Fusion Installation

Version 2013R1

© 2013 Ping Communication AS

Table of Contents

1 Document Introduction

1.1 Name of system

The Fusion system was previously named xAPS. This will be reflected in the installation procedure at the end of this document.

1.2 Document Purpose

The purpose of the document is to explain how to install or migrate Fusion (chapter 4 or 5).

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

1.5 References

Document
[1]   Fusion Web User Manual
[2]   Fusion Shell User Manual
[3]   Fusion Core User Manual
[4]   Fusion Monitor Server User Manual
[5]   Fusion Web Services User Manual
[6]   Fusion TR-069 Server User Manual
[7]   Fusion Syslog Server User Manual
[8]   Fusion SPP Server User Manual
[9]   Fusion Database User Manual

2 Quick Overview

A complete installation of a Default Setup is provided in chapter 5.

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 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 12.04 LTS, 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. Should it be necessary, Fusion can be made to run on Oracle relatively easy, since we've supported Oracle in previous versions. Such a support will have to be initiated by a customer.

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

Important!! If you are running Fusion 2012R1, go to next chapter for migration instructions.

The goal is to install a standard Fusion Server (Default Setup), which requires installation of Ubuntu Server 12.04 LTS 64-bit, MySQL Server 5.5 (latest update), Tomcat 7 and JRE 1.7 (latest update). How to install Ubuntu 12.04 LTS 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.

Important!! In this installation document we use **bold font** to signal variable input. Do not copy and run a command with **bold font**, but exchange it with your own carefully selected input (like username and password).

Important!! The comments in the right column will help you understand what is happening. Sometimes the comments are crucial to get the installation right.

4.1 *Ubuntu 12.04*

Step	Command	Comment
1.1	sudo -s	Enter your account password and become root. This installation document expects you to be root throughout the whole process.
1.2	aptitude update && sudo aptitude safe-upgrade	Will make your system up to date

4.2 *MySQL 5.5 and JRE 1.7*

Step	Command	Comment
2.1	aptitude install mysql-server-5.5	During installation you must enter a MySQL root password. **Remember this password!!** – you will need it in step 2.7 and 4.3. You can stop/start/restart with this command: service mysql start/stop/restart
2.2	pico /etc/mysql/my.cnf	This file is the essential MySQL configuration file.
2.3	bind-address = 0.0.0.0	**Optional!** 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).

2.4	max_allowed_packet = 32M	Should be at least 32M, to allow adding firmwares up until this size into the Fusion database
2.5	innodb_buffer_pool_size=1024M	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.
2.6	service mysql restart	Restart MySQL after changes
2.7	mysql -uroot -p**password**	Log on to MySQL with the MySQL root-password. Perform a simple SQL "select user from mysql.user;" to make sure MySQL is running fine.
2.7	aptitude install openjdk-7-jre-headless	Will install JRE 7
2.8	update-java-alternatives -s /usr/lib/jvm/java-1.7.0-openjdk-amd64	Make sure this newly installed Java version is the default on the system. Some error messages may be shown, but most likely you can ignore them.
2.4	java -version	Should return something like java version "1.7.0_09" OpenJDK Runtime Environment (IcedTea7 2.3.3) (7u9-2.3.3-0ubuntu1~12.04.1) OpenJDK 64-Bit Server VM (build 23.2-b09, mixed mode)" This can be tricky, if you have older/other java versions installed. You can try to uninstall the older versions.

4.3 Tomcat 7

Step	Command	Comment
3.1	aptitude install tomcat7	Install Tomcat 7 You can stop/start/restart with this command: service tomcat7 start/stop/restart
3.3	pico /var/lib/tomcat7/conf/catalina.properties Append the following: ,${catalina.base}/common,${catalina.base}/common/*.properties to the property "common.loader"	Edit the configuration file of tomcat to point to a directory where Fusion properties will be placed.
3.4	pico /etc/default/tomcat7	AUTHBIND=yes will make it possible for Tomcat to run on lower ports (80 and

