



# NetLinx Module Interface Specification

for

## EchoSystem v3.3.0



*Latest version available at: <http://opax.swin.edu.au/~romclean/amx/echo>*

## TABLE OF CONTENTS

Revision History .....	3
Introduction.....	4
AMX Implementation .....	5
Control Messages .....	6
String Feedback Messages.....	10
String Feedback (Warning) Messages.....	12
Debugging with the module .....	13
Channel Feedback .....	14
Level Feedback .....	15
EchoSystem Server Configuration.....	16
Trusted Systems .....	16
Device Configuration (HTTP).....	16
Notes For AV Integrators .....	18
Fully-qualified room names.....	18
Current Product Group .....	18
Known Issues.....	18

## LIST OF TABLES

Table 1 Request/Control Messages .....	9
Table 2 Response/Feedback Messages .....	11
Table 3 Warning Messages.....	12
Table 4 Channel Feedback.....	14
Table 5 Level Feedback.....	15

## **Revision History**

Date	Version	Initials	Comments
2014-01-23	3.3.0	RM	Added Recording Progress level. Added PRESENTERS? command and feedback. Added support for the ScheduleAPI ?filter parameter introduced in ESS v5.3. Fixed issue when ESS reports multiple (historical) appliances for a given room. Fixed appliance persistent socket code. Improved appliance health monitoring and error reporting. Removed WARINING=None message. Removed CAPTURE PROFILES terminology.
2013-09-10	3.2.2	RM	Added optional port for Server. <del>Fixed bug with persistent socket code for appliances, if/when Echo360 chose to support persistent sockets on appliances.</del>
2013-01-30	3.2.1	RM	<i>Not released to public.</i>
2012-11-14	3.2.0	RM	Added CAMPUSES? and BUILDINGS? commands for wizard-style configuration. Added various PRODUCT GROUP commands. Added WAITING channel for pre-record state. Changed TITLE behaviour to start of recording only. Fixed various character encoding bugs. Deprecated CAPTURE PROFILE terminology. Removed RECORDING INPUTS feedback.
2012-09-14	3.1.1	RM	Fixed bug with monitoring Capture Appliance (Gen1) general health.
2012-09-14	3.1.0	RM	Added LIVE channel feedback. Fixed various OAuth bugs. Fixed channel feedback for recording types.
2011-11-25	3.0.0	RM	Initial release. Supersedes all previous Lectopia v1 & v2 modules. Support for Lectopia is discontinued.

## **Introduction**

This is a reference manual for anyone wanting to integrate an AMX NetLinX master with an Echo360 Capture Appliance. Such integrations require appropriate configuration by both the AMX integrator and EchoSystem administrator.

A typical EchoSystem deployment consists of an EchoSystem Server (ESS) and multiple capture solutions. Whilst EchoSystem supports both software and hardware capture solutions, this module is designed for use with the hardware solutions only. At the time of writing, two hardware models exist: Capture Appliance and SafeCapture HD. Unless specified, this document shall refer to both models as CA.

The module can operate in one of two ways:

- 1 Server-based configuration. In this mode, the module contacts the ESS to retrieve hardware details for a given room. It then uses the provided details to control the capture hardware. This approach can provide sufficient flexibility for a fully DHCP-deployed fleet of capture hardware.
- 2 Stand-alone configuration. In this mode, the module must be configured with the specific details for the capture hardware, including credentials and IP address. This approach requires the capture hardware to have a predictable IP address, either through static IPs or DHCP-controlled/reserved IPs.

Regardless of which way the module is configured, the module then attempts to monitor and control the associated capture hardware. This includes:

- Current recording status of the capture hardware
- Pause/Resume/Extend/Stop control of the current recording
- Time remaining and percentage complete
- Presence of AV signals at hardware inputs
- Background health monitoring of capture hardware (every minute if device-polling is enabled)

Communication between the NetLinX master and EchoSystem products are via IP sockets. The module communicates using HTTP, not HTTPS. ESS must be configured correctly to allow requests from the NetLinX via HTTP. Credentials between the NetLinX and ESS are securely hashed using a shared secret to prevent replay attacks on the ESS. Additional network configurations may also be required to suit the deployed network environment, and are beyond the scope of this document.

