

Kinesys Media Server Packet Format

K2 Versions Supported

This data format is valid for K2 V1.2.xxx and above. K2 V1.1.xxx does not support Media servers. Please contact Kinesys about upgrading if you are using K2 V1.1.xxx

K2 V1.3.xxx and above has a protocol version of 0x0001 and specifies degrees to 2 decimal places. K2 V1.2.219 has a protocol version of 0x0000 and specifies degrees to 0 decimal places.

Vector Versions Supported

Vector V2.1.0.54 and later supports Media servers. Vector implements the later protocol version 0x0001 by default, although this can be changed in the Vector.ini file.

Vector V2.1.0.85 and later supports a reduced packet format where only one axis is transmitted per construct, meaning that the number of channels supported has been increased from 24 to 144

Data Format

Data is sent in UDP format to a multicast address that can be changed as required. The data will be coming from a dedicated network card that can be set to any appropriate IP address and subnet mask.

Multicast Address

Any valid multicast address can be used for protocol transmission. This is set in the Network Setup window within K2, or the Vector.ini file in Vector. The default address is 239.195.0.1. This address is by default a Multicast address, but can be changed to the IP address of the media server if direct point to point communication is required.

Port Number

Kinesys use port 6061 for Media Server Packet transmissions

Update Rate

In K2 multicast packets are sent every K2 Tick, so this depends of the tick rate set in the show. By default this is every 100ms. In Vector multicast packets are always sent every 100ms.

Ethernet Ports

Kinesys strongly recommend using a separate Ethernet port for the media server output, on the computer running K2 or Vector. This is to keep the Automation network free of Media traffic. USB to Ethernet adapters would work for this application.

Overall Packet Format [Header size 30 bytes]

Field	Byte Offset	Name	Size	Description
1	0	ID [12]	Byte	Fixed array of 12 characters terminated by a null 'K' 'I' 'N' 'E' 'S' 'Y' 'S' ' ' 'N' 'E' 'T' 0x00
2	12	Opcode	2 byte	Multicast = 0x0003
3	14	ProtVer	2 byte	Protocol Version 0x0000 = Degree's are specified to 0 decimal places 0x0001 = Degree's are specified to 2 decimal places 0x0002 = Only one Axis parameter per construct sent (Vector option only) Current Value = 0x0001
4	16	MinProtVer	2 byte	Minimum Protocol Version compatible node that can receive this packet Current Value = 0x0001
5	18	FrameID	2 byte	Frame identifier, incremented for every packet of messages sent. (rolls over to 0x0000 after 0xFFFF). Note the Frame ID is sent in Little Endian Format. Least significant byte first
6	20	NumDataMsgs	Byte	The number of communication messages contained in the data area (complete messages – not bytes)
7	21	Pad	Byte	Transmit as zero, receivers must not test
8	22	Spare [8]	Byte	Transmit as zero, receivers must not test
9	30	Data [#]	Byte	Data packets, one after the other

All data is sent in **big** endian format (**most** significant byte first) except the FrameID.

Individual Message Format [Message size 10 bytes]

Field	Byte Offset	Name	Size	Description
1	0	AxisID	2 byte	Unique axis ID K2 This made from Construct ID and Parameter ID = $((\text{Construct ID} * 100) + \text{Parameter ID})$ Parameter ID limited to 0, 1, 2, 3, 4, 5 = X, Y, Z, P, T, R Vector This is made from Row number and Media setting $=((\text{Row number} * 100) + \text{Media setting})$ Parameter ID limited to 0, 1, 2, 3, 4, 5 = X, Y, Z, P, T, R
2	2	Position/Value	4 byte	Signed current position or value In mm or degrees to two decimal places
3	6	Speed	2 byte	+/- 32,000 for mm/s +/- 32,000 for degrees/s to two decimal places (i.e. 3000 = 30 degree/s)
4	8	Errors	2 byte	Construct error condition 0 = No errors 0 <> Construct in Error state

All data is sent in **big** endian format (**most** significant byte first).

