ADAPTIVE RECOGNITION



Carmen[®] User Manual

ANPR RESULT:	MMR RESULT:
CU 22 069 🤡 CU 22 069	MAKE: 🕑 AUDI
COUNTRY: 🤣 DENMARK	MODEL: 🥥 A8 AUOI
MAIN BACKG. COLOR: 🥑 WHITE	COLOR: 🥑 GREY
MAIN FONT COLOR: 🥑 BLACK	CATEGORY: 🥑 CAR 🚗
DATE: 🤗 16:36:21 08.04.23	VIEWPOINT: 🥑 FRONT SIDE 🔗
31 // Load ANPR engine for	european region
32 cmAnpr anpr = new cmAnpr	("eur");
33	
24 fonoach (imago in imagos	×
³⁴ Toreach (Image In Images	,

User Manual for Carmen[®] related applications/tools/demos.

Carmen[®] User Manual

Document version: 2024-11-13

Table of Contents

CARMEN LICENSING	6					
LICENSE MANAGER						
1. INTRODUCTION	10					
2. WHAT IS THE PURPOSE OF THE LICENSE MANAGER	10					
3. INSTALLATION OF THE LICENSE MANAGER	11					
4. OPENING THE LICENSE MANAGER	11					
4.1. ON WINDOWS	11					
4.2. ON LINUX	11					
5. STRUCTURE OF THE LICENSE MANAGER	12					
5.1. STATIC LICENSES	12					
5.2. DYNAMICALLY UPLOADABLE LICENSES	13					
5.3. UPLOAD LICENSES	14					
6. LICENSE INSTALLATION	15					
7. LICMAN	16					
8. TROUBLESHOOTING	20					
ENGINE MANAGER	21					
1. INTRODUCTION	21					
2. ENGINE INSTALLATION	22					
2.1. INSTALLATION ON WINDOWS	23					
2.2. INSTALLATION ON LINUX	24					
3. ENGINE MANAGER PRO APPLICATION	25					
4. UNINSTALLING ENGINE(S)	28					
4.1. FROM WINDOWS	28					

4.2. FROM LINUX	
5. ENGMAN	29
LICENSE SERVER	32
1. INTRODUCTION	32
2. SERVER APPLICATION	
2.1. SERVER APPLICATION AS CONSOLE APPLICATION	
2.2. SERVER APPLICATION AS A SERVICE	
2.3. SERVER APPLICATION FOR MORE THAN ONE CLIENT	
3. ConfigLSClient	
VIDEO SDK	40
1. INTRODUCTION	40
2. ENGINE AND LICENSE REQUIREMENTS	40
3. BUILD REQUIREMENTS	41
3.1. LINUX	41
3.2. WINDOWS	43
4. VIDEO INPUT	45
4.1. NETWORK STREAM	45
4.2. VIDEO FILE	46
5. A MINIMAL APPLICATION EXAMPLE	47
5.1. CREATE ANPR	47
5.2. ONEVENTCALLBACK	
5.3. CREATE STREAMPROCESSOR OBJECT	49
5.4. START STREAMPROCESSING	
6. CONSTRUCTING COMPONENTS	51
6.1. ANPR BUILDER	51
6.2. MMR BUILDER	
6.3. ADAPTIVE RECOGNITION CLOUD BUILDER	
6.4. STREAMPROCESSOR BUILDER	53
6.5. LOGGER	
6.6. LICENSING	
7. RESULT CLASSES	60
7.1. IMAGE CLASSES	60
7.2. ANPR (AUTOMATIC NUMBER PLATE RECOGNITION) CLASSES	60
7.3. MMR (MAKE & MODEL RECOGNITION) CLASSES	60
7.4. EVENT CLASSES	61
8. REGION LIST	62
9. KNOWN ISSUES	63
ADI DEMO	64
1. INTRODUCTION	64

2. MAIN SCREEN	
3. FILE MENU	
3.1. OPEN IMAGE(S)	
3.2. OPEN DIRECTORY	
3.3. SAVE LOG	
4. EDIT MENU	67
4.1. ANPR / MMR SETTINGS	
4.2. DEMO PREFERENCES	
5. PROCESS MENU	
6. HELP MENU	71
7. SET ROI/ROU ON THE IMAGE	72
8. MAGNIFIER	
9. ANPR RESULT TREE	
10. ANPR DATA	
11. DATA LOGGING	
ADV DEMO	
1. INTRODUCTION	
2. MAIN SCREEN	
3. FILE MENU	
4. EDIT MENU	
4.1. TRIGGER CONFIGURATION	
4.2. ANPR THREADS	
4.3. ANPR SETTINGS	
4.4. ANPR PROCESSING PREFERENCES	
4.5. LOG SETTINGS	
5. VIEW	
5.1. FONTS	
5.2. RESULT IMAGE	
5.3. TABLE COLUMNS:	
5.4. TABLE ROW HEIGHT	
6. CAMERA MENU	
7. VIDEO PLAYER MENU	94
8. PROCESS MENU	
9. HELP MENU	
10. ANPR RESULT TREE	96
11. ANPR DATA	
12. DATA LOGGING	
ODI DEMO	
1. INTRODUCTION	

2. MAIN SCREEN	
3. FILE MENU	
3.1. OPEN DIRECTORY	
3.2. OPEN IMAGES	
3.3. CLOSE	
4. EDIT MENU	
4.1. ENGINE SETTINGS	
4.2. DEMO PREFERENCES	
5. PROCESSING MENU	
6. HELP MENU	
7. OCR RESULT TREE	
8. OCR DATA	
9. DATA LOGGING	
SDK DESCRIPTION	
1. ANPR SAMPLE CODES	
CMANPR01	
CMANPR02	
CMANPR03	
CMANPR04	
CMANPR05	
CMANPR06	
CMANPR07	
CMANPR08	
CMANPR09	
MMR01	
2. OCR SAMPLE CODES	
CMOCR01	
CMOCR02	
CMOCR03	
3. GX SAMPLE CODES	
GXDEVICES01	
GXDEVICES02	
GXLICENSES01	
4. EXAMPLE CODES AVAILABILITY	
ON WINDOWS	
ON LINUX	
CONTACT INFORMATION	

CARMEN LICENSING

Adaptive Recognition's Carmen® engines, known for their advanced ANPR, OCR, and document recognition capabilities, are designed with robust hardware key protection to ensure secure usage. These engines are embedded into products such as SDKs, middleware, service-type applications, and hardware devices like cameras and scanners, providing reliable, seamless performance in a variety of applications.

There are several methods available for utilizing these engines. One option involves using a hardware key, which can be connected directly to the computer running the engine. This ensures that the necessary licenses are applied effectively.

