

**CIEFFE Nettuno
HTTP Commands
Reference Manual**

Created: 2005-01-31 by: Marco Braga
Last modified: 2005-12-16 by: Marco Braga Revision: 15

Copyright

This manual is the intellectual property of Insignis Technologies and is protected by copyright. All rights are reserved. No part of this document may be reproduced or transmitted for any purpose, by whatever means, be they electronic or mechanical, without the express written permission of Insignis Technologies.

Note

This manual was compiled with the greatest of care and all information double checked. At the time of printing the description was complete and correct. Because of the further development of products, the content of the manual might change without prior notice. Insignis Technologies will not be liable for damage which is directly or indirectly due to errors, incompleteness, or discrepancies between the manual and the product described.

Trademarks

All names used in this manual for hardware and software could be registered trademarks and must be treated as such.





Table of contents

About this manual.....	6
The protocol.....	7
Overview.....	7
HTTP post request.....	7
HTTP post reply.....	8
Commands and status reference.....	10
Syntax specification.....	10
Status.....	11
Connect.....	12
Disconnect.....	13
Play.....	14
Speed.....	15
Pause.....	16
Begin.....	17
End.....	18
Sector.....	19
Find.....	20
Live.....	21
Status text.....	22
Text.....	24



About this manual

Nettuno is CIEFFE's new standalone, DSP based, network encoder / decoder solution. While in decoder mode it can receive compressed video streams from other CIEFFE's products and output composite audio/video to the cctv monitor. In this working mode Nettuno can be entirely controlled by an external software through simple HTTP post/response sequences sent to its internal web server.

This reference manual explains the protocol and commands supported by Nettuno.



The protocol

Overview

Nettuno working as a decoder can receive external commands using HTTP posts directed to its internal web server. The content of each post request is the command (with its parameters) to be executed.

The HTTP “post” is a standard way to send informations to a web server, mainly used to return the informations entered in an HTML form. A complete reference on HTTP can be found on RFC2616:

<http://www.ietf.org/rfc/rfc2616.txt>

HTTP post requests can be easily generated by almost every programming and scripting language. Simple libraries and/or modules are available (if not already included in the standard packages) for C, C++, Visual Basic, Delphi, Python, PERL and many others. Post request can also be generated manually without using any external library or module, by any language able to manipulate strings of text and send them through a TCP connection. Of course every web browser can generate post requests through the “<form>” HTML tag contained in web pages.

HTTP post request

The web server IP address is the same used by Nettuno's main network connection and can be changed in the “network” section of the configuration page. The port used by the web server is the standard TCP port 80.

The post request must be directed to a specified page on Nettuno: “clicmd.cgi”, so supposing Nettuno's IP address is configured as “192.168.135.64” the complete URL for the request is:

<http://192.168.135.64/clicmd.cgi>

The body of the post request defines the command to be executed and is described in the commands reference section of this manual. Nettuno will reply to a request with an HTTP status code and a Nettuno status text contained in the web page returned to the post request.

The description of the contents of an HTTP post request is out of the scopes of this manual, but many examples can be found on the Internet. An example is:

```

POST /clicmd.cgi HTTP/1.1
Connection: Keep-Alive
Content-Type: application/x-www-form-urlencoded
Content-Length: 75
Cache-control: no-cache
Host: 192.168.135.64
Accept: image/gif, image/x-xbitmap, image/jpeg, image/pjpeg, */*
Accept-Encoding: gzip, deflate
Referer: http://192.168.135.64/clicmd.cgi
User-Agent: Mozilla/3.0 (compatible; Indy Library)

cmd=connect&ipaddr=192.168.135.149&ipport=1194&login=admin&passwd=&camera=1

```

The first group of lines is the HTTP header, while the last line is the HTTP post body, containing a Nettuno command. The header is normally generated automatically by the library or module function used to send the post and should not normally be a concern to the programmer.

The last line is the body of the request and contains the command to be sent to Nettuno. Each command line is merged by the library or module post function into a single line with each element separated by a '&' character, according to the HTTP standard. **Notice that the '&' separator is not part of the Nettuno command syntax and is only used as a separator in HTTP post requests.**

HTTP post reply

Nettuno will reply to the request sending an HTTP status code and some lines describing the current status of the Nettuno. An example reply is:

```

HTTP/1.0 200 OK
Content-type: text/html

ipaddr=192.168.140.77
port=1194
camera=2
status=live

```

The HTTP status code is can be:

HTTP status code	HTTP status name	Comments
200	OK	The request syntax is correct and the command has been accepted
400	Bad Request	The request syntax is incorrect, the command has been ignored



The body of the reply will contain the status of the Nettuno after the command. The possible status text lines are described in the reference section of the manual.



Commands and status reference

Syntax specification

This section contains a complete reference of Nettuno commands. Each command is composed by one or more lines of text. The syntax of a line is:

```
parametername=<value>
```

For example:

```
cmd=connect
```

Each command must contain a specific set of lines as described in the following reference. The parameter name is a fixed text, while the value enclosed between the '<' and '>' characters is a variable parameter value. The '<' and '>' characters must NOT be inserted in the command text, they are only used to specify a variable element of the command in the syntax specification.

As an example, a complete command syntax can be:

```
cmd=connect
ipaddr=<server IP address>
ipport=<server video port>
login=<login name>
passwd=<password, can be empty>
camera=<camera number>
```

and a possible use of this command can be:

```
cmd=connect
ipaddr=192.168.135.149
ipport=1194
login=admin
passwd=
camera=1
```

This is the entire text to be sent to Nettuno. Notice that as explained in the previous section, the HTTP post will contain the lines merged in a single line using the '&' character as a separator.