X = horizontal movement in the X direction

Y = horizontal movement in the Y direction

Z = horizontal movement in the Z direction

P = PITCH – angular movement around the X axis

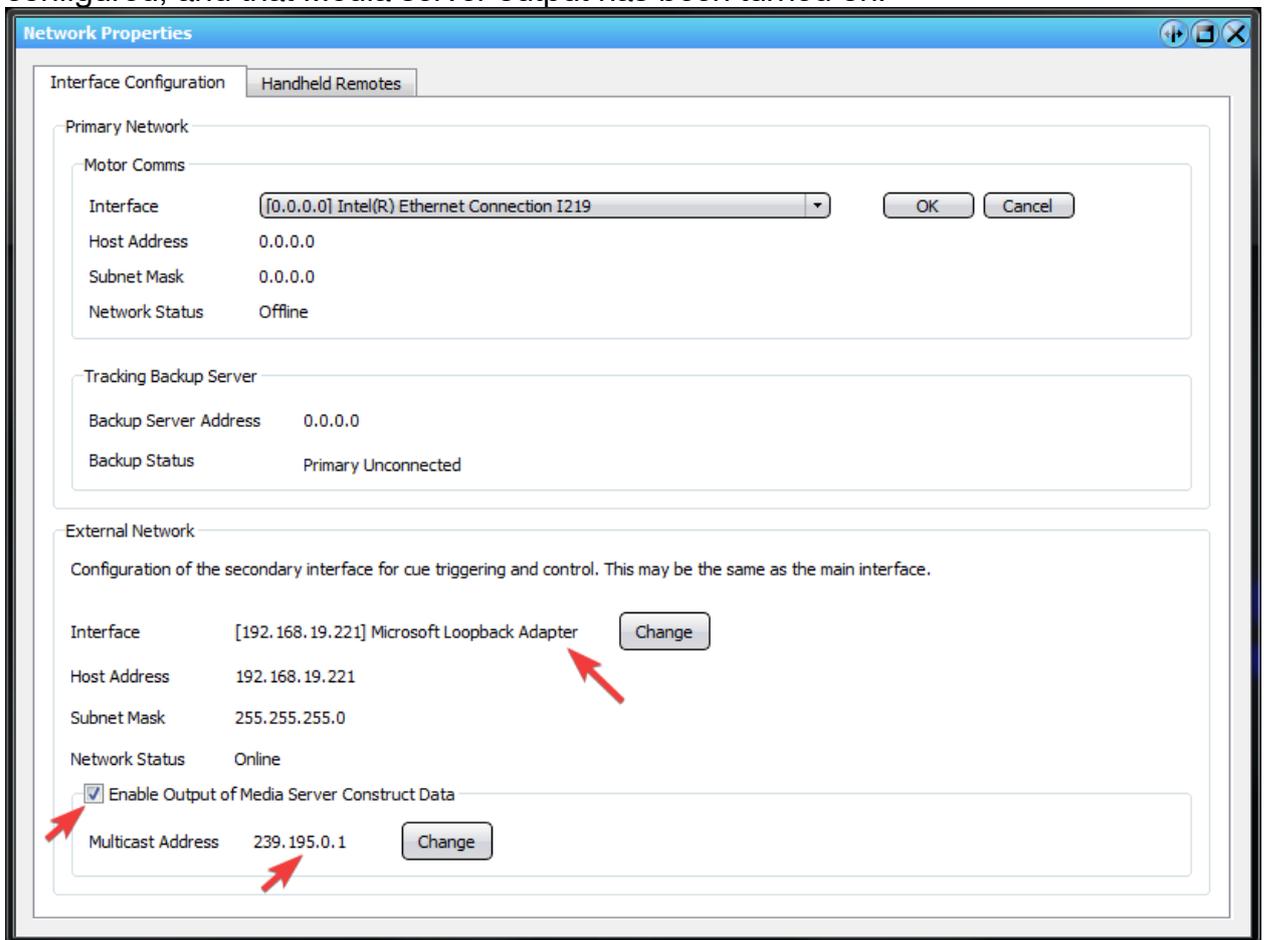
T = TILT – angular movement around the Y axis