In addition to direct hardware key use, the License Server offers a flexible alternative. It allows licenses to be managed centrally, providing added convenience by removing the need for a direct hardware key connection to each device.

Additionally, a cloud-based solution is now available, offering even greater flexibility. This model requires only an internet connection, eliminating the need for physical hardware keys while providing the same reliable access to the engines and their capabilities.

In these setups, the Hardware Key, License Server, and License Server Client all work seamlessly to provide various options for managing and deploying the Carmen® engines based on specific needs.

As it was mentioned before, there are two methods of utilizing these engines, both necessitating the purchase of a lifetime license file and its installation on the **hardware key**. The first method involved using the key directly by plugging it into the computer where the engine ran (with USB and PCI Express versions available).





The second method required setting up a License Server application to share the licenses over the network, allowing multiple engine-using devices to work.

This model consists of a customer network with three main hosts:



- 1. A machine running customer software and the recognition engine.
- 2. A camera that also runs the engine.
- 3. A License Server application that serves both the machine and the camera, hosting the hardware key(s) for them.

This arrangement enables centralized management of licenses and offers flexibility in distributing the engine across multiple devices within the network.

Recently, a third, **cloud-based option** has been introduced, in addition to the existing two. With this model, customers are not required to purchase a hardware key or license. Instead, they must register on the Carmen Cloud website and subscribe to the Carmen® License Key Rental service. For Carmen Go, use the "Carmen GO" option; for all other products (e.g. Carmen FreeFlow SDK, Carmen Video SDK) choose "Carmen FreeFlow". The number of options you purchase determines the maximum number of license locks (image reads) in parallel. After obtaining an API key and running the **License Service Client** executable or deploying its lightweight docker image within their own infrastructure, they can begin using the product immediately. For more information, please visit our page here: https://carmencloud.com/docs/content/downloadable-products/license-service-client/tutorial



In the cloud-based option, the customer network still comprises three hosts:

- 1. A machine running customer software and the recognition engine.
- 2. A camera running the engine.
- 3. The License Service Client serving both the machine and the camera, with a connection to the Carmen Cloud' public license service.

This modern approach leverages cloud technology to manage licenses, eliminating the need for physical hardware keys. The cloud-based infrastructure allows for scalable, remote management, and provides an accessible pathway to adopting newer licensing models.

The License Service Client executable or docker image can be used as an alternative to the License Server application. However, it does not require a local hardware key since it utilizes Carmen Cloud's public license service.

For more information, please visit our page here:

https://carmencloud.com/docs/content/downloadable-products/license-service-client/tutorialexecutable



REQUESTINFO@ADAPTIVERECOGNITION.COM



LICENSE MANAGER

1. INTRODUCTION

Thank you for choosing Adaptive Recognition's Optical Character Recognition technology. This short document will detail the use of the License Manager utility. If you have further questions after going through this manual, feel free to contact Adaptive Recognition's Support Team at http://www.adaptiverecognition.com/support/.

This application can be found in the following folder:

- On Windows: "C:\Program Files\Adaptive Recognition\Common Utils\LicenseManager\"
- On Linux: "\opt\gx64\LicenseManager\"

🖵 Note

The latest available version from License Manager: 7.3.2.0

Important

Please note, that this application is not available for ARM and CenOS6 packages on Linux.

🕻 Note

There are some videos on <u>YouTube</u> how to use our License Manager application. Feel free to check them as well.

2. WHAT IS THE PURPOSE OF THE LICENSE MANAGER

CARMEN[®] requires licenses to run. These licenses define which region-specific engines you are capable to run in what system configuration (single-core, dual-core, quad-core, etc.). The License Manager's main purpose is to help you install, update or delete these licenses by accessing your Hardware Key.

🛛 Note

Since your licenses are located on your Hardware Key, your already installed licenses will be visible in the License Manager application only if your Hardware Key is connected to your computer. Therefore, make sure that your Hardware Key (USB-dongle, PCI-e card, internal USB-key, etc.) is connected to your system.

3. INSTALLATION OF THE LICENSE MANAGER

The License Manager utility is part of all CARMEN® ANPR and OCR Software Package Installers above version 7.2.7.26 and installed automatically with your CARMEN® product. It is designed to provide the same look and features both under Windows and Linux operating systems. Further information about installation can be found in our installation manual.

OPENING THE LICENSE MANAGER 4.

4.1. ON WINDOWS

In order to launch the software, run the LicenseManager_x64.exe. You can start it either by locating the Imicon at the Adaptive Recognition > Common Utils > LicenseManager folder or Windows should be able to locate the application for you after pressing the 'Windows' key and typing 'License Manager'.

The default file path for the License Manager is: "C:\Program Files\Adaptive Recognition\Common Utils\LicenseManager\LicenseManager_x64.exe".

In older versions of CARMEN[®] (before 7.3.1.23) the application was in these folders:

- in case of 32bit OS: "C:\Program Files (x86)\ARH\Common Utils\LicenseManager"
- in case of 64bit OS: "C:\Program Files\ARH\Common Utils\LicenseManager"

4.2. **ON LINUX**

In case of Linux OS, the License Manager can be found under the following path: /opt/gx64/LicenseManager/

🛛 Note

In older Carmen[®] versions (before 7.3.1.23), on 32bit OS you can find the License Manager here: /opt/gx32/LicenseManager/

5. STRUCTURE OF THE LICENSE MANAGER

	(M License Manager 7.3.2.0 (64 bit)		n -					×
	ſ	-Static licenses Old license storage devices:	Licenses:						
		Serial num. Device type Lic. No. Lic. type Description							
(1) -			2.a					2.	b
		New license storage devices:	Licenses:						
		Serial num. Device type HW group	Lic. ID	Lic. date	HWID/HWGRP	Lic. type	Expiry date	Description	
		1190500 USB key	1084459	2023.05.09	1190500	Normal	2030.12.31	CARMEN Go Anpr (UNI)	- 11
			1084460	2023.05.09	1190500	Normal	2030.12.31	CARMEN Go Anpr (UNI)	
\frown			1084461	2023.05.09	1190500	Normal	2030.12.31	CARMEN Go Anpr (UNI)	
(2)			1084462	2023.05.09	1190500	Normal	2030.12.31	CARMEN Go Anpr (UNI)	
\smile			1084463	2023.05.09	1190500	Normal	2030.12.31	CARMEN Go Anpr (UNI)	
l		♥ Upload licenses 2.c 2.e	Last 1 Auto sel	icenses 🕜	Best licenses ad licenses	3	2.f	Refresh devices and license Online manua	1
	гI	License directory: G:/git/LicenseManage	r/bin/x64/lic	censes				Brows	se
		,							
		Saved user licenses:	Tin tumo F	un i nu dana	Da			load licenses Clear licens	
		LIC. ID LIC. date nwiD/hwGRP	LIC. Cype E	xpiry date	De	scription	3.c		
								Save changes Summary	
\bigcirc							5.e	Auto save after upload	
$(3)^{-}$	1						3.0	Auto summary after upload	0
\sim								create log file	
							(3.h)	Brows	e
		\frown					3.i		
		(3.b)							

5.1. STATIC LICENSES

Static licenses were used in older systems, but for backwards compatibility we are keeping this segment in the License Manager. This part will be empty for you most probably. If not, please consider to contact your Sales manager and get an update. Many improvements have been made in our engines since static licenses were in use.

