Amberdms

ABS Manual Installation Guide

This document includes all the information required to install, upgrade and configure the Amberdms Billing System manually on non-offically supported platforms.

If you are a user of RHEL, CentOS or Fedora, you should instead refer to our supported platforms installation guide since the install path is considerably easier.

Version 8 March 2010

Table of Contents

1. Introduction

The Amberdms Billing System is a powerful, user friendly, open-source billing system designed for small and medium businesses and is fully open source under the GNU AGPL software license.

This guide details the steps to install the Amberdms Billing System onto your own servers when running a non-supported operating system.

Amberdms provides official support for a number of distributions, if you are running any of them, please read the ABS Supported OS Install Guide documentation instead:

1. RHEL versions 4 and 5
2. CentOS versions 4 and 5
3. Fedora versions 11 and 12

If your platform is not listed above then you are running on an unsupported platform and this is the correct document for you.

Note that whilst anything not listed above is an “unsupported” platform, this simply means that Amberdms won't provide support for issues experienced on these platforms, however in typical operation, ABS will work fine on most Linux and unix-like operating systems.

If you are looking for user documentation or developer information, check out the full list of manuals online at:
http://www.amberdms.com/?cms=opensource_billing_download

2. Application Requirements

The Amberdms Billing System is a large application with many features. The information in this section of the document details all the components that it requires.

2.1. Operating System

The Amberdms Billing System has been reported by users to work on:

- Debian
- Ubuntu
- MacOS Leopard

With the right package installs, it should also work on:

- Solaris/OpenSolaris
- FreeBSD/OpenBSD/NetBSD
- Any other reasonably standard Linux distribution.

It is unlikely that it will operate correct on:

- Microsoft Windows – The base of the application should work, but features such as emails or PDF generate are likely to fail due to dependence on nix-style filesystem structures and applications.

If you wish to gain official support for a particular operating system/distribution, Amberdms is always interested in performing funded development or accepting contributions – email support@amberdms.com if you are interested to discuss further.

2.2. OS Security Features

Some operating systems such as Linux now include security features such as SELinux, SMACK or AppArmor. These features can often interfere with the Amberdms Billing System, since ABS requires access to feature normally forbidden such as GNU tar, pdflatex and others.

In particular, the two most common features that get blocked by OS security features are:

- Execute GNU tar
- Execute PDFLatex/TeTex/Texlive – this can be complex, since it also needs permission to all the fonts for Tex.

This document includes information on how to adjust the policies or disable SELinux, however users of other security features will either need to learn how to adjust their policies or disable them.

2.3. PHP Versions

Amberdms recommends running this application with PHP versions 5.2.0 and higher in order to receive the full feature set of the application. It is possible to run this product with older PHP releases, but some features will not be available.

We have a list of PHP versions below and what is supported with each specific release:

2.3.1. OLDER THAN 4.3.0

There is no support for versions of PHP older than 4.3.0. If you do use it with an older release, the majority may work OK, but you could experience unexpected and undesirable behaviour.

2.3.2. PHP VERSION > 4.3.0

Under PHP 4 all features except the following are supported:

1. User-specific timezones are not supported. All users are limited to using the default server timezone.

2. The SOAP API is not supported.

2.3.3. PHP VERSION 5.0.0 -> 5.2.0

Under PHP 5 up till 5.2.0, all the features except the following are supported:

1. User-specific timezones are not supported. All users are limited to using the default server timezone.

2.3.4. PHP VERSION 5.2.0+

Under PHP 5.2.0 and above, all feature of the Amberdms Billing System are fully supported.

2.4. PHP Modules

2.4.1. REQUIRED MODULES

PHP is a very modular platform and often distributions split it up into multiple packages. The Amberdms Billing System requires the following components as well as the core of PHP itself:

1. php-curl
2. php-mysql
3. php-pear

If you install PHP yourself from source, make sure you have used the appropriate configure options to include these modules when building.