R = ROTATE – angular movement around the Z axis

Details on using K2 for media output

In K2 each Multicast activated construct will send all six axis parameters, even if they are not all moving. Each activated construct will therefore use 60 bytes (6 packets of 10 bytes each). The maximum size of a transmitted UDP packet is 1500 bytes. So K2 can output multicast data for up to 24 constructs per UDP packet. K2 versions V1.4 and above support sending multiple UDP packets, so there is no limit to the number constructs that can have data media server data output.

To Enable Media server output in K2, make sure that an external network has been configured, and that Media server output has been turned on.



For every parameter of each construct you can easily see its associated Media Server ID as shown in the screen shot below.

Edit T1

- ✓ Identification
- ✓ Origin
- ✓ Safety
- ✓ Z parameter
- ✓ Pitch parameter
- ✓ Tilt parameter
- ✓ Model
- ✓ Drop Links

Construct Parameter

Value

Minimum	Min Flown Height	...	= 0mm
Maximum	13000	...	= 13000mm
Initial	Fly-Out Height	...	= 13000mm

Speed

Minimum	0	...	= 0mm/s
Maximum	400	...	= 400mm/s
Default	100	...	= 100mm/s

Acceleration

Minimum	0	...	= 0mm/s/s
Maximum	1500	...	= 1500mm/s/s
Default	100	...	= 100mm/s/s

Deceleration

Copy Accel to Decel

Minimum	0	...	= 0mm/s/s
Maximum	1500	...	= 1500mm/s/s
Default	100	...	= 100mm/s/s
Error	1500	...	= 1500mm/s/s

Display and Calculation

Edit value as percent Invert percent calc (min 100%, max 0%) Default Table Column 1

