell directory content inside another directory:

```
cp -r <source-directory> <test-directory>
```

2.2 Merge your customizations with new Asterisell version

If you have installed the application using Git (the suggested way), these are the commands for free Asterisell version:

```
git commit -a -m "<DESCRIBE YOUR CUSTOM CHANGES>"
git pull http://github.com/massimo-zaniboni/Asterisell.git
```

These are the commands for the commercial Asterisell version:

```
git commit -a -m "<DESCRIBE YOUR CUSTOM CHANGES>"
git pull http://your-license:your-license@support.asterisell.com/asterisell/git/asterisell-commercial
```

where "your-license", must be replaced with your license code, obtained after purchasing the commercial version.

Then follow the Git procedure to resolve merge conflicts. Usually there will no merge conflicts, because Git tries to preserve both

- edit the files with problem,
- fix the >>>>>> .... <<<<<<< sections,
- signal to git that the conflict on the file is resolved using *git add <fixed-file>*,
- check pending conflict with *git status*,
- commit the final merged version when done with *git commit -a*

In case of troubles, ask to the forum. I will update these installation notes, according the signaled problems/suggestions.

2.3 Database Upgrade

Versions with the same major version number, use compatible database schema with minor changes, so they can be easily upgraded.

For security reasons, make first a backup of current database using something like:

```
mysqldump -u <your-user> <your-database> --single-transaction > asterisell3.sql
```

Note that the *–single-transaction* option allows to make the database backup, without locking MySQL database, and so new CDRs can be processed.

Then upgrade the current database using:

```
cd scripts
php upgradedb.php
```

2.4 Code refresh

After changing some code, for example after resolving merging conflicts, or after applying customization, you *must execute*:

```
sh generate.sh
```

in order to flush the Symfony cache and fix the permission on "log" and "cache" directories, and for generating some PHP files depending from application settings.

2.5 Application testing

If you use the same database both for production and new version testing, you must first disable the cron job processing of the production version, in order avoiding two job processors running on the same database.

If you use different databases, make sure to use again the production database, when you deploy the application.

2.6 Deploy

- Disable cron job processing.
- Rename the current production directory to another name.
- Rename the current testing directory to the name of production directory.
- Make sure using the correct production database.
- Use the correct read/write/ownership permissions for new production directory.
- Test web-site.
- Enable cron job processing.

CHAPTER 3

Configurations

3.1 Main Configurations

`apps/asterisell/config/app.yml` contains the main configurations of Asterisell. Once they are properly configured, they rarely change.

The meaning of various settings are explained directly inside the file comments. Details of some settings will be explained in other tasks. Initially set only parameters of witch you are sure, leaving default values otherwise.

In order to make active the changes, you must execute **./configure.sh**.

Main Configuration File Content.

3.2 Asterisell Owner Params

Use *Params* → *Params* → *Default* for defining the params of *Asterisell owner*.

3.3 Web Site Appearances

Logo, slogan, title, footer and similar aspects of the web-site can be customized directly from *Asterisell Owner Params*.

Web Site appearance can be further customized changing *apps/asterisell/templates/asterisell_layout.php* and *web/css* content.

3.4 Vendors Specification

Use *Parties* → *Customers / Vendors* for defining a *VoIP vendor*.

Use *Rates* for creating rates associated to the *VoIP vendor*, calculating the cost of calls.

You can define one or more vendors, and you can use different vendors for different calls, associating cost rates to corresponding VoIP vendor.

3.5 Customers Specification

Use *Parties* → *Customers / Vendors* for defining a *customer*.

Use *Parties* → *Customer Offices* for defining one or more *customer office* associated to a customer.

Use *Parties* → *VoIP Accounts* for defining one or more *VoIP account*, associated to an office.

The *call report* will display office and account information only when they are relevant. They are not displayed in case of logged customers with only one office, or only one VoIP account.

If a customer is not more active, disable the `Is Active` flag. Invoices associated to the customer will not generated.

3.6 Telephone Prefixes in Call Report

The call report contains something like

Tel.Tr.	Location	Connection
3932894234XXX 1707782466XXX	Italy USA	Mobile Line Fixed Line

The table associates a *location* and a *connection type* to a telephone number.

This association is done inspecting the *Params* → *Telephone Prefixes* table:

- A prefix like "39" can be associated to a generic "Italian Operator".
- A prefix like "393" can be associated to a generic (but more specific) "Mobile Italian Operator".
- A prefix like "39328" can be associated to the specific "Wind Mobile Italian Operator".
- A telephone number is associated to its more specific prefix. So "3944444" is associated to "Italian Operator", while "39344444" to "Mobile Italian Operator" and finally "393284444" to "Wind Mobile Italian Operator".

CDRs must be rerated again, in order to see the effects of changes in the table.

Telephone Prefix table is not used in rating process, but only in call report.

Initial Asterisell installation already load a complete list of standard world-wide prefixes. You can manually add custom prefixes associated to your specific settings.

The table can be completed also importing from a CSV file:

- *Params* → *Rates*.
- Create a pseudo rate of type *CSV File with Tel.Prefixes and Rates*.
- Set the rate according the format of the CSV file.
- Make sure activating *Update also prefix table*.
- Press *Save* button.
- Note that in default settings, local numbers are stored internally in their complete form with the international prefix. Make sure that it is part of the table.
- Delete the rate.
- Check the prefix table.

