

# **Fusion FreeACS Installation**

*Version 2014R1*

## *Table of Contents*

## **Table of Contents**

.....	1
Fusion FreeACS Installation .....	1
Document Introduction .....	3
Name of the system .....	3
Document Purpose .....	3
Document Audience .....	3
Document History .....	3
Quick Overview .....	4
Infrastructure: .....	4
Modules .....	4
Customer requirements .....	5
Hardware .....	5
OS .....	6
Database .....	6
Java and Web container .....	6
Installation from scratch.....	8
/etc/mysql/my.cnf .....	8
/var/lib/tomcat7/conf/catalina.properties.....	8
/etc/default/tomcat7 .....	9
/etc/passwd .....	9
/var/lib/tomcat7/conf/server.xml .....	9
/etc/init.d/tomcat7.....	9
/var/lib/tomcat7/common/xaps-monitor.properties .....	9
/var/lib/tomcat7/common/xaps-stun.properties.....	10
/var/lib/tomcat7/common/xaps-web.properties.....	10
Restart, firewalls and checks .....	10
Optional steps – Tomcat on HTTPS (SSL) .....	10
Technical reference and documentation .....	13
Property files .....	13
Log files.....	13
Fusion logs .....	13
Firewalls .....	13
Documentation .....	14

# 1 Document Introduction

## 1.1 Name of the system

The current name of the system is “Fusion FreeACS”. As this is a relatively new name, the old name “Fusion” is in frequent use and may continue to be for a very long time. An even older name “xAPS” is also in use.

## 1.2 Document Purpose

The purpose of the document is to explain how to install Fusion FreeACS (chapter 4).

## 1.3 Document Audience

The readers will be Fusion Administrators and System Operators.

## 1.4 Document History

Version	Editor	Date	Changes
2009-R1	Morten Simonsen	18-Feb-09	Initial public version.
2009-R1-U1	Fredrik Gratte	31-Mar-09	Updated platform requirements.
2009-R2	Morten Simonsen	02-Jul-09	Revised edition
2011R1	Morten Simonsen	21-Jan-11	Revised edition
2012R1	Morten Simonsen	28-Dec-11	Name change/upgrade from 2011R1 procedure. Added a chapter.
2013R1	Morten Simonsen	17-Jan-13	Updated to latest release
2014R1	Morten Simonsen	03-Feb-14	System is no longer a commercial product – is licensed under the MIT license for free usage. The differences between 2013R1 and 2014R1 are otherwise small.
2014R1	Morten Simonsen	07-Jul-14	Major overhaul. Has created an install-script to do most of the work automatically. Updated to run on Ubuntu 14.04. The installation procedure has been brought down to minimum 5-6 minutes.

## 2 Quick Overview

**A complete installation of a Default Setup is provided in chapter 4, you may skip chapter 2 and 3.**

Fusion can be run in several configurations depending on your needs. The following section will list all modules and comment on where there is a choice to be made.

### ***2.1 Infrastructure:***

This list can also be read as requirements from Ping Communication to the customer, as the customer needs to be knowledgeable about these infrastructure parts, or at the very least be able to acquire the necessary knowledge, to maintain all these components.

- Fusion can run on one physical server if necessary. Several factors come into play to decide how many servers is optimal.
- Operating system which can run JRE 1.7 (see details and exception to this below in the 'OS' chapter).
- MySQL 5.5
- JRE 1.7 (latest update).
- Tomcat 7 (other web containers are possible, but not described)

### ***2.2 Modules***

North-side modules (user interface modules):

- Fusion Web (Standard web interface for management)
- Fusion Shell (CLI, script automation, management)
- Fusion Web Services (if system integration is needed)

Core modules:

- Fusion DB (table definitions)
- Fusion Core
- Fusion Syslog Server
- Fusion Monitor Server

South-side modules (CPE interface modules):

- Fusion TR-069 Server (if you have TR-069 devices)
- Fusion STUN (needed to support TR-111)
- Fusion SPP (needed to provision HTTP/TFTP/Telnet)

The following chapters will explain how to install these modules.

