MUSE Controllers

HControl Protocol Document

HARMAN Pro HControl

The HControl protocol is available for third parties to query certain aspects of the MUSE product. HControl is a text-based protocol using a JSON-like syntax. Simple commands can be used to query certain parameters.

GET

The GET command is used to obtain information from the device. A simple get command has the following syntax:

```
get {"path":"$endpoint"}\n
```

All get commands have responses. A response has the following syntax:

```
@get {"path":"$endpoint","value":"$value"}
```

You can use telnet to experiment with the command/response values. An example is shown below:

```
telnet 10.35.92.85 4197

get {"path":"/configuration/device/name"}\n
@get {"path":"/configuration/device/name","value":"MUSE-950ca0"}

get {"path":"/configuration/device/location"}\n
@get {"path":"/configuration/device/location","value":"Richardson Lab"}
```

Booleans will return true/false (case sensitive) by default, if no specific format is specified. The format can be specified to return alternate values:

```
get {"path":" /configuration/network/interface/1/enable"}\n
@get {"path":"/configuration/network/interface/1/enable","value":true}

get {"path":"/configuration/network/interface/1/enable","format":"string"}\n
@get {"path":"/configuration/network/interface/1/enable","value":"true"}
```

Enumerations can return either the string value or the index value of the enumerated value. Returns will depend on the format requested. By default the index value is returned:

DHCP *['DHCP','STATIC']*:

```
get {"path":"/configuration/network/interface/1/ipv4/dhcp"}\n
@get {"path":"/configuration/network/interface/1/ipv4/dhcp","value":0}

get {"path":"/configuration/network/interface/1/ipv4/dhcp","format":"string"}\n
@get {"path":"/configuration/network/interface/1/ipv4/dhcp","value":"DHCP"}
```

SET

Parameters are set via the SET command. SETs generally have an immediate impact. The following command will change the LED color. The SET response will confirm the value that was set.

```
set {"path":"/configuration/device/location","value":"New York"}\n
@set {"path":"/configuration/device/location","value":"New York"}
```

Booleans use true/false (case sensitive) in SETs by default, if no specific format is specified. The format can be specified for alternate values:

```
set {"path":"/configuration/ntp/enable","value":true}\n
@set {"path":"/configuration/ntp/enable","value":true}

set {"path":"/configuration/ntp/enable","format":"string","value":"true"}\n
@set {"path":"/configuration/ntp/enable","value":"true"}
```

Enumerations can use either the string value or the index value of the enumerated value. The following are equivalent SET commands:

DHCP *['DHCP','STATIC']*:

```
set {"path":"/configuration/network/interface/1/ipv4/dhcp","value":0}\n
@set {"path":"/configuration/network/interface/1/ipv4/dhcp","value":0}

set {"path":"/configuration/network/interface/1/ipv4/dhcp","format":"string","value":"DHCP"}\n
@set {"path":"/configuration/network/interface/1/ipv4/dhcp","value":"DHCP"}
```

HControl	Description	Data Type	Get	Set	Example
DEVICE CONFIGURATION					

/configuration/device/version	OS Version	String	GET		`get {"path":"/configuration/device/version"}\n`
					`@get {"path":"/configuration/device/version","value":"1.2.65"}`

/configuration/device/serialnumber	Serial Number	String	GET		

/configuration/device/name	Device Name	String	GET	SET	`set {"path":"/configuration/device/name","value":" MUSE-950ca0"}\n`
					`@set {"path":"/configuration/device/name","value":" MUSE-950ca0"}`

DATE & TIME					

/configuration/ntp/enable	Enable/disable NTP connection	Boolean	GET	SET	`get {"path":"configuration/ntp/enable"}\n`
					`@get {"path":"/configuration/ntp/enable","value":true}`
					
					`set {"path":"/configuration/ntp/enable","value":true}\n`
					`@set {"path":"/configuration/ntp/enable","value":true}`

/configuration/ntp/server/#/host
 	
where # = 1-10
	Value is a either a resolvable hostname or IP address
	
	Note, setting the NTP server will also enable the NTP server.
		String	GET	SET	Setting an NTP server will automatically enable it. Syncing time can take some time. It is generally recommended to reboot once all configuration is complete when changing the time server.
					
