
V6 Telephony Software



Installation Guide

PROPRIETARY INFORMATION

The information contained in this document is the property of Aculab plc and may be the subject of patents pending or granted, and must not be copied or disclosed without prior written permission. It should not be used for commercial purposes without prior agreement in writing.

All trademarks recognised and acknowledged.

Aculab plc endeavours to ensure that the information in this document is correct and fairly stated but does not accept liability for any error or omission.

The development of Aculab's products and services is continuous and published information may not be up to date. It is important to check the current position with Aculab plc.

Copyright © Aculab plc. 2018-2022 all rights reserved.

Document Revision

Rev	Date	By	Detail
6.0	06.12.02	CJL	Interim release
6.1.0 A	19.06.03	DJL	Controlled release of initial revision 6 software
6.1.0	15.01.04	DJL	updated installation and configuration tools
6.1.1	18.06.04	DJL	Review updates
6.2.2	07.09.04	DJL	Beta release
6.2.2	15.09.04	DJL	Full release
6.2.3	28.10.04	DJL	Updates for Linux
6.2.4	12.11.04	DJL	Addition of SS7 signalling link and ISUP bearer ACT info.
6.3.0	24.12.04	DJL	Various updates including support for new hardware
6.3.1	26.01.05	DJL	Corrections to appendix B
6.3.2	10.02.05	DJL	Removal of Solaris restrictions
6.3.3	22.02.05	DJL	Addition of Linux/Solaris configuration notes
6.3.4	19.04.05	DJL	Correction of typological errors
6.3.5	26.05.05	DJL	Review of ACT functions
6.4.0	03.11.05	DJL	Updates for V6.4 release
6.4.1	18.01.06	DJL	Correction of ACT images/definitions
6.4.2	10.04.06	DJL	Review updates – small changes
6.4.3	06.07.06	DJL	Update to document cross reference descriptions
6.4.4	15.08.06	DJL	Update to Prosody X information
6.4.5	21.08.06	DJL	New example added – installing a Prosody X card.
6.4.6	05.10.06	DJL	Updates to Linux installation example

6.4.7	14.12.06	DJL	Updates to include Solaris specific information
6.4.8	27.09.07	EBJ	Removal of Download and install of the Aculab telephony-software and Telephony-software download and installation.
6.4.9	21.09.09	DRG	Updated Aculab Configuration Tool (ACT) screenshots.
6.4.10	02.11.09	EBJ	Updated to corporate fonts, reformatted and minor corrections.
6.6.0	17.03.15	BWM	Updated for 6.6 release, incorporating Ipv6
6.7.0	11.04.17	PGD	Updated supported O/S list.
6.7.1	06.09.18	PGD	Delete Fetex, update O/S list
6.7.2	29.11.19	PGD	Update O/S support list, correct a weblink
6.8.0	3.11.22	PGD	Align with 6.8 release and Evo, delegate signalling system protocol switch descriptions to Firmware *.txt files.
6.8.1	10.11.22	PGD	Reformatting

CONTENTS

1	Installation and role of V6 telephony software	5
2	Installation process overview	6
2.1	Install the Aculab hardware	6
2.2	Download and install the V6 telephony software	6
2.3	Completing the installation	7
2.4	Configuring V6 telephony software	7
2.5	Getting started using example code.....	7
3	Procedure for completing a Linux installation	8
3.1	Preparation.....	8
3.2	Driver build.....	8
3.3	Loading and unloading driver	9
4	Command line card registration and configuration	10
4.1	Chassis card registration	10
4.2	Chassis card deregistration	11
4.3	Chassis card configuration	11
4.4	The voip_rm file.....	12
4.5	Applying card configuration.....	13
4.6	Signalling firmware download.....	13
5	Card configuration file format.....	15
5.1	General section.....	15
5.2	Call section	15
5.3	Switch section	16
5.4	Speech section	17
5.5	VoIP section.....	17
6	Flash upgrades.....	19

1 Installation and role of V6 telephony software

The various types of Prosody X chassis require a controlling host to run Aculab “V6 telephony software” in order to configure, manage, and allow application access to chassis TDM and media processing resources through Aculab APIs.