The module has been tested with the following components:

Component	Application/Firmware recommended minimum version
NetLinX firmware	3.60.453
EchoSystem Server	5.3.2013-10-28.1
Capture Appliance (Gen 1)	4.0.25278
SafeCapture HD	5.3.38179

## **AMX Implementation**

The module utilises two IP connections – one for connecting to the ESS and another one for connecting to the CA. The module also requires a virtual device to embody the EchoSystem behaviour of both the ESS & CA. The AMX integrator performs all configuration, control and monitoring via the virtual device. While the module was developed on a NetLinX master with Duet firmware, the module itself does not require Duet.

To use the module, the AMX integrator must do the following:

1. Define a virtual device for your program to interact with. NetLinX virtual devices must have a device number in the 32,768 – 36,863 range. *The device number must be unique across the entire AMX workspace.*
2. Define two IP-port devices that the above virtual device will use, one for the ESS and one for the CA. *IP-port devices must be unique across the entire AMX workspace.* Ensure that they do not conflict with ports used in other IP-based modules (RMS, i!-EquipmentMonitor, etc.). It is recommended to use the lowest available ports, since the AMX will reserve memory for *all* ports up to the highest IP port number.
3. Copy the EchoSystem.tko file into the same directory as the main program .axs file or into the shared TKOs folder.
4. Use the DEFINE\_MODULE command to include the EchoSystem module in the program. The module requires: the virtual device, the CA IP-port device and the ESS IP-port device.
5. Initialise the virtual device in the virtual device online event.

An example of how to do this is shown below.

```

DEFINE_DEVICE
    vdvEcho          = 33001:1:0
    vdvEchoCAPort    = 0:2:0    // IP-port for CA
    vdvEchoESSPort   = 0:3:0    // IP-port for ESS

// Put this between DEFINE_START and DEFINE_EVENT sections
DEFINE_MODULE 'EchoSystem' modEcho(vdvEcho, vdvEchoCAPort, vdvEchoESSPort)

DEFINE_EVENT
DATA_EVENT[vdvEcho]{
    online:{
        wait 50{ //Delay recommended to allow CPU to get through reboot load first.
            send_command vdvEcho, 'SERVER=ess.my.server.edu'
            send_command vdvEcho, 'SERVER CONSUMER KEY=key'
            send_command vdvEcho, 'SERVER CONSUMER SECRET=krn8HwqesXMskjhweg...tIQ1w=='
            send_command vdvEcho, 'SERVER REFRESH=60' //Every hour
            send_command vdvEcho, 'ROOM=myRoom'
            send_command vdvEcho, 'COUNTDOWN=2' //Last 2 minutes
            send_command vdvEcho, 'DEVICE REFRESH=5' //Every 5 seconds
        }
    }
}

```

## Control Messages

All control messages are sent to the module via `send_command vdvEcho, {command}`.

The following commands can be sent to the module. Notes:

- Where numeric values are sent, they should be ASCII strings, not integer values.
- Optional filters are shown in [square brackets].
- Parameters are case sensitive, but commands and filters are not.
- Space on either side of the '=' is automatically removed.

