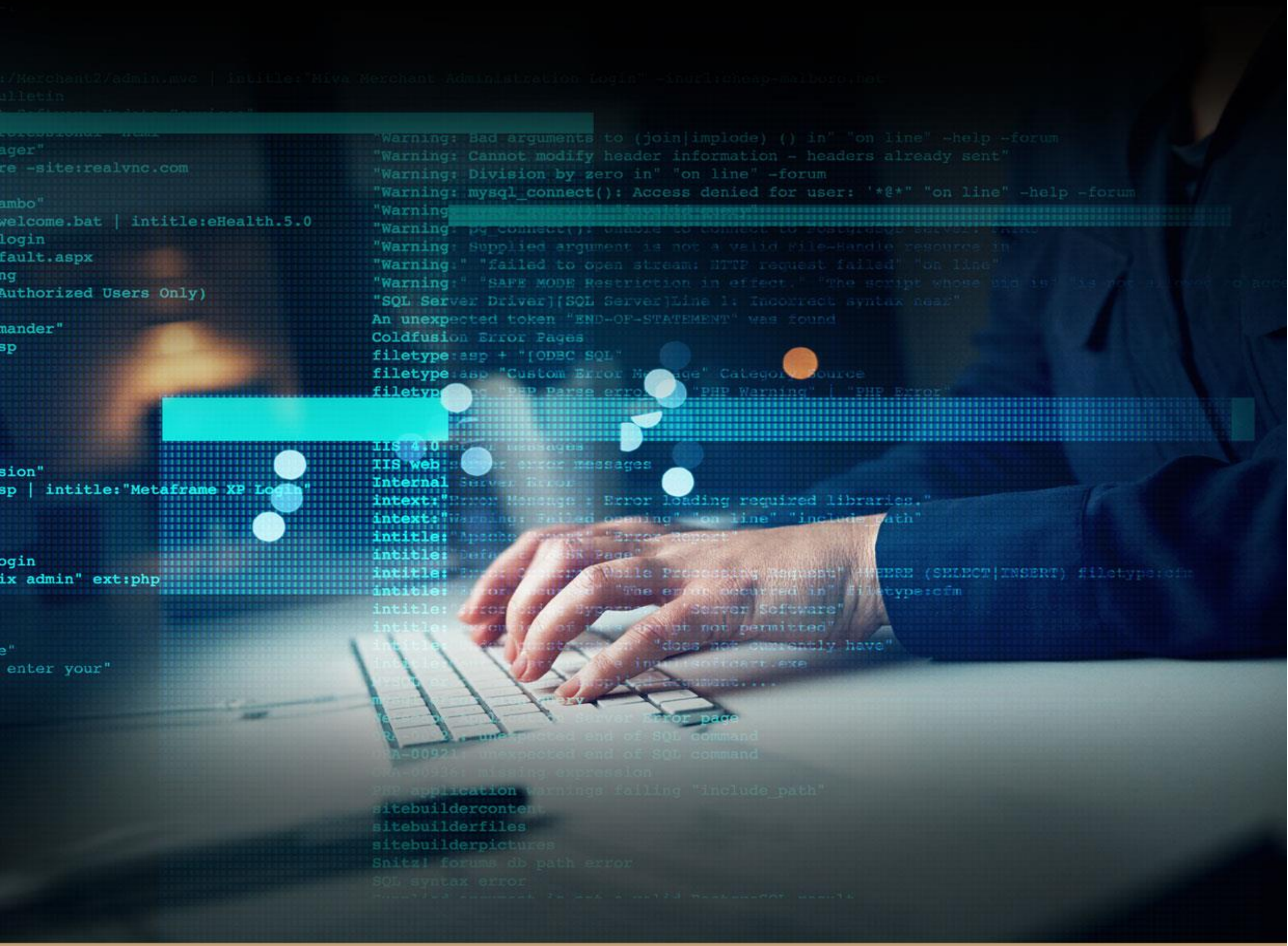
ADAPTIVE RECOGNITION

AUTOMATIC NUMBER PLATE RECOGNITION

Programmer's Guide

Carmen®Go Data Stream Output SDK



CARMEN® GO Data Stream Output SDK

Programmer's Guide

Document version: 1.0.0

Table of Contents

1. INTRODUCTION

This document helps you create the various SDKs required to handle the Event Data Interface of CARMEN® GO software.

2. DESCRIPTION OF CARMEN® GO DATA INTERFACE

CARMEN® GO transmits an event data package through the data output for each license number detection, if it is activated. The data packages consist of two parts: a property descriptor that is transmitted in a compressed picture that contains the image of the event in JPEG format.

Event data includes the following information:

- Video source ID *(sourceId) [integer, 32bit]:*
  This is the position number of the video source in which the license plate number was identified.
- Timestamp *(timestamp) [unsigned integer, 64bit]:*
- Point of time when the frame was received. It is given according to the Epoch / Unix time and is corrected for local time.
- License plate number *(plateText) [C string, ASCII encoding]:*
- License plate number: the actual characters detected on the frame.
- License plate country code *(plateType) [C string, ASCII coding]:*
- Identifies the country / region where the license plate is used, based on the format / characteristic of the license number.
- License number State Code - if available *(plateState) [string, ASCII encoding]:*
- Identifies the State or Region within the country where the license plate is used, based on the format / characteristic of the license number.
- Image size *(image number) [integer, 32bit]:*
  The compressed size of the source frame in bytes and in compressed JPEG data.
- Image data *(byte array):*
  Compressed image data is created based on the JPEG data size.