	Make sure these settings are set: AUTHBIND=yes JAVA_OPTS="-Djava.awt.headless=true -Xmx768m -XX:+UseConcMarkSweepGC"	443). 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.
3.5	pico /etc/passwd Find the row containing "/usr/share/tomcat7". Change "/usr/share/tomcat7" to "/var/lib/tomcat7".	This changes Tomcat's home directory from /usr/share/tomcat7 to /var/lib/tomcat7.
3.6	chown tomcat7:tomcat7 /etc/ssl/certs/java/cacerts	Let Tomcat own the Java CA certificate file.
3.7	pico /var/lib/tomcat7/conf/server.xml Make sure the HTTP Connector is set to use port 80: <Connector port="80" protocol="HTTP/1.1" ….	Default setup of Tomcat is port 8080, we'll change it to 80 (default HTTP port).
3.8	pico /etc/init.d/tomcat7 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.
3.9	service tomcat7 restart	Check /var/lib/tomcat7/logs/catalina.out to make sure Tomcat7 starts without errors
3.10		If you have a firewall, open for TCP/80

4.4 Fusion 2013R1

Step	Command	Comment
4.1	mkdir /var/lib/tomcat7/deploytmp /var/lib/tomcat7/shell	Will create a couple of directories need for Fusion. We place everything in the default tomcat folder (~tomcat7), for easy access.
4.2	rm -rf /var/lib/tomcat7/webapps/ROOT	Remove the default ROOT webapp.
4.3	ln -s /var/lib/tomcat7/webapps/xapsweb /var/lib/tomcat7/webapps/ROOT	Set Fusion Web to be the default webapp.
4.4	Download Fusion release file (we assume the name is fusion-2013R1.zip)	Download/copy Fusion into your home folder.
4.5	unzip fusion-2013R1.zip	
4.6	unzip xapsdatabase-1.3.0/xapsdatabase.zip	
4.7	Install Fusion Database	Read and execute installation described in Fusion Database User Manual.pdf [8]

4.8	cp -v */xaps*.properties /var/lib/tomcat7/common	Run from home folder. Copies all Fusion properties files into correct Tomcat folder.
4.9	sed -i 's/xaps\/xaps/xaps\/**xapsdatabasepassword**/g' /var/lib/tomcat7/common/*.properties	During step 4.3 you created an Fusion database with a password. Make sure to use that in this command, to set the correct property in the Fusion configuration.
4.10	cp -v */xaps*.war /var/lib/tomcat7/webapps	Copy all Fusion WAR files into correct Tomcat folder. The WAR files will hot-deploy if Tomcat is already running.
4.11	cp -v */xaps*.jar /var/lib/tomcat7/shell	Copy Fusion Shell to /opt/xaps
4.12	mv /var/lib/tomcat7/common/*shell* /var/lib/tomcat7/shell	Move the property files needed for Fusion Shell to the correct folder. It should have the correct settings for database password (see 4.5)
4.13	chown -R tomcat7:tomcat7 /var/lib/tomcat7	Let Tomcat own its home/working directory.
4.14	service tomcat7 restart	Restart Tomcat to make sure all changes are taken into effect. Check logs under /var/lib/tomcat7/logs/catalina.out
4.15		Start a browser and locate the Fusion Web interface. Usually http://localhost Then log in using admin/xaps Then choose "Certificates" on the upper right menu, and add the two certificates that accompanies the release-zip.
4.16		If you have a firewall, you must now open UDP/9116 for syslog-traffic.
4.17	chmod g+w /var/lib/tomcat7/common /var/lib/tomcat7/deploytmp /var/lib/tomcat7/shell	Make sure all files in these folder keep the correct group
4.18	chmod g+s /var/lib/tomcat7/common /var/lib/tomcat7/deploytmp /var/lib/tomcat7/shell	Make sure all files in these folder keep the correct group

4.5 Optional steps – Tomcat on HTTPS

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 (http://www.startssl.com/).
5.1-alt1-2	unzip build_jks.zip	
5.1-alt1-3	chmod 755 *.sh	Make the scripts runnable
5.1-alt1-4	./build_jks_class1.sh	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 keytool -import). You may of course use the suggested certificate from StartSLL (5.1-alt1-1)
5.1-alt3	keytool -genkey -alias xaps -keyalg RSA -validity 10000 -keystore /var/lib/tomcat7/.keystore	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	keytool -list -keystore /var/lib/tomcat7/.keystore -storepass importkey	List the key aliases, you will need it in the next step.
5.3	pico /var/lib/tomcat7/conf/server.xml	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	service tomcat7 restart	Start a browser and locate the Fusion Web interface. Usually https://localhost
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:/#
```

```
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="443" 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" />
    -->
(Samma fråga igen? Samma svar.)