### 3 Customer requirements

#### 3.1 Hardware

There are many ways one could organize the hardware to satisfy Fusion. You could do with one physical server at start up. As the number of devices connected to Fusion grows, you should probably split the processes/modules on several servers. This table should give you a quick overview of how Ping Communication thinks about this issue:

CPEs	Connects pr 24h	Devices w/syslog	Fusion Web available on internet	Servers required	Comments
50K	1	Y	Y	1	The minimum requirement, see below for spec. for server.
500K	1	N	N	2	You should have server with the provisioning server (TR-069 or OPP) in DMZ and the rest of the modules on another server within your intranet. This requirement is mainly due to security reasons.
500K	1	Y	N	3	Same as for the above, but syslog could generate a huge load, so it could be smart to have a separate server for the Fusion Syslog Server & Fusion Syslog DB.
500K	24	N	N	5	You should have 3 provisioning servers, since the CPEs connect 24 times a day. The database would be put under some load here, so the database should also be place on its own server. The rest of the modules could be placed on one server.
12M	1	Y	Y	7	The same load as in the previous example, but syslog is turned on, so it will require a database server extra. And don't put Fusion Web on one of the provisioning server, just because they are both located in the DMZ, use a separate server for that module.

A server is expected to have a decent multi-core processor, minimum 8GB RAM, minimum 100 Mbit network interface and hard disk capacity of at least 500GB (this last requirement is only important for the database server). We expect the usage of fast HDD, since this is critical for the database. The minimum specification translates to some of the test servers we have used. Looking closely at these figures you should realize that this specification is a low-end system these days. A state-of-the-art system today (jan 2013) would probably have more capacity. So if you think the number of servers will grow too rapidly with increasing numbers of connects pr 24h, keep in mind that in that situation you would probably use a state-of-the-art system, minimizing the number of servers required.

The tests we have done to come up with this list will of course not represent the absolute truth about how a potential customer will use the system. Particularly the number of parameters in

the database, jobs activated, logging scheme, number of end users, number of interconnecting systems, will influence the performance. That said, we think these figures give a reasonable and reliable picture of the situation.

If you decide to run on multiple servers, the first split should be between Fusion DB and the provisioning servers (SPP or TR-69), since these components are affected the most by an increase in devices. Another important point is that you can add provisioning servers to scale up the system, all of them connecting to the same Fusion DB Server. There is another reason for this split as well, and that is that the provisioning servers must be reachable for all the devices, a requirement which you might not want for your database!

Another split would be to put all the interface modules (Fusion Web, Fusion Shell and Fusion Web Services) on a separate server. A trigger for this move would be to secure these interfaces from direct access from the Internet.

Yet another split would be to put a syslog server and the syslog database on its own server, but that is something one does only if there's a significant load on the syslog server.

The bottleneck of this system will eventually be the database. However we believe that this bottleneck will not be hit before at least 10M CPEs are connected, possibly not before 30-50M CPEs are connected (it all depends on many factors). But this does not take into account that the database server may run in a cluster. We have not experimented with this, but we still believe this is an option, an option that no customer today is likely to reach without a very aggressive provisioning policy (e.g. many connects pr CPE every 24h).

### **3.2 OS**

All modules in Fusion are Java applications. In theory they can be installed on any OS that supports JRE 1.7 and has an available web-container (like Tomcat 7). We have chosen to run on Linux, Ubuntu Server 14.04, 64 bit, and we suggest that our customers do the same. If they do, it's easy to follow the installation procedure in chapter 5.

### **3.3 Database**

Currently Fusion will only run on MySQL 5.5.

