Table of Contents

1 Introduction

OpenEMM is web-based enterprise application for email marketing, marketing automation, email newsletter and service emails (transaction emails and event or time triggered emails). To summarize it, OpenEMM is a tool for customer relationship management by email.

OpenEMM offers tons of features for professional users, among them: a great user interface, template-based HTML mailings, tracking of mail openings and link clicks, automated bounce management, lots of graphical realtime statistics, self-defined and behaviour-based target groups, an integrated content management system, a multiqueue mail backend for maximum sending performance, an extensive set of webservices (including bulk operations), a plugin interface to easily extend the core functionality and a scripting feature to enhance the functionality of OpenEMM with triggerable customized actions and an audit-proof user activity log..

OpenEMM is mainly written in Java, Javascript and Python. OpenEMM employs Java frameworks like Spring, Struts and Tiles. Some performance-sensitive code is written in C. OpenEMM runs on top of a well proven Open Source software stack: Linux, Sendmail, MySQL and Tomcat.

This document will guide you through some necessary steps, which are needed to install and configure OpenEMM. It requires a basic knowledge of Linux system administration and (in case you need it) of Domain Name Services (DNS). The command-line examples are based on RedHat, Ubuntu and SuSE Linux.

Except as otherwise noted, you should run all commands as the user root to make sure you have the required permissions.

1.1 Requirements:

This is the software stack required by OpenEMM:

- Red Hat Enterprise **Linux** 5 or later, CentOS 5 or later, Ubuntu 10 or later, Suse Linux 10 or later
- Sun **Java SE JDK 8** (see chapter 3 for details)
- **Apache Tomcat 7 or 8** (see chapter 4 for details)
- **MySQL** 5.1 or later
- **Python** 2.x (not 3.x!)
- **Sendmail** 8.9 or later

2 Operating System

2.1 Operating System: Updates

If you use Ubuntu and want to install Java later, you need to add a new package source in file *sources.list* in directory */etc/apt* first. Enter these two lines at the end:
deb http://archive.canonical.com/ **lucid partner**
deb-src http://archive.canonical.com/ **lucid partner**

Update the operating system to its latest release. This will keep your system in the most stable state and harden it against various intrusion attempts.

- RedHat Linux: **yum update**
- Ubuntu: **apt-get update; apt-get upgrade**
- SuSE Linux: **yast->Software->Online Update**

2.2 Operating System: 64 Bit

If you work with a 64 Bit Linux you have to install additional packages for compatibility with OpenEMM's sub-programs *bav, bavwrap, smctrl, updater* and *xmlback* (all written in C):

- RedHat Linux: **yum install ld-linux.so.2 sqlite glibc.i686 libxml2.i686 zlib.i686**
  (if packages with suffix i686 do not exist, try suffix i386, and make sure that line **exclude=*.i386 *.i486 *.i586 *.i686** in file */etc/yum.conf* is uncommented)
- SuSE Linux: **yast -> Software -> Software Management -> 32-bit runtime environment**
- Ubuntu: **apt-get install ia32-libs**

Alternatively, you can compile the C files for your platform yourself. Download the source tarball of OpenEMM and execute its build script as user root to create your own customized binary tarball:
ant -f openemm_build.xml build

2.3 Operating System: Package Installation

(run as super user)

Install the required packages. Further dependencies will be resolved and installed automatically by the repository management software.

- RedHat Linux: **yum install mysql-server sendmail-cf MySQL-python libxml2**
- Ubuntu: **apt-get install sendmail mysql-server python-mysqldb**
- SuSE Linux: **yast -i mysql python-mysql sendmail libxml2**
  (If you use a 64 bit installation of SuSE, replace libxml2 with libxml2-32bit)

If package python-mysql is not available in OpenSuse, it is probably not needed. Make sure you install a version 2 of Python, not 3. If you do not want to use OpenEMM with Sendmail, you do not need to install package *sendmail-cf* and/or *sendmail*.

2.4 Operating System: Create User 'openemm'

(run as super user)

Create a special group and user for OpenEMM:
groupadd openemm
RedHat and Suse Linux: **useradd -m -g openemm -d /home/openemm -c "OpenEMM-2015" openemm**
Ubuntu: **useradd -m -g openemm -G adm -d /home/openemm -s /bin/bash -c "OpenEMM-2015" openemm**

The default directory */home/openemm* is used by OpenEMM. OpenEMM runs with the permissions of that user. Only the email sending component, which requires the root TCP port 25, will be run with super user permissions. OpenEMM's userspace concept adds more safety to your server and its services.

3 Installation: Sun Java JDK

OpenEMM's web container Tomcat requires the installation of Oracle's Standard Edition Java Development Kit (SE JDK) - not the GNU version of Java! If Sun's Java SE SDK is not included in your distribution and has not been installed yet, you have to install it by yourself. For OpenEMM 2015 you should use version 8 because Java 7 will no longer be supported by Oracle after April 2015:

Point your browser to *java.oracle.com* and visit the download section, subsection Java SE (Standard Edition). Download the tarball (*.tar.gz) of the latest Java SE JDK 8 (Java Development Kit).

- Create a directory for software required by OpenEMM: **mkdir -p /opt/openemm**
- Copy the file to the new directory: **cp jdk-8u25-linux-i586.tar.gz /opt/openemm**
- Change to this directory: **cd /opt/openemm**
- Unpack the JDK **tar -xvzf jdk-8u25-linux-i586.tar.gz**
- Create a symbolic link *java* for the JDK directory: **ln -s jdk1.8.0_25 java**
- Test the JDK: **/opt/openemm/java/bin/java -version**

You should get output similar to the following:
java version "1.8.0_25"
Java(TM) SE Runtime Environment (build 1.8.0_25-b17)
Java HotSpot(TM) 64-Bit Server VM (build 25.25-b02, mixed mode)

If you want to use an installed JDK, simply edit */home/openemm/bin/openemm.sh* after the installation of the OpenEMM tarball and adjust the parameter *JAVA_HOME* accordingly.

4 Installation: Tomcat

Since OpenEMM is a web application using Java, it requires a web container like Tomcat. OpenEMM 2015 was thoroughly tested with Tomcat 7, but it should run with Tomcat 6 or 8, too. However, if you still use Tomcat 6, we recommend to update to Tomcat 7 or 8 rather sooner than later.

NOTE: If you want to run OpenEMM with Tomcat 8, you have to replace line
<Listener className="org.apache.catalina.core.JasperListener" />
in file *server.xml* in directory */home/openemm/conf/* with this line:
<Listener className="org.apache.catalina.core.ThreadLocalLeakPreventionListener"/>

To install Tomcat, download the latest binary distribution of release **7 or 8** from *http://tomcat.apache.org*. The core package is sufficient. The examples here use Tomcat 7.0.57 – you should adapt them to the latest version of Tomcat **7 or 8** as needed.

- Create a directory for software required by OpenEMM: **mkdir -p /opt/openemm**
- Copy the file to the new directory: **cp apache-tomcat-7.0.57.tar.gz /opt/openemm**
- Change to this directory: **cd /opt/openemm**
- unpack the Tomcat file: **tar -xvzf apache-tomcat-7.0.57.tar.gz**
- Create a symbolic link *tomcat* for the new directory: **ln -s apache-tomcat-7.0.57 tomcat**

If you want to make sure that Tomcat works, enter the following commands (and make sure to stop an existing installation of OpenEMM first):

- Set environment variable JAVA_HOME: **export JAVA_HOME=/opt/openemm/java**
- Change into Tomcat directory: **cd tomcat**
- Start Tomcat: **bin/startup.sh**
- Check for Tomcat installation screen: **http://localhost:8080** (URL for browser or using wget)
- Stop Tomcat: **bin/shutdown.sh**

If you can not connect to Tomcat with your browser you may have to adapt your firewall rules first (see next section).

NOTE: If you want to use a pre-installed Tomcat, simply edit */home/openemm/bin/openemm.sh* after the installation of the OpenEMM tarball and adjust parameter *CATALINA_HOME* accordingly.