The following chassis types are supported in the 6.8 distribution of the V6 telephony software:

- Aculab Prosody X Evo
- Aculab Prosody X HA
- Aculab Prosody X 1U Enterprise

The above chassis types present their resources to controlling hosts as Prosody X cards, with the Prosody X HA chassis showing as one or two cards depending on type, and the other chassis types showing as a single card.

Following attachment of Prosody X chassis to network, the V6 telephony software must be installed on the controlling host system and then configured to set up use of Prosody X resources in the desired way. The same process is used to install and configure a Prosody S host media processor, and to install the services and libraries that provide API access to applications.

NOTE

The controlling host must either be a x64 Linux or recent x64 Windows server system, and Linux systems must have support for glibc 2.17 or later. Please check with Aculab for exact details of supported systems.

V6 telephony software installation is achieved through use of Aculab Installation Tool (AIT) which is freely downloadable package from the Aculab website. The package provides both a graphical user interface (AIT_GUI) and command line (AIT_CMD) tool to install V6 software either over the network from an Aculab distribution server or from a local distribution file.

The resulting installed V6 telephony software will include device drivers, management and configuration tools, firmware for Prosody X Evo, HA and 1U chassis types, Prosody S software, services and libraries for interaction with applications, and licence management software. On Linux based systems some scripts will need to be run to start this software running prior to configuration, whereas with Windows based systems it will start running automatically following successful installation.

Once V6 telephony software has been installed and is running on a controlling host, in order to bring Prosody X / Prosody S resources into an operational state, Prosody X / Prosody S elements will require configuration either through use of the graphical Aculab Configuration Tool (ACT) or through use of configuration text files and various command line tools.

The configuration process will include such steps as selection of IP addresses to be used, signalling systems for E1/T1 trunks, and system clock source specification. For the HA and 1U enterprise chassis types, selection of TiNG algorithms to download onto ProsodyX DSPs will also be required, this step is not required for the Prosody X Evo chassis.

The V6 telephony software installation also provides tools that can be used to verify a Prosody X chassis has up-to-date flash programming, and update it if required.

2 Installation process overview

Installing and configuring Aculab hardware and any associated telephony software involves a number of key steps:

1. Install the Aculab Prosody X chassis and attach to suitable LAN.
2. Download and install the required Aculab telephony software distribution package on controlling host system attached to same LAN.
3. Configure the Aculab telephony software.

Once you have completed these steps, you will be able to utilise the Aculab card telephony software and hardware resources through the various Aculab application programming interfaces (APIs).

2.1 Install the Aculab hardware

Install the Aculab Prosody X chassis in accordance with the appropriate hardware installation guide. For configuration purposes the chassis just needs to be powered up and be reachable on the same subnet as the designated controlling host. Assignment of a Prosody X chassis to a particular controlling host and selection of the IP addresses for Prosody X cards presented by the chassis are achieved either through the ACT or the `prosody_ip_card_mgr` command. In both cases, you will need to know the serial numbers for the presented Prosody X cards.

NOTE

Cards presented by a Prosody X HA chassis will have serial number suffixes "PX1" and (if second card present) "PX2". Other chassis types just present a single card which has no serial number suffix.

2.2 Download and install the V6 telephony software

The required V6 telephony software is available for download from the Aculab web site at www.aculab.com. To assist you with downloading and installation, Aculab provide a tool called the Aculab installation tool (AIT) in both GUI and command line versions. The tool and its usage instructions can be found in the Windows/Linux tools tab of the support/downloads page on the Aculab website.

The AIT application can be run on most systems, however Internet access would normally be required to enable you to select and download the individual V6 telephony software components required for your particular system configuration. Should you require to run the Aculab installer on a stand alone system, the AIT has an option to create an Aculab Package File (APF), which can be used to install software on a system without Internet connectivity.

The AIT will allow selection of a directory into which Aculab telephony software is installed, this is referred to in this document as the ACULAB_ROOT directory.

NOTE

The default location of ACULAB_ROOT for Windows is

`"C:\Program Files (x86)\Aculab\v6"`,

and for Linux

`"/usr/local/aculab/v6"`.