3.7 Rates

Rates classify every *CDR*, and associate a cost and an income to them.

3.7.1 Classification Rates

A new *CDR* starts with state unprocessed, and they are initially managed from *classification rate*. They inspect mainly the amaflags and disposition fields where Asterisk server put billing hints/information.

A classification rate classifies a call:

- as `outgoing`, if the call is made from the customer to an external telephone number;
- as `incoming`, it the call is from an external telephone number to the customer VoIP number;
- as `internal`, if the call is made from the customer to an internal VoIP account;
- as `ignore`, if the CDR can be ignored;

3.7.2 Telephone Number Normalizations

Advanced CDR processing can be performed from customized `classification rates`, that are usually created from devel

- recognition of complex patterns for local/mobile/interstate/intrastate telephone numbers;
- ad-hoc formatting of telephone numbers according their type;
- recognition of off-peak, peak or week-end calls, according the call-time;

These operations are performed by custom PHP code, so you must consult Asterisell assistance and support.

3.7.3 Ignored CDRs

CDRs of type `ignore` are not further processed:

- they are not displayed in the *call report*;
- they are not billed;
- their fields can be in a bad format;

They remain in the *CDR table*, but they are ignored.

Note that they are not deleted, and in case of change of the classification rate, usually during initial phase of Asterisell customized deployment, the ignored CDRs can be re-rated as all other CDRs, and in case they can become CDRs of other type, if the classification logic is changed.

3.7.4 Internal CDRs

CDRs of type `internal` are calls between two local VoIP users.

Usually they have no cost, so they are usually associated to rates saying their cost and income is simply 0.

They can be displayed or not displayed in the *call report*, according the setting `show_internal_calls` in *Main Configurations*. If they are not displayed in the call report, then they are also ignored during the invoice phase. The administrator views always all the type of calls, except ignored calls, because he must have a complete vision of CDR table content.

3.7.5 Incoming Calls

Incoming calls usually have no cost, so they are usually associated to rates saying their cost and income is simply 0.

They can be displayed or not displayed in the *call report*, according the setting `show_incoming_calls` in *Main Configurations*. If they are not displayed in the call report, then they are also ignored during the invoice phase. The administrator views always all the type of calls, except ignored calls, because he must have a complete vision of CDR table content.

3.7.6 Revenue Sharing

A call can have a negative cost. In this case you (as *Asterisell owner*) are earning something for the call. There can be a revenue sharing between the *VoIP vendor* and you (as *Asterisell owner*). This can be the case for incoming calls for mobile telephone numbers.

A call can have a negative income. In this case there can be a revenue sharing between your customer and you.

Separate invoices can be created for revenue sharing, setting the `Is Revenue Sharing` field, during invoice generation. All the CDRs with negative incomes are part of revenue sharing invoices.

3.7.7 Normal Rates

CDRs of type `outgoing`, `incoming`, `internal` are processed from `normal rates`, for determining their cost and income.

Income rates are associated to the customers of a specific price category.

Cost rates are associated to a specific *VoIP vendor*.

3.7.8 Rate Priorities

There can be one or more rates applicable to a CDR. It is always used the most specific rate. When there are doubts about which is the most specific rate, an error is generated, and the CDR remains unprocessed.

Every rate have a priority method. Two rates are comparable if they have the same priority method. If there are two rates applicable to the same CDR, but with two different priority method, an error is generated, because there is no way for determining what is the most specific rate.

Usually classification rates use destination channel as priority method, and normal rates the *external telephone number*.

For example given:

- a rate `A` with the *external telephone number* priority method, matching numbers like `123`;
- a rate `B` matching numbers like `1234`;
- a telephone number `AA` like `12355555`;
- a telephone number `BB` like `12344444`;

then the rate `A` is the best match for `AA`, while `B` is the best match for `BB`.

A rate can have the `exception` field activated. They have a greater priority respect no exception rates. The most specific exception is selected, if there is any. This allows creating special rating rules, having priority methods different from other normal rates.

3.7.9 Price Categories

Income rates have a price category. They are applicable only to customers of the same price category. Example of price categories can be `normal`, `discounted`, `reseller`.

Price categories must be assigned to every *customer*, and optionally to *customer office* when they are different from the price category of the owner customer, and optionally to *VoIP account* when they are different from the price category of the owner office. The price categories are obviously specified inside *Parties* → *Customers / Vendors*, *Parties* → *Customer Offices*, and *Parties* → *VoIP Accounts*.

3.7.10 Vendor Rates

Vendors are typically identified by the `dstchannel` field of the *CDR table*. This field is set from the Asterisk server and it identifies the channel used to route the call.

If you have different *VoIP vendors*, then you must specify different cost rates, associated to different `dstchannels` and *VoIP vendors*.

3.7.11 Rates Specified Using CSV Files

CSV files are useful for specifying rates associated to *external telephone number* according their initial prefix. A single CSV file can contain the rating parameters of thousands of telephone prefixes, and it can be managed from Asterisell and from *Asterisell owner* in a lot more efficient way respect thousands of different rates.