-UUU:**--F1   server.xml     57% L91    (nXML Valid)------------------------------
sudo update-java-alternatives -s /usr/lib/jvm/java-1.7.0-openjdk-amd64
```

Screenshot 7.2

5 Migration from 2012R1 to 2013R1

Important!! If you are running Fusion 2012R1, go to next chapter for migration instructions.

The goal is to install a standard Fusion Server (Default Setup), which requires installation of Ubuntu Server 12.04 LTS 64-bit, MySQL Server 5.5 (latest update), Tomcat 7 and JRE 1.7 (latest update). How to install Ubuntu 12.04 LTS 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.

Important!! In this installation document we use **bold font** to signal variable input. Do not copy and run a command with **bold font**, but exchange it with your own carefully selected input (like username and password).

Important!! The comments in the right column will help you understand what is happening. It's a good idea to know what is happening.

5.1 Ubuntu 12.04

Step	Command	Comment
1.1	sudo –s	Enter your account password and become root (your account must be in the "admin" group). This installation document expects you to be root.
1.2	/etc/init.d/jboss stop	Make sure you stop the service. From this point on, until the installation is complete, you will not have any provisioning service. You can still access Fusion database through the usage of Fusion Shell (going directly towards MySQL DB).
1.2	aptitude update && sudo aptitude safe-upgrade	Will make your system up to date
1.3	do-release-upgrade -c	Will upgrade from Ubuntu 10.04 to 12.04: Extra sshd on port 1022? (y) Lots of updates, can't abort after this: continue? (y) restart services without asking? (default, No) Services to restart? rsync exim4 cron atd (default, OK) Restart libpam services? exim4 cron atd (default, OK) Replace my.cnf? (default, N) Replace ntp.conf? (Y) Remove obsolete packages? (y) Restart? (y)

5.2 MySQL 5.5 and JRE 1.7

Step	Command	Comment
2.1	aptitude install mysql-server-5.5	During installation you must enter a MySQL root password. **Remember this password!!** – you will need it in step 4.3. You can stop/start/restart with this command: service mysql start/stop/restart
2.5	aptitude install openjdk-7-jre-headless	Will install JRE 7
2.6	update-java-alternatives -s /usr/lib/jvm/java-1.7.0-openjdk-amd64	Make sure this newly installed Java version is the default on the system. Some errors may occur, but most likely you can ignore them.
2.7	java -version	Should return something like java version "1.7.0_09" OpenJDK Runtime Environment (IcedTea7 2.3.3) (7u9-2.3.3-0ubuntu1~12.04.1) OpenJDK 64-Bit Server VM (build 23.2-b09, mixed mode)" This can be tricky, if you have older/other java versions installed. You can try to uninstall the older versions.

5.3 Tomcat 7

Step	Command	Comment
3.1	aptitude install tomcat7	Install Tomcat 7 You can stop/start/restart with this command: service tomcat7 start/stop/restart
3.2	pico /etc/passwd	Change the line containing "/usr/share/tomcat7" to "/var/lib/tomcat7".
3.3	pico /var/lib/tomcat7/conf/catalina.properties Append the following: ,${catalina.base}/common,${catalina.base}/common/*.properties	Edit the configuration file of tomcat to point to a directory where Fusion properties will be placed.

		to the property "common.loader"	
3.4	pico /etc/default/tomcat7 Make sure these settings are set: AUTHBIND=yes JAVA_OPTS="-Djava.awt.headless=true -Xmx768m -XX:+UseConcMarkSweepGC"	AUTHBIND=yes will make it possible for Tomcat to run on lower ports (80 and 443). JAVA_OPTS is only changed slightly, to increase maximum memory usage from 128 megabyte till 768 megabyte. This should be sufficient for 10-20K devices.
	pico /etc/passwd Find the row containing "/usr/share/tomcat7". Change "/usr/share/tomcat7" to "/var/lib/tomcat7/".	This changes Tomcat's home directory from /usr/share/tomcat7 to /var/lib/tomcat7.
	chown tomcat7:tomcat7 /etc/ssl/certs/java/cacerts	Let Tomcat own the Java CA certificate file.
3.5	pico /var/lib/tomcat7/conf/server.xml Make sure the HTTP Connector is set to use port 80: <Connector port="80" protocol="HTTP/1.1" ….	Default setup of Tomcat is port 8080, we'll change it to 80 (default HTTP port).