How to install MySQL is considered the responsibility of the customer. Furthermore it may be necessary to tweak the database somewhat as the load grows. This competence should be found within your company. That is to say that Fusion is not a fool-proof system and will require some technical people to take part in the installation and operation.

That said we do have an installation procedure for a complete set up of a standard Fusion Server which includes a reasonably good set up of MySQL 5.5. This was done to minimize our own support effort in the installation process, but also serves us well because the installation of Fusion becomes more coherent across customers. Chapter 5 contains the detailed installation procedure of such a standard set up.

### **3.4 Java and Web container**

As for the databases, you need to be able to install Java and a web container on your system. Fusion requires JRE 1.7, preferably the latest update. When this is installed you can install the

web container. Tomcat 7 has been used in development but other web containers can also be used, since they offer the same runtime environment for Fusion applications.

## 4 Installation from scratch

**You can have Fusion FreeACS up and running in 30 minutes, or possibly even in just 5 minutes (if you do this for the second time). Just read on.**

The goal is to install a standard Fusion Server (Default Setup), which requires installation of Ubuntu Server 14.04 64-bit, MySQL Server 5.6 (latest update), Tomcat 7 and JRE 1.7 (latest update). How to install Ubuntu 14.04 64-bit is beyond the scope of this simple document, but otherwise all other software installation is described. This is not to say that one cannot run on any other OS or J2EE server, but this is the standard/default Fusion installation recommended for most users.

Do the following:

1. **Download `install-or-update-freeacs-ubuntu.sh` from <http://freeacs.com/download> and run the script from your home folder on your ubuntu server. You must have root access. This covers 90% of the installation, and can be done in 1-5 minutes.**
2. **Go through the rest of the modifications described in this chapter. Should be possible to do in 5-30 minutes.**
3. **The server should be ready**

**Important!!** Yellow color indicates an optional step, but it's wise to read the comments before skipping.

### 4.1 `/etc/mysql/my.cnf`

Step	Command/Text	Comment
4.1.1	<code>bind-address = 0.0.0.0</code>	If you want your database to be accessible from outside localhost, set to 0.0.0.0. Else, the database will only be accessible for applications running on localhost (like Fusion server).
4.1.2	<code>max_allowed_packet = 32M</code>	Should be at least 32M, to allow adding firmwares up until this size into the Fusion database
4.1.3	<code>innodb_buffer_pool_size=1024M</code>	This is the most important memory setting, MySQL should have access to perhaps 50% av of total memory on server. If you set this setting to high MySQLs InnoDB engine may silently fail! Check in 2.7.
4.1.4	<code>service mysql restart</code>	Restart MySQL after changes

### 4.2 `/var/lib/tomcat7/conf/catalina.properties`

Step	Command/Text	Comment
4.2.1	Append the following:  <code>,\${catalina.base}/common,\${catalina.bas</code>	Edit the configuration file of tomcat to point to a directory where Fusion properties will be placed. There should



	e}/common/*.properties to the property "common.loader"	not be any line breaks or spaces in the appended text!
--	---	--

### 4.3 /etc/default/tomcat7

Step	Command/Text	Comment
4.3.1	AUTHBIND=yes	<b>Optional!</b> AUTHBIND=yes will make it possible for Tomcat to run on lower ports (80 and 443). Take care to remove the # comment at the beginning of the line.
4.3.2	JAVA_OPTS="-Djava.awt.headless=true -Xmx768m -XX:+UseConcMarkSweepGC"	JAVA_OPTS is only changed slightly, to increase maximum memory usage from 128 megabyte till 768 megabyte. This should be sufficient for 10-50K devices.

### 4.4 /etc/passwd

Step	Command/Text	Comment
4.4.1	Change "/usr/share/tomcat7" to "/var/lib/tomcat7"	This changes Tomcat's home directory from /usr/share/tomcat7 to /var/lib/tomcat7.

