Fusion Use Cases and Training

Version 2012R1

Table of Contents

1 Document Introduction

1.1 Document Purpose

The purpose of the document is to explain how to use Fusion to solve various cases. The document can also be used as a way of Fusion Administrator/Operator training.

1.2 Document Audience

The readers will be Fusion Administrators and system operators.

1.3 Document History

Version	Editor	Date	Changes
2011R1-SP1	Morten Simonsen	04-Feb-11	Revised edition
2012R1	Morten Simonsen	21-Nov-11	Updated to next release.

1.4 Acronyms and Abbreviations

Acronym	Explanation
APS	Automatic Provisioning System.
Fusion	eXtensible APS with advanced features such as Service Windows, Job Control and Smart Groups.
CPE	Customer Premises Equipment
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.
OPP	Owera (now Ping Communication) Provisioning Protocol. Highly compact, efficient and extensible provisioning protocol developed by Owera. Handles configuration, upgrades, logging and connection forwarding to arbitrary CPE services (through NAT), all over an encrypted and authenticated connection.
JRE	Java Runtime Environment

1.5 References

Document
[1] Fusion Web User Manual
[2] Fusion Shell User Manual
[3] Fusion JCS 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

[10] Fusion OPP Server User Manual

2 Introduction

Fusion offers three north-bound interfaces: Web, Shell and Web Services. Only the first two are of interest in this document. If you are running on Fusion Hosting Server, the Shell interface is not available.

When you read this document, you should have Shell/Web/TR-069 manuals available, since there will be references to those documents.

3 Let's connect

The first task after a complete installation of Fusion, is to get started with provisioning from Fusion. To do so we need a Device (the terms "CPE" and "Unit" are also used) supporting TR-069 and the TR-069 server must be running.

Step	Module/Entity	Operation
1	TR-069	Change the property "discovery.mode" to "true" in xaps-tr069.properties. This will enable the server to automatically accept and store necessary data from a connecting device. Note that discovery mode is not recommended for production, only for testing.
2	Device	Log in to the web interface (or any other interface) and locate the provisioning settings. These settings are usually found under the "Advanced" section, another common name is "Remote management". Change the URL to Fusion TR-069 Server. This URL is also mentioned in Fusion Installation.doc. This change **should** trigger the device to connect to the server. To be absolutely sure, restart the device.
3	TR-069	Open the URL mentioned in the previous step in a browser. You should see "some activity" (explained in TR-069 manual). If all is well you will see a complete session in the first table.
4	TR-069	Inspect xaps-tr069.log and xaps-tr069-event.log (log files are found in working directory of the application server: /opt/jboss/bin)
5	Web	Log in and search for the unit using the MAC address of the device. You should see that some parameters are read from the device. You're now ready to start provisioning.

3.1 Alternative I

In case you will not set "discovery.mode" to "true", as in step 1, you must do it all manually. You must then create a Unit Type, setting it to a TR-069 protocol. The other settings for Unit Type creation doesn't matter. Then create a Profile as part of this new Unit Type. The Profile is just a name/container for Units. Lastly create a Unit. The Unit id must be the same as the unit id specified in the Device (ofter called "ACS-Username" in the Web Gui of the device). Furthermore you must create a Unit Parameter Value for "System.X_OWERA-COM.Secret" and specify the ACS-Password found in the device. Then continue from step 2 above.

3.2 Alternative II

Construct a SOAP-client which can communicate with Fusion WebService interface. Complete the steps in alternative I using the SOAP-client. The client must be provided with information about Unit Ids and the ACS-passwords.

3.3 Alternative III

If you buy Pingcom device, you may set up an agreement for using Fusion Staging Server. In this case, the units will be automatically registered on your Fusion server once you buy the Pingcom devices.

4 Simple provisioning

Simple provisioning is about how to set a parameter in Fusion and get it provisioned to the device as soon as possible.

Step	Module/Entity	Operation
1	Web	Look up the Unit you want to provision. Clik on the device and locate the "Unit Configuration" page. Set a parameter value (you need to click on the create-checkbox). Make sure this parameter has the RW flag set. Press the "Update parameters" button.
2	Device	Restart the device or wait until the device will provision again. Default provisioning interval is usually 24h.
3	Device	Log into the web interface and inspect the change of the parameter.
4	Web	Check the Unit dashboard to confirm that the device has indeed been in contact with the server.

