FAQ

FreeACS now offers a forum: http://freeacs.freeforums.org/

Latest updates:
2014-Nov-27: An Empty response (HTTP 204) is the only response - what is going on?
2014-Nov-25: How can I see what's going on in the server?
2014-Nov-25: The provision/reboot/reset button on Unit Configuration does not work!?
2014-Nov-25: What is the principle for provisioning with FreeACS?
2014-Nov-25: How can FreeACS discover my CPE and add it to the system?

General questions

Troubleshooting questions

Provisioning questions

General questions

Is it really free?

Yes - it's licensed with the MIT license - for maximum freedom. Even so it's not like you can install and take advantage of the system in 5 minutes - you must invest some of your time. If you follow the installation guide you may be able to install everything in a fairly quickly (~ hour?). If you like some assistance, it's possible - at a cost. The code is also available on GitHub, so you download it and start to make your own contributions and changes - no limitations!

How long does it take to install?

It can take as little as 5 minutes (if you know what you're doing), but most likely it will take ~60 minutes. Everything is explained in detail in the installation document - so there is no need to worry!

How can I modify the source code to suit my needs?

You can download the code from GitHub and install it in your IDE. Each repository contains a README.md file which explains how to proceed.

Does it support TR-111/TR-069 Annex G?

Yes. Using Fusion FreeACS you can initiate contact to your TR-111/TR-069A.G-enabled device, even though the device is hidden behind a firewall or a NATted router. This makes for quick changes from the server.

Where can I find documentation?

I've updated the Download-page with information about documentation. The Fusion Web User Guide contains some screenshots.

Do I need Fusion-certificates to run the system?

You no longer need to have any kind of Fusion-certificates to run the system. If you find documentation that suggest you need to, it's just because it was necessary when the was not open-source. However, if you want to run the system using HTTPS, you need to add SSL-sertificates (of course). The installation procedure tries to explain how to do this.

Does it support TR-131?

TR-131 is supported, but not 100%. The Web Service Module is the Northbound Interface (NBI) of Fusion FreeACS and supports the features that has been requested by customers so far. You can check out the Web Service Test Client in GitHub if you need some tips on how to make this use/integrate with this interface.

Installation/Troubleshooting questions

I cannot login to the Web interface - what's the problem?

There is several possible explanations for this issue:

- Try login with admin/xaps - this is the default user/pass to the Web/Shell login. This user/password is NOT the same as the user/password specified for the MySQL user in the installation script. In other words there is exactly ONE MySQL user specified in this system: 'xaps', and there can be MULTIPLE "fusion" users specified within the Web/Shell system. A fusion user may or may not have all the privileges to access/change/delete things in the FreeACS database. However, at the beginning there is simply one fusion user: 'admin' with the password 'xaps'. If this doesn't solve it, go to next bullet.
- Open a SSH-console to the server and try login to mysql like this: "mysql -uxaps -pYOURMYSQLPASSWORD xaps". If you succeed in this, proceed to next bullet. If not, one of two things have happened: You have not installed the MySQL-user properly (during the installation script) or you have forgotten the password you specified for the MySQL user in the installation script. The best solution for both these issues is to login into MySQL using the root-user, and set the password for the 'xaps' users (yes, there are two such users! - don't ask me why!). You can also check the database documentation for details on setup of MySQL users. If you have forgotten the root password for the MySQL database, then you must dig deeper on the internet on how to reset that password first (it is possible I think) or reinstall MySQL.
- Ok, so you are able to log into MySQL on the console, but still you cannot login into the Web. Then the issue is most likely to be found in your xaps-web.property file. Check there and identify a property called "db.xaps.url". This property should be found only once and the value should be on this form: xaps/YOURMYSQLPASSWORD@jdbc:mysql://localhost:3306/xaps . If it is not, change it. In other words, the Web-application are to use the 'xaps' user with the password specified in the installation script to login into the 'xaps' database. Now, cross check all the other property files and make sure they also have got the right password. You may have to restart the Tomcat server (service tomcat-7 restart) for the changes to take effect. When this is in place, hopefully the login into Web will work allright.

What is the ACS URL to insert into the CPE?