2.4.2. OPTIONAL MODULES

Optional modules are needed to enable specific features, however the base application will operate without them.

The following modules are required for PHP to allow generation of emails by the Amberdms Billing System:

1. PHP Mail Module
2. PHP Mail:Mime Module

Both of these modules can be downloaded by pear or they may be already included in your distributions repository.

2.5. Additional Requirements

The following other applications are required:

2.5.1. MYSQL SERVER

The Amberdms Billing System required MySQL server version 4 or above with InnoDB support enabled.

2.5.2. PDFLATEX (OPTIONAL)

In order to be able to generate PDF versions of invoices and reports, *pdflatex* must be installed along with it's font packages. It is possible to run the Amberdms Billing System without this program, but PDF generation will not work.

Depending on the distribution you are using, you can find pdflatex in the older *tetex* packages or it's successor, *texlive*.

Debian users will need to install *texlive-latex-recommended* and *texlive-latex-base* packages.

3. Important Notices

3.1. Avoid Shared Hosting

The install instructions in this document are designed for users running their own servers or virtual machines and assumes root access to the system with privileges to install packages as desired as well as security policies.

Amberdms does not recommend or support installing on “Shared Hosting” providers, these providers often have restricted PHP's features and prevent installation of components such as pdflatex or SELinux policies causing undesirable and broken application behaviour.

If you wish to avoid the cost of a virtual machine or dedicated server, consider signing up for Amberdms's hosted platform instead, this will provide a low cost and reliable, managed platform. For more details, please visit www.amberdms.com/billing or email sales@amberdms.com

3.2. Phone Home Disclaimer

Before installing, please be aware that the Amberdms Billing System includes a phone home feature, used to send back a few details to Amberdms so that we can better work out the needs and requirements of our user base.

This information is anonymous and is only used so that we can provide better services and make better development decisions to meet the needs of our user base.

We believe in openness and privacy, so below is a list of all the information that gets sent to Amberdms:

Information Sent	What we use it for
Amberdms Billing System version	To see how quickly users update and if older versions remain in use for significant time.
OS and webserver version.	Allows us to decide what platforms to officially support and make packaging tools for easier deployment for our users.
1. PHP version	By knowing the common versions of PHP used, we can make logical decisions on what versions to test with and how long to support older releases for.
2. Subscription ID	Used to differentiate between our paid and open source customers.

In future, this feature will be extended to check for updates as well, to inform administrators when there is a new version of the software available (with security or bug fixes).

If you wish to disable the phone home feature, you can do so once installed, by logging in as an administrator, going to the control panel and then disabling the “PHONE_HOME” option (although it means a lot to us if you didn't).

If you have any concerns regarding this feature and Amberdms's commitment to privacy, please feel welcome to email support@amberdms.com with any questions.

4. Installation Instructions

Before beginning installation, please make sure you have satisfied all requirements as per “Section 2: Application Requirements” above.

These instructions assume the following:

- Application install path to /usr/share/amberdms/billing_system/
- Configuration install path to /etc/amberdms/billing_system/

It is possible to deploy to other locations, you just need to substitute the paths in the install instructions below to those you desire.

4.1. Download Tarball

First download the latest source tarball from http://www.amberdms.com/?cms=opensource_billing_download

Note that you do not need the extras tarball unless you are planning to do development work and want the sample code for integrating with the SOAP API or some of the beta tools.