2.3 Completing the installation

Once the required V6 telephony software packages have been installed by the AIT, on Windows systems the installed software will start automatically. On Linux systems however, further installation steps are required (involving running of a “dacpinst” script). See section 3 for details on completing a Linux installation.

2.4 Configuring V6 telephony software

The process of configuring firmware required to utilise the Aculab card resources is detailed in the [Aculab configuration tool](#) (ACT) user guide. Where use of a GUI tool is not possible, the same configuration process can also be achieved using command line tools, and set up of various text configuration files. See section 4 for details.

2.5 Getting started using example code

The software installed by the AIT includes a package called ‘example code’. This package contains generic examples of applications developed for test purposes only. These examples may be of assistance when getting started with the Aculab APIs. For further information, please see the Aculab SDK documentation that accompanies the example code files

3 Procedure for completing a Linux installation

After the AIT has been used to install the required set of packages, on Linux systems additional steps are required to start the installed software running, including the building of driver modules.

3.1 Preparation

In order to build these drivers, the Linux kernel headers need to be installed and configured. Your Linux distribution documentation will explain how to do this.

NOTE

See `ACULAB_ROOT/driver/readme.install` for distribution-specific information before building the Aculab drivers on Linux.

NOTE

Linux installation scripts are kept in `ACULAB_ROOT/driver`

The environment variable `$ACULAB_ROOT` must be set to the root directory of the Aculab distribution (e.g., `/usr/local/aculab/v6`). A provided script, `setV6.sh`, in `ACULAB_ROOT` directory will set the all of the environment variables required for building drivers correctly. To run it, invoke:

```
source setV6.sh
```

This file may also be modified and copied into `/etc/profile.d` to ensure settings are loaded for each bash shell.

The many variations of Linux distributions preclude distribution of a complete pre-built driver, so a driver kit is provided that contains precompiled objects with the final compilation of the kernel-specific code and the linking together of the loadable driver module completed on the host machine. The script files `dacpinst` for Linux are used to automate this process and configure the driver to your needs.

To avoid compilation errors, please ensure that the sources for the intended kernel have been installed, prior to building the driver. For most distributions, this required package would be called `kernel-source`.

The commands for `dacpinst` are as follows:

```
./dacpinst build      - builds, then configures, the driver.
./dacpinst clean     - removes the objects, allowing a clean rebuild.
./dacpinst version   - gives the version of dacpinst.
```

At each prompt, `dacpinst` will often give options (i.e., "yes/no") in brackets "(...)", and the default actions in square brackets "[...]". The default actions will predict the desired input according to a typical build. If you'd like the default input, hit return, or else type in what you'd like instead.

3.2 Driver build

Running "`./dacpinst build`" asks you what components you would like to include. Select the components as necessary. If an invalid combination is selected, you will be asked to enter an alternative combination.

Once you have successfully completed the previous part there will be a loadable module created that represents the driver for your system.

You will then be asked for a non-privileged user to run the Aculab tools. The Aculab Resource Manager (`acuresmgr`) will be run as root, but will spawn any tools as this user minimizing the amount of root access required for the Aculab tools.

3.3 Loading and unloading driver

After successfully configuring the driver, you will have been provided with a new script file called `aculab_dacp` for Linux.

The commands for Linux `aculab_dacp` are as follows:

```
./aculab_dacp start      - loads, then starts, the driver.  
./aculab_dacp stop      - stops, then unloads, the driver.  
./aculab_dacp restart   - restarts the driver.
```

`aculab_dacp` also creates the device nodes (in `/dev`) required for correct operation.

The `aculab_dacp` script file can be used to automatically load the driver at the boot-up sequence of Linux, for instance being invoked from a system unit file or through some other Linux distribution specific startup script system such as “init.d” or “cron”.

Once the drivers have been loaded in the controlling host machine, the system should be ready for registering of Prosody X chassis reachable on the same subnet as the controlling host, or set up of a Prosody S media server.