The proper URL is "http://yourhost/tr069". Add portnumber if the port is not 80. All the modules follow the same pattern: http://yourhost/MODULENAME.

How can I see what's going on in the server?

All the logs are written to the standard working directory on your application server. If you've followed the installation procedure with the Tomcat server it should be /var/lib/tomcat7. Here you will find logs from all the various server. You should look at the fusion-tr069-logs, especially the fusion-tr069-event.log and fusion-tr069-conversation.log. To turn on logging into the conversation log, check in the /var/lib/tomcat7 /common - you should find properties files to control all kinds of logging for each server - the conversation log must be turned on (it's off by default). To really capture all traffic between the device and the server, use wireshark/tcpdump/tshark (or the like - search the internet on to how use these very useful tools). Such traces are extremely valuable if you want someone else to look at the conversations.

http://localhost/ returns a blank page

Requests to / should be redirected to /web (which is the Web interface of FreeACS). You can check if this is the case by testing http://localhost/web . If you receive a login-page response from that URL, the redirection is not working, but that is a minor issue (everything will work fine still). The installation script may have failed at this point (you can try to run the l 290-291 in the installation script again). If the /web URL does not return a login page, then you should check the webapps/ directory in Tomcat (/var/lib /tomcat7/webapps). There you should find web.war file and a directory "web" (which is the web.war unzipped). If the file is not there, then the installation failed to copy them into that location. That could be due to missing permission (Linux is always picky about those.) You must run the installation script again with as sudo. If the file is there, but the corresponding directory is missing, then Tomcat may not be running. Check Tomcat's server log (/var/lib/tomcat7/logs/catalina.out). Try to restart Tomcat (command: "service tomcat7 restart").

The provision/reboot/reset button on Unit Configuration does not work!?

The provisioning button will try to initate a provisioning from the ACS - which it is the opposite of the usual behaviour of a CPE-ACS system. Thus is "special" behaviour, not always supported. For it to work, one of these conditions must be met:

- The FreeACS STUN server is set up and the CPE STUN client is connected to it.
- The ConnectionRequestURL parameter is read by the ACS and it is a public IP address. For the ACS to read a parameter by default, it must be already populated in the ACS-database (you can set it manually) or you can set a unittype parameter flag to 'A' for 'Always Read'.
- The ConnectionRequestURL and System.X_OWERA-COM.Device.PublicIPAddress parameters are found and the property "kick.expect-port-forwarding" in xaps-stun.properties is set to "true". In this case FreeACS will try to connect to the ConnectionRequestURL with the hostname exchanged with the PublicIPAddress found.

Another approach to get a device provisioned quickly is to change the provisioning interval by changing the System.X_OWERA-COM.Frequency parameter to f.ex. 10000 (10000 provisionings pr week).

An Empty response (HTTP 204) is the only response - what is going on?

If you experience a situation where the TR-069 server always return an Empty response (HTTP 204) after the initial Inform request/response exchange, you may be the victim of a missing cookie called "JSESSIONID". To make sure that this really is the problem, turn on tracing (wireshark, tshark, tcpdump) of the traffic and inspect the HTTP headers being sent between the parties. If the CPE does not return the cookie "JSESSIONID" after first receiving it, then you found the culprit. This could be a problem on the CPE side, but it seems like two server side settings could help:

1. The path-element of the cookie. Change it by editing context.xml and add . Then restart server. (http://stackoverflow.com/questions/3980392/tomcat-7-session-cookie-path)
2. The HttpOnly element of the cookie. Change it by editing context.xml and add . Then restart server. (http://stackoverflow.com/questions/17991090/tomcat-7-sessionid-cookie-disable-http-only-and-secure)

Provisioning questions

What is the principle for provisioning with FreeACS?