So, if you would like to use the latest engine, thereby increase the overall accuracy and achieve better recognition with the newer license plate types, a hardware key change and license update would be necessary.



5.2. DYNAMICALLY UPLOADABLE LICENSES

Here you could see all your connected Hardware Keys and the installed licenses on them.

2.a: New license storage devices

You will find your connected Hardware Keys here. The key's serial number, device type (e.g.: USB key or PCIe card) and hardware group is shown.

2.b: Licenses

Under licenses you will find the currently installed licenses on your selected Hardware Key. You can run engines which were released before the 'Expire date' and having the same region as it is indicated in the 'Description'. For example, a CARMEN[®] ANPR (EUR) license with an "2020.12.31" expiry date will be able to run an EUR engine which was released before the end of December 2020.

2.c: Auto refresh devices

When enabled, License Manager refreshes your devices and licenses when a new Hardware Key is connected.

2.d: Upload licenses

Enable to upload new licenses on your Hardware Key.

2.e: Auto upload licenses

License manager will upload the selected Licenses automatically when this function is enabled. Last license means that when multiple licenses are available, the software will install the one with the latest 'License date'. When 'Best License' is selected, the program uploads the license with the latest Expiry date.

2.f Refresh devices and licenses

This button reloads your Hardware Keys and Licenses manually.

2.g Online manual

This button opens this manual.



5.3. UPLOAD LICENSES

Under this segment you are capable to upload licenses on your Hardware Key. Remember, this section is only visible when either the 'Upload licenses' or 'Auto upload licenses' is enabled.

3.a: License directory

Before you can select a license to upload, you have to choose the folder were that specific license is located.

🛛 Note

Please note, that you have to select the directory which contains the license and not the license itself!

3.b: Saved user licenses

License manager will show you the available licenses for the connected Hardware Key which are located in the *License directory* folder.

3.c: Upload licenses

This button will upload the selected licenses to your Hardware Key. If you would like to upload all of your licenses, select them all, but make sure that their 'License date' are the same, otherwise you will not be able to upload them!

Note

Please note, that in case you have some licenses uploaded to your Hardware Key, but you are trying to upload licenses with different "Lic. date", then it will delete all the current licenses from your Hardware Key and upload the newly selected ones on them.

3.d: Clear licenses

To delete all your licenses from your Hardware Key.

3.e: Save changes

Manually saves the previously made changes (when Auto save is not enabled).

3.f: Summary

To copy or save the current state of your Hardware Devices and Licenses.

3.g: Auto save after upload

Enable it to save the changes automatically after a license upload.

3.h: Auto summary after upload

Shows the summary after a license upload.

3.i: Create log file

When enabled the License Manager creates a log file to the selected directory.

6. LICENSE INSTALLATION

1) Make sure, that the Hardware Key, where you wish to upload your licenses is connected and visible in the License Manager!

Ι	Dinamically uploadable licenses								
П	New license storage devices:			licenses:					
	Serial num.	Device type	HW group	Lic. ID	Lic. date	HWID/HWGRP	HW date	Expiry date	Description
	1200005			111116	2020.07.01	1200005	2013.01.01	2021.06.30	CARMEN Go Anpr (NAM)
	1100000	COD NOT		111117	2020.07.01	1200005	2013.01.01	2021.06.30	CARMEN Accr
				111118	2020.07.01	1200005	2013.01.01	2021.06.30	CARMEN Accr
				111119	2020.07.01	1200005	2013.01.01	2021.06.30	CARMEN Accr
				111110	2020.07.01	1200005	2013.01.01	2021.06.30	CARMEN Accr
				111111	2020.07.01	1200005	2013.01.01	2021.06.30	CARMEN Anpr (NAM)

- 2) When Upload Licenses section not visible, enable Upload licenses checkbox (2.d).
 ✓ Upload licenses
- 3) Hit Browse (3.a) and Choose the directory where your license files are stored.

As soon as you chose the folder, the available licenses should be visible in the **Saved user licenses** segment.

Saved user li	censes:						
Lic. ID	Lic. date	HWID/HWGRP	HW date	Expiry date	Description	Upload licenses	Clear licenses
111116	2020.07.01	1200005	2013.01.01	2021.06.30	CARMEN Go Anpr (NAM)	Saue changes	Cummo ru
111117	2020.07.01	1200005	2013.01.01	2021.06.30	CARMEN Accr	Save changes Summary	
111118	2020.07.01	1200005	2013.01.01	2021.06.30	CARMEN Accr	✓ Auto save after	upload
111119	2020.07.01	1200005	2013.01.01	2021.06.30	CARMEN Accr	🗸 Auto summary af	ter upload
111110	2020.07.01	1200005	2013.01.01	2021.06.30	CARMEN Accr	Create log file	
111111	2020.07.01	1200005	2013.01.01	2021.06.30	CARMEN Anpr (NAM)		

🛛 Note

You cannot see the licenses listed? You may have chosen the license file, **not the folder itself! Select** only the **directory** and click on **Choose** again! Also, only those licenses will be visible which are dedicated to your connected Hardware Key. Check whether the Hardware Keys Serial number matches the serials indicated in the license (HWID/HWGRP).

- 4) When you are uploading new licenses, your old licenses will be removed automatically from your Hardware Key unless the License dates are the same. Yet, if you wish to delete your old licenses manually, you may do now by pressing the **Clear licenses** (3.d) button.
- 5) Select the licenses you wish to upload to your Hardware Key by click and drag your mouse over the requested licenses. You can also select all the licenses with the Ctrl+A shortkey. By pressing Ctrl while selecting the licenses you are capable to select multiple licenses as well.
- 6) Click on **Upload licenses** (3.c) to upload the highlighted licenses. Your new licenses should be visible under the **Licenses** section.

Your new licenses are now installed on your Hardware Key!

7. LICMAN

CONSOLE APPLICATION

On Linux you can find this application in this folder: **/opt/gx64/LicMan/** On Windows this application is not part of the installed package, please turn to <u>AR support</u> if you need it.

