jsonBI USERS GUIDE

Created by Ken Burkhalter (Copyright ©2013-All Rights Reserved)

Release History

```
' [130630] v1.0.1    -released
' [130713] v1.1.0.0  -fixed sendReq sequence with KeepAlive=False and added proper {} syntax.
' [130717] v2.0b     -Added capture of BI Return Codes. Added STATUS cmd (reads work but writes
                      are still being developed)
' [130731] v2.0.1    -Enhanced error msg when cmmd not recognized to help users debug.
' [130912] v2.1.0    -Added STATUS command set
' [130925] v2.2.0    -Chgd write method for rtnDataFile to try and fix file locked issue that
                      sometimes occured.
' [130927] v2.2.1    -Added DateTime to error comments for better correlation with events.
' [131110] v2.2.2    -bug fix to correct PTZ command failure
 ' [131211] v2.2.3   -bug fix. Chgd TRIGGER from camconfig class to standalone TRIGGER class, as
                      required.
```

This program (jsonBI) is a bi-directional application using the JSON (JavaScript Object Notation) interface to provide methods for remote administration, handling both camera control and functionality of the Blue Iris (BI) Video Security and Webcam software.

This program has no implied warranties for performance, functionality, or suitability for user's purpose. It is designed as a non-showing Windows Forms Application so it runs without screen interruptions.

Description

Included in the software release are both a (Windows) executable as well as the native VB .Net 4.5 code. You are free to use and modify the code as you wish for personal and non-commercial uses, however any distribution of this software to others shall include credit to Ken Burkhalter as the author.

The executable (jsonBI.exe) is called by Command Console, Script, or external application with capabilities to call other programs. ***Following the program call to jsonBI.exe is the command list.*** Up to 15 commands can be chained in a call. They are executed serially one after another.

See also the excellent posting by fastone01 (Senior Member) here . . . http://www.cam-it.org/index.php?topic=2687.0 for an excellent write-up on how to call via scripts or other means, applications like “jsonBI”.

An example of the normal calling syntax for this program would be . . .

```
jsonBI.exe login http://192.168.1.90:8080/json keb qwert1234; camconfig pause ThtrCam -1; ptz,
DrvCam,  5; status
```

NOTE: ***Spaces (as in the camconfig command above) or commas (as in the ptz command above) may be used between arguments. Use a semicolon (;) to separate commands.***

Individual Commands are illustrated below. See also the Appendix to this document, which is reproduced from the Blue Iris Help Guide about the JSON interface. This goes into detail on how Blue Iris treats each command. *This should be referenced to understand the details of each command’s parameters.*

NOTE: The syntax for this App is ***different*** from the actual Blue Iris json command syntax. I have attempted to make the entry of commands less complex and structured and much simpler for users to do. This App converts the simpler command syntax into the more complex BI json requirements.

Examples are documented below for these simpler commands, so that the command strings are clear.

jsonBI Implemented Commands

Note: Each of the following commands are arguments following the jsonBI.exe call. Except for the LOGOUT command, each command sequence sent to BI ***must begin*** with a LOGIN command. At the end of every command sequence sent to BI, all return code information sent back from BI is recorded in a file at C:\biRtnCodes.txt as shown in this example . . .

```
pause:4294967
motion:true
schedule:false
ptzcycle:false
ptzevents:false
alerts:0
record:2
```

LOGIN **login, Blue Iris Computer IP Addr:port/json, userName, userPW**

Definitions:

Blue Iris Computer IP Addr:port =LAN IP addr of BI hosting computer & BI port number plus '/json' (eg, 192.168.1.90:80/json)
userName=your user ID (login name) to BI
userPW=your BI login password

There is normally no reason to ONLY login. The login string is typically part of a normal initial command string. That is . . a login is the first part of each command. It is ended by a ";" and then followed by the one, or more, commands to be executed.