E.g. `send_command vdvEcho, 'ServEr rOoM= myRoom' //Legitimate for myRoom`

Command	Description
ROOM=<value>	Set the room name. This must either match the room name in ESS, or be in the fully-qualified format campus:building:room.  <code>send_command vdvEcho, 'ROOM=myRoom'</code> <code>send_command vdvEcho, 'ROOM=myCampus:myBuilding:myRoom'</code>
ROOM?	Get the room name that the module is using for server-based config and room name cross-checking.  <code>send_command vdvEcho, 'ROOM?'</code>
ROOMS?[filter]	Get a list of rooms from the ESS, with optional filter. When specified, the results are filtered for rooms starting with [filter].  <code>send_command vdvEcho, 'ROOMS?'</code> <code>send_command vdvEcho, 'ROOMS?myRoom'</code>
BUILDINGS?[filter]	Get a list of buildings from the ESS, with optional filter. When specified, the results are filtered for an exact match with [filter].  <code>send_command vdvEcho, 'BUILDINGS?'</code> <code>send_command vdvEcho, 'BUILDINGS?myBuilding'</code>
CAMPUSES?[filter]	Get a list of campuses from the ESS, with optional filter. When specified, the results are filtered for an exact match with [filter].  <code>send_command vdvEcho, 'CAMPUSES?'</code> <code>send_command vdvEcho, 'CAMPUSES?myCampus'</code>
BUILDINGS ON CAMPUS?<campus>	Get a list of buildings for the specified campus.  <code>send_command vdvEcho, 'BUILDINGS ON CAMPUS?myCampus'</code>
ROOMS IN BUILDING ?<campus>:<building>	Get a list of rooms for the specified building. N.B. The campus must be specified.  <code>send_command vdvEcho, 'ROOMS IN BUILDING?myCampus:myBuilding'</code>
SERVER=<address>[:port]	Set the ESS address. Can be a DNS entry or an IP. (Max 128 char) Only specify the port if you are using non-standard ports.  <code>send_command vdvEcho, 'SERVER=ess.my.server.edu'</code>
SERVER?	Get the server address specified in the module.  <code>send_command vdvEcho, 'SERVER?'</code>
SERVER REFRESH=<minutes>	Set the server poll time. Time is in minutes. Default = 0 min.  <code>send_command vdvEcho, 'SERVER REFRESH=1440' //daily</code>

SERVER REFRESH?	Get the server poll time.  send_command vdvEcho, 'SERVER REFRESH?'
SERVER REFRESH	Get room details from the server immediately.  send_command vdvEcho, 'SERVER REFRESH'
SERVER CONSUMER KEY=<key>	Set the server consumer key. (Max 100 chars)  send_command vdvEcho, 'SERVER CONSUMER KEY=myKey'
SERVER CONSUMER KEY?	Get the server consumer key specified in the module.  send_command vdvEcho, 'SERVER CONSUMER KEY?'
SERVER CONSUMER SECRET=<secret>	Set the server consumer secret. (Max 100 chars)  send_command vdvEcho, 'SERVER CONSUMER SECRET=kkjsOIUY7ykjhsgd/skjhw/hjsy=='
SERVER CONSUMER SECRET?	Get the server consumer secret specified in the module.  send_command vdvEcho, 'SERVER CONSUMER SECRET?'
SERVER PRODUCT GROUPS?	Get a list of Product Groups from the server. <i>Each Product Group will generate a separate feedback string event.</i>  send_command vdvEcho, 'SERVER PRODUCT GROUPS?'
DEVICE=<address>	Set the IP address of the device. <i>This is only required if you are deploying the system <b>without</b> auto-configure via the ESS. Responses from the ESS will overwrite this.</i>  send_command vdvEcho, 'DEVICE=192.168.0.101'
DEVICE?	Get the CA IP address.  send_command vdvEcho, 'DEVICE?'
DEVICE USERNAME=<value>	Set the CA username. Default credentials are used in absence of specific credentials.  send_command vdvEcho, 'DEVICE USERNAME=myUser'
DEVICE USERNAME?	Get the CA username.  send_command vdvEcho, 'DEVICE USERNAME?'
DEVICE PASSWORD=<value>	Get the CA password. Default credentials are used in absence of specific credentials.  send_command vdvEcho, 'DEVICE PASSWORD=myPassword'
DEVICE REFRESH=<seconds>	Set the CA poll time. Time is in seconds. Default = 0.  send_command vdvEcho, 'DEVICE REFRESH=5' //5 sec
DEVICE REFRESH?	Get the CA poll time.  send_command vdvEcho, 'DEVICE REFRESH?'
DEVICE PRODUCT GROUPS?	Get a list of the supported Product Groups from the CA. <i>Each Product Group will generate a separate feedback string event.</i>  send_command vdvEcho, 'DEVICE PRODUCT GROUPS?'
CURRENT PRODUCT GROUP?	Get the Product Group being used for the current recording.  send_command vdvEcho, 'CURRENT PRODUCT GROUP?'