5 Enable OpenEMM Access in an iptables Firewall

5.1 RedHat Linux

Edit the file */etc/sysconfig/iptables* to open ports 25 (SMTP), 8080 (OpenEMM console and redirection) and 8044 (OpenEMM update service). Add the following lines in section *-A RH-Firewall-1-INPUT*:
-A RH-Firewall-1-INPUT -m state --state NEW -m tcp -p tcp --dport 25 -j ACCEPT
-A RH-Firewall-1-INPUT -m state --state NEW -m tcp -p tcp --dport 8044 -j ACCEPT
-A RH-Firewall-1-INPUT -m state --state NEW -m tcp -p tcp --dport 8080 -j ACCEPT

If you plan to use the internal SMTP server of OpenEMM instead of Sendmail (see chapter 9) you have to add this line to open port 8025 (OpenEMM SMTP server):
-A RH-Firewall-1-INPUT -m state --state NEW -m tcp -p tcp --dport 8025 -j ACCEPT
Additionally, you have to enable a prerouting forwarding rule from port 25 to 8025. This is done by adding the following code after the comments at the top of the file */etc/sysconfig/iptables*:
***nat**
:PREROUTING ACCEPT [0:0]
:POSTROUTING ACCEPT [0:0]
:OUTPUT ACCEPT [0:0]
-A PREROUTING -i eth+ -p tcp --dport 25 -j REDIRECT --to-port 8025
COMMIT

Committing all these changes requires a restart of iptables, which is done with
/etc/init.d/iptables restart

5.2 Ubuntu

The Ubuntu Firewall is not enabled by default, because no externally reachable services are running. You can check the status of the firewall with
ufw status
Enable the firewall with
ufw enable
and open port 25 (SMTP), 8080 (OpenEMM console & redirection) and 8044 (OpenEMM update service):
ufw allow 25/tcp
ufw allow 8080/tcp
ufw allow 8044/tcp
Verifiy your settings with
ufw status

If you plan to use the internal SMTP server of OpenEMM instead of Sendmail (see chapter 9) you have to open port 8025 (OpenEMM SMTP server) as well:
ufw allow 8025/tcp
Additionally, you have to enable a prerouting forwarding rule from port 25 to 8025. This is done by adding the following code after the comments at the top in file *user.rules* in directory */lib/ufw*:
***nat**
:PREROUTING ACCEPT [0:0]
:POSTROUTING ACCEPT [0:0]
:OUTPUT ACCEPT [0:0]
-A PREROUTING -i eth+ -p tcp --dport 25 -j REDIRECT --to-port 8025
COMMIT

Committing all these changes requires a restart of the Ubuntu firewall, which is done with
service ufw restart

5.3 SuSE Linux

Use Yast to open ports 25 (SMTP), 8080 (OpenEMM) and 8044 (update):
yast -> Security and Users -> Firewall -> Allowed Services
Select *Mail Server*. After that add permission for port 8080, 8025 and 8044:
-> Advance -> Settings for Zone: External Zone -> TCP Ports: 8080 8025 8044

You can omit port 8025 if you plan to use Sendmail (see chapter 9). If you want to use the internal SMTP server of OpenEMM you have to enable a prerouting forwarding rule from port 25 to 8025 by setting parameter *FW_REDIRECT* in file */etc/sysconfig/SuSEfirewall2* to
"0/0,0/0,tcp,25,8025"

Committing this change is accomplished with
/etc/init.d/SuSEfirewall2_setup restart

6 Installation of OpenEMM 2015

Get the latest version of the OpenEMM binary code from
http://www.sourceforge.net/projects/openemm/files

Copy the tarball to a temporary location - */tmp* is a good choice. Change to the home directory and run the following commands to create a version specific directory for the new OpenEMM version and to create a symbolic link from *openemm* to that directory:
cd /home
mv openemm openemm-2015
ln -s openemm-2015 openemm
Change to OpenEMM's directory and unpack the OpenEMM tarball. Do not forget option "p" for the tar command, because some files need to have owner and group set to *root* or special permissions which are preset in the tarball!
cd /home/openemm
tar xzvpf /tmp/OpenEMM-2015-bin.tar.gz
Finally, in order to follow best practise, move the documentation folder to */usr/share/doc* where doc files are usually located on a Linux system:
mkdir -p /usr/share/doc/OpenEMM-2015
mv USR_SHARE/* /usr/share/doc/OpenEMM-2015
rm -r USR_SHARE
If you decide to install OpenEMM in a directory other than */home/openemm* please make sure that your home directory contains a symbolic link to that directory and grant the required file access permissions with
chown -R openemm:openemm {path to OpenEMM install directory}

6.1 Read Access to Maillog

OpenEMM requires read access to the log file */var/log/maillog*.

For RedHat Linux open file */etc/logrotate.d/syslog* and add the following line after the line *sharedscripts*:
create 0604
Also run
chmod 604 /var/log/maillog
to set the permissions of the current maillog.

For Ubuntu nothing has to be done because openemm was added to group *adm* and, therefore, can already access the mailog. But since Ubuntu's maillog is named *mail.log* you have to create a symlink for *maillog*:
ln -s /var/log/mail.log /var/log/maillog

For SuSE Linux open file */etc/syslog-ng/syslog-ng.conf.in* and change the line
options { long_hostnames(off); sync(0); perm(0640); stats(3600); };
to
options { long_hostnames(off); sync(0); perm(0644); stats(3600); };
Also change the line
destination mail { file("/var/log/mail"); };
to
destination mail { file("/var/log/maillog"); };
Finally, activate the changes with
/sbin/SuSEconfig

6.2 Initialize/Update the OpenEMM and the OpenEMM CMS Database

Make sure that MySQL is running.
- RedHat Linux: **/etc/init.d/mysqld restart**
- Ubuntu: **service mysql restart**
- SuSE Linux: **/etc/init.d/mysql restart**

Since OpenEMM 2015 works with a CMS database which did not exist before version 6.0, you have to setup this database and load its layout if you update OpenEMM from a version before 6.0 or if you are installing OpenEMM from scratch:
cd /usr/share/doc/OpenEMM-2015

mysqladmin -u root -p create openemm_cms
(omit option –p in case your MySQL system is not password protected)
mysql -u root -p openemm_cms < openemm_cms-2015.sql

If you would rather install a demo CMS database with sample values, use the file *openemm_demo-cms.sql* instead of the *openemm_cms.sql* file mentioned above. This file contains a CM template, 10 content module types and 12 content modules as samples to work with.

You now have three options:

A.) If you want to set up the OpenEMM database from scratch, use the following commands:
cd /usr/share/doc/OpenEMM-2015
mysqladmin -u root -p create openemm
(omit option –p if your MySQL system is not password protected)
If you plan to use the redirection service of OpenEMM, open file *openemm-2015.sql* with a text editor like edit or vim, and find and replace the string
http://localhost:8080
with a valid redirection URL. In our example it is
http://www.openemm.org:8080
Next, replace the empty mailloop string '' right after the redirection URL you just entered in the step before, with the host name of your server, like
www.openemm.org
If you plan to use the built-in bounce management for asynchronous bounces, you have to use the sender hostname (see section 12.2). In our example it is
news.openemm.org
The sender hostname you enter for the mailloop string will be used as the domain name for the forward addresses generated by the bounce filter.
Finally, load the OpenEMM database layout with
mysql -u root -p openemm < openemm-2015.sql

B.) In case your old OpenEMM databases are somehow lost but you made backup files *openemm.sql* and *openemm_cms.sql* from a former installation (see chapter 10), import the databases with
mysqladmin -u root -p create openemm
mysqladmin -u root -p create openemm_cms
mysql –u root –p openemm < /home/openemm/openemm.sql
mysql –u root –p openemm_cms < /home/openemm/openemm_cms.sql
You may have to update the database schemas. If so, please also follow the instructions in the following paragraph.