If you run this application in Terminal, or Command Prompt: you will get the following help: Use: LicMan_x64.exe [version | -h | -?] | [list [-s -d | -fw | -raw | -SN:serial {-t:type}] | summary [-raw] | add -p:path {-SN:serial {-t:type}} {-save} | clear {-SN:serial {-t:type}} {-save}] {-json} {-o:outputfile}] Copyright © 2017-2022, Adaptive Recognition

Parameters:			
version	Displays the version information of the program.		
-h, -?	Shows this help.		
list	Lists the license storage devices and the licenses		
-s	static licenses		
-d	dinamically uploadable licenses		
-fw	Shows the firmware code version		
-raw	Lists licenses in raw data format		
summary	Create the summary of the dinamically uploadable licenses.		
-raw	Lists licenses in raw data format		
add	Uploads the licenses from a specified directory.		
-p:path	the path of the directory contains the licenses.		
clear	Clear the licenses from the device.		
Global paramete	r (all commands except version and help):		
-json	The program create json format output.		
-o:outputfile	The program redirects the output from display to the outputfile.		

Common parameters (add and clear):

-SN:serial -t:type	dinamically uploadable licenses selected by the serial number and/or device type Device type to select a device. Usable values: "USB key" "Combo Smart" "Combo Scan" "PRMc" "PCIe card" "CARMEN SPI" "CAPMEN 120"
-save	After successfully uploading or clearing the program saves the changes.

Some examples:

Gets the summary about the devices and licenses into summary.txt:

Command: LicMan_x64.exe/LicMan_x86_64.out summary -o:summary.txt

Output: The "summary.txt" file in the same folder with the following content: Summary of the devices and licenses

-

Devices:	
=======	
[1]: Device Type: USB	key (1), Serial: 1111111
Code version:	3.8
Code subversion:	0.1
Code type:	USB key
Max. license:	32
Memory size:	128 K
Time credit:	Supported
*****	***************************************

Licenses:

====	====		
Num.	License Type	Count	Description
1 2 3 4	99901004 1affffff 30ffffff 16ffffff	1 4 1 1	CARMEN Core 4 CARMEN Anpr (EUR) MMR (EUR) CARMEN Ocr (ISO)

Check the version of this application:

Command: LicMan_x64.exe/LicMan_x86_64.out version Output: License Manager (console version) Version: 7.3.1.18 Copyright © 2017-2022, Adaptive Recognition

Print the dynamically uploadable licenses in raw format:

Command: LicMan_x64.exe/LicMan_x86_64.out list -d -raw Output: Licenses in the system:

[1]: HWID: 01111111, Device Type: USB key (1), Serial: 1111111 :

606376-20210914-01111111-20130101--16ffffff-ffffffff-20220630-0000000--00000000-CARMEN Ocr (ISO) 606380-20210914-01111111-20130101--99901004-ffffffff-20220630-00000000--00000000-CARMEN Core 4 606381-20210914-01111111-20130101--1affffff-ffffffff-20220630-00000000--00000000-CARMEN Anpr (EUR) 606382-20210914-01111111-20130101--1affffff-ffffffff-20220630-00000000--00000000-CARMEN Anpr (EUR) 606383-20210914-01111111-20130101--1affffff-ffffffff-20220630-00000000--00000000-CARMEN Anpr (EUR) 606383-20210914-01111111-20130101--1affffff-fffffffff-20220630-00000000--00000000-CARMEN Anpr (EUR)

606384-20210914-01111111-20130101--1affffff-ffffffff-20220630-00000000--00000000-CARMEN Anpr (EUR)

606385-20210914-01111111-20130101--30ffffff-ffffffff-20220630-00000002--00000000-MMR (EUR)

Print the dynamically uploadable licenses, firmware (CodeVersion) in json format: Command: LicMan_x64.exe/LicMan_x86_64.out list -d -fw -json Output: {

```
"NewLicdevs": [
         {
              "Num": 1,
              "HWID": "01111111",
              "DevType": "USB key",
              "DevTypeID": 1,
              "Serial": 1111111,
              "CodeVersion": "3.8",
              "Licenses": [
                  {
                       "LicID": " 606376",
                       "LicDate": "2021.09.14",
                       "ExpDate": "2022.06.30",
                       "Desc": "CARMEN Ocr ( ISO )"
                  }
                  {
                       "LicID": " 606380",
                       "LicDate": "2021.09.14",
                       "ExpDate": "2022.06.30",
                       "Desc": "CARMEN Core 4"
                  },
                  {
                       "LicID": " 606381",
                       "LicDate": "2021.09.14",
                       "ExpDate": "2022.06.30",
                       "Desc": "CARMEN Anpr ( EUR )"
                  },
                  {
                       "LicID": " 606382",
                       "LicDate": "2021.09.14",
                       "ExpDate": "2022.06.30",
                       "Desc": "CARMEN Anpr ( EUR )"
                  },
                  {
                       "LicID": " 606383",
                       "LicDate": "2021.09.14",
                       "ExpDate": "2022.06.30",
                       "Desc": "CARMEN Anpr ( EUR )"
                  },
                  {
                       "LicID": " 606384",
                       "LicDate": "2021.09.14",
                       "ExpDate": "2022.06.30",
                       "Desc": "CARMEN Anpr ( EUR )"
                  },
                  {
                       "LicID": " 606385",
                       "LicDate": "2021.09.14",
                       "ExpDate": "2022.06.30",
                       "Desc": "MMR (EUR)"
                  }
             ]
        }
    ]
}
```

Clear licenses from all attached devices and save this modification: Command: LicMan_x64.exe/LicMan_x86_64.out clear -save Output: [1] Device type: USB key, Serial: 1111111 clearing licenses...OK

Important

If you want to make this change as permanent do not forget to put '-save' at the end of the command, otherwise you will lose the modification. The HW key will hold the modification until the HW key is attached, but once it is detached it will fall back to the original configuration.