DEVICE TYPE?	Get the CA hardware type.  send_command vdvEcho, 'DEVICE TYPE?'
DEVICE SERIAL?	Get the CA serial.  send_command vdvEcho, 'DEVICE SERIAL?'
DEVICE TEMPERATURE?	Get the CA temperature.  send_command vdvEcho, 'DEVICE TEMPERATURE?'
DEVICE FIRMWARE?	Get the CA firmware version.  send_command vdvEcho, 'DEVICE FIRMWARE?'
DEVICE REBOOT	Reboot the CA.  send_command vdvEcho, 'DEVICE REBOOT'
COUNTDOWN=<minutes>	Set the countdown threshold in minutes. When the recording reaches this threshold, the module reports the remaining time in hh:mm:ss every second. The module maintains an internal timer to generate the TIME= strings, and re-syncs with the CA on every poll.  send_command vdvEcho, 'COUNTDOWN=5' //5 min
COUNTDOWN?	Get the countdown threshold in the module.  send_command vdvEcho, 'COUNTDOWN?'
TITLE?	Get the title of the current recording.  send_command vdvEcho, 'TITLE?'
PRESENTERS?	Get the presenters for the current recording.  send_command vdvEcho, 'PRESENTERS?'
STOP	Stop the current recording. The recording may be running, paused or waiting to start.  send_command vdvEcho, 'STOP'
PAUSE	Pause the current recording. Recording must already be running.  send_command vdvEcho, 'PAUSE'
RESUME	Resume a paused recording.  send_command vdvEcho, 'RESUME'
EXTEND=<minutes>	Extend the current recording by the specified minutes. If =<minutes> is not specified, 5 minutes is assumed. <i>Maximum &lt;minutes&gt; is 30. Values greater than 30 are rounded down to 30.</i>  send_command vdvEcho, 'EXTEND=10' //10 min send_command vdvEcho, 'EXTEND' //5 min
AD HOC=<decription>;<minutes>;<product group>	Create an Ad Hoc recording. Product Groups can be obtained from the CA using the DEVICE PRODUCT GROUPS? command.  send_command vdvEcho, 'AD HOC=Test recording;10; Audio Only (Podcast). Balanced between file size & quality'
DEBUG=<level>	Set the debug level. When on, debug messages are sent to vdvEcho and to device 0. In a telnet session these can be seen by entering the 'mgs on' command. Default is 0 (off).  send_command vdvEcho, 'DEBUG=1'



DEBUG?	Get the current debug level.  send_command vdvEcho, 'DEBUG?'
VERSION?	Get the module version.  send_command vdvEcho, 'VERSION?'
INFO?	Get developer contact information.  send_command vdvEcho, 'INFO?'

**Table 1 Request/Control Messages**

## String Feedback Messages

The module provides feedback from the ESS & CA by sending string events to vdvEcho. This is not an exhaustive list of responses from the module. There are additional “debug” responses to assist with fault-finding.

