

How to get recognition events from Overseer\Autocode LPR server

1. Overseer\Autocode works as a CLENT software.	2
1.1 An old style (starting Overseer 1.14 ver.).....	2
1.1.1 How to enable and configure	2
1.1.2 Packet structure.....	2
1.1.3 Examples.....	2
1.2 New style (starting Overseer 1.18.1.42 ver.).....	3
1.2.1 How to enable.....	3
1.2.2 Recognition event format.....	3
1.2.3 Event format content definition	4
1.2.4 Examples.....	4
2. Overseer\Autocode works as a SERVER software	5
2.1 New style (starting Overseer 1.18.1.42 ver.).....	5
2.1.1 How to enable and configure	5
2.1.2 Recognition event format.....	6
2.1.3 Event format content definition	6
2.1.4 Event format example.....	6
3. How to get data from Overseer\Autocode database.....	7
3.1 Prototype.....	7
3.2 function arguments by order	7
3.3 function return structure by order:	7
3.4 Examples:	7
3.5 PostgreSQL credentials:	8

1. Overseer\Autocode works as a CLENT software.

Overseer send UDP packets into defined socket without any delivery proof. So 3rd party software shall open socket and manage packets.

1.1 An old style (starting Overseer 1.14 ver.)

1.1.1 How to enable and configure (after software installation): edit "C:\Program Files (x86)\VIT\Overseer\ip.ose" or "C:\Program Files (x86)\VIT\Overseer\ip.ose" file depending on product type as following:

- enabled = true; // enable\disable feature
- ip = "127.0.0.1"; // destination socket IP addr
- port = 3344; // destination socket port

1.1.2 Packet structure: post delimiter cap delimiter time delimiter platenum delimiter media_id delimiter reservedParamater1 delimiter reservedParameter2 delimiter platear_id delimiter GPScoordinates comment;

- (1) post (decimal_number) - server (Overseer instance) ID;
- (2) cap (decimal_number) - camera ID (starting from 0);
- (3) time (decimal_number 64-bit integer) - event timestamp in unix-timestamp format in accurate to microseconds (ordinary unix timestamp + 6 digits);
- (4) platenum (utf8_string) - number plate string;
- (5) media_id (decimal_number) - image identifier. has the same ID for full, body and plate images;
- (6) reservedParamater1 (decimal_number) - 0 by default;
- (7) reservedParameter2 (decimal_number) - 0 by default;
- (8) platear_id (decimal_number) - number plate ID;
- (9) GPS coordinates (utf8_string).
- (10) delimiter (,) - parameters delimiter (comma)

How to use this data to get additional metadata from LPR database read plate_attr_get() function definition below.

1.1.3 Examples:

GPS device enabled:

553346752,2,1438947888275266,AA5888AB,1075,0,0,1074,N50.458271E30.446288

GPS device disabled:

553346752,2,1438947901455100,AI1667HH,1076,0,0,1075,S0.000000W0.000000.

1.2 New style (starting Overseer 1.18.1.42 ver.)

1.2.1 How to enable

Configure "c:\ProgramData\VIT\Autocode VMS\resources2\user\bridge.plist" file. Insert the next node:

```
incoming = [  
  default = 1000;  
  0 = 10;  
];
```

after `srv_encoding = "1251"; string.`

Config node example

```
NUUO = [  
  class = "Osbornstone";  
  settings = [  
    version = "3";  
    srv-encoding = "1251";  
    incoming = [  
      default = 1000;  
      0 = 10;  
    ];  
  ];  
  description = "Server";  
  log_parm = {  
    fname = "NUUO.log";  
    log_path = "mstone";  
    opts = "AorpF_LOG_FILE";  
    primask = "AorpF_LOG_EMERG_PRI";  
    attrmask = "AorpK_LOG_ATTRMASK_ALL";  
  };  
];
```

Parameters

- default - default events port
- 0 = 10 - means 1st channel (channel numerations starts from 0) will send events on 10 port.