Examples: login http://192.168.1.90:8080/json keb qwert1234 (login only)

login http://192.168.1.90:8080/json keb qwert1234; camconfig trigger houseCAM (Full useful command string)

NOTE: every one of the following commands must start with a LOGIN command as the first item, as in the example above. The commands will not function in the absence of a login command. For simplicity, the examples below are given just for the specific command, assuming that a login string will precede them. A **LOGOUT** does not need to ever be executed as BlueIris will automatically log the session out after a few seconds of inactivity.

ENABLE **camconfig, enable, camera, true/**

camconfig [required]
camera=shortname for camera to be controled (eg, driveCam)
true/false=use *true*, or *false* as value to enable or disable camera

Example: camconfig enable houseCAM true

MOTION **camconfig motion camera true/false**

camconfig [required]
camera=shortname for camera to be controled (eg, driveCam)
true/false=use *true*, or *false* as value to enable or disable camera's motion detection

Example: camconfig motion houseCAM true

PAUSE **camconfig pause camera value**

camconfig [required]
camera=shortname for camera to be controled (eg, driveCam)
value=
-1: pause indefinitely
0: un-pause
1..3: add 30 seconds [1], 1 minute [2], 1 hour [3] to the pause time

Example: camconfig pause houseCAM -1

PTZCYCLE **camconfig ptzcycle camera true/false**

camera=shortname for camera to be controled (eg, driveCam)
true/false=use *true*, or *false* as value to enable or disable camera's preset cycle feature

Example: camconfig ptzcycle houseCAM true

PTZEVENTS **camconfig ptzevents camera true/false**

camera=shortname for camera to be controled (eg, driveCam)
true/false=use *true*, or *false* as value to enable or disable camera's ptzevent schedule

Example: camconfig ptzevents houseCAM false

PTZ **ptz camera button**

camera=shortname for camera to be controled (eg, driveCam)
button: this value determines the PTZ operation performed:
0 : Pan(Left)
1 : Pan(Right)
2 : Tilt(up)
3 : Tilt(down)
4 : Center or home (if supported by camera)
5 : Zoom in
6 : Zoom(out)
8..10 : Power mode, 50, 60, or outdoor
11..26 : Brightness 0-15
27..33 : Contrast 0-6
34..35 : IR on, off
101..120: Go to preset position 1..20
updown: Send a value of 1 to indicate that a complementary "stop" event will follow;
send 0 otherwise and the camera will be moved for a preset duration

Example: ptz houseCAM 3

STATUS **status**

Example: status

The return data from a simple **status** command looks like this . . .

```
signal:1
cpu:10
mem:211.8M
lock:1
profile:1
schedule:Default
dio:[0,0,0,0,0,0,0,0]
 uptime:4:07:59:40
 clips:New: 33 [2%] +491.6G; Stored: 71 [2%] +491.6G
tzone:-240
```

NOTE: the dio states will always be “0” unless there is a controlling interface connected.

The next level of **status** handles the *signal* and *profile* sub-commands. These sub-commands require the status command followed by the sub-command and a parameter.

```
Example:    status signal 1
            status profile 3
```

The highest level **status** command is the *dio*. (Which requires an Arduino or SeaLevel interface before any status will be returned.)

The *dio* sub-command is followed either by a short pulse time value (in milliseconds) or by a hard on/off value in addition to the contact to be controlled.

```
Example:    status dio 5000 3 (Triggers output 3 for 5 seconds)
            status dio true 2  (Forces output 2 to the ON state)
```

NOTE: BlueIris will only respond to dio commands of there is an attached digital I/O device such as the Arduino or SealLevel Systems unit.

TRIGGER **trigger camera**

camera=shortname of camera to be triggered

```
Example:  trigger houseCAM
```

LOGOUT **logout**

Definitions: None

```
Example: logout
```

Logouts are not actually required as BI automatically breaks the connection if no commands are sent after the initial login sequence.

NOTE: These are general commands and may or may not work with a particular camera depending on BI and camera functionality.