<i>String</i>	<i>Description</i>
ROOM=	Feedback on the ROOM? command. The module will use this name for server-based config and room name cross-checking.  ROOM=myRoom ROOM=myCampus:myBuilding:myRoom
ROOMS=	Feedback on the ROOMS? and ROOMS IN BUILDING? commands. The response contains a list of room names, separated by a semicolon.  ROOMS=Room 1;myRoom;Big theatre;Room 3
BUILDINGS=	Feedback on the BUILDINGS? and BUILDINGS ON CAMPUS? commands. The response contains a list of building names, separated by a semicolon.  BUILDINGS=Building 1;myBuilding;Big building
CAMPUSES=	Feedback on the CAMPUSES? command. The response contains a list of campus names, separated by a semicolon.  ROOMS=Main Campus;Remote campus;CubeSat
SERVER=	Feedback on the SERVER? command.  SERVER=my.server.edu.au
SERVER REFRESH=	Feedback on the SERVER REFRESH? command. Time is in minutes  SERVER REFRSH=60 minutes
SERVER CONSUMER KEY=	Feedback on the SERVER CONSUMER KEY? command.  SERVER CONSUMER KEY=myKey
SERVER CONSUMER SECRET=	Feedback on the SERVER CONSUMER SECRET? command.  SERVER CONSUMER SECRET= kkjsOIUY7ykjhsdg/skjhw/hjsy==
SERVER PRODUCT GROUP=	Feedback on the SERVER PRODUCT GROUPS? command. <i>A separate string event will occur for each Product Group.</i>  SERVER PRODUCT GROUP=Audio Only (Podcast). Balanced between file size & quality
DEVICE=	Feedback on the DEVICE? command.  DEVICE=192.168.0.34
DEVICE USERNAME=	Feedback on the DEVICE USERNAME? command.  DEVICE USERNAME=myUser
DEVICE REFRESH=	Feedback on the DEVICE REFRESH? command. Time is in seconds.  DEVICE REFRESH=5 seconds

DEVICE PRODUCT GROUP=	<p>Feedback on the DEVICE PRODUCT GROUP? command. <i>A separate string event will occur for each supported Product Group.</i></p> <p>DEVICE PRODUCT GROUP=Audio Only (Podcast). Balanced between file size &amp; quality</p>
CURRENT PRODUCT GROUP=	<p>Feedback on the CURRENT PRODUT GROUP? command. Also generated when a new recording starts.</p> <p>CURRENT PRODUCT GROUP=Audio Only (Podcast). Balanced between file size &amp; quality</p>
DEVICE TYPE=	<p>Feedback on the DEVICE TYPE? command.</p> <p>DEVICE TYPE=SafeCapture HD</p>
DEVICE SERIAL=	<p>Feedback on the DEVICE SERIAL? command.</p> <p>DEVICE SERIAL=AA-BB-CC-DD-EE-FF</p>
DEVICE TEMPERATURE=	<p>Feedback on the DEVICE TEMPERATURE? command. Also generated when temperature change is detected in background health monitoring.</p> <p>DEVICE TEMPERATURE=53.0C</p>
DEVICE FIRMWARE=	<p>Feedback on the DEVICE FIRMWARE? command.</p> <p>DEVICE FIRMWARE=v5.4.38811</p>
COUNTDOWN=	<p>Feedback on the COUNTDOWN? command. Time is in minutes.</p> <p>DIGI COUNTDOWN=5 minutes</p>
TIME=	<p>When the module detects that the time remaining is less than COUNTDOWN time, the module will generate a string each second to show the remaining time. If the time reaches 0, the recording is stopped early, or the time is extended beyond the threshold, an additional 'TIME=' string is sent to assist with clearing the feedback. Time is in hh:mm:ss.</p> <p>TIME=00:04:30</p>
TITLE=	<p>Feedback on the TITLE? command. Also generated when a new recording starts. When the recording ends, an additional 'TITLE=' string is sent to assist with clearing the feedback.</p> <p>TITLE=AMX Programming 101 TITLE=Confidence Monitoring <i>*while monitoring*</i></p>
PRESENTERS=	<p>Feedback on the PRESENTERS? command. Also generated when a new recording starts. When the recording ends, an additional 'PRESENTERS=' string is sent to assist with clearing the feedback.</p> <p>PRESENTERS=Anne;Bill;Charles</p>
WARNING=	<p>A possible issue has been detected with the capture hardware. See "Warning Messages" table below for some possible issues.</p>
DEBUG=	<p>Feedback on the DEBUG? command.</p> <p>DEBUG=1</p>
VERSION=	<p>Feedback on the VERSION? command.</p> <p>VERSION=v3.2.0</p>
INFO=	<p>Feedback on the INFO? command. Multi-line reply</p>