Add licenses from a folder which contains the ukeys file for an exact attached HW Key temporarily Command: LicMan_x64.exe/LicMan_x86_64.out add -p: "c:\licenses" -SN:1111111 Output:

[1] Device type: USB key, Serial: 1111111:

```
Adding license: LicID:606366, Lic. Date: 2021.09.14, Expiry Date: 2022.06.30, Description: CARMEN Go 8 stream...OKAdding license: LicID:606367, Lic. Date: 2021.09.14, Expiry Date: 2022.06.30, Description: CARMEN Go 8 stream...OKAdding license: LicID:606368, Lic. Date: 2021.09.14, Expiry Date: 2022.06.30, Description: CARMEN Go Anpr (.NAM )...OKAdding license: LicID:606369, Lic. Date: 2021.09.14, Expiry Date: 2022.06.30, Description: CARMEN Go Anpr (.NAM )...OKAdding license: LicID:606369, Lic. Date: 2021.09.14, Expiry Date: 2022.06.30, Description: CARMEN Go Anpr (.NAM )...OKAdding license: LicID:606370, Lic. Date: 2021.09.14, Expiry Date: 2022.06.30, Description: CARMEN Go Anpr (.NAM )...OKAdding license: LicID:606371, Lic. Date: 2021.09.14, Expiry Date: 2022.06.30, Description: CARMEN Go Anpr (.NAM )...OKAdding license: LicID:606372, Lic. Date: 2021.09.14, Expiry Date: 2022.06.30, Description: CARMEN Go Anpr (.NAM )...OKAdding license: LicID:606373, Lic. Date: 2021.09.14, Expiry Date: 2022.06.30, Description: CARMEN Go Anpr (.NAM )...OKAdding license: LicID:606373, Lic. Date: 2021.09.14, Expiry Date: 2022.06.30, Description: CARMEN Go Anpr (.NAM )...OKAdding license: LicID:606374, Lic. Date: 2021.09.14, Expiry Date: 2022.06.30, Description: CARMEN Go Anpr (.NAM )...OKAdding license: LicID:606374, Lic. Date: 2021.09.14, Expiry Date: 2022.06.30, Description: CARMEN Go Anpr (.NAM )...OKAdding license: LicID:606374, Lic. Date: 2021.09.14, Expiry Date: 2022.06.30, Description: CARMEN Go Anpr (.NAM )...OKAdding license: LicID:606375, Lic. Date: 2021.09.14, Expiry Date: 2022.06.30, Description: CARMEN Go Anpr (.NAM )...OKAdding l
```

Important

If you just forgot to put '**-save**' at the end of the command, just run the previous command again, but using the '-save' parameter. The command will show you ERRORS, because the licenses are already on the HW key, but at the end of the process the command will save successfully, and you will not lose the modifications, once the HW Key is detached.

Important

It is allowed to upload only those licenses to the HW Key which has the same Lic Date.

Note

If you would like to update your licenses then the process would be the following:

- 1. Clear the current licenses from the attached HW Key(s) or from an exact HW Key
- 2. Add the licenses from a folder which contains the ukeys file(s) and save the changes

8. TROUBLESHOOTING

Problem observed	Solution
Installation	Disease to visit the Opening the Linence Menager section and palse sure
T cannot find the application!	 Please re-visit the <u>opening the License Manager</u> section and make sure that you were thoroughly followed the instructions written there! Make sure that Carmen[®] is installed on your computer! Please, run the Installer again. If it offers to install, choose that option. If it offers to Change, choose that option. Check whether the License Tools option is installed! If not, install it! Otherwise please contact support at <u>https://adaptiverecognition.com/support/</u>. If you are using Carmen[®] 7.3.1.22 or older, maybe you have installed the 32-bit variant of the License Manager. In that case find them at either the Program Files (x86) folder [Windows] or at /opt/gx32/LicenseManager/ [Linux].
License upload	
The choose button is greyed out! I cannot select my license file!	Select the directory which contains the license and not the license file itself. When the folder is selected the Choose button will be active!
I chose the directory where	• Only those licenses will be visible which are dedicated to your connected
not visible under the Saved User Licenses part, therefor I	Hardware Key. The first 8 numbers from the name of the license file should be the dedicated Hardware Key's ID. If they don't match, it means that license is generated for another Hardware Key!
cannot upload!	• Are you sure that your Hardware Key is connected? The present of the dedicated Hardware Key is a must, otherwise the License Manager will not show your licenses!
I uploaded some licenses to my Hardware Key, but my old licenses are gone! I want both the old and the new licenses on my Hardware Key!	 The Hardware Key can only hold licenses which are having the same Lic. Date. If your licenses are having different license dates, then unfortunately you cannot have them both on your Hardware Key. Please contact your sales person or our <u>support team</u> to request a new license with the same date as your old one!
	 Your 'deleted' licenses are not lost! You can reupload them anytime by locating the other license file and upload those licenses again.
Licenses	
My licenses are highlighted with yellow! What does it mean?	 It means that your licenses currently updated on your Hardware Key are passed their 'Expiry Date'. Your license is still functional; all your ANPR or OCR processes will run indefinitely, however you will not be able to run engines which were released later then the Expiry Date. Please contact your sales person to request a new license with an
	extended Expiry date if you would wish to use recently released up-to-date engines!
My licenses are highlighted with red! What does it mean?	When "K"-license is in use, the License Manager is constantly monitoring the date and time of your system. If it finds an inconsistent change it will block your license. Please contact our <u>support</u> to resolve this issue!

ENGINE MANAGER

INTRODUCTION 1.

Engine Manager Pro is a utility for Windows and Linux based systems, that enables the management of all kinds of OCR engines for the CARMEN® ANPR and CARMEN® OCR Software's main module. This utility is part of the CARMEN® ANPR / OCR Software's download package, and it is automatically installed along with the software.

This document will detail the following:

- How to install engines on Windows and Linux systems
- How to use the Engine Manager PRO application _
- How to remove/uninstall engines from Windows and Linux systems

This application can be found in the following folder:

- On Windows: "C:\Program Files\Adaptive Recognition\CARMEN _ softwares\EngineManagerPro\"
- _ On Linux: "\opt\gx64\EngineManagerPro\"

🕻 Note

The latest available version from Engine Manager Pro: 7.3.2.0

Important

Please note, that this application is not available for ARM and CenOS6 packages on Linux.



2. **ENGINE INSTALLATION**

Important

Before installing the engine(s) make sure that no other process on the PC is using CARMEN® ANPR / OCR.

Engines are to be downloaded from ATSS and arrive in a zip file. For example, the 2020 Q3 general engine: cmanpr-gen-7.3.12.169_20Q3.zip. This zip file contains 2 different folders:

- _ linux
- windows _

Note

CARMEN® ANPR and CARMEN® OCR engines are the same from installation point of view so all the rest is true for the CARMEN® OCR engines as well.



2.1. INSTALLATION ON WINDOWS

In the "windows" folder in the zip file which you have downloaded from ATSS you can find the installer files. For example, the 2023 Q2 general engine: *cmanpr_gen_7.3.15.241_23Q2_x86.msi* or *cmanpr_gen_7.3.15.241_23Q2_x64.msi* – (32-bit and 64-bit versions). This folder also contains a text file which refers to this document and the previous version of it.

After locating the engine(s) in the selected download folder, simply double click on it to begin installation.



As a first step, you need to **accept** the EULA by checking the **"I accept the terms in the License Agreement"** box and click on **Install button**.

🛛 Note

The newly installed engine will be the default engine.

To check the installed engine(s), open the Engine Manager Pro application (see chapter 2).