C.) If you used OpenEMM before and an OpenEMM database already exists, you may have to update your database schema to add new tables and/or columns. Change to directory */usr/share/doc/OpenEMM-2015* and look for files with names like *update_openemm-2011-2013.sql*. To update your database to the latest version you have to apply some or all of these files (depending on the OpenEMM version you used before) in the right sequence to your database. This is done by the (generic) command:
mysql -u root -p openemm < update_openemm-{old version}-{new version}.sql
For example: If you want to update from OpenEMM 6.2 to 2015 you have to run the following two commands in exactly that sequence:
mysql -u root -p openemm < update_openemm-6.2-2011.sql
mysql -u root -p openemm < update_openemm-2011-2013.sql
mysql -u root -p openemm < update_openemm-2013_R2-2015.sql
Do not skip an intermediate version! However, if you did not install a release candidate (RC) of OpenEMM, you should omit all update files concerning release candidate versions (like **update_openemm-5.3.2-5.4.0rc1.sql** or **update_openemm-5.4.0rc1-5.4.0rc2.sql**).

6.3 Basic configuration

The property *system.url* in file *emm.properties* in directory */home/openemm/webapps/openemm/WEB-INF/classes* must be set to the URL of your OpenEMM installation, which is usually identical to your redirection URL, in the form
http://www.openemm.org:8080
The property *cms.ccr.url* in file *cms.properties* in the same directory should be set to the identical URL unless the content manager module (central content repository) runs on a different server - which is possible due to its webservices interface.

6.4 Start and Stop OpenEMM

Change to user *openemm* with
su – openemm
Do not forget the hyphen in the first line!
To start the OpenEMM environment, change to the home directory of OpenEMM and launch the start script with
openemm.sh start
and to stop OpenEMM
openemm.sh stop
If the script openemm.sh is not found make sure that file *.bash_profile* in directory */home/openemm* contains line
PATH=$PATH:$HOME/bin

If OpenEMM reports errors at startup indicating a problem with your operating system's version of library *glibc*, you have to compile OpenEMM yourself. See instructions in build script *openemm_build.xml* in source code tarball *OpenEMM-2015-bin.tar.gz* for details.

To invoke the GUI of OpenEMM, point your webbrowser to
http://{your FQDN}:8080
and log into OpenEMM as
Username: admin
Password: openemm

OpenEMM detects the language setting of your browser and shows the appropriate login page. Obviously, your first step should be to change the password and user name in the settings menu to a new name and a better password.

By default, OpenEMM's menus are shown in English. To change to your local language, click on menu *Settings* and choose sub-menu *User*. Select user *admin* (or the new name you have chosen) and change the language field from English to your language. Retype your password twice (password and confirm field) and press the *Save* button. You have to log out and in again to activate the change of the user language.

7 Configuration of OpenEMM 2015

7.1 Advanced Configuration

If you use the CMS module of OpenEMM to build mailings and want to change the default text for text mails, change the content of the field *text* in table *cm_text_version_tbl* on database *openemm_cms* accordingly. At a minimum you should change the domain name of the links from *localhost* to your redirect domain name.

If you want to work with more than 200,000 addresses in your database, change the value of the corresponding property in file *emm.properties*:
recipient.maxRows=200000
However, the bigger your database, the more the performance of your OpenEMM installation may degrade! If you want to use the import wizard to import more than 60,000 recipients in one chunk (which could take some time), please adjust the following property in the same file accordingly:
import.maxRows=60000

If the OpenEMM database holds more than 10,000 recipients and you open the recipient list you will be greeted with message *The option you selected is too large to be displayed completely. Please limit your selection to reduce the amount of recipients.*
If you want more than 10,000 recipients to be processed for the recipient list (which will take longer to display), set field *max_recipients* in database table *company_tbl* to the value you want:
UPDATE company_tbl SET max_recipients = 100000 WHERE company_id = 1;

OpenEMM 2015 comes with a link checker to verify that all links in your emails lead to an existing target page. You can modify the behaviour of the link checker in database table *config_tbl* through fields *linktimeout* (default value: 20,000 milliseconds) and *threadcount* (default value: 20).

By default, OpenEMM uses domain *openemm.net* for the message ID in the header of the mails sent out by OpenEMM. If you want to change that value to your own domain name, modify property *mailgun.ini.domain* in file *emm.properties* accordingly.

7.2 Configuration for MySQL database

OpenEMM is able to send emails with attachments. The maximum size for email attachments is defined by property *attachment.maxSize*. It is set to 2 MByte by default in file *emm.properties*:
attachment.maxSize=2097152
Please be aware that the default value of MySQL parameter *max_allowed_packet* is just 1 MByte, i.e. by default you can not load a single data packet (file) bigger than 1 MByte into the database. To do this you need to set the parameter *max_allowed_packet* in section *[mysqld]* of MySQL's configuration file (usually *my.cnf* in directory */etc*) to something like the following:
max_allowed_packet=2M
and restart MySQL afterwards, with
/etc/init.d/mysqld restart
Since the transfer of data to the database has some overhead, the value for *max_allowed_packet* should be a little bit higher than the value for *attachment.maxSize*. You can check the current value of *max_allowed_packet* in MySQL with statement
SELECT @@max_allowed_packet;

If your MySQL instance does not run on the same server as OpenEMM you might have to change certain parameters in MySQL's configuration file (usually *my.cnf* in directory */etc*) like commenting out *skip-networking* in section *[mysqld]* or setting *bind-address* to the IP address of your database server.

Especially important is parameter *wait_timeout*, which is set to 28800 by default. This means that MySQL automatically cuts the connection to OpenEMM after 8 hours of inactivity. This leads to an initial connection error when OpenEMM attemtps to contact the MySQL database next time. If your OpenEMM installation does not access its MySQL database all the time, you should increase this value to at least one day (86400) or even a whole week (604800).

For more advice on how to configure the database, please check out MySQL's documentation.

7.3 Configuration of Webservices 1.0 (deprecated)

To be able to access the legacy OpenEMM webservices (1.0) you have to create a webservice user with a password first:
INSERT INTO `ws_admin_tbl` (`username`, `password`) VALUES ('ws-user', 'openemm');
For security reasons you should choose a more elaborate password, of course.

7.4 Configuration of Webservices 2.0

While the interface for OpenEMM webservices 1.0 is part of OpenEMM, the new and more powerful webservice interface 2.0 introduced with OpenEMM 2013 runs as a separate web application in directory */home/openemm/webapps/openemm-ws*. If you want to use the new webservices you have to copy your version of file *emm.properties* from */home/openemm/webapps/openemm/WEB-INF/classes* to */home/openemm/webapps/openemm-ws/WEB-INF/classes* and you have to change the URL of property *wsdlLocationUri* in file *emm-ws.properties* in the same directory to (do not forget the trailing slash!)
http://{your domain}:8080/openemm-ws/

After OpenEMM 2015 has been launched you may request the WSDL file for the new webservices via URL
http://{your domain}:8080/openemm-ws/emmservices.wsdl

To be able to access the new webservices of OpenEMM 2015 you have to create a webservice user with a password first:
INSERT INTO `webservice_user_tbl` (`username`, `password`) VALUES ('ws-user', 'openemm');
(For security reasons please choose a more sophisticated password.)

7.5 Creating Customized Date Formats

If you want OpenEMM tag *[agnDATE]* to work with customized date formats, open MySQL and insert your new date format in table *date_tbl* like
INSERT INTO `date_tbl` (`type`, `format`) VALUES (4, 'dd/MM/yyyy');
where 4 is the value for tag parameter *type* and *dd/MM/yyyy* is the new date format. For details of available date formats see the documentation for the Java class *java.text.SimpleDateFormat*.