4.1 Connect to the device from the server

The device can be reached from the server and a provisioning can be initiated immediately (the server performs a "kick", to kick-start the normal provisioning cycle), making a restart of the device unnecessary (step 2). Depending upon the connection status of the device, parameters have to be read/written from/to the device

4.1.1 Public IP

If the device is on public IP (directly connect to the Internet) the following parameters must be read:

- InternetGatewayDevice.ManagementServer.ConnectionRequestURL

A suggestion is to specify the Unit Type Flag to include "AR" (= Always Read). If this parameter is known, the server will try to contact the device using this URL. The device may also require password/username to allow a connection.

- InternetGatewayDevice.ManagementServer.ConnectionRequestPassword
- InternetGatewayDevice.ManagementServer.ConnectionRequestUsername

These parameters are set from the server.

4.1.2 Local IP (behind a NAT-Gatway)

If the device is on local IP (behind a NAT-Gateway), the following parameter must be read:

- InternetGatewayDevice.ManagementServer.UDPConnectionRequestAddress

If you cannot find this parameter in the device, it's most likely because it doesn't support TR-111, part II. If you find it, you must also specify a host of other parameters, to point to the STUN-server.

- InternetGatewayDevice.ManagementServer.STUNEnable
- InternetGatewayDevice.ManagementServer.STUNServerAddress
- InternetGatewayDevice.ManagementServer.STUNServerPort

These parameters must point to the Fusion STUN-server. Make sure that the STUN-server is setup correctly and was able to bind to the server-port.

4.2 Alternative I (Inspection Mode)

The idea of Inspection mode is to 'inspect' or 'debug' the device. The usual scenario is that only a small subset of parameters are provisioned. However, you may specify the Unit Type Parameter Flag 'I' (=Inspection) on those parameters that you want to inspect in special cases. The good thing is that you do not have to provision more parameters than necessary (which lowers the performance) , but can still have access to interesting data from the device upon Inspection.

When the connect parameters are set up properly, we can try a provisioning without using restart. Go to Unit Configuration of the device (in Web) and locate and set the radio button for "Inspection Mode". In this mode, the device will connect to the server. You may now write changes to the device just by changing parameters and then press a button for "Write changes". You may also inspect changes on the device (for parameters with the R(ead) only and I(nspection) flag set) - just press the button for "Read from device".

4.3 Alternative II (Kick)

The idea of Kick is simply to initiate a regular provisioning, without the rigour of Inspection Mode. Locate Unit Configuration in the Web, and find the radio button marked "Kick". The device should connect to the server within 30 seconds.

5 Advanced provisioning

Advanced provisioning is about Groups and Jobs. The idea is to control precisely which devices are changed, and to control the progress of that change.

5.1 Group

A group is essentially a search. A group can then be used to define a set of devices. This set may change every minute, depending upon the changes in Fusion and the parameters stored every minute. What we want to do is to make a group that defines one single unit (for the purpose of demonstration).

Step	Module/Entity	Operation
1	Web	Make a group, choose default values.
2	Web	Make a group parameter. This parameter must uniquely identify one single device. (ex: SerialNumber/MAC/IP/Seceret)
3	Web	Check current size of the group, it should be 1. Also click on the link "Search for units that matches" to confirm that it is indeed the device you had in mind.

5.2 Job

A Job tells what to do with a certain Group. A Job can also be assigned certain stop rules to control the progress of the change.

Step	Module/Entity	Operation
1	Web	Make a job. Select the group you just made. Select "Any software version" Set job type to CONFIG.
2	Web	Change job by adding a stop rule. The rule must be designed to stop the job if any failure occurs.
3	Web	Make a job parameter. Choose an RW-parameter of no particular interest (perhaps "ProvisioningCode") and set this parameter to a value of your choice.
4	Web	Start the job
5	Device	Restart the device (or kick device from server, see 4.3)
6	Web	Check job status and job counters.

5.3 Alternative I

Instead of using a parameter like SerialNumber or Secret to uniquely identify a unit in the Group chapter, you can create your own special parameter. To do so, simply create a new Unit Type Parameter with the name

- System.X_YOURCOMP-COM.ASpecialParameter

Then specify a unique value for this parameter on the unit. Lastly, change the Group parameter to match this parameter and the unit value.