Asterisell is shipped with a rather powerful rate, accepting a CSV file in many configurable formats. If it is not enough, it is possible *adding custom CSV based rates* following the format of the *VoIP vendor*. This is an example of CSV content:

```
"", "Mobile line" , "Afghanistan" ,  9375 , "0.1"
"", "Mobile line" , "Afghanistan" ,  9377 , "0.1"
"", "Mobile line" , "Afghanistan" ,  9378 , "0.1"
"", "Mobile line" , "Afghanistan" ,  9379 , "0.1"
"", "Fixed line" , "Albania" ,  355 , "0.1"
"", "Mobile line" , "Albania" ,  35567 , "0.1"
"", "Mobile line" , "Albania" ,  35568 , "0.1"
"", "Mobile line" , "Albania" ,  35569 , "0.1"
"", "Fixed line" , "Algeria" ,  213 , "0.1"
"", "Mobile line" , "Algeria" ,  2136 , "0.1"
"", "Mobile line" , "Algeria" ,  2137 , "0.1"
"", "Mobile line" , "Algeria" ,  2139 , "0.1"
"", "Fixed line" , "Algeria [Wataniya]" ,  2135 , "0.1"
"", "Fixed line" , "Algeria [Wataniya]" ,  21355 , "0.1"
"", "Fixed line" , "American Samoa" ,  1684 , "0.1"
"", "Fixed line" , "American Samoa" ,  684 , "0.1"
"", "Mobile line" , "American Samoa" ,  1684252 , "0.1"
"", "Mobile line" , "American Samoa" ,  1684254 , "0.1"
```

Up to date, there is no way in Asterisell for retrieving a loaded CSV file, again in his original format. So make sure having a copy somewhere of loaded CSV files.

3.7.12 Bundle Rates

Bundle Rates can be used for specifying things like the minimum income, or line rental costs. They are associated to an invoicing period, and all the CDRs of this period, but not to any CDR in particular.

3.8 Invoicing

Inside *Main Configurations*, set the options:

- `invoice_date_format`
- `long_details_in_invoice`
- `use_inclusive_end_date_in_invoice`
- `currency`
- `currency_ascii_char`
- `currency_decimal_places_in_invoices`
- `culture`

Invoices are generated from the jobs:

- `GenerateInvoices`
- `GenerateInvoiceDetails`
- `GenerateInvoiceReportAsHTML`
- `GenerateInvoiceReportAsPDF`
- `GenerateCallReportAsPDF`
- `GenerateInvoiceEmail`
- `SendInvoiceEmails`
- `SendInvoiceEmail`
- `SendInvoiceNotificationToAccountant`

In case of invoices with a customized layout, you can substitute the job `GenerateInvoiceReportAsPDF` with another custom job.

Set *Asterisell Owner Params* for setting invoice seller params.

Emails and invoices are generated using the culture specified in `apps/asterisell/config/app.yml` file.

Emails con be customized replacing the job `GenerateInvoiceEmail`, with a custom job:

- a new job like `GenerateAcmeInvoiceEmail` must be created;
- the job must be added to `jobs` list, inside *Main Configurations*, and the old `GenerateInvoiceEmail` must be removed;

3.8.1 Report about Calls

The job `GenerateCallReportAsPDF` generates a detailed call report about the type of calls:

- 30 Most Expensive Outgoing Calls;
- 30 Longest Duration Outgoing Calls;
- 30 Most Frequently Dialled Numbers;
- 30 Most Expensive Dialled Numbers;
- calls ordered by destination type;

It can be sent to customers, in order show them how they used your services.

CHAPTER 4

Usage

4.1 User Interface

4.1.1 Filters

All textual filters accessible from the administrator, require an ending `*`. For example a filter like `abc*`, select all the strings having `abc` as prefix.

Filters inside *call report* add implicitly the `*` character to each search term, in order to simplify its usage for end users.

4.1.2 Online Help

Read carefully the notes of fields, and the comments at the bottom of each form.

4.2 Rating

Make sure understanding the *basics of CDRs rating process* before defining and changing rates.

4.2.1 Error in Rate Configurations

During rating process, Asterisell signals rate errors and conflicts. Rates are usually the most difficult thing to configure in Asterisell, so at least initially, errors are the norm.

Asterisell has a safe approach during rating: a *CDR* is rated only if there is exactly one applicable classification rate, one cost rate, and one income rate, otherwise an error is signaled, and the CDR rating is postponed until rate configurations are fixed. So Asterisell prefers annoying the administrator with error-messages, than silently rating in the wrong way the VoIP calls.

Error notifications contain hints about the solution of the problem, so read them carefully.

If properly configured, errors are notified to Asterisell administrator also by email:

- job `AdviseAdminOfNewProblems` in *Main Configurations* must be active;
- `Customers support email, and automatic problem notification email` field in *Asterisell Owner Params* must be specified;

Unrated CDRs

CDRs with problems are not displayed to customers, but they are only visible to administrators in the *Calls → Unprocessed Calls* web form. Note that in this section there are also the few CDRs that were inserted in the CDR table, after the last cron-job processing job. After the next rating process, they will be moved to the *call report*.

Examples of Problem Solution

An example of error notification:

> Description: No Rate to apply at date 2011-06-08 of type vendor (calculating a cost) on CDR with id `11906`, and destination_type `outgoing` for customer of price category `Normal` for external telephone number `0033631988888` (with number portability applied it is `0033631988888`), and for dstchannel `SIP/abcd-12`.
>
> Effect: CDRs in the given rate interval will not be rated.
>
> Suggested Solution: Complete the rate table and wait for the next rate pass. Use menu entry Calls/Unprocessed Calls, for inspecting all details of unprocessed calls.