DMX Input

Enable DMX Input Treat Input As Signed Start Channel Address 1

Media Server ID

Parameter Address	102
-------------------	-----

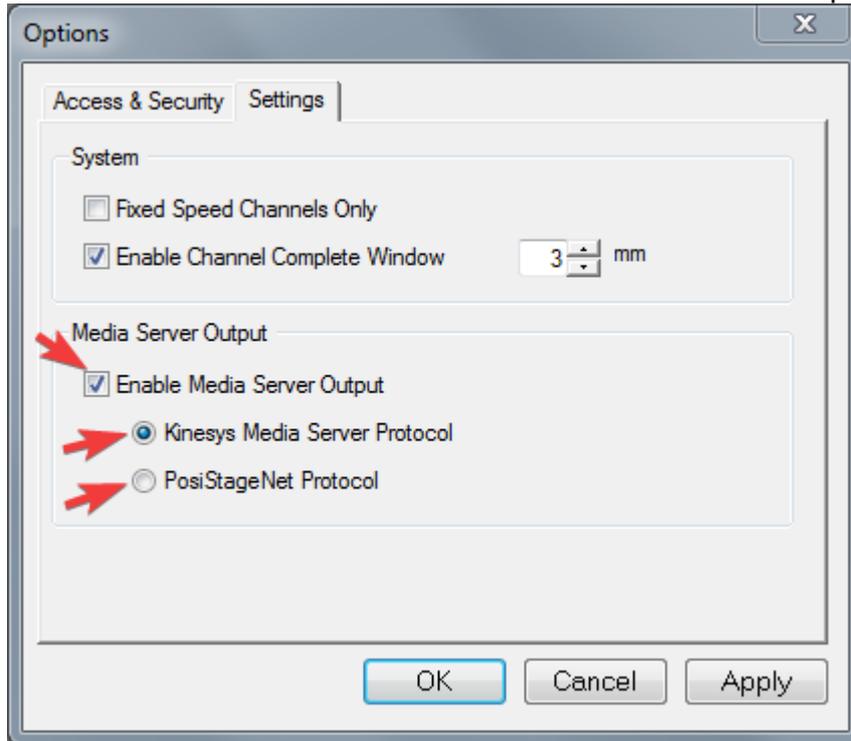
Previous Next Show Errors Ok Cancel

Details on using Vector for media output

Vector version V2.1.87 and above supports outputting media data in two formats.

1. Kinesys UDP format as described in this document
2. PosiStageNet format as described here <http://www.posistage.net/>

The choice of media server format is set under Tools > Options



Vector.ini settings for Media Server output

The Vector.ini file located here “C:\ProgramData\Kinesys\Vector” contains a number of settings related to Media server output, the options are described in the ini file. If any changes are made to the ini file Vector must be restarted to use them.

```

; Media Server Multicast IP = an example is 239.195.0.1
; Can set normal IP address if need send to just one machine
Media Server Multicast IP = 239.195.0.1
Kinesys Media Output=True
Use PosiStageNet Protocol=False

; Media Protocol Versions
; 0 = Original K2 (Degrees specified to 1 degree
; 1 = New K2 (Degrees specified to 100th Degrees
; 2 = Special Vector version, only one axis per construct output