This shows how you can customize parameter to fit your set of groups.

5.4 Exercises

1. Add three dummy softwares to the system, going from version 1 to 3. Then set up Jobs to ensure that all Units within this Unit Type is upgraded from v1 to v2 to v3. You don't have to start the job.

2. Change the previously added jobs to stop if one Unit does not respond 100 seconds after an upgrade.

3. Change the order of the job execution, so that all device are downgraded from v3 to v2 to v1, without deleting the jobs or creating new jobs.

4. Make a repeatable job that runs once every week. This job should set the parameter "System.X_OWERA-COM.Debug" to 0.

6 Syslog - mostly if Pingcom devices are used

Syslog is a standard within the Unix/Linux world, and Fusion takes advantage of this concept. Both the devices and the server modules can be set up to log into the Fusion Syslog server. We can use syslog to investigate/debug/analyze our deployment.

6.1 See syslog from device & server

Step	Module/Entity	Operation
1	Device	Log in to the web interface (or any other interface) and locate the syslog settings. Make sure you specify the correct IP-address of the Fusion Syslog server and set the port to 9116 (the default syslog port in Fusion). If possible also specify log level to "debug".
2	Syslog	Open the propertyfile "xaps-syslog-logs.properties". Locate the setting "log.Messages" and set it to "debug, MESSAGES". Wait min. 30 sec. The run a command to tail the log file "xaps-syslog-messages.log" (tail -f). Restart the device to trigger some messages, you should see some activity in the console.
3	Syslog	The message may include MAC-address as part of the message. If it does, update (if necessary) the propertyfile xaps-syslog.properties with "mac-regex-pattern.<idx>" properties. This will ensure that the syslog messages will be connected to the correct device, since MAC information is available in the Fusion database. If no MAC is available, the server will try to connect the syslog messages with the units in the database, based on IP address match.
4	Web	On the Unit Configuration page, click on the link "Last 100 entries from syslog". You should now see the syslog messages from the device.
5	TR-069	In the propertyfile "xaps-tr069-logs.properties" you may locate the setting: "com.owera". Set it to "debug, TR069, SYSLOG". You should now, after a maximum of 30 seconds delay, see activity from the TR-069 Server in the syslog page of Web. You might need to restart the device to see some activity. The messages will not be shown in the tail of xaps-syslog-messages.log, because the messages are written directly to the syslog table.

6.2 Syslog filter

Step	Module/Entity	Operation
1	Web	Click on "Advanced" in the Syslog search page. Set Facility to "TR-069". You should now see the messages from the server only.
2	Web	Change the Severity level filer to show Error and Warning only. You should now see the warnings and errors from the server (if any).

6.3 Syslog Event

Let's assume that the filtering above showed a large number of syslog warning messages with the content: "gw: Blocked". We would like the syslog server to drop these messages, because they are so frequent and because they do not give us a lot of necessary or interesting information.

Step	Module/Entity	Operation
1	Web	Locate the Syslog Event page under Unit Type Actions. Create a Syslog Event with the EventId set to 2000 and the expression set to "gw: Blocked" (or another phrase that has occurred several times in the syslog messages - you may use regular expressions). The task must be DISCARD. Delete limit should be 0.
2	Web	Inspect syslog page to see that the Syslog Event has taken effect
3	Web	Change the Syslog Event to task DUCT. Set timeout value to 1 (minute). This task tells the syslog server to lookout for Duplicate UnitId & Content and save to syslog every Timeout minutes. In other words, you should now see a summary of equal messages every minute.
4	Web	Make a new Syslog Event, set EventId to 3000. The expression should be "sip" and the task should be STORE. Change the delete limit to 100. This will override the delete limit specified for messages containing "sip" and set it to 100 days. Check xaps-core.properties to view the default deletion limits based on severity.

7 Reports - requires the "Report certificate"

7.1 Unit Report

This report is default and provides a simple starting point for our "Report Tour".

Step	Module/Entity	Operation
1	Web	Locate the Report menu choice, and click on "XAPS reports" and "Unit". You should see a chart showing a few days and the unit count these days.
2	Web	Click on "Advanced" and change period type to HOUR. Choose the check box "Status" and "SoftwareVersion". You should now see one line for each combination of provisioning status and software version. This should give you a quick overview of the provisioning status of your system.