If you want the import wizard to work with customized date formats, open file *DateFormat.java* in directory */src/java/org/agnitas/util/importvalues* of the source tarball of OpenEMM, replace the semicolon at the end of line 41 with a comma, insert a line like
ddMMyyyyHHmmss("dd.MM.yyyy HH:mm:ss", "import.date.format.ddMMyyyyHHmmss"); // 5
afterwards, re-compile the file with
javac DateFormat.java
to a class file and use it to replace the old class file in your OpenEMM installation.
In this example *dd.MM.yyyy HH:mm:ss* is the new date format and *ddMMyyyyHHmmss* is its key. To avoid an error message in the user interface add this key in file *messages.properties*:
import.date.format.MMddyyyy = dd.MM.yyyy HH:mm:ss
Make sure that you use only format fragments “yyyy”, “MM”, “dd”, “HH”, “mm” and “ss” in your customized date format.

7.6 Adjust Color Codes for Heatmap

The default percentage values for the color codes of the heatmap are stored in the OpenEMM database in table *click_stat_colors_tbl*. You may modify the values for *range_start* (minimum percentage value for a color code), *range_end* (maximum percentage value for a color code) and *color* (hex code for the color code). Use this SQL code to change the precentage ranges:
UPDATE click_stat_colors_tbl SET range_start = 0, range_end = 1 WHERE id = 1;
UPDATE click_stat_colors_tbl SET range_start = 1, range_end = 2 WHERE id = 2;
UPDATE click_stat_colors_tbl SET range_start = 2, range_end = 3 WHERE id = 3;
UPDATE click_stat_colors_tbl SET range_start = 3, range_end = 5 WHERE id = 4;
UPDATE click_stat_colors_tbl SET range_start = 5, range_end = 10 WHERE id = 5;
UPDATE click_stat_colors_tbl SET range_start = 10, range_end = 100 WHERE id = 6;
and adjust the values for *range_start* and *range_end* as needed. If you want to change the color values too, make sure to choose light colors.

8 Administration of OpenEMM 2015

8.1 Database Backup

For MySQL there exist plenty of strategies for database backups and tons of books and Internet resources on that subject. However, if you run only a medium MySQL database with a few GByte of data and if you can live with an interruption of services of very few minutes, you may simply use tool mysqldump:
mysqldump -aCceQx --hex-blob --routines --triggers -u root -p -r openemm.sql openemm
Executed at the command line, this statement copies a database dump in a very robust format into text file openemm.sql. The database dump can be imported back into an emtpy database emm simply with
mysql -u root -p openemm < openemm.sql
To backup the OpenEMM CMS database, simply replace *openemm* with *openemm_cms*.

8.2 Database Cleanup

OpenEMM bounce management stores all bounce information in the database. After one or two years of operation, bounce information can account for 80% or even 90% of the size of your database. However, it is not necessary to store bounce information forever. You can set a limit of how many days bounce information should be stored with the parameter *bounce.maxRemain*. We recommend the following setting (90 days):
bounces.maxRemain.days=90

You can also set a limit of how many days subscribers who did not confirm their double opt-in mail should be stored in the database. (If you do not delete them, they can not restart the subscription process.) We recommend the following setting (30 days):
pending.maxRemain.days=30

All parameters are set in the text file **emm.properties** in directory */home/openemm/webapps/openemm/WEB-INF/classes*.

These two cleanup jobs are executed by the JobQueue of OpenEMM. Table *job_queue_tbl* lists all jobs periodically executed by the JobQueue. You can set the execution times of the cleanup jobs via their entries in *job_queue_tbl* because the mass deletion of information can place serious strain on your database resources. The default entry for the cleanup of old bounces and pending confirmations is
INSERT INTO job_queue_tbl
(description, created, laststart, running, lastresult, startaftererror, lastduration, interval, nextstart, hostname, runclass, deleted)
VALUES ('DBCleaner', CURRENT_TIMESTAMP, null, '0', 'OK', '0', '0', '0300', CURRENT_TIMESTAMP, null, 'org.agnitas.util.quartz.DBCleanerJobWorker', '0');
This SQL statements defines a start time of 3:00 AM. If you want the cleanup to start at a different time, you have to update field *interval* accordingly.

To increase security, OpenEMM now blocks logins when the same IP address generates a certain number of failed logins. The default value for the max. number of failed logins is 3 and the default value for the lock out time is 300 seconds. You can change both values in the database in table *company_tbl*, field *max_login_fails* and *login_block_time*.

All login tries are logged in table *login_track_tbl*. Since brute force attacks from evil hackers to log into OpenEMM could flood this table, all entries older than 60 days are deleted from this table automatically. This cleanup job is executed by the JobQueue, too. The default entry for this cleanup is:
INSERT INTO job_queue_tbl
(description, created, laststart, running, lastresult, startaftererror, lastduration, interval, nextstart, hostname, runclass, deleted)
VALUES('LoginTrackTableCleaner', CURRENT_TIMESTAMP, null, '0', 'OK', '0', '0', '0400', CURRENT_TIMESTAMP, null, 'org.agnitas.util.quartz.LoginTrackTableCleanerJobWorker', '0');
This SQL statements defines a start time of 4:00 AM. If you want the cleanup to start at a different time, you have to update field *interval* accordingly.

8.3 Advanced Database Cleanup

If, despite the cleanup described above, at some point in time your database becomes simply too big (and because of it, too slow!), you should delete old mailings and related statistical data from the

database. If you delete a mailing in the GUI it is only set to status “deleted”. We recommend to delete entries from these 10 tables:

mailing_tbl	Contains mailing information, this is the basic table which references all tables below, you can safely delete all mailings with field *deleted* set to 1
mailing_account_tbl	While sending a mailing, for every sent block a record with the number, size and type of block is written
mailing_backend_log_tbl	Contains information on how many emails of a mailing have already been generated
component_tbl	Contains content components (like images and attachments) of mailings
dyn_name_tbl	Contains text module names (content for text modules is saved in table *dyn_content_tbl*)
dyn_content_tbl	Contains content blocks for text modules, use field *dyn_name_id* to identify entries to be deleted (table *dyn_name_tbl* maps *dyn_name_ids* to *mailing_ids*)
mailtrack_tbl	Contains a record for every recipient and every mailing he/she got
onepixellog_tbl	Contains a record for every recipient who opened a mail
rdir_log_tbl	Logs clicks on redirected links in sent mails
rdir_url_tbl	Contains all trackable mailing links

Unless otherwise noted use the field *mailing_id* to identify the entries to be deleted from each table.

Usually, *component_tbl* and *mailtrack_tbl* tend to be the biggest tables. You can check the size of all OpenEMM tables with SQL statement (output in MByte):

```
SELECT table_name, ROUND((data_length+index_length)/1024/1024, 2) table_size FROM
information_schema.tables WHERE table_schema='openemm' ORDER BY table_size DESC;
```

If you want to delete mailings not set to status “deleted” make sure to not delete date- or event-based mailings (*mailing_type* = 1 or 2) without checking first if they are still in use. If you do not want to delete templates, make sure that field *is_template* is set to 0. And if you want to make sure to only delete sent mailings (because you might still work on the unsent ones), check field *status_field* of table *mailing_account_tbl* for “W” to identify those mailings. Bringing it all together in one SQL statement:

```
SELECT mailing_id FROM mailing_tbl WHERE
   creation_date LIKE '2010%'
   AND mailing_type = 0
   AND (
      is_template = 0
      AND mailing_id IN (
         SELECT mailing_id FROM mailing_account_tbl WHERE status_field = 'W'
      )
      OR deleted = 1
   );
```

But because sub-selects in MySQL are quite slow, be patient with this statement, it may take a few minutes or even longer. An alternative would be to create a temporary table first with the IDs of all mailings to be deleted:

```
CREATE TABLE tmp_mailing_delete AS SELECT mailing_id FROM mailing_tbl WHERE
   creation_date LIKE '2010%'
   AND mailing_type = 0
   AND is_template = 0;
DELETE FROM tmp_mailing_delete WHERE
   mailing_id NOT IN (
      SELECT mailing_id FROM mailing_account_tbl WHERE status_field = 'W'
   );
INSERT INTO tmp_mailing_delete (
   SELECT mailing_id FROM mailing_tbl WHERE
      creation_date LIKE '2010%'
      AND mailing_type = 0
      AND deleted = 1
   );
```

Both examples delete mailing from the year 2010. To choose a different year, just change lines

```
creation_date LIKE '2010%'
```

8.4 Database Tuning

80% of all application performance problems are really database performance problems. If you run a big OpenEMM installation and you are not satisfied with the application's performance, here are some database tuning tips you should try.