3.6		If you have a firewall, open for TCP/80

5.4 *Fusion*

Step	Command	Comment
4.1	mkdir /var/lib/tomcat7/deploytmp /var/lib/tomcat7/shell /var/lib/tomcat7/shell	Will create a couple of directories need for Fusion. We place everything in the default tomcat folder (~tomcat7), for easy access.
4.2	chmod g+w /var/lib/tomcat7/common /var/lib/tomcat7/deploytmp /var/lib/tomcat7/shell	Make sure all files in these folder keep the correct group
4.3	chmod g+s /var/lib/tomcat7/common /var/lib/tomcat7/deploytmp /var/lib/tomcat7/shell	Make sure all files in these folder keep the correct group
	rm -rf /var/lib/tomcat7/webapps/ROOT	Remove the default ROOT webapp.
	ln -s /var/lib/tomcat7/webapps/xapsweb /var/lib/tomcat7/webapps/ROOT	Set Fusion Web to be the default webapp.
4.1	Download Fusion release file (we assume the name is fusion-2013R1.zip)	Download/copy Fusion into your home folder.
4.2	unzip fusion-2013R1.zip	
4.3	unzip xapsdatabase-1.2.2/xapsdatabase.zip	

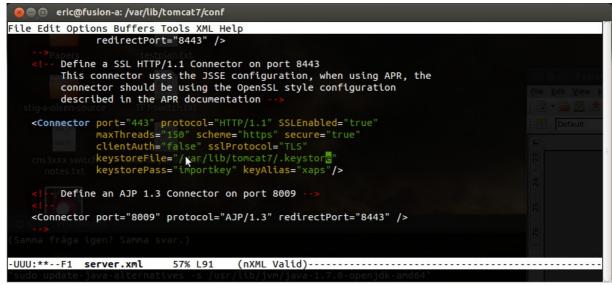| 4.4	Install Fusion Database	Read and execute installation described in Fusion Database User Manual.pdf.
4.5	cp -v */xaps*.properties /var/lib/tomcat7/common	Run from home folder. Copies all Fusion properties files into correct Tomcat folder.
4.6	sed -i 's/xaps\/xaps/xaps\/**xapsdatabasepassword**/g' /var/lib/tomcat/common/*.properties	During step 4.3 you created an Fusion database with a password. Make sure to use that in this command, to set the correct property in the Fusion configuration.
4.7	cp -v */xaps*.war /var/lib/tomcat7/webapps	Copy all Fusion WAR files into correct Tomcat folder. The WAR files will hot-deploy if Tomcat is already running.
4.11	cp -v */xaps*.jar /var/lib/tomcat7/shell	Copy Fusion Shell to /opt/xaps
4.12	mv /var/lib/tomcat7/common/*shell* /var/lib/tomcat7/shell	Move the property files needed for Fusion Shell to the correct folder. It should have the correct settings for database password (see 4.5)
	chown -R tomcat7:tomcat7 /var/lib/tomcat7	Let Tomcat own its home/working directory.
4.13		Start a browser and locate the Fusion Web interface. Usually http://localhost Then log in using admin/xaps Then choose "Certificates" on the upper right menu, and add the two certificates that accompanies the release-zip.
4.14		If you have a firewall, you must now open UDP/9116 for syslog-traffic.
4.9	wget http://localhost	Should return some Fusion HTML within 60 seconds after deployment of WAR files.

5.5 Optional steps – Tomcat on HTTPS

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
7.1-alt1-1		Buy (class2) or get for free (class1) a certificate from StartSSL (http://www.startssl.com/).
7.1-alt1-2	unzip build_jks.zip	
7.1-alt1-3	chmod 755 *.sh	Make the scripts runnable
7.1-alt1-4	./build_jks_class1.sh	The command will show you help text and how to run the script. If you bought class2 certificates, run the other script.
7.1-alt2		Buy your own certificate and import it into a java keystore (using keytool -import)
7.1-alt3	keytool -genkey -alias xaps -keyalg RSA -validity 10000 -keystore /var/lib/tomcat7/.keystore	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.
7.1-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.
	keytool -list -keystore /var/lib/tomcat7/.keystore -storepass importkey	List the key aliases, you will need it in the next step.
7.2	pico /var/lib/tomcat7/conf/server.xml	Edit the file to match screenshot 7.2, using the alias acquired in the previous step. If you allow low ports you can change 8443 to 443. NB! This change will still allow traffic on HTTP, to avoid that uncomment the connector for port 8080 (or 80)
7.4	service tomcat7 restart	
7.6		If you have a firewall, open for TCP/443 or TCP/8443 (depending upon your choices)

Screenshot 7.2

5.6 Optional steps – MySQL tweakings

Step	Command	Comment
8.1	pico /etc/mysql/my.cnf	This file is the essential MySQL configuration file.
8.2	bind-address = 0.0.0.0	Replace or add this line to be able to access MySQL from remote hosts. This is very valuable for Ping Communication in an extended test/pilot period. Also open TCP/3306 in firewall.
8.3	max_allowed_packet = 16M	Should be at least 16M, to allow adding firmwares up until this size into the Fusion database