7.2 Syslog Report

This report shows how we're able to use a top-down approach in our investigation for causes of syslog messages.

Step	Module/Entity	Operation
1	Web	Locate the Report menu choice, and click on "Syslog reports" and "Syslog". You should see a chart showing a few days and the message count these days.
2	Web	Click on "Advanced". Choose the check box "Severity" and "Facility". You should now see one line for each combination of severity and syslog producer.
3	Web	An interesting aspect with Syslog reports, both this and Voip and others, are the ability to zoom into the data. Click on the the line coordinate which has the highest value and observe that period type shifts from DAY to HOUR and that the time period changes from many days to one single day.
4	Web	Again, click on the line coordinate with the highest value. You should now see a list of devices with a message-count and error/warning-message-count. The time period has again changed to one single hour.
5	Web	Try to change the order of the list, to show the device with most error messages on top. Click on this device-id's link.You should now see the dashboard of the unit.

7.3 Dashboard

The Dashboard is meant to give support desk a quick device status overview.

Step	Module/Entity	Operation
1	Web	If you completed the previous task, you should now see a dashboard. If not, search for a unit and choose "Unit Dashboard".
2	Web	The dashboard shows status for provisioning (for all devices), VoIP and Hardware status (for Pingcom devices) and Syslog status (for all Syslog emitting devices). All these states are summed in the Speedometer for overall score. See if you can understand how the overall score has been calculated (roughly).
3	Web	Click on Unit History. You should see 3 tabs: Voip, Hardware and Syslog. By clicking around a little you should recognize the system from the report system. Try to inspect another time period.

7.4 Voip

Prepare this by setting up a Pingcom device with a working Voip service and syslogging to this server.

Step	Module/Entity	Operation
1	Web	Open the dashboard for this unit. Make a phone call, do not hang up. Wait 15 seconds, then you should see that the dashboard shows an "active line". You may click on the link.
2	Web	A chart will appear, showing the MOS (Mean Opinion Score) for the call. It is refreshed every 5 seconds. Hang up.
3	Web	Enter Unit History and you should see the latest conversation. You can click on the link to the conversation and view chart again.

7.5 Exercises

1. Identify the three devices with the highest amount of "NoSipServiceTime" in the previous 6 hours. Investigate what might be the cause of this situation.
2. Identify the facility which sends most syslog messages in the previous hour
3. Use two metrics drop-down of the Voip chart to identify the correlating value for any low value of the metric "VoipQuality".

8 Advanced

8.1 Monitor a certain syslog message

The report system will monitor certain predefined metrics. You may however also use to monitor metrics of your own choosing. By following this procedure you can know how many units are experiencing a certain syslog message, hour by hour.

Step	Module/Entity	Operation
1	Web	Make a Unit Type Parameter called "System.X_YOURCOMP-COM.DNSErrorCount". It should have the flag "X"
2	Web	Make a Group, check the “Enable” checkbox in “Time rolling” and set Parameter to be the newly created Unit Type Parameter. Format should "yyyyMMddHH" and then choose "No offset".
3	Web	Make a Syslog Event. Set event id to 4000, expression should be "DNS Error" and task must be GROUPSYNC. Choose the group you just created. Delete limit can be 0. At this point we have created a syslog event which will synchronize the group-parameter to the Unit Parameter of the device that sent the syslog message "DNS Error". This group parameter is changed to a new timestamp every hour by Fusion Core Server. Furthermore, the group is counted in the Group Report. A few minutes into the next hour you should see the Group report for this particular hour, showing how many devices experienced a "DNS Error".

8.2 Trigger debug-mode for a certain syslog message

Some syslog messages might indicate a serious problem. In those case it could be a good idea if the device changed log level to "DEBUG", to try to identify the source of the problem. To complete this, you need to be able to run commands in Fusion Shell. The task also shows how to turn off the debug log level after 3 days.