If your OpenEMM database holds a long list of recipients, you may speed up certain database operations with a combined index on four fields of table *customer_1_binding_tbl*:
CREATE INDEX custbind$cuid_mlid_user$idx ON customer_1_binding_tbl (customer_id, mailinglist_id, user_status, user_type);

If you use any other profile field than *email* for duplicate checks, you should put an index on this field in *customer_1_tbl*:
CREATE INDEX cust$<fieldname>$idx ON customer_1_tbl (<fieldname>)

If you work with a big database you can speed up database performance for tracking mails openings with an index on table *onepixel_log_tbl* for the combination of *mailing_id* and *customer_id*:
CREATE INDEX onpxmid_cuididx ON onepixel_log_tbl (mailing_id, customer_id);
In this case we also recommend an index on table *rdir_log_tbl* for the combination of *mailing_id*, *customer_id* and *url_id* to speed up processing of link clicks:
CREATE INDEX rlogmid_cuid_urlididx ON rdir_log_tbl (mailing_id, customer_id, url_id);

While MySQL's default database engine MyISAM works fine with the default configuration, we suggest these settings in section *[mysqld]* of MySQL's configuration file (usually *my.cnf* in directory */etc*) for an OpenEMM stand-alone database:
key_buffer_size=64M
max_connections=50
max_heap_table_size=32M
query_cache_size=32M
read_buffer_size=512K
table_cache=128
To check if the settings fit your needs, you could use the tuning-primer script available at http://www.day32.com/MySQL.

Due to a bug in OpenEMM < 6.0, some temporary tables were not always deleted. You can identify these tables by the prefix "TMP_CRT_" and safely drop them from your database with
DROP TABLE TMP_CRT_..._TBL;

Since version 5.5 InnoDB is the default engine of MySQL. While InnoDB supports row locking and real transactions for better crash protection (opposed to MyISAM), the internal data structure is more complex than MyISAM's, which leads to significantly larger table sizes, slower writes, slower full table scans and slower handling of BLOBs and CLOBs. Also, backup and recovery via mysqldump/mysql is much slower.

The choice between MyISAM and InnoDB depends on the size and usage profile of your OpenEMM database and there is no panacea for the decision of which engine to select. Since converting a table from MyISAM to InnoDB is easy, you could simply give it a try (after a backup). The best table candidates are *customer_1_binding_tbl*, *customer_1_tbl*, *mailtrack_tbl* and *onepixel_log_tbl*. You could convert table *customer_1_binding_tbl* to InnoDB with
ALTER TABLE customer_1_binding_tbl type = InnoDB;
But please be aware that this conversion will be done line by line and that it needs some time. So, either do it at night or check the time demand first with a copy of your production database.

Because InnoDB is much more sensitive to configuration parameters than MyISAM, you should at least add properties *innodb_buffer_pool_size* and *innodb_log_file_size* in section [mysqld], because the default values of 128 MByte and 5 MByte are much too small for bigger databases with lots of InnoDB tables. As a rule of thumb: If your whole OpenEMM database was converted to InnoDB and runs on a dedicated server, *innodb_buffer_pool_size* should be set to 75% of the RAM of your server and *innodb_log_file_size* should be set to ¼ of the size of *innodb_buffer_pool_size*, but not higher than 256 MByte to limit recovery time after a database crash.

If OpenEMM runs on a Linux operating system you should add property
innodb_flush_method=O_DIRECT
and to prevent the InnoDB engine from saving all table data into system tablespace file *ibdata1* in directory */var/lib/mysql* you may add property
innodb_file_per_table=1
in section *[mysqld]* of MySQL's configuration file *my.cnf* (usually to be found in directory */etc*).

8.5 Out of Memory

If you work with big lists and experience an error message like "Java.lang.OutOfMemoryError: Java heap space", you have to allocate more memory to the Java Virtual machine (JVM). You can increase the minimum and maximum memory in file *openemm.sh* in directory */home/openemm/bin* by changing the parameter **-Xms256m** for minimum and **-Xmx512m** for maximum memory of *JAVA_OPTS*. If you have allocated all memory available and the error remains, you should increase your server RAM to at least 2 GByte (better: 4 GByte) and modify the parameter accordingly.

9 SMTP Server/MTA

OpenEMM relies on a SMTP server to send out mails and to accept bounces and replies. OpenEMM uses Sendmail for that task by default, because Sendmail is a proven, secure, and fast MTA and OpenEMM is deeply integrated with Sendmail: For sending emails, OpenEMM creates spool files which can be processed directly by Sendmail, and EMM is able to read Sendmail's log files directly to collect information on the delivery status of each individual email.

Furthermore, OpenEMM uses a multi mailqueue architecture directly build on Sendmail's queue concept. To process the response received by Sendmail, OpenEMM uses a plugin based on Sendmail's milter API. And to be able to handle several domains in parallel, EMM uses Sendmail's mailertable mechanism.

However, if you do not want to (or can not) use Sendmail, you can disable its use after installation of OpenEMM. In this case OpenEMM uses an internal SMTP server (like the Windows version of OpenEMM). If you use the internal SMTP server of OpenEMM, please make sure that no other MTA (like Postfix, qmail or Exim) is active on your machine. OpenEMM does not work with other MTAs than Sendmail or its internal SMTP server!

If you use Sendmail, you do not have to open port 8025 (see chapter 5), but you might have to change some Sendmail configuration files to adapt Sendmail for OpenEMM before installing OpenEMM. Please see appendix A for further details.

9.1 Enable or Disable Sendmail

The use of Sendmail is enabled by default. Depending on your choice whether to use Sendmail or not, you enable Sendmail with
/home/openemm/bin/sendmail-enable.sh
and you disable it with
/home/openemm/bin/sendmail-disable.sh
This has to be done as user *openemm* before starting OpenEMM or after stopping OpenEMM (see section 6.4).

If you plan to use Sendmail you do not have to start (or stop) it, since this is already done by the start script of OpenEMM. When not using Sendmail you can define a smart mailer. To do this create a file named *smart-relay* in directory */home/openemm/conf* with the syntax
{username}:{password}@{smart-relay-domainname}
The use of a smart-relay may be helpful for dial-up users to send out mails via their ISP. The name of the smart-relay is provided by your ISP. In case your ISP's smart-relay does not support TLS, you have to remove the two code lines
smtp.starttls ()
smtp.ehlo ()
in file *semu.py* in directory */home/openemm/bin/scripts*.

Depending on the configuration of the smart-relay you are using, synchronous bounces are either passed through directly back to the sender (OpenEMM) or these instant bounces are sent back to the sender via email and have to be processed by the bounce management for asynchronous bounces (see section 11.2 for details).

9.2 Sendmail Performance Tuning

If you have configured OpenEMM to use Sendmail, it works with several mail queues in parallel to maximize the mail output. Queue ADMIN takes care of all admin and test mailings, which have to be delivered very quickly. The queue named QUEUE is the entry point for all “real” mailings, queue MIDQUEUE holds mails which could not be delivered quickly and, therefore, were shifted to it, and queue SLOWQUEUE holds all mails that even MIDQUEUE could not deliver.