Its solution:

- check all the active rates of type vendor, active at date 2011-06-08;
- check if they are applicable to the telephone number `0033631988888`, and price category `Normal`;
- fix the rate for supporting also this type, or add a new rate;
- delete the problems from the list;
- re-rate old CDRs that can be affected by the changes in rates;
- check if the error is signaled again in the error table;
- note that errors generated during on-line rerating are not sent to the administrators emails, in order to reducing the number of sent emails during administration and maintenance work.

4.2.2 CDRs Re-rating

- open *Calls → Call Report*;
- select the time frame of CDRs you want re-rate;
- note that other filters take no effect during re-rating, only the time-frame criteria;
- click on *Re-rate Calls in Time-frame*;

4.2.3 Rates Updating

In Advance

If your *VoIP vendor* communicates you the changes of rates before they take effect, you can create the corresponding new rates, specifying the start date and time. They will be applied only to CDRs after the starting date. You must also set the ending date of current rates, because after the new rates are active, they must be disabled.

This is the best way for updating the rates, because if you rerate CDRs, Asterisell can always apply the correct rate.

Current

If you update current rates, because there are errors, you must rerate current CDRs, otherwise the modified rates will be applied only to new inserted CDRs. CDRs are rerated *in this way*.

Changing Customer Price Category

Up to date customer price category has no start and end date, as in the case of rates. A customer has only one price category, and if you change it, it takes effect immediately. All new CDRs associated to the customer will be calculated according the new price category. If you want apply the price category also to old CDRs, you must rerate them *in this way*.

A drawback of this approach it is that if you rerate old CDRs of the customer, they will be rated according the current price category, and not the price category when the CDRs were made. This is obviously a problem in Asterisell specification of price categories. In practice this problem it is not severe, because usually a customer changes his price category at the beginning of the new invoicing period, and rarely CDRs already billed are re-rated. So the typically way for changing price category, is rerating all the CDRs of the current month.

4.3 Invoices

4.3.1 Generation

Accounting → *Customer Invoices* for generating the invoice of a single customer.

Accounting → *Batch Invoice Creation* for generating all invoices of active customers, of a specific billing period.

Accounting → *Vendor Invoices* for generating the amount you owe to your *VoIP vendor*, according the costs calculated from Asterisell. The generated pseudo invoices should correspond to the invoices received from vendors.

If you change the cost of calls already invoiced, then the invoice is not updated automatically. You can force a regeneration of an invoice using the `Regenerate Invoice button` which is displayed in the edit form of an invoice. It can be used also for batch invoices, and all invoices are simply updated.

4.3.2 Sending Emails

Email is sent only if the `Send email to customer` button is pressed and the customer has an associated email.

If there were a problem sending an email, then Asterisell signal the problem, and mark the invoice as not sent.

If SMTP params are left incomplete, then Asterisell will simply not send emails, and it does not consider this an error.

If SMTP params are not correct then Asterisell signal the problem inside problem table.

CHAPTER 5

Partners and Resellers

5.1 Resellers

A *reseller* is a company selling VoIP calls to his customers, and where you play the role of his VoIP vendor.

In case of resellers, there are two running instances of Asterisell:

- your main instance;
- reseller instance;

A reseller has distinct:

- Asterisell website address;
- Asterisell database;
- customers;
- rates;
- invoices;

Usually a reseller uses your VoIP servers for managing the calls of his customers, and usually you install the Asterisell reseller instance on the same server of your main instance, but this is not mandatory.

5.1.1 License

You need a separate license of Asterisell for each reseller.

5.1.2 Main Asterisell Instance Configuration

In your main Asterisell instance you must:

- make sure that in *always_scheduled_jobs* of *Main Configurations*, the job *ExportCDRToReseller* is active.
- create a customer for the reseller;
- enable the *Is Reseller* flag in the *Parties* → *Customers / Vendors* form;
- configure the *reseller short code*;

Usually resellers have also a reseller price category.

From now all the CDRs of VoIP accounts associated to the reseller will be exported to CSV files, and put inside the directory *cdr-exported-to-resellers/reseller-short-code*. The directory will be created automatically from Asterisell. The CSV files have name like *cdr-12345678.csv* where *123456789* is a unique and progressive number.

In case of installation of reseller and main instance on the same machine, make sure that the main instance export directory is readable and writable from the cron-job.

Supposing your reseller *Gold* has customer *Acme*, with VoIP account *acme001*:

- you must enable VoIP account *acme01* on your VoIP server;
- in Asterisell, you must create the VoIP account *acme01* associated to customer *Gold*;
- you *must not* create customer *Acme* on your Asterisell instance;

So you don't know anything about the customers of your reseller, and the applied rates. You only know about the VoIP accounts you manage, and what a reseller must pay you for this service.

Asterisell will produce a unique invoice with all the calls of the VoIP accounts associated to the reseller customer. This is the usual Asterisell work-flow, because a reseller is a normal customer inside Asterisell main instance.

5.1.3 Reseller Instance Configuration

- Install Asterisell for the reseller.
- Set *external_asterisell_voip_provider* options inside *Main Configurations*. If the two instances are put on separated computers/network, you must create some script copying new CSV files in the main Asterisell instance, in a local directory of the reseller instance.
- Make sure that in *always_scheduled_jobs*, the jobs *ImportCDRFromAsterisellProvider* and *CompareProviderCostWithCalculatedCost* are active.
- Set vendor rates equal to the rates you apply to reseller on the main Asterisell instance. Note: CDRs imported on the reseller instance will be checked, and the reseller will be informed if there is some difference between the cost of VoIP calls calculated from main instance, and the cost calculated from reseller instance, applying the vendor rates.
- Remove all classification rates, and add the *ProcessImportedCDR* classification rate. This rate recognizes CDRs imported from the main Asterisell instance.
- Create initial customers, using the same VoIP accounts enabled on the main instance. Note: it is not important activating accounts first on reseller or first on main instance, because data will be not lost in any case.