Step	Module/Entity	Operation
1	Web	Make a Unit Type Parameter called "System.X_YOURCOMP-COM.DebugCleanupAt". It should have the flag "X"
2	Web	Make a Group with the name "TimeOffset", check the “Enable” checkbox in “Time rolling” and set Parameter to be the newly created Unit Type Parameter. Format should "yyyyMMdd" and Offset should be "3". We have now created a group which will be updated with the date 3 days ahead.

3	Shell	Make a script called debug.xss (a simple text file) and enter the following: setparam System.X_OWERA-COM.Debug 1 /ut:NPA201E/gr:TimeOffset/listparams > GROUP_PARAMS setparam < GROUP_PARAMS The NPA201E is supposed to be the Unit Type you're working within. Change this part of the script if necessary.
4	Shell	Upload the script file into Fusion using the shell command "importfile". Make sure to set the filetype to SCRIPT.
5	Web	Make a Syslog Event with the event id 5000. Expression should/could be "`Reboot reason.*(Watchdog|Destination memory)`" and Task should be "CALL". Choose the script previously created in CALL. Delete limit can be 0. Any syslog message with this special kind of "reboot" should trigger the script debug.xss for that particular unit. The group parameters from the group TimeOffset will be copied to this particular unit.
6	Web	Make a new group called "DebugCleanup", check the “Enabled” checkbox in “Time rolling” and set Parameter to the same Unit Type Parameter created in step 1. The format should still be "yyyyMMdd", but the offset should be "No offset". This group is then updated with the date of today.
7	Web	Make a job which works on "DebugCleanup" and set the parameter "System.X_OWERA-COM.Debug" to 0.

9 Shell (only for admins)

9.1 Startup

Shell is a powerful tool, and can do a lot of things far easier and faster than using Web. Let's look at how to start Shell and navigate the various contexts.

Step	Module/Entity	Operation
1	Shell	Start Fusion Shell by this command: java -jar xapsshell.jar
2	Shell	Choose the database which you have setup for Fusion. You should now see some ASCII art showing the xAPS Shell logo. You're in!
3	Shell	Type help to show the available commands. Note that you can get detailed help of a command by typing "help <commandname>". By typing just enough to avoid ambiguity, the help command will respond. A valid example is "help g" as short for "help generic".
4	Shell	You are now in the root context of Shell. To change context, lookup the command "cc" (change-context) and change context to a Unit Type.
5	Shell	Run help again to see the commands available in this context. Note that you can make a profile or a group or a syslog event in this context.
6	Shell	Change back to root context by entering the command "cc .." followed by enter. At this point you know how to log in, to get help and to navigate the shell.

9.2 export/import/delete

Some of the most useful commands is the ability to run pre-made scripts to export/import and delete the entire Fusion database or a Unit Type at once. These commands can only be triggered using command line option.

Step	Module/Entity	Operation
1	Shell	Type "unittypeexport ALL" on the root context of the shell.
2	Shell	To import a specific Unit Type to an Fusion database, that particular Unit Type must be exported first (from another Fusion database). The entire content of that Unit Type is expected to be

		stored in a directory with the same name as the Unit Type using this command: unittypeimport ALL
3	Shell	Careful with this one! This is the same principle as for the two previous commands: unittypecompletedelete ALL

9.3 Shell - simple looping/scripting

Step	Module/Entity	Operation
1	Shell	To export the output of a command, just do like this (assuming a Unit Type context): listparams > params.txt You may now inspect the content of the params.txt file: cat params.txt
2	Shell	To loop over a file (e.g. copy Unit Type Parameters), assuming you have changed to another Unit Type context. setparam < params.txt The command will loop over each line in the params.txt file. Each line will simply be appended to the "setparam" command and then executed.
3	Shell	You may also grab certain parts of the file and use. Each line is automatically delimited by use of space and double quotes. Therefore, if you need only need the names (for example) of the params.txt file, but want to set all flags to X, you can run the command like this: setparam ${1} X < params.txt The ${1} argument will make sure that only the first argument (delimited by space and double quotes) of each line of the params.txt file is used.

9.4 Exercises

1. Create a script to list the Unit Parameters (not the Profile Parameters) of one single unit. Run the script using the command "call".

2. Create a script to add a new Profile and a new Unit in this new Profile and then copy the Unit Parameters found in the previous task.

3. List all units which have Unit Parameter "System.X_OWERA-COM.LastConnectTms" set to yesterday and export the list of units to a file.