



Fusion Release Changes

Version 2013R1

© 2007-2013 Ping Communication AS

Table of Contents

Fusion Release Changes.....	1
Version 2013R1.....	1
1 Document Introduction.....	3
1.1 Document Purpose.....	3
1.2 Document Audience.....	3
1.3 Document History.....	3
1.4 Acronyms and Abbreviations.....	3
2 2013R1.....	5
2.1 Permission system.....	5
2.2 File store.....	5
2.3 Script Execution Web.....	5
2.4 Script Execution Overall.....	5
2.5 Syslog Event.....	6
2.6 Trigger/Alarm.....	6
2.7 Heartbeat Detection.....	7
2.8 Monitor.....	7
2.9 Reports	7
2.10 Cloud.....	7
2.11 TR-069 Data model enumeration generation.....	7
2.12 Extraction.....	8
2.13 TR069 Test Case system.....	8
2.14 TR-069 Server.....	8
2.15 Misc.....	8
3 2012R1	9
3.1 All/Many modules.....	9
3.1.1 Search capability.....	9
3.1.2 Jobs.....	9
3.1.3 Performance.....	9
3.1.4 Bugfix.....	9
3.2 TR069	9
3.3 STUN.....	10
3.4 Web.....	10
3.4.1 Reports.....	10
3.4.2 Navigation.....	10
3.4.3 Search.....	10
3.4.4 Various.....	10
3.5 Shell.....	10
3.6 Monitor.....	11
3.7 Core.....	11
3.8 SPP.....	11
3.9 Syslog.....	11

1 Document Introduction

1.1 Document Purpose

The purpose of this document is to list the changes in the system compared to the previous release.

1.2 Document Audience

Fusion Operators and Administrators

1.3 Document History

Version	Editor	Date	Changes
2012R1	M. Simonsen	19-Dec-11	Initial version
2013R1	M. Simonsen	02-Mar-13	Updated

1.4 Acronyms and Abbreviations

Acronym	Explanation
ACS	Auto Configuration Server.
APS	Automatic Provisioning System.
Fusion	Ping Communication's eXtensible APS with advanced features such as Service Windows, Job Control and Smart Groups.
Fusion Module	The Fusion consists of several independently running Modules. The Modules may run on separate hosts in the network.
North Side	Common term describing the part of the Fusion which includes all Fusion Modules that provide management interfaces.
South Side	Common term describing the part of the Fusion which includes all Fusion Modules that provide CPE communication protocols.
Core	Common term describing the part of the Fusion which includes all Fusion Modules that provide neither management interfaces nor CPE communication protocols.
CPE	Customer Premises Equipment. Used in this document to refer to a single physical device. Same as the term "Device".
Parameter	Each individual configuration setting is represented in the Fusion Data Model as a Parameter. A Parameter consists of a name and usually (but not always) a value.
Unit	A dataset in the Fusion database consisting of Parameter values relating to a single CPE. This dataset may extend beyond the Parameter values actually sent to the CPE, as some Parameter values may only be useful or needed by the Fusion itself. Also, the dataset may represent only a subset of all the configurable settings in the CPE. For these reasons, it is important to distinguish the term "Unit" from the terms "CPE" and "Device".
Profile	Dataset stored in the Fusion containing Parameter values shared by multiple Units of the same Unit Type. A Unit is always

	assigned to a single profile. Multiple Profiles may be created for a Unit Type.
Unit Type	Units that represent CPEs of the same model share a common definition of that CPE model named Unit Type. The Unit Type definition is a list of Parameter names only, as the Unit Type never contains any Parameter values (values are stored in the Unit and/or Profile).
Group	A set of matching criteria used to search for Units. Commonly referred to as Smart Group.
Job	Automates and controls changes to Units within a Group. Partitions the changes over time according to rules to limit network load.
Job Chain	Multiple Jobs being automatically executed in a designated sequence.
Periodic Mode	Provisioning Mode where the Fusion automatically configures all CPEs based on their combined Unit and Profile parameter values.
Inspection Mode	Provisioning Mode where an Fusion Operator manually inspects and configures a single CPE through the Fusion Web Interface.
Staging	Fusion functionality used for optimizations in manufacturing, logistics and time-to-market for CPEs.
TR-069	Industry standard provisioning protocol used by the Fusion to read and write configurations from and to the CPEs, in addition to handle upgrades.
SPP	Simple Protocol Provisioning, covers HTTP(S)/TFTP

2 2013R1

The name change introduced in 2012R1 (from "xAPS" to "Fusion") is still not completed entirely. Thus, you will still find references to "xaps" from time to time.

The main goal of this release has been to create Trigger/Alarms and improve the Script Execution capabilities. All of this points to a direction where Fusion is more flexible to act upon various events and automatically make attempts on a solution. In one word, Fusion is heading towards "Self Healing" - in order to minimize the ill effects of CPEs with problematic software and to lower the support cost.