The configuration of these mail queues is done in script *mailer.sh* in directory */home/openemm/bin*. We took great care to choose the best values possible for OpenEMM and we recommend to change the default values only if you run into a real delivery problem.
If you want to tweak the mail queue default values, search for this line:
for stage in 1 2 3 4; do

In the loop that follows, the four mail queues used by OpenEMM are created and configured:

- Parameter *-q* defines how often a certain queue is processed. To achieve a high delivery rate, OpenEMM processes queues ADMIN and QUEUE in 1 minute cycles, but this also clogs the maillog file. You may change cycle time for QUEUE to 5 or 10 minutes. In this case you have to replace parameter *-q1m* in the parameter line with *-q5m* or *-q10m*. MIDQUEUE is processed every 30 minutes and SLOWQUEUE every 90 minutes by default. We do not recommend to change these values.
- Parameters *-OTimeout.iconnect* and *-OTimeout.connect* define the time Sendmail waits for the receiving mail servers to respond. The first parameter is used for the initial try to send out an email and the second parameter is used for all later tries. A short timeout will allow Sendmail to close unused connections earlier so that it has more capacity for new connections to more responsive mail servers. However, if a lot of mail servers are busy, the number of failed mail deliveries will rise.
  Since no timeout values are defined for MIDQUEUE and SLOWQUEUE, Sendmail uses the default values (usually quite high values of 3 or 5 minutes).
- Parameter *count* defines how many processes are created to process the corresponding mail queue.

A few lines further down in script *mailer.sh*, the queue control programm *qctrl* is called three times to define the conditions for moving mails from QUEUE to MIDQUEUE, from MIDQUEUE to SLOWQUEUE and from SLOWQUEUE to Nirwana (aka */dev/null*):

- Parameter *-d* defines the delay in seconds after that a queue is scanned again for mails to be moved to a slower queue (or to */dev/null*). By default, QUEUE is scanned every 13 minutes, MIDQUEUE is scanned every 54 minutes and SLOWQUEUE is scanned about every 6 hours.
- Parameter *tries* defines the number of delivery tries which triggers the move of a mail to a slower queue or the removal from the queue system altogether. By default, a mail is moved from QUEUE to MIDQUEUE after at least 3 failed delivery tries and from MIDQUEUE to SLOWQUEUE after a total of at least 10 failed delivery tries.
- Parameter *maxage* defines the time after which a mail is dropped from SLOWQUEUE. By default, SLOWQUEUE is cleaned from all mails older than 6 days. You may shorten this period to keep the SLOWQUEUE short. However, the lower you go with the value for *maxage*, the higher will be the number of undelivered softbounces. So, you are trading in performance for deliverability here.

10 Upgrade/De-Installation

For security reasons, make a backup of the OpenEMM database and the OpenEMM CMS database first (omit option –p in case your MySQL system is not password protected):
mysqldump -aCceQx --hex-blob –u root -p -r /home/openemm.sql openemm
mysqldump -aCceQx --hex-blob –u root -p -r /home/openemm_cms.sql openemm_cms

Whether you do an automatic or manual upgrade, since you made changes to the default content of file *emm.properties* and *cms.properties* make sure to copy those changes to the new file versions after the upgrade. While the online update of OpenEMM tries to copy your changes into the new files itself we recommend checking them afterwards in order to be sure the values have been copied correctly.

10.1 Automatic Upgrade

If you use OpenEMM 5.4.0 RC1 or later you can use the online update feature in the settings menu of the user interface to upgrade OpenEMM with a single click. If, after you agreed to the update, your browser claims that it can not access the next page, please wait a few seconds for the update process to launch and try again.

If the selected download server causes a problem and the download of the new release hangs, you must kill the upgrade process at the command line. First, find the PID of the process with
ps -u openemm -fww | grep upgrade
This statement should deliver a list with at least one process initiated by *python /home/openemm/bin/scripts/upgrade.py*. Kill this process softly with
kill {pid}
Replace *{pid}* with the PID of the upgrade process. If the process is still alive afterwards, you have to hard kill it with
kill -9 {pid}

After that you can restart OpenEMM, log in and try to start the upgrade again. If you want to go back to the former version of OpenEMM change directory with
cd /home
and check for a directory named *openemm-x.y* (with x.y being the release number). Delete the current directory *openemm* with
rm -rf openemm
and rename the old directory back to *openemm* with
mv openemm-x.y openemm
When you start OpenEMM now, the old version x.y of OpenEMM is started. While changes to the database are not rolled back with this approach this should not cause any problems because the database changes are only important for new features (which are missing in the former version).
However, if you want to start the automatic update again you have to reset your databases to the state before you started the upgrade (when you made your backup):
mysqladmin -u root -p drop openemm
mysqladmin -u root -p drop openemm_cms
mysqladmin -u root -p create openemm
mysqladmin -u root -p create openemm_cms
mysql -u root -p openemm < openemm.sql
mysql -u root -p openemm_cms < openemm_cms.sql

If you have created a file *bav.conf-local* (see section 12.2 below), do not forget to re-create it after every update of OpenEMM – otherwise it will be missing and the management of asynchronous bounces will not work correctly!

10.2 Manual Upgrade and De-Installation

If you want to upgrade to a new version of OpenEMM but you do not want to use the online update feature of OpenEMM, you have to uninstall the current version first. This is done by a few simple steps:
Change to user openemm:
su – openemm
Stop OpenEMM:
openemm.sh stop
Exit openemm user and change back to root:
exit

Uninstall OpenEMM files:
rm -f README.txt UPDATE.txt
rm -rf bin conf logs temp var webapps webservices work
rm -rf /usr/share/doc/OpenEMM-2015

If you want to start your next installation from scratch, simply delete both databases:
mysqladmin -u root -p drop openemm
mysqladmin -u root -p drop openemm_cms

If you want to install a new version of OpenEMM, continue with chapter 6 and omit section 6.1. Otherwise delete the home directory of OpenEMM:
rm -rf /home/openemm
and delete user *openemm* and group *openemm*:
userdel openemm
groupdel openemm

11 Extensions for OpenEMM

11.1 Extending Sendmail Emulation with Plugins

The Sendmail emulation of OpenEMM (*semu.py*) uses a plugin framework (*aps.py*) to implement a plugin manager and to provide extension points (right now method *handleOutgoingMail*). These extension points can be implemented by plugins in order to extend the functionality of the Sendmail emulation. Readme file *aps.readme* in directory */src/script/lib* of the source tarball of OpenEMM provides documentation for the plugin framework like how to implement it and how to write your own extensions.

As an example, Python script *listUnsubscribeHeader.py* demonstrates how to implement extension point *handleOutgoingMail* with your own code. The code of this script adds a line with a global list unsubscribe link to the header of all outgoing mails (method *main* is only implemented to provide a test case). Configuration file *semu.cfg* defines the basic URL used for the unsubscribe link and may be modified to point to an OpenEMM form. Both files are located in directory */home/openemm/conf/semu*.

11.2 Extending the OpenEMM GUI with Plugins

OpenEMM comes with an extension architecture which allows developers to enhance the functionality of OpenEMM with plugins. The plugin manager of OpenEMM let users install and activate plugins with a single click. These plugins fit seamlessly into the GUI of OpenEMM. If you want to know more about the extension architecture of OpenEMM, download the official OpenEMM extension architecture documentation at https://sourceforge.net/projects/openemm/files/OpenEMM%20development/.

12 Domain Name Service (DNS) Configuration

If you need background information on terms like FQDN, hostnames, domains and DNS entries, please see appendix B.

12.1 Reverse DNS

Make sure that a Reverse DNS entry exists for the IP address of your OpenEMM server and that it matches the FQDN of this server. This is important, because most MTAs that receive mails from your OpenEMM installation will do a Reverse DNS lookup in order to check if the FQDN of your server and the reverse DNS entry of your server's IP address match. If not, this is an indication of a spambot network and as a result quite often your emails will be rejected.

12.2 Redirect Service

Basically, OpenEMM runs out of the box. It just requires a simple FQDN, which has to be mapped via a DNS entry to an available (fixed) IP address provided by your ISP. You can use that FQDN for the redirection service provided by OpenEMM. Example: Your machine's hostname is *www* and your domain is *openemm.org*. In that case simply add that FQDN, as described in section 6.2 A. It would look like *http://www.openemm.org:8080*, since the redirection service of OpenEMM usually uses port 8080. If you use port 8080, do not forget to include it in external links pointing to OpenEMM (like subscribe links in forms on your website). Hint: You can map that port to any other port - see appendix C for further details.