To unload the driver, ensure that no Prosody applications or tools are running, and then enter:

```
./aculap_dacp stop
```

4 Command line card registration and configuration

In case of a server machine, where use of the graphical ACT configuration tool is not practical, chassis registration and card configuration can be achieved using command line tools and creation of card configuration text files. On a Linux system, the script in ACULAB_ROOT directory should be run as below to set up a path to these command line tools:

```
source setV6.sh
```

4.1 Chassis card registration

The Prosody X cards presented by a Prosody X chassis must be added to list of cards that a controlling host will boot and supply software.

In the simplest case, a Prosody X card presented by a chassis may be added using one of the following commands::

```
prosody_ip_card_mgr add <serial_no> <card_key> <card_ip_address> <netmask>
```

to provide an explicit configuration or:

```
prosody_ip_card_mgr add <serial_no> <card_key> dhcp
```

to configure using DHCP.

The `card_key` is a password which ensures only hosts knowing this password can access the resources of the Prosody X card. More than one system may use resources on a Prosody X card, but they must all use the same `card_key` to interact with the card.

For example, If you want to assign a Prosody X card an IP address of 192.168.1.42 and card key "theKey", you could use:

```
prosody_ip_card_mgr add 179119 theKey 192.168.1.42 255.255.255.0
```

More complex card configuration setups are possible through specification of flags to the above commands. To see all the available options, invoke `prosody_ip_card_mgr` without any parameters.

To list all Prosody X cards currently known by the controlling host use the following command:

```
prosody_ip_card_mgr -list
```

Example output:

```
prosody_ip_card_mgr V6.8.6
4 Prosody IP cards registered on this system:
221896 (Prosody X) (R):(10.202.205.48, will boot) In service
226159 (Prosody X) (R):(10.202.206.37, will boot) In service
241411 (Prosody X) (R):(10.202.205.43, will boot) In service
100000 (Prosody S) (L):127.0.0.1 U:6583 C:2030 L:2546 Connected
```

To obtain card details for a specific card, use the following command:

```
prosody_ip_card_mgr --info --serial <serial_no>
```

For example, to obtain information for card serial number 221411, use the following:

```
prosody_ip_card_mgr --info --serial 241411
```

This will return information similar to the following for a Prosody X Evo chassis:

```
prosody_ip_card_mgr V6.8.6

Card 241411:
  Type:          Aculab Prosody X Evo
  Version:       V1.0
  MAC address:   3C:EC:EF:21:16:5E

  Configure:    yes
  Persistent:   no
  Remote:       yes
  Key:          theKey
  IPv4:         Dynamic
  IPv6:         Not configured
  Controlling IP: 10.202.205.43
  Status:       In service

  2 ethernet ports:
    0: Connected at 1000Mbit full duplex and active
    1: Disconnected

Device 0:
  Model:          AC2100 Prosody T
  Serial number: 241420
  Version:        1.0
  Bootloader:
  Firmware:      Aculab Prosody X Evo Firmware 6.8.6.20220927
  3.3V:          3.246V
  12V:           12.014V
  Temperature 0: 33C
  Temperature 1: 41C
  FPGA version:  14.10

Device 1:
  Model:          AC2100 Prosody T
  Serial number: 241421
  Version:        1.0
  Bootloader:
  Firmware:      FPGA 14.10 built at 14:16:39 on 9-Sep-2022
  3.3V:          3.252V
  12V:           12.000V
  Temperature 0: 36C
  Temperature 1: 42C
  FPGA version:  14.10
```

The devices listed as “AC2100 Prosody T” refers to internal element of a Prosody X Evo chassis, the listed serial numbers of these devices are not normally relevant to Prosody X Evo configuration procedures, it is the serial number of the chassis which is the one that should be specified for command line tools (241411 in this case).

4.2 Chassis card deregistration

To remove a card use the following command:

```
prosody_ip_card_mgr remove <serial_no>
```

4.3 Chassis card configuration

The configuration of Prosody X cards is determined using a series of card configuration files. These determine, among other things, the signalling firmware that is automatically downloaded to card TDM ports on V6 telephony software startup.

The configuration files have a name consisting of serial number of a Prosody X card followed by a suffix of “.cfg”. For example, a card with the serial number 12345 would

have a corresponding `cfg` file of `12345.cfg`.

The Resource Manager element of the V6 telephony software looks for card configuration files in the `ACULAB_ROOT/cfg` directory.