**Table 2 Response/Feedback Messages**

## String Feedback (Warning) Messages

The string feedback WARNING=<message> is used to provide a variety of warnings regarding the system configuration, health, and responses from the ESS. The following warnings may be of particular interest for managing/monitoring the system deployment.

<i>Warning message</i>	<i>Description</i>
Hardware room name <room> does not match module room name <room>.	The response from the CA contains the room name. This warning is generated when this room name differs from the room name supplied to the module. It is possible that the AMX is connecting to the wrong CA due to IP changes with the CA.
Room <room> not found.	The specified room name was not found in the ESS.
Multiple matches for Room <room> found.	The specified room name is not unique within the ESS deployment. You will have to use the campus:building:room format instead.
Room <room> has no associated devices.	The room exists in ESS, but no capture hardware has been assigned to it.
The device <serial> does not yet have an IP address.	The device serial has been registered with ESS, but the hardware has not "checked in" to ESS. ESS does not know the device IP address, if any.
Can not control <xyz> devices.	The module will only work with Capture Appliance (Gen 1) and SafeCapture HD (Gen 2) devices.
Device credentials are not authorized.	The Username and/or Password for the device are incorrect.
No campuses found for <filter>.	There are no campuses matching the filter in ESS.
No campuses found.	No campuses have been created in ESS.
No buildings found for <filter>.	There are no buildings matching the filter in ESS.
No buildings found.	No buildings have been created in ESS.
No rooms found for <filter>.	There are no rooms matching the filter in ESS.
No rooms found.	No rooms have been created in ESS.
No campus:building specified.	The query requires an appropriate campus:building filter.
No product groups found.	The ESS does not have any Product Groups specified.
Device rebooting. Please standby...	The reboot command has been issued to the device from the AMX.
Fan <fan #> rpm: <speed>	The fan speed has dropped below 1000 rpm. Possible fan stall. SafeCapture HD hardware only.
Possible disc mount failure for /data (<capacity>)	The data volume on the CA may not have mounted correctly. The volume capacity for /data partition appears unusually small (< 10Gb). This could result in recording failures.
I2C bus lockup	I2C bus on appliance has locked up, requiring hard power cycle.
Possible clock drift on appliance.	Comms between the appliance and ESS have timed out. Typical cause is clock drift > 60 seconds between the two.

**Table 3 Warning Messages**

In the case of the room name mismatch, you can configure the module to re-query the ESS to obtain the correct hardware details for your room. E.g.

```
DATA_EVENT[vdvEcho]{
  string:{
    if(find_string(data.text,'WARNING=',1) == 1){ //We have a warning
      if(find_string(data.text,'Hardware room name',1) &&
        find_string(data.text,'does not match module room name',1)){
        send_string 0,'Room name mismatch - Refreshing from ESS'
        send_command vdvEcho,'SERVER REFRESH'
      }
    }
  }
}
```

Tip: When using the module in a stand-alone configuration, the above warning can be suppressed by configuring the module with the ROOM=<value> command.

## Debugging with the module

When debugging mode is enabled for the module, all string messages from the module are also sent to device 0. These can be viewed via a telnet/ssh session to the AMX master:

- Open a telnet session to the AMX master.
- Type `send_command vdvEcho,'DEBUG=1'`
- Type `msg on`

You can issue direct commands to the module from the command line:

- Open a telnet session to the AMX master.
- Type `msg on`
- Type `send_command vdvEcho,'SERVER PRODUCT GROUPS?'`

Debug messages will still be generated unless you turn them off. To turn off the debug messages:

- Open a telnet session to the AMX master.
- Type `send_command vdvEcho,'DEBUG=0'`

## Channel Feedback

The module provides feedback for the CA via channel events to vdvEcho.