Reseller instance will produce an invoice for each of his customers. This is the normal Asterisell work-flow, because every reseller customer, is a normal customer inside reseller instance.

5.1.4 Rerating of CDRs

If already exported CDRs will be rated again on the Asterisell main instance (for example after changing rates, or fixing errors), they will be exported again to the reseller, and reseller instance will update the old CDRs with new values, without creating duplicated/conflicting CDRs.

So reseller instance can recognize and manage properly all events like:

- a CDR changed cost/income;
- a CDR changed VoIP account;
- a CDR changed outgoing/incoming/internal state;

- a CDR changed destination/source telephone number;
- and so on..;

The only exception is in case of CDRs becoming *ignored call* on the Asterisell main instance, after they were already sent as a different CDR type. In practice this is not a big problem because rules about CDRs to ignore, rarely change.

5.2 Partners

A *partner* is another company collaborating with you, that uses your same Asterisell instance, and the same VoIP server.

Partners can have distinct:

- customers;
- invoices;
- login page;
- web site logo, header, footer;
- rates;

All partners have the same administrator privileges on the same Asterisell instance, so they can read and change all the data of other partners.

5.2.1 License

On the contrary of resellers, partners can use both the same Asterisell license, because it is the same Asterisell instance.

5.2.2 Configuration

- Create a partner on *Params* → *Params*.
- Associate customers to their partner.

5.2.3 Invoices

Each customer invoice will use as seller the corresponding partner.

5.2.4 Custom Login Page

After login a customer see always the logo, header and footers of his associated partner.

Before login it is not possible, because there is a unique Asterisell web address for each partner. So you must use a common header for all partners.

It is possible creating a custom login page, with a specific address, for each partner.. In *Params* → *Params* web form, you can specify in *Login URN* a URL-friendly short name for the partner. Then the custom login page with the headers and footers of the partner is accessible on something like *https://www.example.com/partner-name*.