Some lines can be optional to a command. These lines will be enclosed between '[' and ']' in the syntax specification.

Status

This command does nothing. It just forces Nettuno to reply with the status text.

SYNTAX:

```
cmd=status
```

PARAMETERS:

NONE

Connect

This command will make Nettuno connect to a specified server, ask for the live video from the specified camera. If Nettuno is already connected to that server, it will only ask the video from the new camera.

SYNTAX:

```
cmd=connect
ipaddr=<server IP address>
ipport=<server video port>
login=<login name>
passwd=<password, can be empty>
camera=<camera number>
```

PARAMETERS:

ipaddr	IP address of the server to connect to, in the format x.y.w.z, for example "192.168.135.64".
ipport	TCP port of the video server. Normally should be set to 1194.
login	Login name on the server to connect with.
passwd	Password to use with the specified login. It can be empty.
camera	Number of the camera to show. typically from 1 to 16.

Disconnect

This command will disconnect Nettuno from the current server. If it's not connected the command will be ignored.

SYNTAX:

```
cmd=disconnect
```

PARAMETERS:

NONE

Play

Activate the play mode on the currently connected server. Nettuno will ask the server for the recorded video on the specified camera and sector number.

SYNTAX:

```
cmd=play
camera=<numerocam>
sector=<numerosettore>
wait=<0|1>
time=<time>
```

PARAMETERS:

camera	Number of the camera to show. typically from 1 to 16.
sector	Sector number: 0 = Prime 1 = Time Lapse 2 = Alarm
wait	Activate the step-by-step mode. If set to 0, play will begin immediately. If set to 1, it will only set the specified parameters and wait for a "speed" command to proceed.
time	Specify the starting play time as 64 bit integer Windows filetime. Use 0 to start from the beginning of the recording.

Speed

While in play mode, change the play speed and eventually resume from the “step-by-step” mode. If used while in “step-by-step” mode and setting the “wait” parameter to 1, it will show only one frame and wait for a new speed command.

SYNTAX:

```
cmd=speed
speed=<speed>
wait=<0|1>
```

PARAMETERS:

speed	New speed value. Positive values are forward, negative are backward. A value of 0 is the same as a pause command. 1 is the normal play speed value. This parameter must be between -512 and 512.
wait	Activate the step-by-step mode. If set to 0, play will begin immediately with the new value. If set to 1, it will only set the specified parameters and wait for a “speed” command to proceed.

Pause

While in play mode, this command will pause the video.

SYNTAX:

cmd=pause

PARAMETERS:

NONE

Begin

While in play mode, jumps to the beginning of the current sector and wait for a “speed” command.

SYNTAX:

cmd=begin

PARAMETERS:

NONE

End

While in play mode, jumps to the end of the current sector and wait for a “speed” command.

SYNTAX:

cmd=end

PARAMETERS:

NONE

Sector

While in play mode, this command changes the current sector.

SYNTAX:

```
cmd=sector  
sector=<sector number 0-2>
```

PARAMETERS:

sector	Sector number: 0 = Prime 1 = Time Lapse 2 = Alarm
--------	------------------------------------------------------------

Find

While in play mode, this command jumps the play time to the specified time position.

SYNTAX:

```
cmd=find
date=<datetime value>
wait=<0|1>
```

PARAMETERS:

wait	Activate the step-by-step mode. If set to 0, play will begin immediately from the new time position. If set to 1, it will only set the specified parameters and wait for a “speed” command to proceed.
time	This parameter line is optional. If not specified, play will start from the beginning of the specified sector. If specified, it must be a 64 bit integer Windows filetime.

Live

Return to live mode from the play mode, switching to the specified camera.

SYNTAX:

```
cmd=live  
camera=<camera number>
```

PARAMETERS:

camera	Number of the camera to show. typically from 1 to 16.
--------	-------------------------------------------------------

Status text

Nettuno replies to commands with a status text that can be used to synchronize the client application with the current status of Nettuno. The status is composed by several lines, as in this example:

```
ipaddr=192.168.140.77
port=1194
camera=2
status=live
```

The lines contained in the status text can change depending on the status of Nettuno:

While disconnected:

```
status=disconnected
```

PARAMETERS:

NONE

While in live mode:

```
ipaddr=<current server IP address>
port=<server video port>
camera=<camera number>
status=live
```

PARAMETERS:

ipaddr	IP address of the server, in the format x.y.w.z, for example "192.168.135.64".
ipport	TCP port of the video server.
camera	Number of the current camera. Typically from 1 to 16.

While in play mode:

```
ipaddr=<current server IP address>
port=<server video port>
camera=<camera number>
status=<play | pause>
speed=<speed value>
sector=<sector number>
```

PARAMETERS:

ipaddr	IP address of the server, in the format x.y.w.z, for example "192.168.135.64".
ipport	TCP port of the video server.
camera	Number of the current camera. Typically from 1 to 16.
status	The text string "play" while playing video or "pause" when in pause mode.
speed	Current speed value. Positive values are forward, negative are backward. 1 is the normal play speed value. This parameter is between -512 and 512.
sector	Sector number: 0 = Prime 1 = Time Lapse 2 = Alarm

Text

Set or remove up to 3 lines of custom text

SYNTAX:

```
cmd=text  
l1=<line1>  
l2=<line2>  
l3=<line3>
```

PARAMETERS:

l1,l2,l3	Text to be shown or an empty line to remove the text
----------	------------------------------------------------------