

A²Billing 1.4

INSTALLATION GUIDE 1.0

Alberto Escudero-Pascual aep@it46.se

Louise Berthilson louise@it46.se

(cc) Creative Commons Share Alike Non-Commercial Attribution 3

This guide focuses on the installation of a billing system for the Asterisk open source PBX. The document covers the installation of *a2billing*. A2billing is an open source implementation of a telecommunication billing and added value services platform.¹

A2billing is a LAMP (Linux Apache Mysql(Postgresql) PHP) application that interfaces with Asterisk using both the AMI and AGI interfaces.

This document has been tested after using a2billing trunk with Debian etch, Debian etch and half, Ubuntu 8.04 and Ubuntu 8.10. (27th March 2009)

¹ <http://www.asterisk2billing.org>

Table of Contents

PART 1. Software Requirements	2
Important note about distributions.....	2
Linux Requirements and a2billing.....	3
Not getting lost, what is the installation about?.....	3
How all fits together?.....	4
PART 2. Installation of a2billing.....	5
STEP 1. Get A2Billing source code.....	6
STEP 2. Prepare the Database.....	6
STEP 3. Edit a2billing configuration file.....	7
STEP 4. Fix permissions, files and folders	8
1. SIP and IAX.....	8
2. Sound files.....	8
3. Configure Asterisk Manager.....	8
STEP 5. Install The AGI components	9
STEP 6. Install web-based Graphical interfaces	9
STEP 7. Create dialplan.....	10
STEP 8. Configure recurring services.....	11
STEP 9. Call back daemon (only for Call backs)	12
PART 3. Default passwords and URL.....	12

PART 1. SOFTWARE REQUIREMENTS

IMPORTANT NOTE ABOUT DISTRIBUTIONS

This documentation assumes that you are using a .deb based Linux distro that has used the folder `/usr/share/asterisk` during asterisk packaging.

Other distributions use the alternate folder `/var/lib/asterisk`. That is the case if you compile from source the path by default is `/var/lib/asterisk`.

The basic assumptions of this documentation is that you are **not** compiling any of the software packages for source.

Other assumptions are:

- your apache2 default root folder is `/var/www`
- your asterisk sounds are under `/usr/share/asterisk/sounds`
- your asterisk AGI folder is expected under `/usr/share/asterisk/agi-bin`

- your apache2 runs as www-data (uid)
- you asterisk runs as asterisk (uid)
- those using subversion to check out the code, are recommended to use symbolic links instead of copying the files to the right directories

LINUX REQUIREMENTS AND A2BILLING

a2billing 1.4 includes the following third party software:

- **phpagi 2.14** Copyright 2003, 2004, 2005 Matthew Asham and David Eder
- **smarty 2.6.22** Copyright 2001-2005 New Digital Group, Inc
- **jgraph 1.27** Copyright 2001, 2002, 2003, 2004 Johan Persson Aditus Consulting
- **adodb 4.991** Copyright (c) 2000, 2001, 2002, 2003, 2004 John Lim
- **html2pdf** Copyright (C) 2004-2005 Renato Coelho
- **famfamfam.com** Icons from Mark James
- **phpconfig** (C) 2003 Dave Packham and Rob Birkinshaw

NOT GETTING LOST, WHAT IS THE INSTALLATION ABOUT?

In a nutshell installing a2billing requires nine steps:

1. Download and unpack source code
2. Setup the database
3. Edit a2billing.conf file. Setting up the database parameters
4. Fix permissions and folders
5. Installing the web based graphical user interfaces (Customer and Admin)
6. Place the AGI files
7. Prepare your dialplan
8. Add your cronjobs
9. Configure your callback daemon (only for callback)

To help you to understand what we are doing consider that:

1. The mother of all components of a2billing is an AGI file (a2billing.php) that is called from asterisk dialplan. (See Step: 6, 7)
2. The overall goal of the installation is to make sure that the AGI file that is placed in the asterisk agi-bin folder and invoked from the dialplan can (1) access a mysql database and (2) connect to Asterisk manager to place and receive call related commands and information. (See Step: 2, 3, 4)

3. A2billing is managed via three different user interfaces (admin, agent and customer) that needs to run via the apache2 web browser with the right owner and permissions (See Step: 5)
4. There are other components as cronjobs, callback daemon, etc that are needed for advance added value services (See Step: 8, 9)

HOW ALL FITS TOGETHER?