```

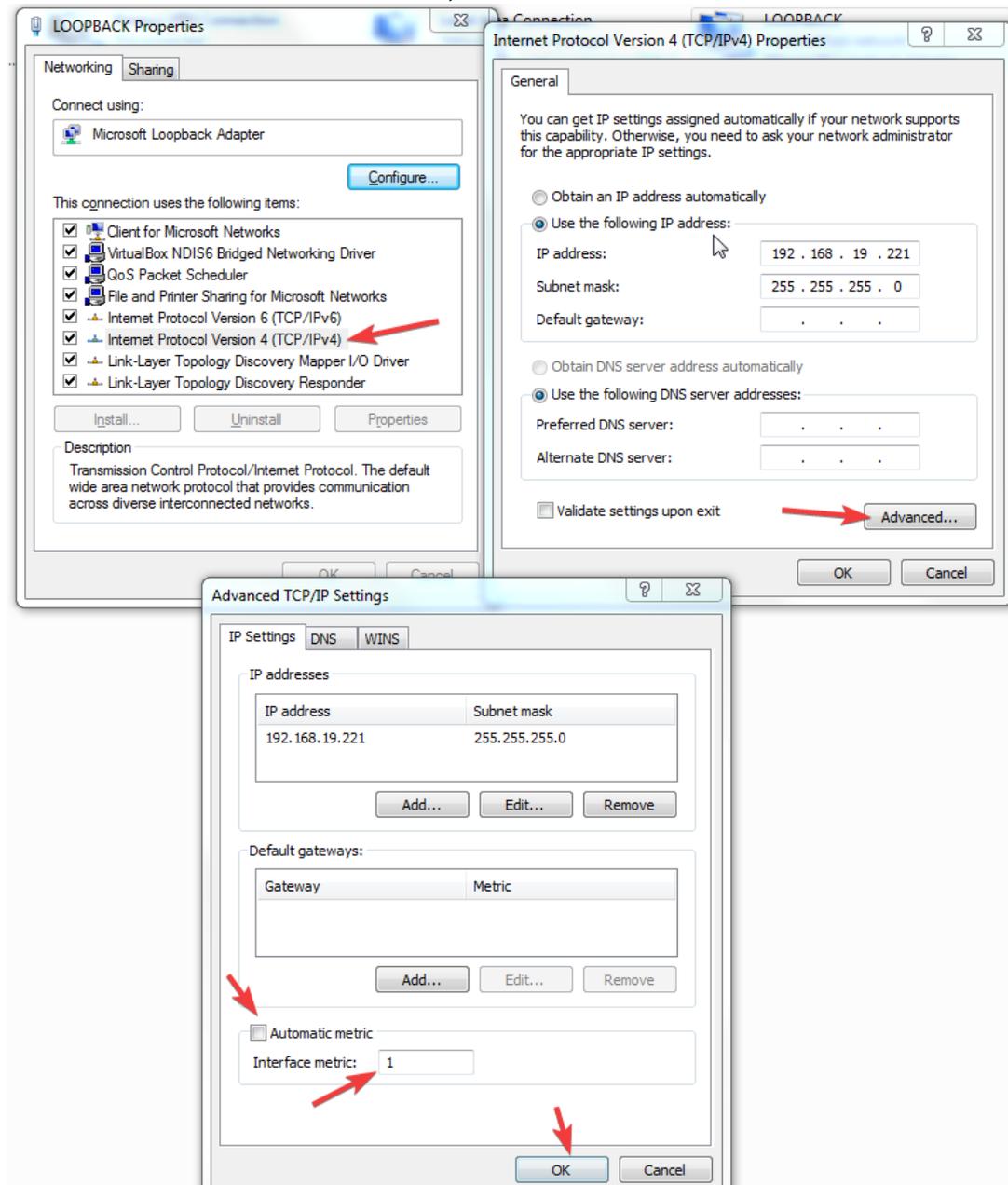
Media Server Protocol Version = 1

Media Protocol Versions

In Vector version V2.1.87 and above, the option is available to send only one Axis parameter for each channel, so 10 bytes per channel. This allows Vector (which is limited to sending one UDP media packet) to output media server information to 144 channels. To do this change “Media Server Protocol Version” to 2 in the Vector.ini file.

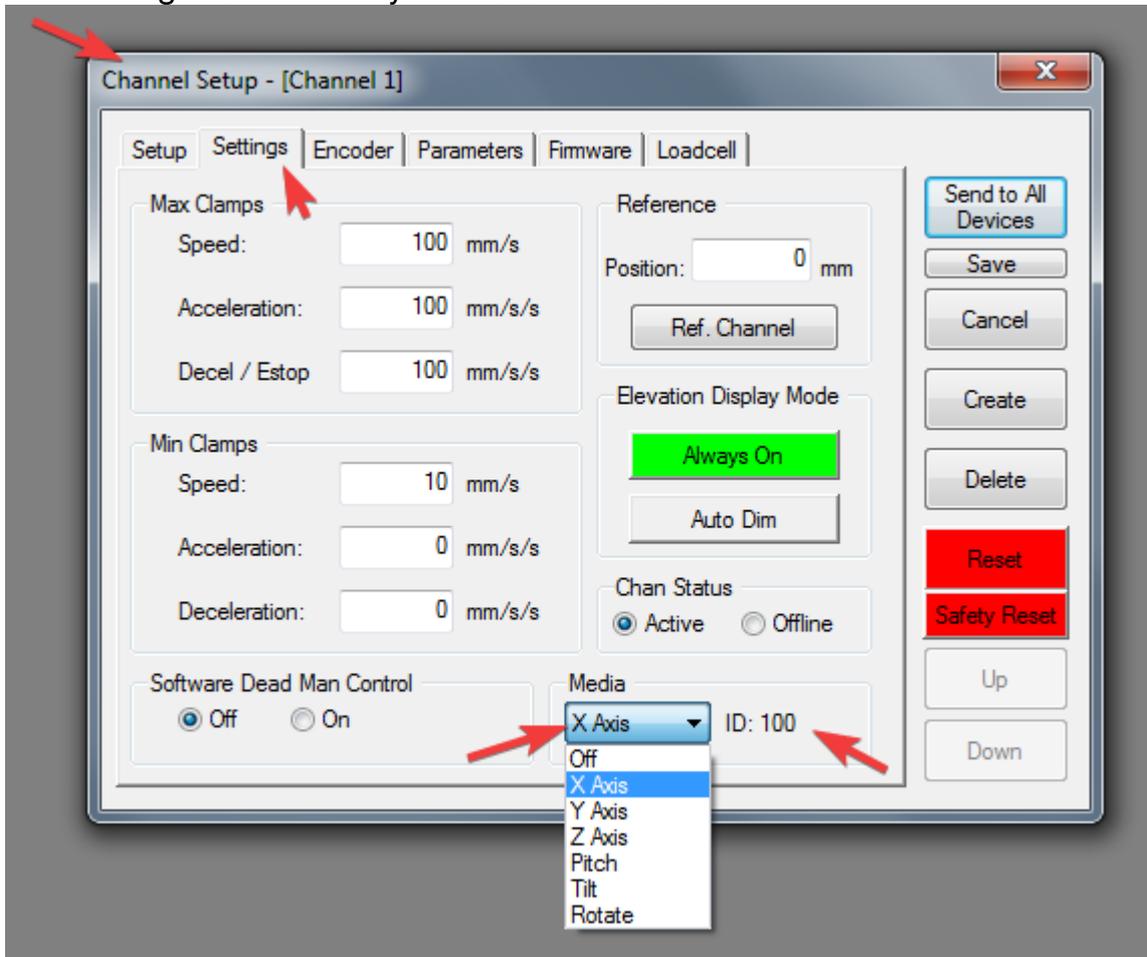
Use of a second Network adaptor for Media server output