12.3 Bounce Management

Bounce management provides you with the capability to keep your mailing lists clean and up-to-date automatically. A bounce message is an error message, which is sent from a mailserver on the recipient's side to the sender if an email is not deliverable. Bounce management administers emails which are undeliverable temporarily (soft bounce) or permanently (hard bounce). It also filters error messages and autoresponder mails.

If you want OpenEMM to process bounces received during the send process (synchronous bounces) no further configuration is required, because bounce management for synchronous bounces works out of the box. However, if you want OpenEMM also to process bounces (and autoresponder mails) which are received hours or even days later (asynchronous bounces), some setup is necessary. This is recommended if you send mailings to large lists because the number of deferred bounces and autoresponder mails will be significant and automated bounce management by OpenEMM will save you a lot of manual work.

If you want to use the bounce management for asynchronous bounces you need to define a dedicated sender hostname for OpenEMM which is different from the server hostname (the existing host name of your server, see file *hosts* in directory */etc*) and you have to set up an A record and a MX (Mail Exchanger) record in your Domain Name Server (DNS) for the sender hostname. The MX record is used to route mail for a domain to one or more IP addresses. OpenEMM needs the new (virtual) host as a destination, to forward all incoming response to, for further processing by OpenEMM's bounce management.

In our example the server hostname is *host* and the sender hostname for OpenEMM will be *news*. The (abbreviated) DNS entry looks like this:

```
---Domain: openemm.org---
host                86400 IN    A     0     83.220.154.85
news                86400 IN    A     0     83.220.154.85
news.openemm.org. 86400 IN    MX    10    host.openemm.org.
---Domain: openemm.org---
```

The first line assigns the IP address for *openemm.org* and the second line defines the regular hostname. The third and fourth line define the A record and MX record for sender hostname *news*, meaning that host *host* accepts emails sent to host *news*.

Validate your setup is correct by using a tool like *host* or *dig*, for example:
host –a openemm.org
host –a host.openemm.org

host –a news.openemm.org

When you send emails and want to take advantage of the bounce management for asynchronous bounces, there are two possibilities for the format of the sender address:

A.) Use whatever sender address you like. Implement a forward mechanism in the email account of this sender address to forward incoming mail sent back to the sender address to a filter address of OpenEMM. In order to create this filter address, set up a bounce filter in OpenEMM (see user manual). This filter will auto-generate a filter address like *ext_1@news.openemm.org*. After processing the incoming mail, the bounce filter will forward the filtered response to a feedback email address of your choice (different from the sender address, of course).
The flow for responses to your mailings works like this:
recipient → sender address → filter address of bounce filter (auto-generated by OpenEMM) -> feedback address
The advantage of this model is that you can choose any sender address you want, but you have to implement an external forward mechanism.

B.) Use a sender address with the sender hostname (in our example *news@news.openemm.org*) Since no real email addresses exist for this sender hostname, normally it would not be possible to reply to an email with this sender address. To forward responses to a valid email address you have to define a bounce filter. In the configuration for the bounce filter set field *Address* to a feedback address of your choice. The forward address generated by the bounce filter (in our example *ext_1@news.openemm.org*) has to be defined as an alias in directory */home/openemm/conf/bav* in a new file named *bav.conf-local*. Our example:

---File: /home/openemm/conf/bav/bav.conf-local----
news@news.openemm.org alias:ext_1@news.openemm.org
---File: /home/openemm/conf/bav/bav.conf-local ----

The flow of responses to your mailings works like this:
recipient → sender address → bav.conf-local → filter address of bounce filter (auto-generated by OpenEMM) → feedback address

If you create the file *bav.conf-local* do not forget to re-create it after every update of OpenEMM – otherwise it will be missing and the management of asynchronous bounces will not work correctly!

12.4 Softbounce Scoring

If an email address generates lots of softbounces (temporary delivery problems) this is actually an indication that the email address is undeliverable permanently (hardbounce). OpenEMM provides softbounce scoring to identify those email addresses and to convert them to hardbounces. To enable this conversion you should run the script *softbounce.sh* daily as user *openemm*. The best way to accomplish this is to create a cron job with
su - openemm
crontab -e
i (to enter edit mode)
0 3 * * * /home/openemm/bin/softbounce.sh
[Esc]
:x
This crontab entry would start *softbounce.sh* at 3:00 am. *softbounce.sh* analyses the bounces and writes all softbounces to a special softbounce table. If the bounce generating email address already exists, its bounce count is incremented.

The rules for converting a softbounce to a hardbounce work like this:

1. Select all email addresses in the softbounce table which generated more than 7 softbounces and where the time-lag between the first and last bounce is longer than 30 days.
2. If no mail opening or link click was registered within the last 30 days for an email address which matchs the before-mentioned conditions, this address is flagged as a hardbounce.
3. If at least one opening or click was registered within the last 30 days, this address is removed from the softbounce table, i.e. its bounce count is reset to zero.

12.5 Hardbounces vs. Softbounces

Some advanced users of OpenEMM have noticed that OpenEMM does not treat all hardbounce messages reported by remote mail servers as hardbounce. In fact, bounce messages with code 500, 550 or 554 are treated as softbounces, although bounce codes starting with 5 would indicate a hardbounce.

The reason for this kind of ignorant behaviour ist intentional, because some mail servers are not properly configured regarding the generation of hardbounce messages and mistakenly report permanent delivery errors - some even by intention to pretend that certain email addresses do not exist. If OpenEMM would handle those fake hardbounce messages as real hardbounces email addresses of existing recipients would be disabled. As result, we only try to accept bounces as hardbounces which are really proved to be hardbounces. These are codes 511 (“user unkown”), 512 (“domain unknown”) and all other hardbounces where no excluding rule is defined.

File *bav.rule*, section *[hard]* in directory */home/openemm/conf/bav* lists bounce text messages which are recognized as hardbounces by OpenEMM's bounce management. You may add your own set of messages here. Bounce messages with code 500, 550 or 554 will still be treated as softbounces, nevertheless.

If, for example, you want to use *bav.rule* to catch all bounces with text messages containing bounce codes from 550 to 559 including DSN (delivery status notification) 5.1.1 or 5.1.2, add this rule in section *[hard]* of *bav.rule*:

Remote server replied: 55[0-9] 5\.1\.[12]

By the way, if a hardbounce message is recognized as a softbounce even if it is a real hardbounce, this is not a problem. Because a real hardbounce is reported for each mailing again and is counted as a softbounce each time, it will be finally caught by the softbounce scoring of OpenEMM (see previous section) and converted to a hardbounce in the end.

13 Appendix A: Configuration of Sendmail

If you want to use OpenEMM bounce management not only for synchronous bounces, but also for asynchronous bounces, some Sendmail configuration is required – when entering the following lines, please make sure that each time the initial apostrophe is a back tick (`), otherwise the M4 preprocessor will fail to interpret the input correctly!

By the way, make sure that SELinux is disabled, so that Sendmail is able to invoke OpenEMM's mail filter (milter). Open file *config* in directory */etc/selinux/* and change property SELINUX to
SELINUX=disabled

13.1 RedHat Linux and Ubuntu:

Open file *sendmail.mc* in directory */etc/mail* and change the line
- RedHat: **DAEMON_OPTIONS(`Port=smtp,Addr=127.0.0.1, Name=MTA')dnl**
- Ubuntu: **DAEMON_OPTIONS(`Family=inet, Name=MTA-v4, Port=smtp, Addr=127.0.0.1')dnl**
to
- RedHat: **dnl DAEMON_OPTIONS(`Port=smtp,Addr=127.0.0.1, Name=MTA')dnl**
- Ubuntu: **dnl DAEMON_OPTIONS(`Family=inet, Name=MTA-v4, Port=smtp, Addr=127.0.0.1')dnl**

This will enable Sendmail to listen on all available network interfaces. By default Sendmail listens only on the local interface *lo0* for connections and “dnl” comments out this directive.