FreeACS always run this basic "conversation" with a few variations: (C=TR-069-Client/CPE/Device/Unit, S=Server/FreeACS/TR-069-Server)

1. C: Inform request to S
2. S: HTTP Basic/Digest Challenge
3. C: Inform request with proper HTTP Authentication information (requst contains information about software version, keyroot, etc)
4. S: Inform response (if authentication is ok)
5. C: Empty request (hand-over to S)
6. S: GetParameterValuesRequest (asks for all parameter which is configured for this particular unit/profile in ACS + all parameters with the A-flag set + ConfigVendorFile-object)
7. C: GetParameterValuesResponse (compare the reponse from the device with the parameters configured in ACS)
8. S: SetParameterValueRequest (send all values which differed between ACS and CPE, and as default set the Periodic Provisioning Interval to a new value every time (86400 +/- 20%))
9. C: SetParameterValueResponse
10. S: Empty request (HTTP 204) (handover/end of session)

This is the default process. The following situations will override this default behaviour, and with the following order of priority:

1. If System.X_OWERA-COM.Reset is set to "1" for this unit, then SPV-SPVr is replaced with FactoryReset-ResetResponse
2. If System.X_OWERA-COM.Reboot is set to "1" for this unit, then SPV-SPVr is replaced with Reboot-RebootResponse
3. If System.X_OWERA-COM.DesiredSoftwareVersion is different from the version reported in the Inform-request, then SPV-SPVr is replaced with Download-DownloadResponse (for the software)
4. If System.X_OWERA-COM.TR069Script.<ScriptName>.Version is different from the version reported in the ConfigFileObject, then SPV-SPVr is replaced with Download-DownloadResponse (for the script)

The idea with this process is that it is flexible enough and very robust (the main goal is to always bring the device in line with the ACS).

How can FreeACS discover my CPE and add it to the system?

The TR-069 server offers a feature called "discovery". The discovery process is really a two-stage process:

- Create the Unittype (if necessary) + Profile (if necessary) + Unit and retrieve the password from the CPE and store it in System.X_OWERA_COM.Secret (on the Unit). The server does this by running HTTP Basic Authentication and simply accepts the password and store it in System.X_OWERA-COM.Secret for future use. This makes this server "open" to all TR-069 enabled devices and is not considered safe to run in a production environment, unless you really know what you are doing.
- Create all the Unittype parameters (using the TR-069 Method GetParameterNames). This procedure may fail!! Many devices are not sufficiently "strong" to handle it.

To perform step 1 and 2 of this process, the are two requirements which must be fulfilled:

- The discovery.mode property in the xaps-tr069.properties file is set to "true"
- The Unit is not found in FreeACS (The Unit-id is unique across the entire FreeACS database, and if it exists in another Unittype, it will not be created again)

To just perform step 2 of this process, there are two other ways to go about it:

1. Set parameter System.X_OWERA-COM.Discover = "1" (set on unit or profile). The server will reset this parameter to "0" on the particular unit, to avoid repeating the process.
2. If a Unittype has been created, but NO Device/InternetGatewayDevice-parameters have been created (only System-parameters).

An alternative to create the parameters this way, is to use the Web/Shell interface to add them manually.

How can I manually add a CPE to FreeACS system?

Run the Shell (check User Guide on how to run it.). When logged into the shell, run the command "help setunittype" to understand how to run this command (or read the doc). Then create a unittype - the name should be the name of the devicetype. A unittype contains profiles (static groups of devices) and firmwares, etc. Change context into this unittype ("cc ut:UNTITYPENAME") and then change into the

Default profile context ("cc pr:Default"). Then add a unit (device) with the setunit command. The name of the unit must be exactly the same as the ACS-username specified in the CPE. Then change context into the unit ("cc un:UNITNAME") and create the ACS-password with this command: "setparam System.X_OWERA-COM.Secret YOUR-CPE-PASSWORD". At this point the server should be ready to accept incoming requests from your CPE. Nothing is actually provisioned at this stage, naturally, but it should connect.

How to setup STUN server?

The STUN server documentation is found in chapter 7 in the TR-069 Server User Manual. Not really intuitive...sorry about that. However, here is brief summary:

- The STUN server must run on the same host as the TR-069 server.
- Your CPE/TR-069 client must support TR-111 part II. If the device has a parameter named UDPConnectionRequestAddress you're probably ok.
- Setup the CPE to connect to the STUN-server. This can be done, either on the CPEs own configuration pages or using Fusion to provision the following parameters:
  - InternetGatewayDevice.ManagementServer.STUNEnable = 1
  - InternetGatewayDevice.ManagementServer.STUNMinimumKeepAlivePeriod = 30
  - InternetGatewayDevice.ManagementServer.STUNServerAddress = STUN-SERVER-IP/HOSTNAME (depending on the CPE, IP-address might be safer to use)
  - InternetGatewayDevice.ManagementServer.STUNServerPort = 3478