SEE APPENDIX ON NEXT PAGE FOR DETAILED BLUE IRIS HANDLING OF JSON COMMANDS.

APPENDIX - JSON interface

The JSON (JavaScript Object Notation) interface exists to provide methods for remote administration of your Blue Iris system. It will be the gateway used by new client apps including those for iOS (iPhone and iPad) and Android mobile devices. It will also be used by the next generation of web pages used to access Blue Iris via browser in order to provide more secure authentication.

For a description of JSON, see http://www.json.org/. It's simply a block of text which is sent by HTTP-POST to the Blue Iris web server page /json. Blue Iris will respond with a JSON formatted response (content-type application/json).

Each JSON object sent to Blue Iris should have a "cmd" value, for example, "cmd":"login". Additional values will depend upon the type of command sent. Here's an example command and response in full JSON format:

```
{"cmd":"login"}
{"result":"fail","session":"182c8a04f7d4ab042ff8e4a2"}
```

Following are the available commands:

alertlist

Get a list of alert images from the Alerts folder

camera: a camera's short name or a group name; "index" will return alerts from all cameras

startdate: expressed as the integer number of seconds since January 1, 1970

reset: if true, will erase all alerts from the alerts folder.

The returned data value is an array of JSON objects each describing a camera or a camera group. For each of these objects, the following values are defined:

- ***camera***: the camera or group name
- ***jpeg***: the JPEG image filename as found in the Alerts folder
- ***path***: the part of the absolute file path that follows the New clips folder path; if there are no subfolders, this is simply \ and the filename.
- ***offset***: the number of kilobytes (for BVR files) or milliseconds (for other formats) into the file at which the alert occurred
- ***flags***: the following flags are defined:
  - 1: the offset is in time (milliseconds)
  - 2: the alert was triggered by the motion detector
  - 4: the camera was in a no-signal state at the time of the alert
  - 8: the alert was triggered by an audio event
  - 16: the alert was triggered by an external source such as DIO, JSON command, or manual trigger
- ***date***: file creation date, expressed as the integer number of seconds since January 1, 1970
- ***color***: 24-bit RGB value (red least significant) representing the camera's display color

camconfig

Get (and optionally set) the state of many camera properties:

- ***reset***:true reset the camera
- ***enable***:true or false enable or disable the camera

pause:n sends a pause command, and returns a value in seconds

-1: pause indefinitely

0: un-pause

1..3: add 30 seconds, 1 minute, 1 hour to the pause time

motion:true or false enable or disable the motion detector

schedule:true or false enable or disable the camera's custom schedule

ptzcycle:true or false enable or disable the preset-cycle feature

ptzevents:true or false enable or disable the PTZ event schedule

alerts:n sets the corresponding alert function

record:n sets the corresponding record function

camlist

Returns a list of cameras on the system ordered by group. Cameras not belonging to any group are shown beneath the "all cameras" group. Disabled cameras are placed at the end of the list.

reset: send a value of true for this argument to reset the statistics for the cameras.

data is an array of objects (note the [] surrounding a JSON array), each describing a camera or a camera group. For each of these objects, the following values are defined:

optionsDisplay: the camera or group name

optionsValue: the camera or group short name, used for other requests and commands requiring a camera short name

FPS: the current number of frames/second delivered from the camera

color: 24-bit RGB value (red least significant) representing the camera's display color

clipsCreated: the number of clips created since the camera stats were last reset

isAlerting: true or false; currently sending an alert

isEnabled: true or false

isOnline true or false

isMotion: true or false

isNoSignal: true or false

isPaused: true or false

isTriggered: true or false

isRecording: true or false

isYellow: true or false; the yellow caution icon

profile: the camera's currently active profile, or as overridden by the global schedule or the UI profile buttons.