A2billing is a complex software but the main idea can be summarized as follows:
When a call comes into Asterisk, it falls into a context (a part of the dialplan) that will execute `a2billing.php` PHP file in one of its operational modes. The PHP file is the AGI (Asterisk Gateway Interface) software that is responsible of retrieving information of the call from the AMI (Asterisk Manager Interface) and interact with the Mysql database. All actions related to the call (routing, playing a IVR, etc) are executed talking to Asterisk via the AMI.

`a2billing.php` (the AGI code in the dialplan) uses a PHP class known as **phpagi** to talk with the AMI and **adodb** to connect with the mysql database. To separate functionality from presentation a2billing 1.4 is using **smarty**.

Pre-required Packages

A2billing requires the packages of a LAMP (PHP5) installation.

To install the necessary packages, run the following commands:

```
#apt-get install libapache2-mod-php5 php5 php5-common  
#apt-get install php5-cli php5-mysql mysql-server apache2 php5-gd  
#apt-get install openssh-server
```

Version 1.4 requires Mcrypt for PHP5

```
#apt-get install php5-mcrypt
```

Asterisk is of course also needed.

```
#apt-get install asterisk
```

PART 2. INSTALLATION OF A2BILLING

In a nutshell installing a2billing requires seven steps:

1. Download and unpack source code
2. Setup the database
3. Edit a2billing.conf file. Setting up the database parameters
4. Fix permissions and folders
5. Place the AGI files
6. Installing the web based graphical user interfaces (Customer, Agent and Admin)
7. Prepare your dialplan
8. Add your cronjobs
9. Configure your callback daemon (only for callback)

STEP 1. GET A2BILLING SOURCE CODE

Download the source code (wget) and unpack it (tar).²

```
#cd /usr/local/src/
#wget http://www.asterisk2billing.org/downloads/A2Billing_1.4.0.tar.gz
#tar zxvf A2Billing_1.3.4.tar.gz
```

We will now create a symbolic link to the new directory

```
#ln -s /usr/local/src/a2billing_1.4.0 /usr/local/src/a2billing
```

Make sure you have all the code under /usr/local/src/a2billing

STEP 2. PREPARE THE DATABASE

We will now create a MySQL database (mya2billing) for the billing software. The file *a2billing-MYSQL-createdb-user.sql* includes a script that creates the database with the correct access control users and permissions.

```
#cd /usr/local/src/a2billing
#mysql -u root -p < DataBase/mysql/Mysql-5.x/a2billing-MYSQL-createdb-
user.sql
```

After creating the database structure, we will create a set of tables and insert some initial basic data that A2Billing needs to work.

```
#mysql -u root -p mya2billing < DataBase/mysql/Mysql-5.x/a2billing-
schema-MYSQL.5.X-v1.4.0.sql
```

Checkpoint 1: Check that the database (my2billing) and that (89 ?) tables have been created.

```
#mysql -u root -p mya2billing
mysql>show tables
mysql>exit
```

² If you are a developer consider using the SVN version
 svn co --username guest --password guest http://svn.a2billing.net/svn/asterisk2billing/trunk

STEP 3. EDIT A2BILLING CONFIGURATION FILE

We need to edit the *A2Billing configuration file* (a2billing.conf file). The file contains all the configuration information for the Asterisk2Billing to connect to the database.