1.2.2 Recognition event format

How to configure

Open file "c:\ProgramData\VIT\Autocode VMS\resources2\user\bridge.plist" Configure node

```
NUUO = [  
  kernel.kernel.user.100.0-osuti.send = [  
  ];  
  kernel.kernel.user.202.0-osuti.send = [  
  ];  
];
```

as

```
NUUO = [  
  kernel.kernel.user.100.0-osuti.send = [  
    format = "...";  
  ];  
];
```

J;
J;

1.2.3 Event format content definition

- (1) {plate_number:plate} - number plate
- (2) {plate_validity} - recognition accuracy
- (3) {timestamp:timestamp} - timestamp in datetime format
- (4) {timestamp} - timestamp in unixtimestamp format
- (5) {post_name} - Overseer instance\server name
- (6) {camera_info>ip} - video source information (based on NUUO software specification)
- (7) {camera_info>global_id} - video source information (based on NUUO software specification)
- (8) {camera_info>local_id} - video source information (based on NUUO software specification)
- (9) {evcode:opcode2eventname} - LPR event type
- (10) {channel_number} - recognition channel ID
- (11) {gate_info} - relay number in case of DIDO usage
- (12) {add_money:money} - money transfer amount in case of Retes feature usage
- (13) {post_id} - Overseer instance\server ID
- (14) {plate_id:guid} - LPR event GUID
- (15) {rn:tag} - \r\n statement
- (16) {n:tag} - \n statement

1.2.4 Examples

NUUO based event format:

```
format =
"<OMSG><Plate>{plate_number:plate}</Plate><Confidenc>{plate_validity}</Confidenc>
<Time>{timestamp:timestamp}</Time><Post>{post_name}</Post><Camera><IP>{camera
_info>ip}</IP><GlobalID>{camera_info>global_id}</GlobalID><LocalID>{camera_info>loc
al_id}</LocalID></Camera><Event>{evcode:opcode2eventname}</Event><Info><channel
>{channel_number}</channel><gate-info>{gate_info}</gate-info><add-
money>{add_money:money}</add_money></Info><PostId>{post_id}</PostId><EventId>{
plate_id:guid}</EventId></OMSG>";
```

2. Overseer\Autocode works as a SERVER software

3rd party software shall connect to Overseer based socket and listen events. There are no specific features to mark event as read or missed.

2.1 New style (starting Overseer 1.18.1.42 ver.)

2.1.1 How to enable and configure

Configure "c:\ProgramData\VIT\Autocode VMS\resources2\user\bridge.plist" file.

1. Insert the next node after `srv_encoding = "1251";` string:

```
target = [  
  port_bind = 5000;  
  dest_addr = "192.168.101.211";  
  dev_name = "/dev/oti/socket/tcp";  
];
```

2. change mode to 2: `version = "2";`

It is possible to create the only 1 destination target.

Config node example:

```
NUUO = [  
  class = "Osbornstone";  
  settings = [  
    version = "2";  
    srv-encoding = "1251";  
    target = [  
      port_bind = 5000;  
      dest_addr = "192.168.101.211";  
      dev_name = "/dev/oti/socket/tcp";  
    ];  
  ];  
  description = "Server";  
  log_parm = {  
    fname = "NUUO.log";  
    log_path = "mstone";  
    opts = "AorpF_LOG_FILE";  
    primask = "AorpF_LOG_EMERG_PRI";  
    attrmask = "AorpK_LOG_ATTRMASK_ALL";  
  };  
];
```

Parameters

- port_bind - target port
- dest_addr - target IP adds
- dev_name - socket type (tcp/udp).

2.1.2 Recognition event format

How to configure

Open file "c:\ProgramData\VIT\Autocode VMS\resources2\user\bridge.plist" Configure node

```
NUUO = [  
    kernel.kernel.user.100.0-osuti.send = [  
        ];  
    kernel.kernel.user.202.0-osuti.send = [  
        ];  
    ];
```

as

```
NUUO = [  
    kernel.kernel.user.100.0-osuti.send = [  
        format = "...";  
    ];  
    ];
```