ptz: is PTZ supported, true or false

audio: is audio supported, true or false

width: width of the standard video frame

height: height of the standard video frame

nTriggers: number of trigger events since last reset

nNoSignal: number of no signal events since last reset

nClips: number of no recording events since last reset

cliplist

Get a list of clips from the New folder

camera: a camera's short name or a group name; "index" will return clips from all cameras

startdate: expressed as the integer number of seconds since January 1, 1970

enddate: expressed as the integer number of seconds since January 1, 1970

tiles: true or false; true to send only 1 entry per day in order to mark tiles on the calendar

The returned data value is an array of JSON objects each describing a camera or a camera group. For each of these objects, the following values are defined:

camera: the camera or group name

path: the part of the absolute file path that follows the New clips folder path; if there are no subfolders, this is simply \ and the filename.

date: file creation date, expressed as the integer number of seconds since January 1, 1970

color: 24-bit RGB value (red least significant) representing the camera's display color

log

Get a list of the status log entries, an array of objects:

date: expressed as the integer number of seconds since January 1, 1970

level: severity, 0=info, 1=warn, 2=error

obj: object name

msg: the text of the log entry

login

Blue Iris will respond with a "result" value of "fail' and a "session" value. Respond with this session value combined with a userid and password and MD5 hash encoded as follows:

response = MD5("userid:session:password")

{"cmd":"login","session":"182c8a04f7d4ab042ff8e4a2","response":"response"}

There are additional login values supported in order to identify a mobile device:

uuid: a unique identifier

token: a code used to send push notifications

devicename: a description of the device

devicetype: for example, "iOS"

if a correct response is received, Blue Iris will respond:

{"result":"success","data":{"system name":"your system name"}}

For all other commands, you must supply a valid "session" value as supplied by the login command.

logout

If you were successfully logged-in, you will receive

{"result":"success"}

ptz

Operate a camera's PTZ functionality

camera: a camera's short name

button: this value determines the PTZ operation performed:

- 0: Pan left
- 1: Pan right
- 2: Tilt up
- 3: Tilt down
- 4: Center or home (if supported by camera)
- 5: Zoom in
- 6: Zoom out
- 8..10: Power mode, 50, 60, or outdoor
- 11..26: Brightness 0-15
- 27..33: Contrast 0-6
- 34..35: IR on, off
- 101..120: Go to preset position 1..20

updown: send a value of 1 to indicate that a complementary "stop" event will follow; send 0 otherwise and the camera will be moved for a preset duration

status

Get (and optionally set) the state of the traffic signal icon, active global profile as well as the schedule's hold/run state:

- ***signal***: a single digit 0 for red, 1 for green, 2 for yellow.
- ***profile***: a single digit 0-7 for the profile number to set; or -1 to change the hold/run state. This functions the same it does on the local UI, so sending a profile change a second time will set the schedule to its "hold" state.
- ***dio***: the state of a DIO output. An array of 0's and 1's is returned, or you may set a particular value by sending an object with:
  - ***output***: an output number 0-7
  - ***force***: true or false
  - ***msec***: the number of milliseconds to hold the output enabled if force is not specified.
- ***play***: play a sound file from the application Sounds folder.

The follow values are also returned:

- ***lock***: the state of the schedule run/hold button: 0 for run, 2 for temp, 1 for hold
- ***clips***: a text value describing the number of clips in the New and Stored folders along with disc usage statistics
- ***warnings***: the number of new warnings since the log command was last used

alerts: the number of new alerts since the alerts command was last used

cpu: the server's CPU usage overall (not just Blue Iris) expressed as a percentage.

mem: a string representation of the Blue Iris process memory usage

uptime: a string representation of the time in days:hours:minutes:seconds that Blue Iris has been running

sysconfig

Get and set system configuration settings. Admin access required.

archive (output and optionally input): enable web archival

schedule (output and optionally input): enable the global schedule

trigger

Trigger the motion sensor on a specific camera. Admin access required. Additional required values:

camera: a valid and enabled camera short name.