2.1 *Permission system*

The permission system has been given a makeover, so the following things are not possible:

- A non-admin user can create other users (with same or less privileges).
- All users may change their own passwords
- User-login is now possible to require in Shell (but not mandatory)
- Permission handling is the same for Web, Web Services and Shell (all North-side modules)

2.2 *File store*

The file store in Fusion now handles all kinds of files. This opens for a more flexible usage of the file store: It can contain firmware-binaries, TR069 scripts, Fusion Shell scripts, Unit Id lists, Telnet scripts and any other file format necessary. For those file types which are not binary, you may also edit the files directly. This makes it possible to change scripts or unit id lists in the Web without having to export/import the files (as in previous edition of Fusion). The file store also retains the owner of the file, which is important for script execution with the correct permissions.

2.3 *Script Execution Web*

Fusion Shell scripts can be uploaded into file store and be run directly in Fusion Web. Thus Web offers a Shell interface of sorts, since it's possible to both edit a script and run it. The main driver behind this is twofold:

1. Enable non-shell users to make and run/test scripts
2. Making it easy to build wizards/scripts for certain repetitive tasks

2.4 *Script Execution Overall*

Several modules in Fusion can trigger a script execution:

- Job as part of a provisioning, thus making provisioning more flexible

- Syslog Event which happens if certain messages arrives, which can enable Fusion to act upon certain events
- Trigger Release which happens if a set of Syslog Events occur in a specific time period (more on Trigger Release later)
- Web can select a script for execution

This capability greatly enhances the flexibility of Fusion, since a script is much more flexible than in-built standard behaviour. It is likely that future use cases for Fusion will only find a solution using scripts. All the scripts are executed in a Fusion Shell daemon inside the Core module – thus logs can be found in fusion-core.log

2.5 Syslog Event

The Syslog Event capability has grown in this release:

- Possible to execute a script for a particular event
- Possible to match with a group, so that only units within a certain group can match a syslog event.
- Necessary to create a Syslog Event to handle Trigger/Alarms and Heartbeat Detection

Previous release had a slightly more advanced duplicate feature, which has been simplified in this release. Thus, a Syslog Event with duplicate policy will ignore duplicates (matching both Unit Id and content of Syslog message) within the same hour of the first match.

2.6 Trigger/Alarm

Trigger is a totally new concept in Fusion: It is configured using a Syslog Event and some limits to tell when to release the Trigger. An example could be that if Syslog Event 1001 has been recorded 10 times within the last 15 minutes on 10 different units, then the Trigger will release. A trigger release can in turn trigger a mail going out to the Operator of Fusion (or someone else). Another possibility is that a Trigger release can run a script to perform some actions. The great thing about this, is that the Trigger script will receive as input a file containing all the Units that were the cause of the Trigger release. This makes it possible to perform an action on those selected few. An example of such an action could be to reboot the devices in question.

Furthermore, Triggers can be built together into Composite Triggers, so that a Trigger release is not based on one single Syslog Events, but rather a set of events. In addition a Trigger may or may not send a mail, and it may or may not be an alarm – it could also be a weekly report.

All together, Trigger/Alarm offers new capacity for “Self Healing” based on the syslog messages that is picked up from the devices. To further improve the situation, all provisioning modules now make a standard syslog message upon provisioning. This message can be used to make certain actions based on the provisioning outcome.

2.7 *Heartbeat Detection*

One problem which is not solved with Syslog Event and Triggers directly is if a service in the device stops. One example could be if a SIP service on an ATA fails, but there are many other possible services that could exist in complex and modern CPEs. Such a situation can only be detected through the absence of what we call "Heartbeat". You may define such a Heartbeat in Fusion and specify a certain timeout period. If the Heartbeat has not been detected in this period, Fusion will then insert a syslog message into the system telling that a Heartbeat is missing. By doing so, you may specify a Syslog Event for this particular Heartbeat missing message. And on top of that you may build a trigger to act upon those devices where a service has stopped.

Using Syslog Events, Triggers/Alarms and Heartbeats, any syslog emitting device may be monitored and problems may be remedied. Above all, this allows for unprecedented flexibility, only limited by the device capabilities and logging behaviour.

2.8 *Monitor*

The Monitor module is totally rewritten. Some of the previous capabilities are removed since it was considered unnecessary. However, the Monitor page is also improved to always show the version number of all modules. In addition, the configuration is of the server is much, much easier. You may decide to hook the Fusion monitoring into your own surveillance center. This is done by monitoring just one URL: <http://yourhost/xapsmonitor/ok>. If this URL returns "XAPSOK <version-number>", then everything is fine with Fusion, otherwise it will return the error status of the modules involved.

2.9 *Reports*

The configuration of reports (in xaps-core.properties) is simplified. Some old reports are removed (they were never used). Also some reports were unnecessary slow in the past, but this has been remedied.

2.10 *Cloud*

Fusion is now possible to run, and is actually running, as a cloud service (available on <https://pingcom.xaps-hosting.net>). In the processing of "cloudification" Fusion has improved many critical paths in the system, to improve the overall performance of the system.