4.2. Application Installation

First copy the download tarball to /tmp and unpack with:

```
$ tar -xkjvf amberdms-bs-VERSION.tar.bz2
```

Create required directories

```
$ mkdir -p /usr/share/amberdms
$ mkdir -p /etc/amberdms/billing_system
```

Copy the tarball's unpacked directory:

```
$ cp -vr amberdms-bs-VERSION /usr/share/amberdms/billing_system
```

Install template configuration file

```
$ cp /usr/share/amberdms/billing_system/include/sample_config.php /etc/amberdms/billing_system/config.php
$ ln -s  /etc/amberdms/billing_system/config.php /usr/share/amberdms/billing_system/include/config-settings.php
```

Install cron file

```
$ cp /usr/share/amberdms/billing_system/help/resources/amberdms-bs-cron /etc/cron.d/amberdms-bs
```

Set permissions – this is important, since we need to limit what users can read files such as configuration. The apache user can be named differently on some systems, typically Apache is run as either:

- www-data (Debian and Ubuntu)
- apache
- apache2
- httpd

If you are unsure, check the User settings in the Apache configuration files. For these instructions we assume “apache” for the webserver and “cron” for the cron daemon.

```
$ chown -R root:apache /etc/amberdms/billing_system
$ chmod -R 750 /etc/amberdms/billing_system
$ chmod -R 644 /etc/cron.d/amberdms-bs
$ chmod -R 755 /usr/share/amberdms/billing_system
```

This completes installation of the application files, now move onto the database installation.

4.3. Database Installation

Database installation can be done in one of two ways – either via the automated utility (recommended) or manually by importing the right schema file.

Before installing the ABS database, if you have only just installed MySQL, you should set a root password for security. To configure a root password, run the following commands:

```
$ /usr/bin/mysqladmin -u root password 'NEWPASSWORD' -p
$ /usr/bin/mysqladmin -u root -h $HOSTNAME password 'NEWPASSWORD'
```

4.3.1. AUTOMATED UTILITY

Amberdms has developed a utility which will import a template database, create a MySQL user account for the DB only and automatically writes the configuration file for the user.

Please note: this tool is dependant on the default paths for the install of ABS, if you are installing to a custom path location, you should follow the manual DB instructions.

To run the installation tool, execute:

```
$ cd /usr/share/amberdms/billing_system/help/resources/; ./autoinstall.pl
```

If you experience any problems, follow through with the manual install steps below instead.

4.3.2. MANUAL INSTALLATION

The Amberdms Billing System database can be imported manually by following these steps:

1. Locate the latest version of /usr/share/amberdms/billing_system/help/schmea/version_*_install.sql

2. Import with:
   $ mysql -u root -p < /usr/share/amberdms/billing_system/help/schmea/version_LATESTDATE_install.sql

3. Create a MySQL user acount for the billing system only:

   $ mysql -u root -p

   GRANT USAGE ON * . * TO 'billing_system@'%' IDENTIFIED BY 'passwordgoeshere' WITH MAX_QUERIES_PER_HOUR 0 MAX_CONNECTIONS_PER_HOUR 0 MAX_UPDATES_PER_HOUR 0 ;

   GRANT SELECT , INSERT , UPDATE , DELETE , CREATE , DROP , INDEX , ALTER , CREATE TEMPORARY TABLES, LOCK TABLES ON `billing_system` . * TO 'billing_system'\@'%';\n";

   \q

4. Update the configuration file in /etc/amberdms/billing_system/config.php with the correct DB name and user settings.

4.4. Apache Configuration

After following steps above, the main application configuration file will have been written with the correct MySQL database configuration.

The remaining step is to create an apache configuration file – the exact location will vary depending on the structure of the apache configuration, but look for conf.d/ or some other module configuration directory.

Create a configuration file with:

```
cat > amberdms-bs.conf << "EOF"
#
# Amberdms Billing System is an open source accounting, service billing and time keeping web application.
#
Alias /billing_system /usr/share/amberdms/billing_system

<Location /billing_system>
	Order deny,allow
	Allow from all
	AllowOverride all
</Location>
EOF
```

Then reload apache with:

```
$ apachectl -k restart
```

4.5. SELinux (optional)

If your operating system runs SELinux, you need to either disable SELinux or update your policies to permit correct operation of ABS. Typically SELinux will break pdflatex PDF generation and database backup/exports.

4.5.1. DISABLING SELINUX

You can disable SELinux by running:

```
$ setenforce permissive
```

And then update /etc/selinux/config and changing the policy from “Targetted” to “Enforcing” to make the change permanently.