### 4.5 /var/lib/tomcat7/conf/server.xml

Step	Command/Text	Comment
4.5.1	<Connector port="80" protocol="HTTP/1.1" ....	Default setup of Tomcat is port 8080, we'll change it to 80 (default HTTP port).

### 4.6 /etc/init.d/tomcat7

Step	Command/Text	Comment
4.6.1	Find the line beginning with "# Required-Start:" and append "\$mysql" to it.	Some Fusion services have a habit of reporting e-mail errors if they cannot connect to the database when they start. (E.g. when rebooting the machine) To avoid this, we make Tomcat depend on MySQL for it to start on boot.

### 4.7 /var/lib/tomcat7/common/xaps-monitor.properties

The monitor server itself is not critical for FreeACS, it's main job is to send email and monitor the other servers in the FreeACS solution.

Step	Property	Comment
4.7.1	mail-settings	Specify in order to get mail about events and errors in FreeACS
4.7.2	fusion.urlbase	This url will be used in mail sent to you; specify a url-base which can reach the FreeACS from "outside"

#### 4.8 /var/lib/tomcat7/common/xaps-stun.properties

The STUN server is fairly important, since all server-side triggering of provisioning goes through this server. Thus, if you try to «kick» the CPE or press the «provisioning» button in the Web interface, the STUN server must have a correct configuration.

Step	Property	Comment
4.8.1	primary.ip	Set it to the IP address of your server. The server will try to bind to this IP on port 3478. If this fails, the server will not start unless you change the test.runwithstun
4.8.2	test.runwithstun	The server will start even if the STUN behaviour is not supported. In this case, the server can still be used to trigger/kick CPEs available on public ConnectionRequestURL addresses.

#### 4.9 /var/lib/tomcat7/common/xaps-web.properties

Step	Property	Comment
4.9.1	monitor.location	It should return a web-page (use wget to test). If not, change the url or check if the Monitor server is actually running.

#### 4.10 Restart, firewalls and checks

Step	Command/Text	Comment
4.10.1	service tomcat7 restart	Check /var/lib/tomcat7/logs/catalina.out to make sure Tomcat7 starts without errors.
4.10.2	wget localhost wget localhost/web wget localhost/tr069	<p>If you have a firewall, open for TCP/80. You can check to see if tomcat is available by using the command. If everything went well you should get the FreeACS Web interface, with an user/password prompt. <b>Login using admin/xaps as user/pass</b>. You may of course change the default password inside the web application.</p> <p>If the FreeACS Web interface does not appear, then try http://localhost/web. The TR-069 server should be available on http://localhost/tr069. The TR-069 clients will connect using HTTP POST, while the “browser” returns the response from HTTP GET.</p>
4.10.3	fusionshell	You should log out of Ubuntu and log in again, before attempting this command, unless you might get some error messages. This shell is crucial is providing a scripting environment to FreeACS.
4.10.4	See chapter 5.3	Several port openings may be expected if a firewall is present
4.10.5	COMPLETE	The server is now ready!

#### 4.11 Optional steps – Tomcat on HTTPS (SSL)

At this point in the installation, all FreeACS servers run on the same Tomcat instance. Let's say you want to have a secure communication with your CPE (most people do), you would then need to create a SSL server-side certificate and all your CPEs would have to accept this certificate. This certificate would be shared by all servers running in the same Tomcat instance. The consequence is that if you, for some reasons want to have a different certificate

for you FreeACS Web interface – that server would have to run on another Tomcat instance, most likely on another host. The same goes for the TR-069 server, if you want to support various CPEs which demands a particular SSL certificate, you need to create several TR-069 servers, each with it's own certificate installed.

If you run Fusion TR-069 server AND you want to provision Ping Communication devices using TR-069, please follow step 7.1-alt1.

If you do not provision Ping Communication devices, but still wish to avoid warnings when you use Fusion Web, please follow step 7.1-alt2

Otherwise follow step 7.1-alt3 or 7.1-alt4.