Kinesys ALWAYS recommend using a second network adaptor for media server output. To ensure that Windows uses this second network adaptor for media output ensure that its “metric” is set to 1, as shown below.



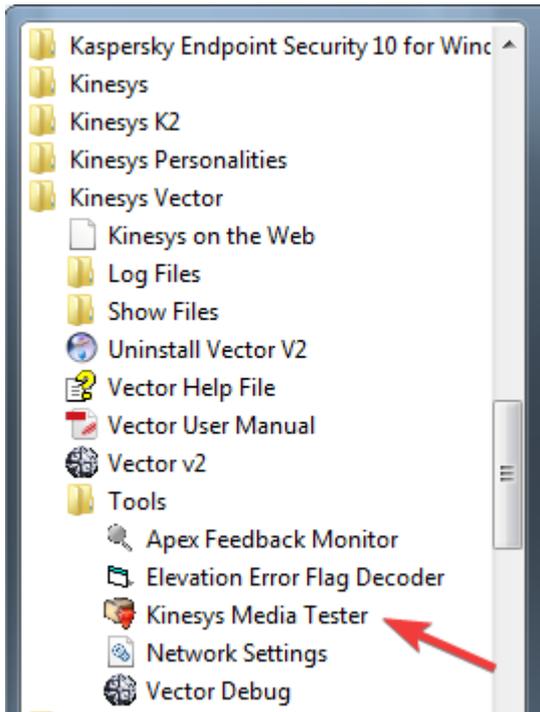
Enable Media output on a specific channel in Vector

To enable a specific channel for media server output in Vector, select the Axis to be represented by the channel in the Media settings. The Media ID will be shown; this is the ID tag that will identify the data in media server.