					`set {"path":"/ configuration/ntp/1/server","value":"pool.ntp.org"}\n`
					`@set {"path":"/configuration/ntp/1/server","value":"pool.ntp.org"}`

/configuration/ntp/server/#/keyid
 	
where # = 1-10
	NTP Server Key ID	Integer	GET	SET	`get {"path":"configuration/ntp/server/1/keyid"}`
					`@get {"path":"/configuration/ntp/server/1/keyid","value":"1"}`
					
					`set {"path":"/configuration/ntp/server/1/keyid","value":"1"}`
					`@set {"path":"/configuration/ntp/server/1/keyid","value":"1"}`

/configuration/ntp/keyseq/#/keyid
 	
where # = 1-10
	NTP Authentication Key Sequence ID	Integer	GET	SET	`get {"path":"configuration/ntp/keyseq/1/keyid"}`
					`@get {"path":"/configuration/ntp/keyseq/1/keyid","value":"1"}`
					
					`set {"path":"/configuration/ntp/keyseq/1/keyid","value":"1"}`
					`@set {"path":"/configuration/ntp/keyseq/1/keyid","value":"1"}`

/configuration/ntp/keyseq/#/keypassword
 	
where # = 1-10
	NTP Authentication Key Sequence Password	String		SET	`set {"path":"/configuration/ntp/keyseq/1/keypassword","value":"password"}`
					`@set {"path":"/configuration/ntp/keyseq/1/keypassword","value":"password"}`

/configuration/ntp/keyseq/#/encryption
 	
where # = 1-10
	NTP Authentication Key Sequence Encryption	Enum
		
		[‘SHA1’,
		‘MD5’,
		‘NONE’]
			GET	SET	`get {"path":"configuration/ntp/keyseq/1/encryption"}`
					`@get {"path":"/configuration/ntp/keyseq/1/encryption","value":"SHA1"}`
					
					`set {"path":"/configuration/ntp/keyseq/1/encryption","value":"SHA1"}`
					`@set {"path":"/configuration/ntp/keyseq/1/encryption","value":"SHA1"}`

HControl	Description	Data Type	Get	Set	Example
NETWORK					
/configuration/network/interface/#/ipv4/ip_address where # = 1-2	Network IP Address. This must be known in order to run any queries	String	GET	SET	
/configuration/network/interface/#/ipv4/subnetmask where # = 1-2	Subnet Mask	String	GET	SET	get {"path":"/configuration/network/interface/1/ipv4/subnetmask"}\n @get {"path":"/configuration/network/interface/1/ipv4/subnetmask", "value":"255.255.255.0"}
/configuration/network/interface/#/ipv4/gateway where # = 1-2	Gateway IP	String	GET		get {"path":"/configuration/network/interface/1/ipv4/gateway"}\n @get {"path":"/configuration/network/interface/1/ipv4/gateway", "value":"192.168.3.1"}
/configuration/network/interface/#/ipv4/dhcp where # = 1-2	DHCP or Static	Enum	GET	SET	

/configuration/network/interface/#/dnsserver/%
where # = 1-2
where % = 1-3
	DNS addresses	String	GET	SET	get {"path":"/configuration/network/interface/1/dnsserver/1"}\n
					@get {"path":"/configuration/network/interface/1/dnsserver/1", "value":"192.168.3.1"}
					set {"path":"/configuration/network/interface/1/dnsserver/2","value":"8.8.8.8"}\n
					@set {"path":"/configuration/network/interface/1/dnsserver/2", "value":"8.8.8.8"}

/configuration/network/interface/#/mac where # = 1-2	MAC Address	String	GET		

Executing Commands

EXEC

HControl	Description	Data Type	Arguments / Example
REBOOT & RESET			
reboot	Reboot	command	`reboot\n`

/configuration/commands/reset	Reset	command	**System Reset:**
			`exec {"path":"/configuration/commands/","command":"reset","format":"string","value":"System"}\n`
			Factory Reset:
			`exec {"path":"/configuration/commands/","command":"reset","format":"string","value":"Factory"}\n`