ipConfigure Embedded LPR API

Version 1.3.5

June 10, 2013

1 Camera Configuration Parameters

ipConfigure Embedded LPR uses several user-adjustable configuration parameters which are exposed by Axis Communication's VAPIX® Version 3 Parameter Management system. These parameters are:

`root.Elpr.FontSet`	License plate training data set used for character recognition. Valid values are dependent on the contents of the Embedded LPR payload file and may be discovered using the VAPIX® API (see below).
`root.Elpr.TargetFramerate`	Rate, in frames per second, at which images are captured for processing. Valid values are the integers $\{1, 2, \ldots, 10\}$.
`root.Elpr.PlateColor`	Determines if Embedded LPR will search for license plates containing light-colored characters on dark backgrounds, dark-colored characters on light backgrounds, or both. Valid values are the strings {`"L"`, `"D"`, `"B"`} for light-on-dark, dark-on-light, and both, respectively.
`root.Elpr.MinimumCharacterConfidence`	Only characters with a recognition confidence greater than this value will be considered part of a license plate. Valid values are the integers $\{0, 1, \ldots, 100\}$.
`root.Elpr.PlateConfidence`	Only license plate results whose median character confidence is greater than this value will be recorded. Valid values are the integers $\{0, 1, \ldots, 100\}$.
`root.Elpr.OverviewFrameCount`	Specifies the number of overview JPEG images to save with each recognized license plate. Valid values are the integers $\{1, 2, \ldots, 5\}$.
`root.Elpr.FlipRawInput`	If true, the camera image is flipped before processing license plates. Valid values are `YES` and `NO`.
`root.Elpr.DisableLED`	If true, the camera's status LED is turned off during Embedded LPR operation, which may be necessary for night time processing in certain enclosures. Valid values are `YES` and `NO`.

`root.Elpr.FIFOProcessing`	If true, image frames are processed on a strict first-in, first-out basis. If false, image frames are processed in the most computationally efficient order (which may result in newer frames being processed before older frames). Valid values are `YES` and `NO`.
`root.Elpr.PushResults`	Controls result push behavior. If set to `off`, push notifications are not sent. `all` pushes every detected plate. `hits` pushes only plates which match the watchlist.
`root.Elpr.PushMethod`	If set to `http`, results are pushed to an HTTP server via POST requests. If set to `tcp`, results are sent directly to a port on a listening IP address.
`root.Elpr.PushFormat`	Specifies format of pushed text. Available options are `form`, for URL-encoded HTTP form data, `serial` for ASCII records, and `json` for JSON-formatted data.
`root.Elpr.PushAddress`	Address to which license plates results will be pushed. See §6 for more information.
`root.Elpr.LogLevel`	Controls the level of detail written to the application log. Valid values are the integers $\{1, 2, \ldots, 100\}$. Values greater than 5 may degrade application performance.

The complete VAPIX® API, including methods to modify, query, and enumerate valid values, is available from Axis Communications AB at `http://www.axis.com`.

2 Storage Status

When Embedded LPR is running, the system's storage status may be queried using the URL:

```
http://camera.ip/local/elpr/storcap.cgi
```

The following JSON-encoded variables are returned:

`used`	Number of bytes used on the camera's SD card.
`total`	Number of bytes available on the camera's SD card.
`percent`	Percentage of used SD card capacity.
`queue`	The number of images saved in the Embedded LPR work queue on the SD card, waiting to be processed.

3 Camera Status

When Embedded LPR is running, the system's runtime status may be queried using the URL:

```
http://camera.ip/local/elpr/status.cgi
```

The following JSON-encoded results are possible:

`{ "elpr_status": "ELPR_READY" }`	Embedded LPR is fully installed. License plates will be recognized and recorded only under this status.
`{ "elpr_status": "ELPR_UNLICENSED" }`	Embedded LPR is installed but not licensed.
`{ "elpr_status": "ELPR_NO_CARD" }`	Could not find an SD card in the camera.
`{ "elpr_status": "ELPR_CARD_UNMOUNT_ERROR" }`	Detected a uninitialized SD card that could not be unmount.
`{ "elpr_status": "ELPR_CARD_MOUNT_ERROR" }`	Failed to mount the installed SD card; a reformat is required.
`{ "elpr_status": "ELPR_CARD_NO_PAYLOAD" }`	The SD card is installed and ready but Embedded LPR payload contents could not be found.