  Make sure these parameters are actually set on the CPE. You can see this several places, like the fusion-tr069-event.log or click on the "Provisioning history"-heading in the Unit Configuration Page.
- When the device is setup to connect to the STUN-server, you must make sure this actually happens. You can try checking out the fusion-stun.log or use wireshark/tshark to trace traffic to port 3478.
- When the device is actually connecting to the STUN-server AND has provisioned another cycle with the ACS, the parameter UDPConnectionRequestAddress will be populated with a value like IP-address:Port - this is very good sign!
- At this point, the Web interface (with it's provisioning buttons) can start to work (try the "Provision" button). The Web interface will issue a "kick" to the STUN-server, which will send the kick over the UDP/STUN-connection to the device. You can follow the progress in the fusion-stun-kick-single.log. The device should then initiate a regular provisioning cycle. The other buttons work the same way, but certain flags are in the System.X_OWERA-COM.-parameters to signal reboot, read-all or factory-reset on the next provisioning cycle.

All my devices have the same ACS-user/pass - can this be fixed?

Yes - very likely. FreeACS can be setup to run a simple script upon contact with the CPE, and this script can actually change the ACS-username and ACS-password of the device. For this to work, the CPE must contain a unique serial number as this would be the basis for building a unique ACS-username. The following steps must be done:

- We assume that the "default" unit/device has been created in FreeACS. We assume that the unit-id is "admin" and that the unit-is located within the "Default" profile.
- Create a group. Read about groups in the Fusion Web User Manual. Make a group with filters to match the ONLY the default unit-id. One way could be to let the group match all device within the "Default" profile, but you must then make sure that no other units are created within this default profile (se below in the script).
- Create a \"file\" (inside the Web interface of FreeACS) - must be of type "SHELL_SCRIPT". A script that has been made for a customer some time ago looked something like this.

```
# We assume that this script is triggered by a SHELL-Job

# This script attempts to change the ACS username
# of a device. The use case is a AirTies router which always
# identifies as "admin" when it connects to the TR-069
# server. We want to change this username "on-the-fly"
# to be on this form: --

# This script will be started in the unit-context, that is
# when reading a parameter like SerialNumber, we will get the
# serialnumber of the device which triggered this script. The
# serialnumber is stored in the "sn" variable
var sn '${_InternetGatewayDevice.DeviceInfo.SerialNumber}'

# Next we must change the ACS-username in this context - this parameter
# will be set on the device.
setparam InternetGatewayDevice.ManagementServer.Username 012345-MyProductClass-${sn}
```

```
# Create unit in FreeACS and make sure the profile DevicesWithProperNames exists
# before running this script. You may of course place the device in the Default
# profile, just make sure that the group doesn't match these new units. If you run
# the server in discovery mode, this may also be skipped, since the unit-id will be
# created on the next connection with the server.
../../pr:DevicesWithProperNames/setunit 012345-MyProductClass-${sn}

# We *could* change the password, to make something more secret
# than the default password. In that case you may run
# Set on device
setparam InternetGatewayDevice.ManagementServer.Password Super-secret-${sn}
# Set the same password in FreeACS
../../pr:DevicesWithProperNames/un:012345-MyProductClass-${sn}/setparam System.X_OWERA-COM.Secret Super-secret-${sn}

# We do some cleanup on the default unit - just in case.
delparam InternetGatewayDevice.DeviceInfo.SerialNumber
```

You should modify the script to suit your unittype name and other needs. The password is very poor using this method, but you can also improve on that by generating passwords outside FreeACS and dump it into the database using the Shell/CLI.

- Create a job. Read about jobs in the Fusion Web User Manual. The essence is that you make a job of SHELL type and uses the group you created in the previous step and the file you created. Then make sure to start the job. Set the repeat counter on the job to 99999999, to make sure the job is run as many times as needed. Set the repeat interval to 0, to allow the job to re-run on every connection made from a "default device".
- Upon next provisioning, the job will be executed on that particular device. After the job has completed, the CPE should have a new user/pass.