These files are created either automatically by the Aculab Configuration Tool (ACT), or using manually the command line using `cfgtemplates` tool and an ASCII text editor.

Creating configuration files with `cfgtemplates`

To run this application the user must provide a serial number into the application, for example,

```
cfgtemplates 241411
```

The above example will start an interactive session where some choices for card configuration are specified:

- If the card is already in the system, during the session you will be asked if you wish to automatically detect the card type.
- You will then be asked to specify the bus mode; H100 should be specified.
- Next you will be asked if the card is a "Bus Master" i.e. is this card to supply the clock source. If yes, then the clock source will be requested. If No, the card will take the CTBus Mode as the clock source. For Prosody X Evo chassis and Prosody 1U enterprise chassis, a response of yes should be given.
- Now you will be asked to confirm if the card is to be bus terminated (at either end of the CTBus) – this is a legacy option, reply no to this.
- If the card is not known to the system, you will be asked to specify the card type. This will be followed by a series of questions relating to the number of ports, Prosody modules and so on as appropriate

Once the session is complete, a ".cfg" file will have been created in the current directory, for example `241411.cfg`

This will not be a completed version of the file; certain fields have been left blank for the user to edit using a standard ASCII text editor. For example, the call section will have no firmware selected for the call ports. Also the Speech section will not have any TiNG firmware specified (this section does not need to be completed for the Prosody Evo chassis).

The file should now be edited to specify missing information and then copied to `ACULAB_ROOT/cfg`.

The sections and fields of this card configuration file are described in section 5.

4.4 The `voip_rm` file

Another file in the `ACULAB_ROOT/cfg` directory `voip_rm.txt` controls if SIP services are started. This file is created by default if not present on telephony software start up, and can be edited. The default content is as below:

```
[SIPService]
USE=TRUE
AddressHold=IPT_ENABLED
EarlyMedia=IPT_ENABLED
[EndSIPService]
```

To disable SIP, the field "USE" in the `[SIPService]` section should be set to FALSE.

4.5 Applying card configuration

When the V6 telephony software is started, for each card for which a “.cfg” file exists in `ACULAB_ROOT/cfg`, the configuration for that card will be applied as specified in the file.

The configuration can be reapplied if changed after startup by invoking the `config` command:

```
config <serial-number> <switches>
```

Where:

`<serial-number>` is the serial number of the card

`<switches>` are the following:

- `-v` Verbose mode.
- `-m <module-no>` Download TiNG firmware to selected module.
- `-p <port-no>` Download signalling firmware to selected port.
- `-s` Apply clocking setting to card.

These switches can be combined, for example:

```
config 241411 -p 0 -p 1 -p 2 -p 3 -m 0 -m 1 -s -v
```

This example will download signalling firmware to ports 0 – 3 & TiNG firmware to modules 0 –1, apply clocking configuration, and run in tool verbose mode.

4.6 Signalling firmware download

`fwdspldr` is a signalling firmware download utility for TDM ports on Prosody X cards. It is both used to download the signalling protocol firmware, and to specify signalling configuration switches. (For previous generations of Aculab hardware it was also used to download signalling DSP firmware but this is now managed automatically so the dsp options are no longer relevant.)

Usage for `fwdspldr` is as follows:

```
fwdspldr <serial-number> <port-number> <filename> <config>
```

Where:

- `<serial-number>` Is the serial number of the card to download firmware to.
- `<port-number>` Is the port on the card to download the firmware to.
- `<filename>` Is the signalling protocol firmware file to download.
- `<config>` Is used to specify any configuration switches to be passed to the signalling protocol firmware.

NOTE

The filename suffix that should be used depends on the chassis type, for Prosody X Evo chassis the suffix should be “.pxi”, for other chassis types the suffix should be “.pmx”

For further details of protocol firmware switches, (-s switches) that can be specified in <config> please refer to the documentation (*.txt file) that accompanies each protocol firmware file in `ACULAB_ROOT/Firmware` installation directory.