4 Query License Plate Results

When Embedded LPR is running, results may be queried using the URL:

```
http://camera.ip/local/elpr/json.cgi
```

The following parameters may be specified:

`begin`	The beginning of the date range in which license plates will be returned. Valid values are any Unix timestamp (defined by `POSIX.1` as the number of seconds since midnight on January 1, 1970, not including leap seconds).
`end`	The end of the date range in which license plates will be returned. Valid values are any Unix timestamp.
`limit`	Number of results to return. Valid values are: `> 0` the maximum number of LPR results to return. `-1` return a single response of the form `{"count": 99}` with the size of the results set.
`offset`	Omits the first n results from the result set. Valid values are integers greater than 0. This parameter is valid only when used in conjunction with parameter `limit`.
`asc`	Ordering of the results set. Valid values are: `> 0` order results in ascending order by timestamp. `<= 0` order results in descending order by timestamp.
`find`	Returns only those results whose plate text matches the specified string, which may include the wildcards `?` (matching any one character) and `%` (matching zero or more characters). The specified string, including wildcards, must be URL encoded.

After querying license plate results, the camera will return JSON-encoded data of the form:

```
{
   "identifier":  "id",
   "items":
   [
      { "id":  0, "watch_hit":  0, "date":  "00/00/00", "time":  "00:00:00",
        "text":  "AAA000", "plt":  "/local/elpr/results/00000/0/plt_000000.bmp",
        "ovr":  [
           "/local/elpr/results/00000/0/ovr_000000.jpg",
           "/local/elpr/results/00000/0/ovr_000001.jpg",
           "/local/elpr/results/00000/0/ovr_000002.jpg
        ]},
      { "id":  1, "watch_hit":  0, "date":  "00/00/00", "time":  "00:00:00",
        "text":  "AAA000", "plt":  "/local/elpr/results/00000/0/plt_000001.bmp",
        "ovr":  [
           "/local/elpr/results/00000/0/ovr_000001.jpg",
           "/local/elpr/results/00000/0/ovr_000002.jpg",
           "/local/elpr/results/00000/0/ovr_000003.jpg
        ]},
       ...
   ]
}
```

Where the resultant data contains the values:

`id`	Unique identifier for this Embedded LPR result. Valid values are integers greater than 0.
`watch_hit`	Specifies if this Embedded LPR result triggered a watch list hit event. Valid values are 0 for false and 1 for true.
`date`	Date on which this Embedded LPR result was captured. If the VAPIX® parameter `root.Image.OwnDateFormatEnabled` is set to `yes` the date is formatted according to `root.Image.OwnDateFormat`, otherwise the date is formatted as "`MM/DD/YYYY`".
`time`	Time at which this Embedded LPR result was captured. If the VAPIX® parameter `root.Image.OwnTimeFormatEnabled` is set to `yes` the time is formatted according to `root.Image.OwnTimeFormat`, otherwise the time is formatted as "`hh:mm:ss`".
`text`	Contents of the recognized license plate.
`plt`	Path of the binary license plate character image for this Embedded LPR result. This image is externally visible by prepending the camera IP address.
`ovr`	Array of paths of the color overview images for this Embedded LPR result. These images are externally visible by prepending the camera IP address.

4.1 Examples

Find the first 50 plates containing the letters "CORTL" anywhere in the license plate text, captured on or after January 23, 2012 at 1:47:12pm EST:

```
http://camera.ip/local/elpr/lpr.cgi?begin=1343069232&limit=50&asc=1&find=%25CORTL%25
```

Assuming your interface displays 50 results per page, retrieve page 3 of the previous result set:

```
http://camera.ip/local/elpr/lpr.cgi?begin=1343069232&limit=50&offset=100&asc=1&find=%25CORTL%25
```

Retrieve the total number of plates in the database containing the letters "CORTL" somewhere in the license plate text:

```
http://camera.ip/local/elpr/lpr.cgi?limit=-1&find=%25CORTL%25
```

5 Watch List Management

License plate tags in the watch list may be retrieved using the URL:

```
http://camera.ip/local/elpr/wl.cgi
```

After querying the watch list, the camera will return JSON-encoded data of the form:

```
{
   "identifier":  "id",
   "items":
   [
      { "id":  0, "text":  "ABC123" },
      { "id":  1, "text":  "XYZ456" }
       ...
   ]
}
```

Where the resultant data contains the values:

`id`	Unique identifier for this watch list entry. Valid values are integers greater than or equal to 0.
`text`	License plate text which will generate a watch list hit event. This string may contain wildcards of the form described in §4.

The following parameters may be appended to the watch list URL for watch list management:

`remove`	Remove the license plate text entry from the watch list specified by identifier n. Valid values are integers greater than or equal to 0. This command will return an empty page.
`add`	Add a specified license plate text entry to the watch list. Valid values are any URL encoded string. This command will return the `id` of the newly added entry.

5.1 Examples

Add the license plate "ABC123" to the watch list:

```
http://camera.ip/local/elpr/wl.cgi?add=ABC123
```

The returned result will be off the form `{ "id":  99 }`.

Remove the previously added license plate from the watch list:

```
http://camera.ip/local/elpr/wl.cgi?remove=99
```

6 Push Notification

Push notifications allow eLPR to actively notify an external server about new license plate results. Whenever a license plate is detected, the result will be sent via an HTTP POST request or TCP.

This section describes the output format and destination address of the push notification. For configuration options, see §1.