5.3 Main Configuration File Content

apps/asterisell/config/app.yml

```
# !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!! #
# !!!                                                                          !!! #
# !!! I M P O R T A N T:                                                       !!! #
# !!!                                                                          !!! #
# !!! 1) if you change these values then execute "configure.sh" in order to    !!! #
# !!! update the application behaviour;                                        !!! #
# !!!                                                                          !!! #
# !!! 2) many parameters need a rerate of old calls in order to be propagated  !!! #
# !!! also to old calls. When this is the case, it is signaled in the parameter !!! #
# !!! comments as "IMPORTANT"                                                  !!! #
# !!!                                                                          !!! #
# !!! 3) avoid TABS inside this file, they are not accepted, use only SPACES   !!! #
# !!!                                                                          !!! #
# !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!! #

all:
  available:

    # The PhpRate class available during the setup of a rate plan.
    # The class are located in apps/asterisell/lib/ directory.
    # You can add your own classed.
    # The format is:
    #
    # > PHP-Class-Name: Rate Description
    #
    phpRates:
      PhpCDRProcessing: CDR initial classification
      PhpRateByDuration: By Duration
      PhpRateInCommercialVersion: CSV File with Tel.Prefixes and Rates

    # These jobs are executed every time the job-queue processor
    # is invoked. They are indipendent from job-data or events.
    # They are executed before "available jobs".
    #
    # These jobs are processed following the order of declaration.
    #
    # These jobs are processed only one time for each activation
    # of the JobQueueProcessor.
    #
    # These jobs are of class FixedJobProcessor.
    #
    # NOTE: if one of this jobs fails, then the next job is executed
    # in any case.
    #
    # IMPORTANT: Changes to this parameter make effect only on new rated calls,
    # so you must force a rerate if you want apply the new mask only to already
    # rated calls.
    #
    always_scheduled_jobs:
      - RateCalls
      - CleanCacheFromOldItems              # maintanance work
      - CheckWebsiteUpdates                 # check if there are news on Asterisell web-site
      - CheckPartiesWithoutParams           # maintanance work
      # - CheckCallCostLimit                # check if customers have reached theirs call cost limit
```

```
    # - CheckSafeLimitForConcurrentCalls  # check if the server has reached his limit about the max
    - AdviseAdminOfNewProblems            # send emails with found problems to administrator


  # The jobs that are activate when there is a compatible event
  # on the job queue.
  #
  # Jobs are iteratively checked for every initial and new event,
  # so if a Job fires new events, they can be immediately processed
  # from other jobs.
  #
  # These jobs are of class JobProcessor.
  #
  # IMPORTANT: Changes to this parameter make effect only on new rated calls,
  # so you must force a rerate if you want apply the new mask only to already
  # rated calls.
  #
  jobs:
    - WarnCustomerForHighCallCost  # This job is executed only if it active also the 'CheckCallCost

# Complete only if this Asterisell instance is a reseller of another
# Asterisell server acting like a VoIP provider.
#
# NOTE: if this is the Asterisell acting like a main VoIP provider, these
# settings must be left empty. They must be completed on the reseller side.
#
# NOTE: these options can be used only for commercial version of Asterisell.
#
external_asterisell_voip_provider:
  - csv_files_source_directory: ""
  - temp_processing_csv_files_directory: ""
  - processed_csv_files_directory: ""
  - insert_cdr_using_dstchannel: ""

# Enable/disable displaying of incoming/incoming/internal
# calls in the customer call report.
# This allows to reduce the information overhead.
#
# This settings influence also INVOICE GENERATION:
# if incoming/internal calls are not displayed in the CALL REPORT
# then they are not displated / summed up in the invoices,
# otherwise they are showed / summed up in the invoices.
#
# IMPORTANT: if you assign an earn to incoming and internal
# calls, but they are not displayed in the call report,
# then these earns are lost because they are not inserted
# in the invoices. The underline philosophy of Asterisell
# is that you can account to customers only showed costs
# in the call report.
#
# Ignored and Unprocessed calls are not displayed by default.
#
# The administrator can all these type of calls,
# in any case, indipendently from these settings.
#
# allowed values: true / false (case sensitive)
#
show_incoming_calls: false
```

```
show_outgoing_calls: true
show_internal_calls: false

# Display in the customer call report also
# the direction of the call (incoming/outgoing/internal)
# for each call.
#
# The administrar is not affected from this setting, because
# he sees always the call direction.
#
# If both incoming and outgoing calls are displayed, the safer
# approach is enabling also displying of call direction, otherwise
# can be difficult for the end-user, distinguish between the type
# of a call.
#
# allowed values: true / false (case sensitive)
#
show_call_direction: false

# Display in the customer call report
# also a filter on the call direction (incoming/outgoing/internal).
#
# This setting influence only the call report
# of customer. The administrator can always
# filter on call direction.
#
show_filter_on_call_direction: false

# How many calls shows in the call report
# containing the list of made calls.
# This is the main report of the program.
#
how_many_calls_in_call_report: 30

# External telephone number with the specified
# prefix are displayed in a short form, without
# the prefix.
#
# This option is useful when all your customers
# reside in the same area. So you can remove
# the standard/default prefix from called numbers
# when they are in the standard/default area.
#
# All other numbers are not modified.
#
# In any case, this option does not change the
# stored numbers inside the database, only their
# visualization.
#
# Use "-", if you want to disable this feature.
#
# IMPORTANT: Changes to this parameter make effect only on new rated calls,
# so you must force a rerate, if you want apply the new settings
# also to already rated calls.
#
not_displayed_telephone_prefix: "-"

# Check cost-limits interval expressed in minutes.
```

```
# Cost-limits check is an heavy operation to do and
# it can be executed after a certain interval of time.
#
check_cost_limits_after_minutes: 60

# In case the job GenerateMailWarningCustomerForHighCallCost
# is enabled (it is inserted in the list of active jobs),
# this params set how often a customer is warned
# about his call cost exceeding his cost limit.
#
# 0 for disable customer advise (only admin is advised).
#
repeat_advise_of_high_cost_limit_after_days: 0

# How to calculate the max cost limit for each customer.
#
# "30" for considering the cost of last 30 days.
# "m" for considering the current month.
#
max_cost_limit_timeframe: m

 # Used from job CheckFrauds (enabled only in commercial version)
 # The job run according the scheduling of `check_cost_limits_after_minutes`
 #
 check_frauds:
   max_unprocessed_calls: 10
   # if there are more than these unprocessed calls,
   # then advise the administrator, because there can be unnoticed high calls costs

   compare_xx_last_days_respect_customer_daily_max_cost: 5
   # note: customer daily max cost is the customer monthly max cost, specified in the user interfac
   # divided by 30. Set to 0 for disabling it.

   compare_x_last_months_respect_current_costs: 3
   # warn if a customer spent a 25% more than the maximum cost of last x months in the
   # same day of week and a comparable hour of the day (hours grouped by 00-06, 06-12, 12-18, 18-24
   # Set to 0 for disabling it.

# How many concurrent calls, the Asterisk server can support in a safely
# manner. If this number of concurrent calls is reached,
# then the administrator is advised.
#
# This number depends from the incoming/outgoing bandwidth of the Asterisell
# server and from the bandwidth reserved/used from each call/connection.
# It is only an indicative/warning value, used for monitoring when
# the system reach a warning usage level.
#
safe_limit_for_concurrent_calls: 5

# The format to use for date/timestamp display in CDR call report.
# The format is specified using PHP "date" function format:
#    http://www.php.net/manual/en/function.date.php
#
date_format: "Y/m/d, G:i:s"