Step	Command	Comment
5.1-alt1-1		Buy (class2) or get for free (class1) a certificate from StartSSL ( <a href="http://www.startssl.com/">http://www.startssl.com/</a> ).
5.1-alt1-2	<code>unzip build_jks.zip</code>	
5.1-alt1-3	<code>chmod 755 *.sh</code>	Make the scripts runnable
5.1-alt1-4	<code>./build_jks_class1.sh</code>	The command will show you help text and how to run the script. If you bought class2 certificates, run the other script.
5.1-alt2		Buy your own certificate and import it into a java keystore (using <code>keytool -import</code> ). You may of course use the suggested certificate from StartSSL (5.1-alt1-1)
5.1-alt3	<code>keytool -genkey -alias xaps -keyalg RSA -validity 10000 -keystore /var/lib/tomcat7/.keystore</code>	Follow the steps show in screenshot 7.1. The information you enter into the certificate will only be shown when you examine the certificate in a browser. The certificate should be valid for 10000 days. Note that browsers do not like this self-signed certificate, if you want a real certificate, you must purchase one.
5.1-alt4		Copy keystore from old server/installation into you're working folder. In that case, you should also use the same keystore-password and alias as you had in the old server.xml.
5.2	<code>keytool -list -keystore /var/lib/tomcat7/.keystore -storepass importkey</code>	List the key aliases, you will need it in the next step.
5.3	<code>pico /var/lib/tomcat7/conf/server.xml</code>	Edit the file to match screenshot 7.2, using the alias acquired in the previous step. This change will still allow traffic on HTTP, to avoid that uncomment the connector for port 8080 (or 80)
5.4	<code>service tomcat7 restart</code>	Start a browser and locate the Fusion Web interface. Usually <code>https://localhost</code>
5.5		If you have a firewall, open for TCP/443

```
eric@fusion-a: /var/lib/tomcat7/conf
root@fusion-a:/# keytool -genkey -alias xaps -keyalg RSA -validity 10000 -keystore /var/lib/tomcat7/.
keystore
Enter keystore password:
Re-enter new password:
What is your first and last name?
[Unknown]: Comp
What is the name of your organizational unit?
[Unknown]: Comp
What is the name of your organization?
[Unknown]: Comp
What is the name of your City or Locality?
[Unknown]:
What is the name of your State or Province?
[Unknown]:
What is the two-letter country code for this unit?
[Unknown]:
Is CN=Comp, OU=Comp, O=Comp, L=Unknown, ST=Unknown, C=Unknown correct?
[no]: y

Enter key password for <xaps>
(RETURN if same as keystore password):
root@fusion-a:/#
```

Screenshot 7.1

```
eric@fusion-a: /var/lib/tomcat7/conf
File Edit Options Buffers Tools XML Help
redirectPort="8443" />
<!-- Define a SSL HTTP/1.1 Connector on port 8443
This connector uses the JSSE configuration, when using APR, the
connector should be using the OpenSSL style configuration
described in the APR documentation -->
<Connector port="8443" protocol="HTTP/1.1" SSLEnabled="true"
maxThreads="150" scheme="https" secure="true"
clientAuth="false" sslProtocol="TLS"
keystoreFile="/var/lib/tomcat7/.keystore"
keystorePass="importkey" keyAlias="xaps" />
<!-- Define an AJP 1.3 Connector on port 8009 -->
<!--
<Connector port="8009" protocol="AJP/1.3" redirectPort="8443" />
-->
-UUU:**--F1 server.xml 57% L91 (nXML Valid)-----
```

Screenshot 7.2

## 5 Technical reference and documentation

In this chapter you'll find important information of a installed Fusion system; where to find log files, firewall settings, etc. By following the instructions in chapter 4, you'll end up with a Default Setup (DS), and for this setup we'll provide exact information.