2.11 *TR-069 Data model enumeration generation*

TR-069 comes with a set of published Data models, describing each parameter with regards to name, type, usage and possibly also an enumerated set of allowed values. This set of values can be useful to put into the Unit Type, and Fusion now offers commands to easily extract the enumerated values directly from the TR069 data model.

However, there is no guarantee that these enumerated values are in fact supported by the device, so this feature should be used with caution.

2.12 *Extraction*

2013R1 comes with an extraction feature (available on the Unit Configuration page in Web and in the Unit context in Shell). The extraction feature will return all parameters from the device and store it in the Fusion database (in a separate storage from the regular provisioned data). This can be an valuable tool to investigate a "troublesome" device.

2.13 *TR069 Test Case system*

All devices supporting TR-069 will need to be tested before going into production provisioning. Fusion now offers a way to test devices with regards to a number of input/output.

2.14 *TR-069 Server*

The server now handles a few more quirks:

- prettyprint: allows logging of XML from a device to be formatted nicely
- xmlcharfilter: allows server to filter out illegal XML characters before parsing the input from the device
- ignorevendorconfigfile: the server has implented a new way to handle TR-069 scripts/configuration files to be download. However, not all devices may handle the query for VendorConfigFile. object with all child nodes, thus it's possible to turn off this feature

2.15 *Misc*

There's been a goodly number of bug fixes and performance enhancements in this release. It's of course not too interesting for the average customer, but there has been some significant clean-up in the code. This is helpful for future developments, and ensure a lean and easy-to-adapt system.

3 2012R1

The overall change is a name change from xAPS to Fusion. Other changes will be listed per module.

3.1 *All/Many modules*

3.1.1 Search capability

Now supports more advanced search, both for group search, unit search and syslog search. The searches can now (with some minor differences):

- specify the same parameter several times
- specify a range of operators: >, >=, =, <=, <, <>
- specify certain control characters: *, %, _, \$, ^, |, !
- specify a type of value, as text or numbers
- search for NULL or not-NULL values

3.1.2 Jobs

New types of jobs added:

- Telnet: Can make scripts and run on a group of devices. Can also retrieve data from running the telnet scripts and store in Fusion DB.
- Shell: Can run shell-script upon provisioning a device. Dramatically increases the logic that can be applied in provisioning.
- Reset: Can perform FactoryReset on TR-069 devices (that supports this method)

The repeatable job feature has been changed to do fixed intervals between those jobs. Also, a number of bugs connected to the calculation of the next periodic inform interval has been fixed.

3.1.3 Performance

- Removed some very CPU consuming and unnecessary forced garbage clean-up, which caused massive usage of CPU every other second.
- Improved caching mechanism and inter-modular change messaging in DBI

3.1.4 Bugfix

A large number of bugs has been fixed throughout all modules. All in all the changes made in this release are fewer than in the previous, which might give the bonus of fewer undetected bugs.

3.2 *TR069*

UnitId can now be any kind of string, not only the standard OUI-Serialnumber or OUI-ProductClass-Serialnumber. This is due to the fact the quite a lot of devices do not adhere strictly to the TR-069 standard.

Improved logging

Added support for FactoryReset method

More robust with devices not sending any data at all

3.3 *STUN*

The kick functionality both using TR-111 (over UDP) and to public IP work fine.
Handles both "single kick" and "group kick" (using Jobs).

3.4 *Web*

3.4.1 Reports

Possible to make Syslog reports based on group selection
Possible to filter on Software-version in all Syslog/TR-reports
Filtering on report-results in improved

3.4.2 Navigation

Changed menu system, should be more intuitive
Context menu navigation added, easier to navigate back and forth among the various contexts.

3.4.3 Search

Possible to search both on Global level (across all unittypes) and within context (context search) and on the search page.
Improved search capabilities (mentioned previously)

3.4.4 Various

Can now add various types of file (SOFTWARE, CONFIG, SCRIPT)
Table sorting is available on a range of pages
Documentation is improved a lot

3.5 *Shell*

- Total makeover of interface: Changed context-switching, command prompt and variable referencing.
- Supports control statements like if, else, elsif, while, continue and break
- Supports variables, can specify both variable by value and by reference in call-commands. Can also specify totally new variable names upon start-up of shell.
- Many new commands, like kick (ask device to provision) and unit (quick unit context switch), setvar (make a variable), cc (change context), syslog (syslog search)
- Commands support options, like -c (list context), -u (use context), -a (list all) and -o (order)
- Output-commands can be piped
- Massive changes of unittype parameters no longer causes excessive load on the other modules.
- Command history available
- Help system is greatly improved
- Output listing is ordered strictly column-wise, much easier to read.

3.6 *Monitor*

Improved monitoring for failures in DBI

Improved monitoring of TR-069, will detect connections which never finishes.

3.7 *Core*

Only library changes, minor bugfixes and logging changes

3.8 *SPP*

TFTP port can now be read from the properties file

Added Telnet provisioning. Can be triggered using a Job.

3.9 *Syslog*

Improved logging (syslog-summary) to get better overview of situation