🛛 Note

You can locate **GXSD.DAT** in this folder: **c:/ProgramData/gx/** This file contains all of the installed engines and their properties. It is NOT RECOMMENDED to change this file manually, please use **Engine Manager Pro**, or the **Demo Applications** to do that.

Important

From 20Q3 engines vcredist (for Visual Studio 2015, 2017 and 2019) is a must on windows systems. You can download it from <u>here</u>.

INSTALLATION ON LINUX 2.2.

In the "linux" folder in the zip file which you have downloaded from ATSS you can find the installer files separated in folders by architecture (arm, arm64, x64 and x86). For example, the 2020 Q3 general engine for x64 contains the following files:

Files in the package: cmanpr-gen-7.3.12.169_20Q3-x86_64.tar.gz _install_cmanpr-gen-7.3.12.169_20Q3-x86_64.sh _uninstall_cmanpr-gen-7.3.12.169_20Q3-x86_64.sh

Where "_install_cmanpr-gen-7.3.12.169_20Q3-x86_64.sh" is the script for installing the engine and " _uninstall_cmanpr-gen-7.3.12.169_20Q3-x86_64.sh" is the script for uninstalling the engine.

The install script does all the necessary file copies to the relevant folders and inserts the engine properties into the gxsd.dat. The engine will be set as the default engine. The uninstall script does the opposite. (For further information please check this chapter)

Note

You can locate GXSD.DAT in this folder: /var/gx This file contains all of the installed engines and their properties. It is NOT RECOMMENDED to change this file manually, please use Engine Manager Pro, or the Demo Applications to do that.

To check the installed engine, open the Engine Manager Pro application (see chapter 2).



3. ENGINE MANAGER PRO APPLICATION

Once the CARMEN® ANPR / OCR Software and the recognition engine(s) have been installed open the Engine Manager Pro application.

On Windows: navigate to **Start / All Programs / Adaptive Recognition / Engine Manager (x64)** or type "Engine Manager" in the search box of your **Start Menu**.

On Linux: navigate to **/opt/gx64/EngineManagerPro/** When you start the utility, the following window will appear:

Important

If you want to take effect your modification in this application, you must run it as an **administrator** or **root**, otherwise the application will not be able to save the changes into **GXSD.DAT** file!

Engine name	Local/Cloud licenses	Property name	Property value
cmanpr-7.3.16.188 : eur	2/2	Iocation	
cmanpr-7.3.17.24 : sas_go	4/4	read_adr_type	1
cmanpr-7.3.16.248 : arab	2/2	read_base_type	0
cmanpr-7.3.17.58 : aus_go	4/4	read_emptyadr_type	0
cmanpr-7.3.17.35 : eur_go	4/4	read_licence_plate_type	1
cmanpr-7.3.17.41 : nam_go	4/4	read_without_type	0
cmanpr-7.3.16.127 : world	2/2	♦ size	33
cmanpr-7.3.17.111 : gen	2/2	size_max	80
cmanpr-7.3.1.2 : cloud	No info / 2	size_min	10
		♦ timeout	500
		adapt_environment	-1
		•	
	<u> </u>	Property functions: Reset Sav	e
ne type: CARMEN ANPR	·	Display properties:	Online ma

Engine name	Local/Cloud licenses	Property name	Property value
mmr-7.3.4.100 : mmr-eur		triggermode	0
mmr-7.3.4.99 : mmr-aus	1/2	independent_classification	0
mmr-7.3.4.93 : mmr-gen	1/2		auto
mmr-7.3.4.98 : mmr-nam	1/2		
mms 7.2.4.02 ; mms and	1/2		
mm-r.3.4.32 . mm-sds	172		
11111-1.34.32 . 11111-3d5	172		
mm-1.34.32 . mm-3d5		Property functions: Reset Save	

Adaptive Recognition AmericaAdaptive Recognition Hungary

Adaptive Recognition Nordic
 Adaptive Recognition Singapore

REQUESTINFO@ADAPTIVERECOGNITION.COM WWW.ADAPTIVERECOGNITION.COM

EngMan Api key feature:

When you subscribe to Carmen Cloud License, the API key becomes essential for authentication purposes. It ensures that your application is authorized to use the services provided by Carmen Cloud. Once the API key is stored, it functions as a bridge between your application and the cloud. Without an API key, unauthorized access is denied, preventing misuse of the services.

The API key feature within Engine Manager is only for requesting licenses within Engine Manager, it does not make any changes to the system.

When you enter the API key, the screen updates within 2-3 seconds to display the licenses. The **Save** button only saves the API Key in the Engine Manager configuration so that it is entered in the application when it is reopened.

Engine name:

List of all successfully installed engines.

Local Licenses:

Available number of licenses for the given engine (Single: 1, Dual: 2, Quad: 4, No available licence for the engine: 0, Missing files: No info)

Cloud licenses:

In the "licenses" column next to the engines, it will also show whether the given engine has a Cloud license.

Property name / Property value:

Properties and their values for finetuning the engine. For quick information about the property, hover the mouse over it.

Engine name		Local/Cloud licenses		Property	name	Property value
cmanpr-7.3.16.188 : eur			•	locat	ion	
cmanpr-7.3.17.24 : sas_g	0	4 / 4	•	siz	e	15
cmanpr-7.3.16.248 : arab)	2/2	•	size		
cmanpr-7.3.17.58 : aus_g	0	4 / 4	•	size	size	
cmanpr-7.3.17.35 : eur_go	0	4 / 4	•	tim	The average he	ight of the number plate characters
cmanpr-7.3.17.41 : nam_g	jo	4 / 4	\$	adapt_er	image in pixels.	
cmanpr-7.3.16.127 : world	t l	2/2	•	confider	Possible values	positive integers (greater than 10)
cmanpr-7.3.17.111 : ge	en	2/2	•	contra	property value of	an be queried by the GetProperty() f
cmanpr-7.3.1.2 : cloud		No info / 2	\$	gapto	Suggested valu	c: leave the default value
			\$	gene	ral	5
			•	heapfre	efreq	0
			-1			
		<u>,</u>	Propert	y functions:	Reset Sav	/e
e type: CARMEN ANPR		•	Display	properties:		Online ma
			1			

For detailed information about the properties, click "Online manual" button which will open <u>this link</u> in case of ANPR engines, <u>this link</u> in case of OCR engines and <u>this link</u> in case of MMR engines.

Engine type:

With this selector, you can filter the installed engines to show the ANPR regional engines (like EUR, CAS, NAF, etc...), the ORC related engines (like ACCR, UIC, ISO, etc ...) and the MMR regional engines (like EUR, CAM, etc ...).

Display properties:

With this selector, you can filter the engine properties by their importance or their impact to the recognition results.