First, we need to copy the default configuration file from the source code (/usr/local/src) to Asterisk (/etc). Notice that in previous versions the a2billing.conf was under /etc/asterisk

```
#cp /usr/local/src/a2billing/a2billing.conf /etc/
```

Open the file with your favorite text editor (vi is used in this example). If you are new to Linux, we recommend you to use the text editor Gedit³.

```
#vi /etc/asterisk/a2billing.conf
```

The only parameters that you need to change here is the database connection information, an example follows:

```
[database]
hostname = localhost
port = 3306
user = a2billinguser
password = a2billing
dbname = mya2billing
;dbtype = postgres
dbtype = mysql
```

³ Gedit is a graphical text editor and can not be run from a console. In that case, we recommend *vi*.

STEP 4. FIX PERMISSIONS, FILES AND FOLDERS

In this step, we will tweak the file permissions of Asterisk to fit the A2Billing software. We will also create a number of additional files and folders that A2Billing needs, which does not come with the default installation.

1. SIP and IAX

First we will set a few file permissions (chmod, chown) and create (touch) the SIP and IAX configuration files for Asterisk.

```
chmod 777 /etc/asterisk
touch /etc/asterisk/additional_a2billing_iax.conf
touch /etc/asterisk/additional_a2billing_sip.conf
echo \#include additional_a2billing_sip.conf >> /etc/asterisk/sip.conf
echo \#include additional_a2billing_iax.conf >> /etc/asterisk/iax.conf
chown -Rf www-data /etc/asterisk/additional_a2billing_iax.conf
chown -Rf www-data /etc/asterisk/additional_a2billing_sip.conf
```

2. Sound files

We need to copy (cp) a few files from A2Billing package to Asterisk sounds folder. Use the installation script as follows:

```
/usr/local/src/a2billing/addons/install_a2b_sounds_deb.sh
chown -R asterisk:asterisk /usr/share/asterisk/sounds/
```

3. Configure Asterisk Manager

Configure the Asterisk Manager by editing the *manager.conf* file.

```
#vi /etc/asterisk/manager.conf
[general]
enabled = yes
port = 5038
bindaddr = 0.0.0.0

[myasterisk]
secret=mycode
read=system,call,log,verbose,command,agent,user
write=system,call,log,verbose,command,agent,user
```

STEP 5. INSTALL THE AGI COMPONENTS

Place the entire content of the directory AGI into your agi-bin directory.

```
mkdir /usr/share/asterisk/agi-bin
cd /usr/local/src/a2billing/AGI
cp a2billing.php /usr/share/asterisk/agi-bin/
cp -Rf lib /usr/share/asterisk/agi-bin/
```

```
Make sure the script is executable
chmod +x /usr/share/asterisk/agi-bin/a2billing.php
```

STEP 6. INSTALL WEB-BASED GRAPHICAL INTERFACES

In this step, we will install the three graphical interfaces of A2Billing, the Administration (admin), Agent (agent) and the Customer interface (customer).

Place the directories *admin*, *customer*, *agent* and *common* into your webserver document root.

```
mkdir /var/www/a2billing
cp -rf /usr/local/src/a2billing/admin /var/www/a2billing
cp -rf /usr/local/src/a2billing/agent /var/www/a2billing
cp -rf /usr/local/src/a2billing/customer /var/www/a2billing
cp -rf /usr/local/src/a2billing/common /var/www/a2billing
```

```
chmod 755 /var/www/a2billing/admin/templates_c
chmod 755 /var/www/a2billing/customer/templates_c
chmod 755 /var/www/a2billing/agent/templates_c
```

```
chown -Rf www-data:www-data /var/www/a2billing/admin/templates_c
chown -Rf www-data:www-data /var/www/a2billing/customer/templates_c
chown -Rf www-data:www-data /var/www/a2billing/agent/templates_c
```

Restart your webserver (Apache).

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

Checkpoint 2: Direct a browser to the administrative web interface (<http://<ip-addr>/a2billing/admin>) and login (check for all default passwords at the end of this document).

STEP 7. CREATE DIALPLAN

The *extensions.conf* is the Asterisk dialplan. Calls that interact with the billing software need to be handled inside of one or many a2billing related contexts.