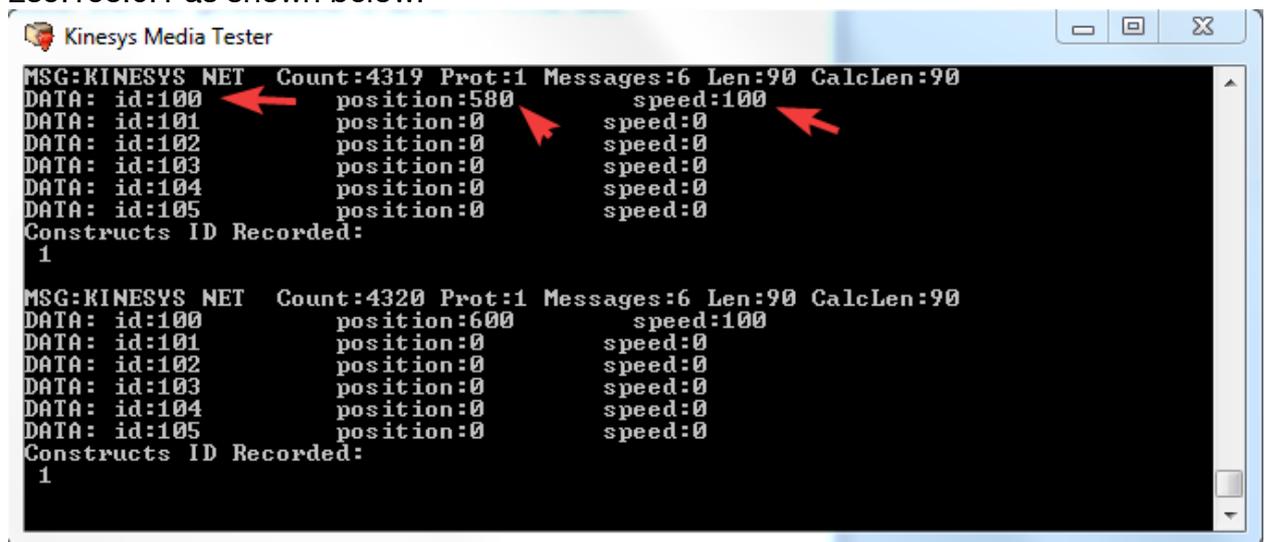


Testing Media Server output

A simple test program has been included in Vector that will display the output being transmitted from Vector. This program will listen to output sent to the multicast address 239.195.0.1 and will only decode Media Server protocols 0 and 1. This test program is called "Kinesys Media Tester". This program is an internal test program used by Kinesys and no support is provided beyond the information in this document.



When run this will sniff any Media server data sent to the multicast address 239.195.0.1 as shown below.



It is also possible to use network sniffing programs like Wireshark <https://www.wireshark.org/> to sniff the data on the network to look for media server packets. The use of Wireshark is beyond the scope of this document. For experienced users of Wireshark a Kinesys Media server protocol decoder is available to download from here <https://kinesys.box.com/v/Wireshark-MediaServer> this decoder is unsupported and should only be used by those with knowledge of using Wireshark.

Products known to support the Kinesys Media server packet format

This is not an exhaustive list but details Visualisers and Media Servers that are known to support Kinesys products.

All the products below support both protocol versions 0 and 1 according to their manufacturers, unless otherwise stated.

Please contact the product manufacturer for more details on specific configuration and setup.

AVOLITES Ai

<https://www.avolites.com/ai-features>

Capture

<http://www.capturesweden.com>

Catalyst

<http://www.samsc.com>

D3 / DISGUISE

<http://www.d3technologies.com>

<https://www.disguise.one/en/>

Green Hippo Hippotizer

<http://www.green-hippo.com/>

(Requires Hippo version 4.5 or greater)

Green Hippo also supports the PosiStageNet protocol, which is supported in a special version of Vector. Please contact Kinesys if PosiStageNet support is required.

Martin P3 System Controllers

<http://www.martin.com/en-US/Product-Details/P3-300-System-Controller>

Mbox

<http://www.prg.com/prg-products/media-servers/mbox-media-server>

Pandoras Box

<http://www.coolux.de>

See

https://www.coolux.de/root/downloads/support/Documentation/Helpfile/index.html#kinesys_in.htm

SMODE Station

<http://smode.fr/>

Wysiwyg

<http://cast-soft.com/wysiwyg/overview>

(Support for newer V1 protocol not confirmed by the manufacturer)