4.5.2. WRITING SELINUX POLICIES

Writing SELinux policies requires solid understanding of how SELinux works, but modern distributions do make it easier.

In short, you need to:

1. Switch SELinux to permisive mode
2. Run all the features that SELinux is breaking, to capture audit log information.
3. Run audit2allow to generate policy.
4. Compile that policy into a module for loading or include into the system base policy.

The exact steps will differ depending on your platform, a good resource for information on writing SELinux policies, see the CentOS 5 manual at http://www.centos.org/docs/5/html/Deployment_Guide-en-US/sec-sel-building-policy-module.html

4.6. Access the Billing System

The Amberdms Billing System is now installed – to begin using it, use your web browser to access:

https://example.com/billing_system

(where example.com is the hostname of your server)

The default login account is:
Username: setup
Password: setup123

Once logged in, you can change your password using the options page.

For further information, including information on configuring program options, please read the Amberdms Billing System User Guide.

5. Upgrade Instructions

Upgrading the Amberdms Billing System is reasonably straight forwards, replace the application files with the latest release, run the MySQL DB update script and all done.

You can skip any number of versions when upgrading and the schema update script will handle that correctly.

Before beginning the upgrade, please make sure you have satisfied all requirements as per “Section 2: Application Requirements” above, the dependencies can sometimes change between releases.

These instructions assume the following:

- Application install path to /usr/share/amberdms/billing_system/
- Configuration install path to /etc/amberdms/billing_system/

It is possible to deploy to other locations, you just need to substitute the paths in the install instructions below to those you desire.

5.1. Backing up the database

Before doing anything, first backup the database. You can do this using the CLI mysqldump tools, or by logging into the Amberdms Billing System as an administrator and going Admin -> Database Backup.

5.2. Upgrading the applications

Upgrading the application involves replacing the source with the new version, in the same way as the initial install.

First copy the download tarball to /tmp and unpack with:

```
$ tar -xkjvf amberdms-bs-VERSION.tar.bz2
```

Remove the existing version:

```
$ rm -rf /usr/share/amberdms/billing_system
```

Copy the new version from the unpacked tarball

```
$ cp -vr amberdms-bs-VERSION /usr/share/amberdms/billing_system
```

Create symlink for configuration file

```
$ ln -s  /etc/amberdms/billing_system/config.php /usr/share/amberdms/billing_system/include/config-settings.php
```

Install new cron file

```
$ cp /usr/share/amberdms/billing_system/help/resources/amberdms-bs-cron /etc/cron.d/amberdms-bs
```

Set permissions – this is important, since we need to limit what users can read files such as configuration. The apache user can be named differently on some systems, typically Apache is run as either:

- www-data (Debian and Ubuntu)
- apache
- apache2
- httpd

If you are unsure, check the User settings in the Apache configuration files. For these instructions we assume “apache” for the webserver and “cron” for the cron daemon.

```
$ chown -R root:apache /etc/amberdms/billing_system
$ chmod -R 750 /etc/amberdms/billing_system
$ chmod -R 644 /etc/cron.d/amberdms-bs
$ chmod -R 755 /usr/share/amberdms/billing_system
```

This completes the upgrade of the application files, now move onto the database installation in the following section.

5.3. Upgrading the database

To upgrade the database, an easy to use script has been provided, which reads the settings in your configuration file, checks the current schema version of the database and then applies all the database configuration changes that are required.

```
$ cd /usr/share/amberdms/billing_system/
$ help/resources/schema_update.pl -s help/schema/ -c /etc/amberdms/billing_system/config.php -v
```

The key options are:

-s	Location of the help/schema/ directory.
-c	Location of the configuration file.
-v	Verbose – details the actions taken.

Once the SQL upgrade has been completed, you will be able to login to the billing system. If the upgrade is unsuccessful, users will be unable to login to prevent unexpected behaviour of the billing system.