Name	Virtual Device Channel	Description
Pre-record	240	Waiting (about to start)
Recording state	241	Idle
	242	Paused
	243	Recording
	245	Scheduled
Recording type	246	Ad Hoc
	247	Monitoring
	249	Live Streaming
AV signal detection	251	Audio
	252	Video 1 sync
	253	Video 2 sync
Comms	254	Comms to CA is okay.

**Table 4 Channel Feedback**

If the Comms channel goes off, all other channels are turned off since the AMX module can not determine the actual state of the capture platform. Typical issues that can cause the Comms channel to turn off are:

- AMX unable to reach capture platform via network. (This can be tested using the ping command from an AMX telnet session.)
- Network failure.
- Hardware failure.

A recording can be started in one of three possible ways: Scheduled, Ad Hoc, or via the Monitoring tab on the CA web GUI. The module can detect which way the recording started, which can be used to customise the feedback for the in-room users. E.g.

```
CHANNEL_EVENT[vdvEcho,243]{ //Recording
  on:{
    //Turn OnAir light ON for Scheduled or Ad Hoc recordings, but not Monitoring.
    if([vdvEcho,245] || [vdvEcho,246])
      on[dvRelay,rlOnAirLight]
  }
  off:{
    off[dvRelay,rlOnAirLight]
  }
}
```

Be aware that the LEDs on the front panel of the CA will still illuminate, regardless of the in-room feedback that the AMX provides to the users. If this is undesirable, thought should be given to hiding the front panel.

If the current recording is being Live Streamed, the Live channel will turn on.

## Level Feedback

The module provides level feedback for a number of hardware parameters. Audio and recording progress levels are updated every DEVICE REFRESH period. Other levels are updated every HEALTH CHECK period (1 minute if device polling is enabled).

*Each hardware model has a unique "effective audio range", which is a nominal range between the noise-floor of the hardware and the point at which distortion will occur. The module scales the input level so that signals within the effective audio range are presented as levels between 0 – 255, with an exponential-to-linear conversion for visual feedback.*

Name	Virtual Device Level	Description	Range
Left	1	Left audio level	0 – 255 SafeCapture HD = -40dBu to +0dBu Capture Appliance = -40dBu to -5dBu
Right	2	Right audio level	0 – 255 SafeCapture HD = -40dBu to +0dBu Capture Appliance = -40dBu to -5dBu
Recording progress	3	% of current recording complete	0 – 100
Temperature	5	Temperature in °C	0 – 255 0°C to 255 °C
/data capacity	6	Capacity used on /data partition	0 – 100 0 = 0% used, 100 = 100% used

**Table 5 Level Feedback**

## **EchoSystem Server Configuration**

For improved efficiency, the NetLinx master uses persistent connections to the ESS. As such, the ESS web server must support HTTP/1.1 persistent connections.

The NetLinx master communicates with the ESS via the Schedule API over HTTP. All requests to the ESS are authenticated using OAuth in a “pre-shared key” manner. A common secret is stored on both the ESS and the NetLinx master. The secret is used as part of the authentication process for every request to the ESS Scheduled API. The secret is not transmitted as clear text, but is used to generate a unique hash signature for each request. The end result is that only authorised clients (NetLinx masters, etc.) can place successful requests with the ESS.

## **Trusted Systems**

The ESS manages access to the Schedule API in the Trusted Systems section. There is no restriction on how many systems can connect using a single set of Trusted System details. You may wish to create a separate trusted system per campus or use one for the entire institute. You may also wish to create a separate trusted system for development purposes, which can later be disabled/modified to prevent access once the development phase is complete. At the time of writing, ESS Trusted Systems have full access to all commands in the Schedule API.