Add the following line at the end of the file:
INPUT_MAIL_FILTER(`bav', `S=unix:/home/openemm/var/run/bav.sock, F=T')dnl
This will enable the dynamic mail loop required by the bounce management to process asynchronous bounces.

For Ubuntu enter the following line in file *sendmail.mc* after the line starting with
FEATURE(`no_default_msa
FEATURE(`mailertable', `hash -o /etc/mail/mailertable.db')dnl
to activate the mailertable feature and create the required database *mailertable.db*.

If file *relay-domains* does not exist in directory */etc/mail*, create the file - for example with
touch relay-domains
and add a line at the end of the file which specifies your DNS entry for the sender hostname (FQDN). In our example it is simply:
news.openemm.org
This will make sure that responses to an email address with domain news.openemm.org are accepted by Sendmail for relaying.

Open file *mailertable* in the same directory or create it if it does not exist with
touch mailertable
and add a line at the end which activates the internal forwarding for the sender hostname to procmail for filtering:
news.openemm.org procmail:/home/openemm/conf/bav/bav.rc

To activate all changes to the Sendmail configuration, make sure to have package *sendmail-cf* installed and run the following commands:
make -C /etc/mail
/etc/init.d/sendmail reload
You may ignore the warning that */home/openemm/var/run/bav.sock* is missing, since this file will be provided during installation of OpenEMM

13.2 SuSE Linux

WARNING: Editing the files mentioned below breaks the YaST configuration capabilities for Sendmail. However, you can later reactivate YaST via
MAIL_CREATE_CONFIG="yes"
in file */etc/sysconfig/mail* and YaST will not overwrite your Sendmail configuration but save the new file as sendmail.cf.{name} so that you can compare the settings (with diff). If there are too many changes to copy them manually into the existing *sendmail.cf*, rename the new file to *sendmail.cf* and run
/sbin/SuSEconfig

and repeat the steps in the following section.

Open file */etc/sysconfig/mail* and change the line
MAIL_CREATE_CONFIG="yes"
to
MAIL_CREATE_CONFIG="no"
This line excludes Yast from Sendmail configuration and allows you to change the configuration manually by yourself.

Open file */etc/mail/linux.mc* and change line
dnl undefine(`confHOST_STATUS_DIRECTORY')dnl
to
undefine(`confHOST_STATUS_DIRECTORY')dnl

Check the file for line
MAILER(procmail)dnl
and add it at the end if is not there and add the following line at the end:
INPUT_MAIL_FILTER(`bav',`S=unix:/home/openemm/var/run/bav.sock,F=T')dnl

If file */etc/mail/relay-domains* does not exist, create the file - for example with
touch relay-domains
and add a line at the end of the file which specifies your DNS entry for the sender hostname (FQDN). In our example it is simply:
news.openemm.org

Open file */etc/mail/mailertable* and add a line at the end which activates bounce management for that FQDN:
news.openemm.org procmail:/home/openemm/conf/bav/bav.rc

To activate all changes to the Sendmail configuration, run the following commands:
cd /etc/mail
m4 linux.mc > /etc/sendmail.cf
m4 linux.submit.mc > submit.mc
make
/etc/init.d/sendmail reload
You may ignore the warning that */home/openemm/var/run/bav.sock* is missing, since this file will be provided during installation of OpenEMM.

IMPORTANT: If you use AppArmor with SuSE, it requires the following entries for the file */etc/apparmor.d/usr.sbin.sendmail*:
/home/openemm/var/spool/ADMIN rwl,
/home/openemm/var/spool/ADMIN/* rwl,
/home/openemm/var/spool/QUEUE rwl,
/home/openemm/var/spool/QUEUE/* rwl,

Otherwise, Sendmail will not be able to communicate with OpenEMM.

Finally, restart the AppArmor Service with
/etc/init.d/boot.apparmor reload

14 Appendix B: DNS Entries, FQDN, Hostnames and Domains

14.1 What is a DNS entry and what is its purpose?

A DNS entry maps the IP address of a server to a human readable address. Example: In place of the IP address 83.220.154.85, which points to the OpenEMM webserver, you can use the DNS address www.openemm.org, which is much more convenient.

14.2 What is a Hostname, a Domain and a FQDN

A Fully Qualified Domain Name (FQDN) links to an IP address of a server. The FQDN may be composed of letters and numbers and by using this option nobody has to remember the difficult number sequence (IP). A FQDN is divided in three levels:
- The affix of the domain is the Top Level Domain (TLD). Example: *com*, *org* or *net*
- The domain name will be inserted in front of the TLD. Example: *openemm* or *agnitas*
- The FQDN starts with the hostname. For webpages this is very often *www*

Example: The FQDN *www.yourdomain.com* is composed of
- www = hostname
- yourdomain = domain name
- com = TLD

As you can see, the FQDN consists of the hostname, the domain name and the top level domain separated by dots. The combination of domain name and TLD is commonly referred as domain. The FQDN can be expanded by a subdomain (like *miami*). The subdomain will be inserted between the hostname and the domain. Example: www.miami.yourdomain.com

14.3 How do I get a Domain and a FQDN?

There a lot of providers where you may host a domain. You will only host the combination of the TLD and the domain name. Example: *yourdomain.com*. You may link a domain name to different targets by using different hostnames. The domain name will be registered with a Domain Name Server (DNS). This server forwards all requests to the particular IP address. After your domain name has been registered, you may set up the FQDN in the provider's web interface. The provider allows you to define several hostnames to create different FQDNs, which will forward to different servers (or - with the help of your firewall - to different ports of the same server). You may set up different addresses like
- web server: *www.yourdomain.com*
- mail server: *mail.yourdomain.com*
- FTP server: *ftp.yourdomain.com*

15 Appendix C: OpenEMM as Redirection Server on Port 80

You can use your server as a redirect server to track mail opening rates and link clicks. This is helpful to determine the success of an email marketing campaign. By default, OpenEMM enables that service on port 8080. If you want to use a URL without an explicit declaration of a port, this section shows one way to achieve this.

To use your system as a redirection server on HTTP default port 80, first make sure that there are no conflicting services running on TCP port 80, like an Apache Httpd server. On RedHat Linux the check is done by running
netstat -ant | grep ':::80'
If you see active services, you have to stop them. Example: To stop an active Apache Httpd server run
/etc/init.d/httpd stop
Also make sure that these services do not start automatically after system reboot (for example by using *chkconfig*).

15.1 RedHat Linux and Ubuntu

Enable a Prerouting Forwarding Rule from port 80 to 8080 by adding the following code after the comments at the top in file */etc/sysconfig/iptables*:
***nat**
:PREROUTING ACCEPT [0:0]
:POSTROUTING ACCEPT [0:0]
:OUTPUT ACCEPT [0:0]
-A PREROUTING -i eth+ -p tcp --dport 80 -j REDIRECT --to-port 8080
COMMIT

Committing the changes requires a restart of the firewall, which is done with
- RedHat: **/etc/init.d/iptables restart**
- Ubuntu: **service ufw restart**

15.2 SuSe Linux

Enable the prerouting forwarding rule from port 80 to 8080 by setting parameter *FW_REDIRECT* in file */etc/sysconfig/SuSEfirewall2* to
"0/0,0/0,tcp,80,8080"

Committing this change is done with
/etc/init.d/SuSEfirewall2_setup restart

15.3 Changes to the Database

When you have implemented port forwarding as described above, the "old" port 8080 still works, of course. Therefore, you do not have to modify the URLs in existing mailings. However you should change the field *rdir_domain* in table *company_tbl* by removing the substring ":8080" at the end of the domain name like so:
update company_tbl set rdir_domain = 'www.openemm.org';

16 Credits

Lead author: Martin Aschoff (maschoff@agnitas.com)
Contributors: Anton Melser, Thomas Wittmann