8.4	innodb_buffer_pool_size=1024M	You should set this pool to as much memory as you have available (free memory), because the syslog table of Fusion can be rather big and therefore needs large indexes (which should fit in memory for maximum performance)
8.5	service mysql restart	Restart MySQL after changes

5.7 Upgrade from xAPS 2011R1 to Fusion 2012R1

Word of caution: Before starting on this operation, a healthy number of backups should be taken. Read Fusion Database User Manual, on how to perform database backups, and backup other files as you go along.

The following steps must be performed when updating from xAPS 2011R1 to Fusion 2012R1:

Step	Command	Comment
9.1	Download Fusion release file (we assume the name is xaps-2012R1.zip)	Download/copy Fusion into your home folder.
9.2	unzip xaps-2012R1.zip	

9.3	Locate these propertyfiles of 2012R1 and update the old property files (from 2011R1) so that the properties have the same order:
	    - xaps-spp.properties:
	    Telnet config (line 102-115) has been added
	    - xaps-tr069.properties
	    Shell-daemon (line 34-43) has been added
	    - xaps-web.properties
	    Other configuration (line 36-EOF) has been replaced.
	    - xaps-syslog-logs.properties:
	    Summary log (line 88-90 and line 118-125) has been added
	    - xaps-stun.properties:
	    Kick-properties (line 17-30) has been added
	    - xaps-stun-logs.properties:
	    Kick-appendere (line 130-EOF) has been removed and 2 new appenders have been added
		Make sure the propertyfiles are updated to 2012R1. Only a few propertyfiles are affected.

9.4	unzip xapsdatabase-1.2.2/xapsdatabase.zip	
9.5	cd xapsdatabase-1.2.2/mysql	
9.6	TIMEOUT - read comment!	Check the indexes on the syslog table in MySQL. Read the comment in the sql-file upgradesyslogindexes.sql and make changes accordingly (if necessary - otherwise run as-is). When/if you run the sql-script, it will take a long time (some hours). You may decide to stop JBoss first, then it will go a little faster (since no syslog messages will be written to Fusion DB), but the downtime can be several hours.
9.7	mysql -u**user** -p**password** xaps < upgradesyslogindexes.sql	The syslog indexes are updated so the Fusion DB perform at it's best.
9.8	cp xapsshell-2.0.7/xapsshell.jar /opt/xaps	Will overwrite the old xAPS Shell with the new Fusion Shell. The property files may remain the same.
9.9	cp xapsshell-2.0.7/scripts/ /opt/xaps	Will copy some important scripts to Fusion Shell directory
9.10	cd /opt/xaps	Change working directory
9.11	java -jar xapsshell.jar Choose correct database, usually "1"	Connect to Fusion DB
9.12	/etc/init.d/jboss stop	You may postpone this until after 9.14, but then you must make sure no one can make changes to jobs/groups in Fusion DB. From this moment until upgrade is finished, it may take 15 minutes,

		which will be the downtime on the system.
9.13	call scripts/2012R1/eb11To12.xss	Run command inside Fusion Shell. This command will export a number of files which should contain all information about jobs/groups in the Fusion DB. It is important that you make sure the files look reasonable, since the tables will be deleted later.
9.14	exit	Exit Fusion Shell.
9.15	mysql -u**user** -p**password** xaps < upgrade2012R1.sql	A few database tables are changed/updated.
9.16	java -jar /opt/xaps/xapsshell.jar	Connect to Fusion DB
9.17	call scripts/2012R1/ia11To12.xss	Run command inside Fusion Shell. This command will import the files you previously exported. If all goes well, all data are imported.
9.18	exit	Exit Fusion Shell
4.19	cp -v */xaps*.ear /opt/jboss/server/default/deploy	Copy all Fusion EAR files into correct JBoss folder. The EAR files will hot-deploy if JBoss is already running.
4.20	sed -i 's/com.owera.xaps.intf.util.SyslogAppender/com.owera.xaps.dbi.util.SyslogAppender/g' /opt/jboss/server/default/conf/*.properties	Changes a classpath to SyslogAppender in all property files
4.20	/opt/xapsperm.sh	Run this script (made in 1.8) to set correct permission on everything you copied into JBoss
9.21	/etc/init.d/jboss start	Check /opt/jboss/server/default/log/server.log to see if everything went well during the startup

5.8 xapsperm.sh script