Default engine:

After installing a new engine, that will be set as the default engine. It is highlighted with green and a checkmark is also visible to the left of the engine name.

Changing the default engine (which is currently in use):

If you want to change the default engine, right click on the engine which you would like to make as default and click "set as default".

P Engine	Manager Pro 7.3.1.5 (64 bit)		
	Engine name	Set a	s default
	cmanpr-7.3.11.227 : gen		4
v	cmanpr-7.3.12.169 : gen		4



4. UNINSTALLING ENGINE(S)

🛛 Note

After deletion of an engine which is the current default engine, the default engine will be the first found engine in the GXSD.DAT file.

4.1. FROM WINDOWS

Go to Windows \rightarrow Settings \rightarrow Apps & features: select the desired engine from the **Installed engines**. Once the engine that you would like to uninstall have been selected, click the **[Uninstall]** button to uninstall the selected engine.

Settings	
Home Find a setting	Apps & features CARMEN Anpr - CAM - 7.3.10.119 (18Q4) (64 15.1 MB ARH Inc. 2019-03-12
Apps	CARMEN Anpr - GEN - 7.3.10.169 (19Q1) (64 12.0 MB ARH Inc. 2019-04-30
E Apps & features	CARMEN Anpr - LATIN - 7.3.10.101 (18Q4) (6 41.5 MB ARH Inc. 2019-03-05
E Default apps	CARMEN Anpr - LATIN - 7.3.10.171 (19Q1) (6 46.8 MB ARH Inc. 2019-04-30
때 Offline maps	CARMEN Anpr - NAF - 7.3.10.163 (19Q1) (64 20.7 MB ARH Inc. 2019-04-09
□ Video playback	CARMEN Anpr - NAM - 7.3.10.190 (19Q1) (64 87.8 MB ARH Inc. 2019-04-23
	CARMEN Anpr - NAM - 7.3.10.51 (18Q3) (64 56.5 MB ARH Inc. 2019-03-08

4.2. FROM LINUX

The *__uninstall_cmanpr-gen-7.3.10-169_19Q1-x86_64.sh*' is the script for uninstalling the engine.

The uninstall script does all the necessary file deletion from the relevant folders and deletes the engine properties from the gxsd.dat.

5. ENGMAN

CONSOLE APPLICATION

On Linux you can find this application in this folder: **/opt/gx64/EngMan/** On windows this application is not part of the installed package, please turn to <u>AR support</u> if you need this application.

If you run this application in Terminal, or Command Prompt: you will get the following help: "Use: EngMan_x64.exe/EngMan_x86_64.out version | -h | -? | [[list {-lic} | getdef | setdef -e:engine] {anpr} {-ocr} {-mmr} {-alltype}] {-json} {-o:outputfile}] Copyright © 2019-2022, Adaptive Recognition

Parameters:

==========	
version -h, -?	Displays the version information of the program. Shows this help.
List	Lists the installed engines. If no -anpr, -ocr and -mmr switches added lists anpr, ocr and mmr engines too.
-lic	checks the licenses for the engine
getdef	Gets the default engine.
setdef -e:engine	Sets the default engine. the name of the engine to set the default engine.

Global parameter (all commands except version and help):

-anpr	Runs the command on the anpr engines
-ocr	Runs the command on the ocr engines
-mmr	Runs the command on the mmr engines
-alltype	Runs the command on the anpr,ocr and mmr engines
-json	The program create json format output.
-o:outputfile	The program redirects the output from display to the outputfile

Some examples:

Gets the installed ANPR engines:

Command: EngMan_x64.exe/EngMan_x86_64.out list -anpr

Output: Engines (ANPR):

cmanpr-7.3.14.101 : eur License: FOUND cmanpr-7.3.14.95 : nam License: NOT FOUND

```
Gets the installed engines (ANPR, OCR, MMR) with license checking in json format:
Command: EngMan_x64.exe/EngMan_x86_64.out list -lic -alltypes -json
Output:
{
```

```
"Engines (ANPR)": [
         {
              "Engine": "cmanpr-7.3.14.51 : arab",
              "License": "FOUND"
         },
         {
              "Engine": "cmanpr-7.3.14.101 : eur",
              "License": "FOUND"
         },
              "Engine": "cmanpr-7.3.14.95 : nam",
              "License": "NOT FOUND"
         }
    ],
    "Engines (MMR)": [
         {
              "Engine": "mmr-7.3.2.31 : mmr-eur",
              "License": "FOUND"
         },
         {
              "Engine": "mmr-7.3.2.27 : mmr-arab",
              "License": "FOUND"
         },
         {
              "Engine": "mmr-7.3.2.32 : mmr-nam",
              "License": "NOT FOUND"
         }
    ],
     "Engines (OCR)": [
         {
              "Engine": "cmocr-7.3.2.100 : isoilu",
              "License": "FOUND"
         },
         {
              "Engine": "cmocr-7.3.2.84 : uic",
              "License": "FOUND"
         },
    1
}
```

Gets the default engines and write the result to the result.txt:

Command: EngMan_x64.exe/EngMan_x86_64.out getdef -alltype -o:result.txt Output: The "result.txt" file in the same folder with the following content: Default anpr: cmanpr-7.3.14.51 : arab Default mmr: mmr-7.3.2.31 : mmr-eur Default ocr: cmocr-7.3.2.100:isoilu

Sets the default ocr engine:

Command: EngMan_x64.exe/EngMan_x86_64.out setdef -e:"cmocr-7.3.2.84 : uic" -ocr"

nothing, but after checking the default 'ocr' engine with this command: Output: EngMan_x64.exe/EngMan_x86_64.out getdef -ocr the output will be: Default ocr: cmocr-7.3.2.84:uic

Check the version of this application:

Command: EngMan_x64.exe/EngMan_x86_64.out version

Engine Manager (console version) Output: Version: 7.3.1.5 Copyright © 2019-2022, Adaptive Recognition



LICENSE SERVER

1. INTRODUCTION

From CARMEN® ANPR version 7.3.1.24 and CARMEN® OCR version 7.3.1.15, License Server is part of the installer package. This application allows users to share any Adaptive Recognition license over their network.

The server computer, which has the hardware key(s) (HW Key or NNC) with the licenses plugged in, runs the Server Application.

Client computer(s) may join this server by using the Client Application that, upon a successful configuration, will make it possible to do ANPR/OCR processes without any HW Key attached to the client computer(s).

This solution is provided free for up to one client.

If you would like to connect more than one client to the server, please contact your Sales Manager to check the possibilities.