  # "r" for something like "Thu, 21 Dec 2000 16:01:07 +0200"
  # "c" for something like "2004-02-12T15:19:21+00:00"
  # "F d, Y G:i:s" for something like "Dec 21, 2000 16:01:07"
  # "d/m/Y, G:i:s" for italian format
```

```
# The format to use for date display in Invoices
#
# IMPORTANT: Changes to this parameter make effect only on new generated invoices.
#
invoice_date_format: "Y-m-d"
# invoice_date_format: "d/m/Y"  for italian format

# true for having invoices with details like:
# > Italy - Fixed Line        200 calls  ....
# > Italy - Mobile Line       100 calls  ...
# (grouped by geographi location and type of operator)
#
# false for having invoices with details like:
# > Fixed Line                 200 calls
# > Mobile Lines               100 calls
# (grouped by type of operator.
#  Maybe you can further create operators like "National Fixed Line",
# "International Fixed Line" and os on...)
#
long_details_in_invoice: true

# true for using the last day of the month, in invoices of the month,
# like "from 2011-06-01 to 2011-06-30".
#
# false for using the first day of the next month, in invoices of the month,
# like "from 2011-06-01 to 2011-07-01".
#
use_inclusive_end_date_in_invoice: false

# the default currency used for rates, cost and incomes.
#
currency: EUR
# currency: USD
# currency: AUD

# The ASCII char to use for representing the currency
# in PDF and textual forms.
#
# EUR: 128
# US: 36
# GBP: 163
#
# IMPORTANT: Changes to this parameter make effect only on new generated invoices.
#
currency_ascii_char: 128

# The decimal separator symbol to use in CSV files exported to customers.
#
decimal_separator_symbol_in_csv: "."

# true for exporting numbers like "1.234" in CSV files exported to customers.
# false for exporting numbers like 1.234 (without \").
#
# true is a more safe option of your customers have different locale
# with different types of decimal separator.
#
# false allows recognizing more easily numbers, during importing of CSV files
# inside a spreadsheet.
```

```
#
numbers_as_strings_in_csv: true

# The field separator used in CSV files exported to customers.
# Normally it is ",".
#
# If you are using "," also as decimal separator,
# and numbers are not surrounded between \", then use something like ";".
#
csv_field_separator: ","

# The default culture to use for formatting numbers, dates and so on.
# Something like en_US, it_IT, ...
#
# IMPORTANT: if you change this value update also the corresponding
# "apps/asterisell/config/settings.yml" file in "prod: default_culture".
# This is not strictly necessary because after login, Asterisell
# set the culture of the session to this value, but it is
# a safe setting.
#
# culture: en_US
# culture: it_IT
culture: en_US

# decimal places to use for currency when stored in the database CDR table
#
# IMPORTANT: if you change this value then force a re-rate of all calls
# because already rated calls will be in an inconsisten format.
#
currency_decimal_places: 4

# decimal places to use for each currency in an invoice
#
# IMPORTANT: Changes to this parameter make effect only on new generated invoices.
#
# IMPORTANT: this parameter is also used for rounding
# costs/incomes in CALL REPORT. Supposing 2 decimal places,
# a cost like "0.005" is rounded to "0.01".
#
currency_decimal_places_in_invoices: 2

# If these field is an empty list ([]) then "cdr.dst" field is the destination
# telephone number of the call.
#
# If the value of this setting is a list then
# cdr.lastdata field is the destination telephone number of the call.
# For each billable call, "cdr.lastapp" must be a value inside the list,
# otherwise the call is not rated and the problem
# is signaled to the administrator.
#
# IMPORTANT: Changes to this parameter make effect only on new rated calls,
# so you must force a rerate if you want apply the new mask only to already
# rated calls.
#
lastapp_accepted_values: []

# The internal telephone number is the VoIP account code,
# managed from the Asterisk server.
```

```
#
# The external telephone number is the number residing on
# external lines, reachable using the services of the
# VoIP service provider.
#
# In an outgoing call, the internal telephone number is the source of the call.
# In an incoming call, the internal telephone number is the destination of the call.
#
# In an outgoing call, the external telephone number is the destination of the call.
# In an incoming call, the external telephone number is the source of the call.
#
# Asterisell displays in CALL REPORT something like:
#
# > | INTERNAL VoIP Account | Call Direction | EXTERNAL Telephone Number |
# > +-----------------------+----------------+---------------------------+
# > | account123            |    outgoing    | 390423XXXX                |
# > | account123            |    incoming    | 39343XXXXX                |
#
# so the distinction is between internal and external telephone number
# and not between call source and destination.
# The same distinction is done also in the rate forms.
#
# The problem is: what fields of the CDR record must Asterisell
# use for displaying in the call report the internal and external telephone number?
#
# Available options:
# * 0: use "accountcode" field as internal telephone number (display in CALL REPORT the account-coo
#      use "dst/lastapp_accepted_values" field as external telephone number.
# * 1: use "src" field as internal telephone number only for outgoing calls,
#      use "src" field as external telephone number only for incoming calls,
#      use "dst/lastapp_accepted_values" as external telephone number only for outgoing calls,
#      use "dst/lastapp_accepted_values" as internal telephone number only for incoming calls.
#      in case of unprocessed/ignored/internal use always "src" field for internal calls.
# * 2: use "src" field as internal telephone number,
#      use "dst/lastapp_accepted_values" field as external telephone number.
#      in case of unprocessed/ignored/internal use always "src" field for internal calls.
# * 3: it is assigned from the CDRInitialProcessing rate, that must complete during initial process
#      these fields:
#        - `cdr.ar_asterisk_account_id` (null if it is missing, and RateProcess will inform the use
#        - `cdr.cached_internal_telephone_number`;
#        - `cdr.cached_external_telephone_number`;
#        - `cdr.cached_masked_external_telephone_number`, can be null (it is completed from the fra
#        - `cdr.ar_telephone_prefix.id` can be null (it is completed from the framework),
#           or with an explicit value, in this case number portability is not automatically transfe
#           and it must be explicitely managed from the CDR processor;
#      See `apps/asterisell/lib/rates/CustomCDRProcessing.php` for a template of associated process
#      A rate like the proposed example must be added to the rate table.
# * 4: use "accountcode" field as internal telephone number (display in CALL REPORT the account-coo
#      use "dst" as external telephone number for outgoing and internal calls,
#      use "src" as external telephone number for incoming calls.
# IMPORTANT: Changes to this parameter make effect only on new rated calls,
# so you must force a rerate if you want apply the new mask only to already
# rated calls.
#
internal_external_telephone_numbers: 0

# An external telephone number can be masked in the CALL REPORT view,
# for privacy reasons.
```

```
#
# This parameter identifies the number of last digits to mask in the
# external telephone number.
#
# A typical value is 3 digits to mask, in order to obtain something
# like "39059567XXX".
#
# 0 stays for no number masking.
#
# Internal calls (in the case they are displayed in the
# CALL REPORT), between two VoIP accounts are no masked.
#
# Unprocessed calls are never masked, but they are viewable
# only from the administrator.
#
# IMPORTANT: Changes to this parameter make effect only on new rated calls,
# so you must force a rerate if you want apply the new mask only to already
# rated calls.
#
# IMPORTANT: Changes to this parameter make effect only on new rated calls,
# so you must force a rerate if you want apply the new mask only to already
# rated calls.
#
mask_for_external_telephone_number: 3

# Remove a log message from the Job Log Table
# when it is older than this number of months.
#
months_after_removing_a_job_log_entry: 2

# Where uploaded files are put/read.
# This is a directory inside "web" directory,
# and it must be web-server readable and writable.
#
# This default directory is already created.
# If you set up a new directory, you must manually
# create it.
#
sfMediaLibrary:
  upload_dir: uploads/assets
```