The following example downloads firmware to the first two ports of a Prosody X Evo chassis with first port set up for network side ETS300 and the second for user side.

```
fwdspldr 241411 0 ets_net.pxi -cNE
fwdspldr 241411 1 ets_usr.pxi
```

5 Card configuration file format

The `.cfg` files are split into a number of sections corresponding separate areas of card functionality, each section delimited by [`<section_name>`] and [`End<section_name>`] delimit each of the components. The sections are in following order:

```
[General]
[EndGeneral]
[Call]
[EndCall]
[Switch]
[EndSwitch]
[Speech]
[EndSpeech]
[VoIP]
[EndVoIP]
```

NOTE

The fields in sections are case sensitive

5.1 General section

This section contains a single field, `CardName`. The field value should contain a name assigned to card.

```
[General]
CardName=TRUNKSET1
[EndGeneral]
```

5.2 Call section

The following fields are required in the call section for each port.

```
Port=
PortName=
Firmware=
Config=
DSPA=
DSPB=
PortInit=
```

The following definitions apply:

Port

Is the port number starting at 0, if you have four ports then they are numbered 0-3, for example:

```
Port=0
```

PortName

Is a user defined 16 character text string, for example:

```
Portname=London Port1.
```

Firmware

The location from `ACULAB_ROOT` of the call firmware file, for example:

```
Firmware=/Firmware/ets_usr.pxi
```

Config

The protocol firmware switch to applied at download time, for example:

```
Config=-CNE
```

DSPA & DSPB

Redundant fields that were used for earlier generation cards.

PortInit

Can be either `TRUE` to perform `PortInit`, or `FALSE` to not perform `PortInit`, for example:

```
PortInit=TRUE
```

A typical configuration would be as follows:

```
[Call]
Port=0
PortName=Port 0
Firmware=/Firmware/ets_net.pmx
Config=-cNE
DSPA=
DSPB=
PortInit=TRUE
Port=1
PortName=Port 1
Firmware=/Firmware/ets_usr.pmx
Config=
DSPA=
DSPB=
PortInit=TRUE
[EndCall]
```

5.3 Switch section

There are 7 fields in the switch section. These fields map directly to the fields in the "h100_config_board_clock_parms" structure, and are populated with exactly the same values (see the Switch API guide for further details).

Example switch section:

```
[Switch]
CtBusTermination=TRUE
CtBus=SWMODE_CTBUS_H100
Source=H100_SOURCE_INTERNAL
Network=0
H100Mode=H100_MASTER_A
AutoFallBack=H100_FALLBACK_DISABLED
NetRefClockSpeed=H100_NETREF_8KHZ
[EndSwitch]
```

Each line in the section consists of a field name followed by equals and then the value assigned to that field.

.

5.4 Speech section

NOTE

This section only applies to Prosody X HA chassis and Prosody X 1U enterprise chassis, it is not relevant for Prosody X Evo chassis.

The speech section is split into "blocks" in a similar fashion to the call section, however in this case the "blocks" are delimited by the "Module" field.

The "Firmwares" field takes the full path from the `ACULAB_ROOT` of the TiNG firmware to be downloaded.

The TiNG firmware files are assumed to be in the following directory:

```
ACULAB_ROOT/TiNG/starcore/gen
```

Example speech section:

```
[Speech]
Module=0
Firmware=datafeed inchan outchan playA
Module=1
Firmware=datafeed inchan outchan playA
[EndSpeech]
```

module – the module number, starting with module 0

Firmware – the firmware to be downloaded to that module.

5.5 VoIP section

These settings are largely redundant and were used for older cards. However some settings are still used for legacy applications that are built to use SIP indirectly through the Media Handler Plugin (MHP). The following definitions apply:

RTP_TOS, RTCP_TOS - Used to set IPv4 type of service for RTP end points set up by Media Handler Plugin, normally should be set to zero.

DEF_JITTER, MAX_JITTER - used to set RTP jitter buffer parameters, set to zero for usual defaults

DTMFDETECT - controls if conversion of inband tones to RFC2833 is enabled

IPADDRESS, SUBNETMASK, GATEWAY, - redundant settings that were used for earlier generation cards, *these should be set to 0.0.0.0 to allow the config tool to parse the files correctly*

ENCODEGAIN, DECODEGAIN, MAX_JITTER_BUFFER - redundant settings that were used for earlier generation cards.