6.1 Push Methods

6.1.1 HTTP POST

When HTTP push notifications are configured, eLPR will send an HTTP POST request to the configured push address. The value of this push address should be of the form:

```
http://server.com/page
```

HTTP basic access authentication is supported using an address of the form:

```
http://user:password@server.com/page
```

For example:

```
http://lpruser:secret@www.ipconfigure.com/elpr/listener.php
```

6.1.2 TCP

The TCP method allows license plate result data to be pushed directly to a listening port on an IP address. When this method is configured, the push address must specify the destination IP address and port using the form:

```
0.0.0.0:000
```

For example:

```
192.168.103.20:1234
```

6.2 Push Formats

All push formats contain seven named fields:

plate	Recognized license plate text
timestamp	Unix time stamp
date	Date string formatted according to configured camera settings
time	Time string formatted according to configured camera settings
plate_path	Full path to result's segmented license plate image
overview_path	Array of full paths to result's overview images
watch_hit	1 if the pushed result was found on the watchlist, 0 otherwise.

6.2.1 URL-Encoded form data

This format is most useful when using the HTTP POST push method, but may be configured for TCP as well. Here the push message body contains the fields described in §6.2, with the members of `overview_path` URL-encoded and specified as `overview_path[0]`, `overview_path[1]`, ..., `overview_path[`n`]`. When using HTTP POST, the `Content-Type` will be set to `application/x-www-form-urlencoded`.

Sample URL-encoded form POST:

```
plate=CORTL&timestamp=1370906629&date=2013%2D06%2D10&time=23%3A23%3A49&plate_path=http%3A%2F%2F \
192%2E168%2E103%2E181%2Flocal%2Felpr%2Fresults%2F13709%2F0%2Fplt%5F662900168%2Ebmp&overview_pat \
h%5B0%5D=http%3A%2F%2F192%2E168%2E103%2E181%2Flocal%2Felpr%2Fresults%2F13709%2F0%2Fovr%5F662900 \
168%2Ejpg&overview_path%5B1%5D=http%3A%2F%2F192%2E168%2E103%2E181%2Flocal%2Felpr%2Fresults%2F13 \
709%2F0%2Fovr%5F662900167%2Ejpg&overview_path%5B2%5D=http%3A%2F%2F192%2E168%2E103%2E181%2Flocal \
%2Felpr%2Fresults%2F13709%2F0%2Fovr%5F662900166%2Ejpg&watchlist_hit=0
```

For example, if the above were posted to a PHP script, the `$_POST` variable would contain:

```
Array
(
     [plate] => CORTL
     [timestamp] => 1370906629
     [date] => 2013-06-10
     [time] => 23:23:49
     [plate_path] => http://192.168.103.181/local/elpr/results/13709/0/plt_662900168.bmp
     [overview_path] => Array
     (
          [0] => http://192.168.103.181/local/elpr/results/13709/0/ovr_662900168.jpg
          [1] => http://192.168.103.181/local/elpr/results/13709/0/ovr_662900167.jpg
          [2] => http://192.168.103.181/local/elpr/results/13709/0/ovr_662900166.jpg
     )
     [watchlist_hit] => 0
)
```

6.2.2 Simple plain text

This format presents the data as a flat ASCII record. When using HTTP POST, the `Content-Type` header will be set to `text/plain`.

Record structure:

```
BEGIN PLATE
License plate
Unix timestamp
Date
Time
Plate URL
BEGIN OVERVIEWS
Overview 1 URL
Overview 2 URL
...
Overview n URL
END OVERVIEWS
watchlist hit
END PLATE
```

Sample record:

```
BEGIN PLATE
CORTL
1370907115
2013-06-10
23:31:55
http://192.168.103.181/local/elpr/results/13709/0/plt_711500004.bmp
BEGIN OVERVIEWS
http://192.168.103.181/local/elpr/results/13709/0/ovr_711500004.jpg
http://192.168.103.181/local/elpr/results/13709/0/ovr_711500003.jpg
http://192.168.103.181/local/elpr/results/13709/0/ovr_711400002.jpg
END OVERVIEWS
0
END PLATE
```

6.2.3 JSON-encoded data

This push format presents the data as a set of JSON-encoded variables. When using HTTP POST, the `Content-Type` header will be set to `application/json`.

Sample JSON data:


```
{
     "plate":  "C0RTL",
     "timestamp":  1370907616,
     "date":  "2013-06-10",
     "time":  "23:40:16",
     "plate_path":  "http://192.168.103.181/local/elpr/results/13709/0/plt_761600004.bmp",

     "overview_path":  [
          "http://192.168.103.181/local/elpr/results/13709/0/ovr_761600004.jpg",
          "http://192.168.103.181/local/elpr/results/13709/0/ovr_761600003.jpg",
          "http://192.168.103.181/local/elpr/results/13709/0/ovr_761600002.jpg"
     ],
     "watch_hit":  0
}
```