### 5.1 Property files

Property files are found in **/var/lib/tomcat7/common** and also in **/var/lib/tomcat7/shell**. Each module has two property files following these conventions:

**xaps-<modulename>.properties**: Contains all properties and control mechanism for the module.

**xaps-<modulename>-logs.properties**: Contains all properties to control logging, number of logs, name of logs, loglevels, backups, etc.

Information about the various property files are found in the User Manuals of each module, but each property file is supposed to be self-documented.

### 5.2 Log files

#### Fusion logs

Log files are found in **/var/lib/tomcat7**. The logs are named following this convention: **fusion-<modulename>(-<optionalname>).log**. Usually every module has a default/regular log: **fusion-<modulename>.log**, but some modules have multiple logs. Each log file can be controlled by settings in the corresponding **xaps-<modulename>-logs.properties** files (see previous chapter).

Old logs can be found in the backup-logs directory. These logs are kept for as long as specified in the logs-properties file.

### 5.3 Firewalls

The following holes in the firewall may/must be opened (for those modules placed behind the firewall):

Module	Port	Type	Comment
Monitor, TR069, SPP, Web, WS	80	TCP	In case you have setup the installation to run on port 80 (see chapter 4.5) To allow requests into TR-069 or HTTP for provisioning. Also access to monitor-server, Web and Web Services.
Monitor, TR069, SPP, Web, WS	8080	TCP	In case you run DS (skipped chapter 4.5). To allow requests into TR-069 or HTTP for provisioning. Also access to monitor-server, Web and Web Services.
Monitor, TR069, SPP, Web, WS	443	TCP	In case you have setup the installation to run on port 443 (see chapter 4.8). To allow requests into TR-069 or HTTP for provisioning. Also access to monitor-server, Web and Web

			Services.
SPP	69	TCP	To allow TFTP-provisioning (offered by the SPP server)
DB	3306	TCP	Allows direct access to MySQL database (see chapter 4.1). This allows Fusion Shell to run on a remote host accessing the DB directly.
STUN	3479 3480	UDP	To support TR-111 (the devices must also support this) and devices access this STUN-server.
Syslog	9116	UDP	To allow syslog messages to be sent to Fusion Syslog server. Should always be open.

## 5.4 Documentation

All modules have a User Manual, to describe how to use the system. Some modules also have additional documentation. These documents are found in GitHub on the following locations.:

Server	URL	Comment
<b>General</b>	<a href="https://github.com/freeacs/readme">https://github.com/freeacs/readme</a>	General documentation
<b>Core</b>	<a href="https://github.com/freeacs/core/tree/master/docs">https://github.com/freeacs/core/tree/master/docs</a>	
<b>Monitor</b>	<a href="https://github.com/freeacs/monitor/tree/master/docs">https://github.com/freeacs/monitor/tree/master/docs</a>	
<b>Shell</b>	<a href="https://github.com/freeacs/shell/tree/master/docs">https://github.com/freeacs/shell/tree/master/docs</a>	
<b>SPP</b>	<a href="https://github.com/freeacs/spp/tree/master/docs">https://github.com/freeacs/spp/tree/master/docs</a>	
<b>STUN</b>	<a href="https://github.com/freeacs/tr069/tree/master/docs">https://github.com/freeacs/tr069/tree/master/docs</a>	Chapter 7
<b>Syslog</b>	<a href="https://github.com/freeacs/syslog/tree/master/docs">https://github.com/freeacs/syslog/tree/master/docs</a>	
<b>TR-069</b>	<a href="https://github.com/freeacs/tr069/tree/master/docs">https://github.com/freeacs/tr069/tree/master/docs</a>	
<b>Web</b>	<a href="https://github.com/freeacs/web/tree/master/docs">https://github.com/freeacs/web/tree/master/docs</a>	
<b>Web Services</b>	<a href="https://github.com/freeacs/ws/tree/master/docs">https://github.com/freeacs/ws/tree/master/docs</a>	