CHAPTER 6

Development

6.1 Project Repository

This is the project page http://github.com/massimo-zaniboni/Asterisell

6.2 Symfony Framework

Asterisell uses the Symfony PHP framework, that is a well designed and documentated framework.

Asterisell distribution contanins a "snapshot" of the Symfony framework, version 1.0.5. The directory *symfony-patch* contains some patches to the framework. They are already applied to the version shipped with Asterisell.

6.3 Debug

In order to enable the development mode execute:

```
symfony enable asterisell dev
```

If Asterisell project is in the */var/www/asterisell* directory and your local apache server is listening on port 3000, then you must open the web page *http://localhost:3000/asterisell/web/asterisell_dev.php/login*.

6.4 Localization

If you want to support a new language / culture you must:

- add to *apps/asterisell/config/app.yml* the new currency and culture;
- copy *apps/asterisell/i18n/messages.it.xml* to *apps/asterisell/i18n/messages.your_culture_code.xml*;
- replace all Italian translations with your locale translations;
- execute "sh configure.sh" in order to clear the cache and view the new messages;

CHAPTER 7

Glossary

Asterisell owner You: a company selling VoIP services to his *customers*. It uses one or more *VoIP vendors*, and it can have optionally one or more *partners*.

bundle rate A rate associated to the set of CDRs of the invoicing period, as the minimum income, or line rentals.

call report The web page displaying all the VoIP calls, and related data to administrator or to end users. It displays specialized information according the type of logged user.

CDR A call detail record of the *CDR table*, containing all the details of a VoIP call.

CDR table A MySQL table where the VoIP server put details about the VoIP calls.

classification rate A rate that classifies the *CDR* as *outgoing call*, or *incoming call*, or *internal call*, or *ignored call*.

cost rate A rate that assigns the cost of the *CDR*, that is what you (the *Asterisell owner*) must pay to another *VoIP vendor* for routing the calls.

customer A party that uses the services of *Asterisell owner* for making VoIP calls. He can have one or more *customer offices*.

customer office It identifies distinct physical locations, or distinct business units, of a *customer*. It can have one or more *VoIP accounts*.

external telephone number It is the number residing on external lines, reachable using the services of the *VoIP vendor*. In case of *outgoing call*, it is the destination of the call. In case of *incoming call*, it is the source of the call.

ignored call A call that has no income, cost, and that is not displayed in the *call report*. It can be completely ignored.

income rate A rate that assigns the income of the *CDR*, that is what the *customer* must pay for it.

incoming call A call coming from a telephone network not directly managed from you, and having one of your customer *VoIP account* as destination.

internal call A call inside a telephone network directly managed from you. The call source and destination are usually both associated to your customers.

internal telephone number It is the *VoIP account* code. In case of *outgoing call*, it is the source of the call. In case of *incoming call*, it is the destination of the call.

outgoing call A call directed to a telephone network not directly managed from you, and having one of your customer *VoIP account* as source of the call.

partner A Partner is a company collaborating with you, that uses your same Asterisell instance for selling VoIP calls to his customers.

reseller A Reseller is a company selling VoIP calls to his customers, and where you play the role of his *VoIP vendor*.

VoIP account A VoIP internal telephone number managed from the Asterisk server. It is an internal account code. Usually Asterisk VoIP server put it on the `accountcode` field of the *CDR table*. Note: the Asterisk standard documentation pretends that this code is specified on a per-channel basis, but this is not the case in Asterisell configuration. So each accountcode has the same associated customer for every channel. This is also useful/practical in case of the same customer using different channels.

VoIP vendor A company that can route VoIP calls to destination end-points, typically traditional telephone networks. It is used from the *Asterisell owner* for routing the VoIP calls.

Index