To configure the ESS appropriately, the ESS administrator must do the following:

1. Log into ESS and navigate to the System->Trusted Systems tab.
2. Add a new Trusted System. Enter a System Name and Description for the trusted system. The Consumer Key will need to be provided to the AMX integrator
3. Save the new trusted system.
4. Provide the Consumer Key and Consumer Secret to the AMX integrator. They will need to configure the module with these details.
5. The AMX integrator will also require the ESS base URL (E.g. `ess.my.institution.edu`) and relevant Room Names.

## **Device Configuration (HTTP)**

The module is written in the native AMX NetLinx language. As such, it only supports HTTP sockets, not HTTPS. The ESS must be configured to allow REST API calls over HTTP for the CA. This can either be done at a global level, or on a per device level. N.B. The ESS maintains separate settings on a per-device-type basis. At the time of writing there are three device types: Capture Appliance, SafeCapture HD and Classroom Capture (software). The module has been tested with the Capture Appliance and SafeCapture HD only.

To configure the ESS globally, the ESS administrator must do the following:



1. Log into ESS and navigate to the System->Device Defaults tab.
2. Click the Edit button to modify the settings.
3. For each hardware type (Capture Appliance and SafeCapture HD) set the following:
  - a. Access Protocol = HTTP
  - b. Port = 8080
  - c. Admin Username/Password = admin/password \*
4. Save the changes to the settings.

*\* If the credentials are changed from the default values of "admin" and "password", each NetLinx master must also provide the appropriate credentials to the EchoSystem module.*

Alternatively to configure an individual CA, the ESS administrator must do the following:

1. Log into ESS and navigate to the Configuration->Devices tab.
2. Hover the mouse over the appropriate device and click the "edit" option.
3. Set the following Local Web User Interface settings:
  - a. Access Protocol = HTTP
  - b. Port = 8080
4. Save the changes to the settings.

## **Notes For AV Integrators**

### **Fully-qualified room names**

EchoSystem does not require room names to be unique across the institute. They only need to be unique within a building. This also applies to building names within a campus. If the deployment has two rooms with the same name, we are not able to resolve the room/CA using just the room name. In such cases we should use the fully-qualified name in the format campus:building:room. The demonstration code in the .zip bundle shows how to do this.

### **Current Product Group**

CA API versions prior to v3.0 provided basic information regarding the current capture, and whether it utilised video (Composite/S-video) as part of the current capture. This may have been used to turn on additional lighting to improve the quality of camera footage. As of CA API v3.0 (approx CA firmware v4.x.x) this is no longer the case. Instead we are provided with the Product Group for the current recording. The module reports the current Product Group at the start of a recording. You *may* be able to use this to determine whether the recording utilises a camera, and control spot/flood lights appropriately, however Product Groups can be renamed by the EchoSystem Admin. If you require automatic control of spot/flood lights you should discuss a Product Group naming convention with your EchoSystem Admin. The demonstration code in the .zip bundle shows how to do this without relying on a keyword in each Product Group.

## **Known Issues**

*Latest known issues:* <http://opax.swin.edu.au/~romclean/amx/echo>

- Some early versions of AMX v4 firmware appear to have a faulty IP stack, which causes all IP-based comms to fail. Ensure you are using either v3 firmware or the latest v4 firmware.
- ESS Schedule API does not currently allow "chunked" requests. (I.e. Retrieve X results at a time.) Buffers within the module can store approx 60 results. If you expect more results than this, consider using the [filter] options and/or use the CAMPUS? and BUILDINGS ON CAMPUS? commands to reduce quantity of results.
- Initial post-boot load on the AMX CPU may cause IP sockets to not respond favourably. Be kind to the AMX CPU by waiting at least 5 seconds after vdvEcho is online before configuring it. Refer to AMX Implementation above for code sample.