```
#!/bin/sh

chown -R jboss:jboss /opt/jboss/
find /opt/jboss/ -type f -print0 | xargs -0 chmod g+w 2>/dev/null
find /opt/jboss/ -type d -print0 | xargs -0 chmod g+wX 2>/dev/null
find /opt/jboss/ -type f -iname "*xaps*" -print0 | xargs -0 chmod 664 2>/dev/null
find /opt/jboss/ -type d -iname "*xaps*" -print0 | xargs -0 chmod 775 2>/dev/null
chown -R xaps:xaps /opt/xaps/
find /opt/xaps/ -type f -iname "*" -print0 | xargs -0 chmod 664 2>/dev/null
```

6 Technical reference

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 5, you'll end up with a Default Setup (DS), and for this setup we'll provide exact information.

6.1 Property files

In DS property files are found in **/opt/jboss/server/default/conf**. You may choose to use other servers than default in JBoss, and then the directory will change accordingly. 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.

6.2 Log files

6.2.1   Fusion logs

Log files are found in the working folder of the Application Server (usually JBoss). For DS you'll find logs in **/opt/jboss/bin** and **/opt/jboss/bin/backup-logs**. The logs are named following this convention: xaps-<modulename>(-<optionalname>).log. Usually every module has a default/regular log: xaps-<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.

6.2.2   JBoss log

Another important log file in DS is /opt/jboss/server/default/logs/server.log. This contains important information about the running of JBoss. Always check this log when problems regarding loading of modules occur or stopping and starting JBoss. When starting up you should see something like this:

6.3 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 5.5) 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 5.5). 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. 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
DB	3306	TCP	Allows direct access to MySQL database. 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.
OPP	9117	TCP	An obsolete module of Fusion, may be necessary to open for older installations
JBoss	8787	TCP/UDP	To allow remote debugging of Fusion/JBoss in case of severe problems.