The data channel for receiving events is based on the primary IP address which runs CARMEN® GO, via the service port configured on CARMEN® GO web interface.

Example

It is running on a computer with an IP address of 192.168.1.15 in a local area network where CARMEN® GO 23 port is configured with the following client socket:

protokoll:	TCP
IP-cím:	192.168.1.15
port:	23

3. CARMEN® GO C LANGUAGE SDK SOFTWARE DEVELOPMENT KIT API

Required files for C LANGUAGE SDK API development:

- *carmengo_clib.h*
  This header file contains the function statements used by the API.
- *libcarmengo_capi.dll*
  This file contains the CARMEN GO API runtime code. This file is required for the program which use CARMEN GO API at runtime.
- *libcarmengo_api.dll.a*
  This statically linked import library file provides runtime loading of exported methods in libcarmengo_api.dll.

The structure of the event data structure used in C API (CMGO_DATACHANNEL_READ_RESULT):

- source_id: video source ID
- timestamp: frame time stamp
- plate_text: license plate with UTF-8 character encoding
- plate_type: nationality code country code
- jpeg_size: frame size in bytes
- jpeg_data: frame binary data (JPEG)

First, use the *cmgo_create_datachannel* method to initialize a new data channel with the desired connection data (the CARMEN GO IP address + data interface port number). The IP address must be entered in [0-255]. [0-255]. [0-255]. [0-255] (e.g. 192.168.1.15). When entering correct data, the recovered *handle* represents the CARMEN GO data interface.

In the next step, *cmgo_datachannel_connect* must initiate the connection structure, which, if it does not finish within the set time, will fail to contact.

Receiving events takes place by repeatedly invoking the *cmgo_datachannel_read* method. Received event packages are delivered through the CMGO_DATACHANNEL_READ_RESULT structure, from which the event data can be extracted directly. The image for the event is available in JPEG format.

You can take initiative for a new event package for each scan. The scanning operation has a timeout, basically, if no new event is received on the data channel after 10 seconds, the call returns with an error.

The CMGO_DATACHANNEL_READ_RESULT structure can be deleted with the *cmgo_datachannel_free_result* method.

After completing these tasks, you must close the connection with the *cmgo_datachannel_disconnect* method and then use the *cmgo_release_datachannel* method to delete the *handle*.

Example

```
#include "carmengo_api_clib.h"

#include <stdio.h>
#include <mem.h>

int main(int argc, char** argv)
{
    int st;
    char address[] = "127.0.0.1";
    unsigned int port = 23;

    CMGO_DATACHANNEL_HANDLE handle = NULL;
    CMGO_DATACHANNEL_READ_RESULT result;

    memset(&result, 0, sizeof(result));

    printf("Creating DataChannel [%s,%u] ...\n", address, port);
    st = cmgo_create_datachannel(address, port, &handle);
    if (0 != st) {
        printf("Couldn't create DataChannel!\n");
        return -1;
    }
    printf("DataChannel Created\n");

    printf("Connecting...\n");
    st = cmgo_datachannel_connect(handle, 10000);
    if (0 != st) {
        printf("Couldn't connect!\n");
        return -1;
    }
    printf("Connected!\n");

    const int N = 10; // max count of the events to read
    for(int i = 0; i < N; i++) {
        printf("Waiting For Result [%d]!\n", i);
        if(0 == cmgo_datachannel_read(handle, &result, 10000)) {
            printf("Result received\n");
            printf(" - Plate: %s\n", result.plate_text);
            printf(" - JPEG Size: %d\n", result.jpeg_size);
            printf("\n");
            fflush(stdout);
        } else {
            printf("Waiting Timed Out!\n");
            fflush(stdout);
        }
    }

    printf("Free Result Structure\n");
    cmgo_datachannel_free_result(&result);

    printf("Disconnect\n");
    cmgo_datachannel_disconnect(handle);

    printf("Release DataChannel\n");
    cmgo_release_datachannel(handle);

    return 0;
}
```

Contact Information

Headquarters:
Adaptive Recognition Hungary Inc.
Alkotás utca 41 HU-
1123 Budapest Hungary
Phone: +36 1 201 9650
Fax: +36 1 201 9651
Web: www.adaptiverecognition.com

Service Address:
Adaptive Recognition Hungary Inc.
Ipari Park HRSZ1113/1 HU
2074 Perbál Hungary
Phone: +36 1 2019650
E-mail: rmarequest@adaptiverecognition.com

Adaptive Recognition Hungary Technical Support System (ATSS) is designed to provide you the fastest and most proficient assistance, so you can quickly get back to business.
For further technical information about our products, please visit our official website.

Information regarding hardware, software, manuals and FAQ are easily accessible for customers who previously registered to enter the dedicated ATSS site. Besides offering assistance, the site is also designed to provide maximum protection while managing your business information and technical solutions utilized.

New User
If this is your first online support request, please create an account by clicking on this link.

Returning User
All registered ATSS customers receive a personal access link via e-mail. If you previously received a confirmation message from ATSS, it contains the embedded link that allows you to securely enter the support site.

If you need assistance with login or registration, please contact atsshelp@adaptiverecognition.com for help.