The calls that reach the context are processed using the a2billing.php AGI script. The a2billing.php script can be invoked in many different modes (standard, did, voucher, callback, etc). In the example, we create two different contexts, the first context [a2billing] handles all the calls from our VoIP clients. When a call arrives, any extension number `_X.` (2 digits or more) reaches the script a2billing.php

The second context [did], will be used to route incoming calls back to the users. Calls to the clients (DID) are handled inside of the [did] context. The script a2billing.php in did mode is responsible of routing the call back to one of our users.⁴

Edit *extension.conf* (/etc/asterisk) and add the following two extensions.

```
[a2billing]
; CallingCard application
exten => _X.,1,Answer
exten => _X.,2,Wait,2
exten => _X.,3,DeadAGI,a2billing.php
exten => _X.,4,Wait,2
exten => _X.,5,Hangup

[did]
; CallingCard application
exten => _X.,1,DeadAGI(a2billing.php|1|did)
```

⁴ DeadAGI is a variant of AGI that you use when the channel is hung up.

STEP 8. CONFIGURE RECURRING SERVICES

Recurring services are handled via the `/etc/crontab`

You can add the following cron jobs to your `/etc/crontab` or create a file with the jobs in `/var/spool/cron/a2billing`

```
# update the currency table
0 6 * * * php /usr/local/src/a2billing/Cronjobs/currencies_update_yahoo.php

# manage the monthly services subscription
0 6 1 * * php
/usr/local/src/a2billing/Cronjobs/a2billing_subscription_fee.php

# To check account of each Users and send an email if the balance is
less than the user have choice.
0 * * * * php /usr/local/src/a2billing/Cronjobs/a2billing_notify_account.php

# To check all the accounts and send an notification email if the
balance is less than the first argument.
0 */6 * * * php /usr/local/src/a2billing/Cronjobs/a2billing_check_account.php

# this script will browse all the DID that are reserve and check if the
customer need to pay for it
# bill them or warn them per email to know if they want to pay in
order to keep their DIDs
0 2 * * * php /usr/local/src/a2billing/Cronjobs/a2billing_bill_diduse.php

# This script will take care of the recurring service.
0 12 * * * php /usr/local/src/a2billing/Cronjobs/a2billing_batch_process.php

# To generate invoices and for each user.
0 6 * * * php /usr/local/src/a2billing/Cronjobs/a2billing_invoice_cront.php

# to proceed the autodialer
*/5 * * * * php
/usr/local/src/a2billing/Cronjobs/a2billing_batch_autodialer.php

# manage alarms
0 * * * * php /usr/local/src/a2billing/Cronjobs/a2billing_alarm.php
```

STEP 9. CALL BACK DAEMON (ONLY FOR CALL BACKS)

The call back daemon is responsible of reading from the database the pool of calls stored for call back and trigger those calls periodically. The daemon is written in Python. Install the python-setuptools and use easy_install to install the callback_daemon

```
apt-get install python-setuptools python-mysqldb python-psycpg2
python-sqlalchemy
cd /usr/local/src/a2billing/CallBack
easy_install callback-daemon-py/dist/callback_daemon-1.0.prod_r1527-
py2.5.egg
```

Install the init.d startup script

```
cd /usr/local/src/a2billing/CallBack/callback-daemon-py/callback_daemon/
cp a2b-callback-daemon.debian /etc/init.d/a2b-callback-daemon
chmod +x /etc/init.d/a2b-callback-daemon
```

Make sure the daemon starts

```
update-rc.d a2b-callback-daemon defaults 40 60
```

If you need to remove the daemon in the future run

```
update-rc.d -f a2b-callback-daemon remove
```

PART 3. DEFAULT PASSWORDS AND URL

Type	Username	Password	Database
Admin UI	root	changepassword	
Manager	myasterisk	mycode	
MySQL database	a2billinguser	a2billing	mya2billing

Note! The first character in all usernames and passwords is a small letter (not capital).