TDMENCODING, ECHOCANCEL, ECHOSPAN, ECHOSUPPRESSION - redundant settings used for no longer supported generic timeslot mode

Example:

```
[VoIP]
IPAddress=0.0.0.0
SubnetMask=0.0.0.0
Gateway=0.0.0.0
EncodeGain=8192
```

```
DecodeGain=8192
RTP_TOS=0x0
RTCP_TOS=0x0
Def_Jitter=30
Max_Jitter=150
Max_Jitter_Buffer=250
TDMEncoding=TDM_ULAW
EchoCancel=EC_OFF
EchoSpan=0
EchoSuppression=ES_OFF
DTMFDetect=1
[EndVoIP]
```

6 Flash upgrades

A Prosody X chassis has flash memory associated with its presented Prosody X cards containing firmware and FPGA data, which allows it to boot into a state where it can be configured over a network and operate its devices.

Aculab may release new versions of the firmware flash images from time to time, for example to fix bugs or enhance performance

Each new distribution of Aculab host software is tested with a specified version of the set of flash images and also contains the files that comprise these images. Having installed a new version of host software, you should find out if the firmware images are up to date - if it is not up to date then a flash upgrade should be carried out.

Flash upgrades are most easily applied using the ACT tool (see ACT user guide), but can also be applied using the command line tool `prosody_ip_card_flash`

In order to determine flash image versions for cards presented by a Prosody X chassis, the output of `configuration_summary` can be consulted, or the command line tool `prosody_ip_card_mgr` can be invoked with the "info" subcommand. For applications, the API call `acu_prosody_ip_get_device_info()` allows the information to be retrieved programmatically.

Example output from tool info subcommand is shown below.

```
prosody_ip_card_mgr info 226159

prosody_ip_card_mgr V6.8.6

Using old format
Card 226159:
  Type:          Aculab Prosody X card
  Version:       V1.2
  MAC address:   00:02:1F:00:F6:1A

  Configure:    yes
  Persistent:   yes
  Remote:       yes
  Key:          mykey
  IPv4:         Dynamic
  IPv6:         Not configured
  Controlling IP: 10.202.206.37
  Status:       In service

  2 ethernet ports:
    0: Connected at 100Mbit full duplex and active
    1: Disconnected

Device 0:
  Model:          AC5850 Prosody X
  Serial number: 226159
  Version:        1.2
  Bootloader:
  Firmware:       Aculab Prosody IP Firmware 1.0.149.391
  3.3V:           3.266V
  Temperature 0: 39C
  Temperature 1: 32C
```

The `prosody_ip_card_flash` utility is used to upgrade the flash images on the card under the control of the host. From a command line prompt run:

```
prosody_ip_card_flash <serial> <index>
```

Where `<serial>` is the serial number of the card and the `<index>` device selector should always be set to zero for current Aculab products.

This will check the card to see if any firmware updates can be applied to the card.

```
prosody_ip_card_flash 226159 0
Flash update tool V6.8.6

Checking for updates to 226159, AC5850 Prosody X 1.2
  FPGA is up to date
  Bootloader is up to date
  Kernel is up to date
  Ramdisk is up to date
  DTB is up to date
Card firmware is up to date
```

If any updates are required, re-run the command with the additional `-u` option to update the flash image:

```
prosody_ip_card_flash <serial> <index> -u
```

CAUTION

Once the upgrade is in progress you must not attempt to abort it: doing so may cause the flash upgrade to fail rendering the card uncontactable. Should a flash upgrade fail for any reason, please contact Aculab support for advice.

The card should come back into service a few minutes after performing a flash upgrade.

NOTE

For Prosody HA chassis fitted with two Prosody X cards, each card (one with suffix PX1, one with suffix PX2) needs to be flash upgraded individually.

Contact us

Phone

+44 (0)1908 273800 (UK)
+1(781) 352 3550 (USA)

Email

Info@aculab.com
Sales@aculab.com
Support@aculab.com

Socials



Certificate number IS 722024
ISO 27001:2013



Certificate number FS722030
ISO 9001:2015