2.1.3 Event format content definition

- (1) {plate_number:plate} - number plate
- (2) {plate_validity} - recognition accuracy
- (3) {timestamp:timestamp} - timestamp in datetime format
- (4) {timestamp} - timestamp in unixtimestamp format
- (5) {post_name} - Overseer instance\server name
- (6) {camera_info>ip} - video source information (based on NUUO software specification)
- (7) {camera_info>global_id} - video source information (based on NUUO software specification)
- (8) {camera_info>local_id} - video source information (based on NUUO software specification)
- (9) {evcode:opcode2eventname} - LPR event type
- (10) {channel_number} - recognition channel ID
- (11) {gate_info} - relay number in case of DIDO usage
- (12) {add_money:money} - money transfer amount in case of Retes feature usage
- (13) {post_id} - Overseer instance\server ID
- (14) {plate_id:guid} - LPR event GUID
- (15) {rn:tag} - \r\n statement
- (16) {n:tag} - \n statement

2.1.4 Event format example

NUUO based event format:

format =

```
"<OMSG><Plate>{plate_number:plate}</Plate><Confidency>{plate_validity}</Confidency>  
<Time>{timestamp:timestamp}</Time><Post>{post_name}</Post><Camera><IP>{camera  
_info>ip}</IP><GlobalID>{camera_info>global_id}</GlobalID><LocalID>{camera_info>loc  
al_id}</LocalID></Camera><Event>{evcode:opcode2eventname}</Event><Info><channel  
>{channel_number}</channel><gate-info>{gate_info}</gate-info><add-  
money>{add_money:money}</add_money></Info><PostId>{post_id}</PostId><EventId>{  
plate_id:guid}</EventId></OMSG>";
```

3. How to get data from Overseer\Autocode database

plate_attr_get() - external PostgreSQL based function to get event metadata from Overseer\Autocode database.

3.1 Prototype

plate_attr_get(text, timestamp without time zone, timestamp without time zone, numeric, character varying);

3.2 function arguments by order

- (1) number plate template (text) - could contain POSIX regexp, wildcard template with "*" char, or ordinary license plate full string to check for exact equality;
- (2) timestamp "from" (timestamp) - datetime in string format based on PostgreSQL settings, or NULL for open "from" interval;
- (3) timestamp "to" (timestamp) - datetime in string format based on PostgreSQL settings, or NULL for open "to" interval;
- (4) "Levenshtein distance" (https://en.wikipedia.org/wiki/Levenshtein_distance) (numeric) - 0.0 by default. Or NULL in case you need not use this feature.
- (5) image format (text) - define image type to return on request. could be chosen among the next:
 - none or NULL - no image data to return
 - full - return full frame image
 - body - return vehicle face crop from full image
 - plate - returns plate number crop from full image

3.3 function return structure by order:

- (1) event_id (integer) - record ID
- (2) event_ctime (timestamp) - date time
- (3) event_media_id (integer) - media (image) ID
- (4) event_device_id (integer) - camera ID
- (5) event_channel (integer) - recognition channel ID
- (6) event_channel_ds (text) - recognition channel description
- (7) event_post_id (integer) - server (Overseer instance) ID
- (8) event_number (text) - plate number
- (9) plate_about (text) - vehicle description from card file, if exist
- (10) event_image (bytea) - image in format depending on image format parameter

3.4 Examples:

- SELECT * FROM plate_attr_get('*',NULL,NULL,NULL,NULL); -- will return full events list without images
- SELECT * FROM plate_attr_get('A*', '01-01-2015', NULL, NULL, 'full') -- will return events for plates starting from "A" letter, starting from 01-01-2015 with full images

- `SELECT * FROM plate_attr_get('AA85AA','01-01-2015',NULL, 2, 'body')` -- will return plates like 'AA85AA' with Levenshtein distance ≤ 2 starting from '01-01-2015' with vehicle face images.
- `SELECT * FROM plate_attr_get('^[AK].*00.*$', '01-03-2015', '01-04-2015', NULL, 'plate')` -- will return events for plate numbers started from A or K letter AND 0 pair inside plate string AND for March month AND with plate crop image

3.5 PostgreSQL credentials:

- server = by default localhost for each Overseer\Autocode instance. It is possible to install on dedicated database server (Linux preferred)
- port = 5432 (default)
- user = autocode
- password = autocode
- default folder = c:\postgres, It is possible to change working folder on installation step, or data folder after installation.