🛛 Note

The latest available version from License Server: 7.3.1.36





2. SERVER APPLICATION

This application can be found here:

- On Windows: "C:\Program Files\Adaptive Recognition\Common Utils\LicenseServer\Server\
- On Linux: "/opt/gx64/LicenseServer/"

The Server application allows the sharing of your licenses over the network. On Windows, there is a **LicenseServer.cfg** file in the same folder where the application is installed. This file is in the following folder on Linux-based computers: **/etc/gxd/licserver**. This file stores the configuration settings, which the user can modify manually.

This file contains the following properties:

Property and its default value	Description
SRVPORT=8998	This is the server port. The Client will connect through this to
	the Server. Modification of the property is permitted. Please
	make sure that the set port is allowed on your firewall.
CHECKPORT=48315	This is for an internal service port. Please DO NOT MODIFY.
LOGPATH=.\logs\	The folder where the Server application stores the logs. If it is
	empty, the log files will be saved into the same folder where the
	application is installed.
IDLETIMEOUT=600	Indicates the idle timeout in milliseconds. The Client Application
	sends "keepalive" messages every 40 seconds to make the
	connection stable. If there is no "keepalive" message, then the
	Server application breaks the connection to prevent a client
	from being stuck.
NNCTIMEOUT_MS=2000	This value defines, in milliseconds, when one NNC lock request
	has to be finished. If the CPU load is high and the number of
	available licenses is low, it may result in a timeout error,
	indicated by the "HW Key lock error" message and the loss of
	the ANPR/AICR/MMR process.
LOCKTIMEOUT=3000	Sets, in milliseconds, the maximum time one license can be
	locked.
NPROC=6	The number of threads not handling license lock calls.
NPROCLOCK=6	The number of threads handling license lock calls.
SAVEDATA=1	If it is set to "1," then every invalid request's binary data will be
	saved. Any other value than "1" will not save invalid requests'
	binary data.

If you properly set all the above properties in this file, please open a Console/Terminal window in the Server application's folder and just start the application.

2.1. SERVER APPLICATION AS CONSOLE APPLICATION

2.1.1. ON WINDOWS

Just go to this folder: "C:\Program Files\Adaptive Recognition\Common

Utils\LicenseServer\Server\" and run this file: LicenseServer_x64.exe in command prompt.

2.1.2. ON LINUX

There are 3 possibilities to start the Server Application on Linux from the Console:

 Go to this folder: "/opt/gx64/LicenseServer/" and this this command: "/LicenseServer_x86_64"

🛛 Note

Please make sure that kernel drivers are running if you are using this option

 Go to this folder: "/opt/gx64/LicenseServer/" and this this command: "./LicenseServer_start.sh"

🖵 Note

If you start the Server application like this, the script will check if the kernel drivers are compiled for the current kernel and running, and if not, it will try to compile and start them to make sure that Server application will be able to run.

3. Hit the following command from any folder: "LicenseServer"

🕻 Note

This will do the same as point #2, but from any folder.

Once the Server application is running, it will write to the console something like this:

```
Checking time functions
t0: 248725978581 -> t1: 248726984480, dt: 1005899
t2: 248725978582 -> t3: 248726984480, dt: 1005898
t4: 1649309382940272 -> t5: 1649309383955655, dt: 1015383
```

Log files:

```
Log files will be saved to .\logs\ directory
[LSLOG] Current logfile: .\logs\/licsrv.20220407 072943 956.log (20220407 072943 956)
                   logs
[LSLOG] Clean
                           in
                                   c:\Program
                                                Files\Adaptive
                                                                  Recognition\Common
Utils\LicenseServer\Server\logs (licsrv.*.log)
[LSLOG] Current logfile: .\logs\/lscons.20220407 072943 956.log (20220407 072943 956)
[LSLOG] Clean logs in c:\Program
                                                Files\Adaptive Recognition\Common
Utils\LicenseServer\Server\logs (lscons.*.log)
[licsrv] Logging started
2022.04.07 7:29:43,956395 [11176] Main started
[lscons] Logging started
```

Searching for licenses:

2022.04.07 7:29:43,956507 Finding licenses type: 99900099, min. expiry date: 20220407 ... 2022.04.07 7:29:43,956761 Found license type: 99900099, licID: 111111, expiry date: 20230331 2022.04.07 7:29:43,956840 Query access to share licenses -> Locking(CARMEN Server license) 2022.04.07 7:29:43,958076 Finding licenses to share... 2022.04.07 7:29:43,958163 Found 1 devices 2022.04.07 7:29:43,958238 [0111111] XX licenses 2022.04.07 7:29:43,987310 nk: 21, nkf: 0 2022.04.07 7:29:43,987440 Found 21 license(s) to share. 2022.04.07 7:29:43,987647 Num. of license types shared: 4 laffffff -> 4 99902000 -> 1 30ffffff -> 1 16ffffff -> 4

Starting the License Server:

2022.04.07 7:29:43,988242 Starting License Server (v7.3.1.27 - License Server (rev: d5944ac)) ... 2022.04.07 7:29:43,988353 Server running mode: Standalone 2022.04.07 7:29:43,988485 Server port: 8998 2022.04.07 7:29:43,988796 Health check port: 48315 2022.04.07 7:29:43,989069 Server status port: 8980 2022.04.07 7:29:43,989284 Max. clients: 20 2022.04.07 7:29:43,989442 idle time out: 600 sec 2022.04.07 7:29:43,989526 Lock time out: 3000 msec 2022.04.07 7:29:43,989681 Set NNC time out -> 2000 msec 2022.04.07 7:29:43,989875 NNC time out: 2 sec 2022.04.07 7:29:43,991339 Num of process / lock threads: 6 / 6 [LSLOG] Current logfile: .\logs\/portcheck.20220407 072943 987.log (20220407 072943 987) 2022.04.07 7:29:43,992648 Wait for end of initialization [LSLOG] Clean logs in c:\Program Files\Adaptive Recognition\Common Utils\LicenseServer\Server\logs (portcheck.*.log) 2022.04.07 7:29:43,993429 Threads initialization is SUCCESS. Init time: 0 msec [portcheck] Logging started

Details about possible connections: 2022.04.07 7:29:43,994417 Server has created. IP: 0.0.0.0:8998 Max. client num.: 20 192.168.6.175 192.168.74.1 192.168.109.1 Waiting for a connection (ESC to exit) ... 2022.04.07 7:29:43,994799 [LICSRV]: Wait for event: -1

Note

If you would like to exit from the Server application, please press 'Esc' key on your keyboard.

Adaptive Recognition America Adaptive Recognition Hungary

Adaptive Recognition Nordic Adaptive Recognition Singapore REQUESTINFO@ADAPTIVERECOGNITION.COM WWW.ADAPTIVERECOGNITION.COM

SERVER APPLICATION AS A SERVICE 2.2.

There is a possibility to use License Server as a service on Windows and Linux.

2.2.1. ON WINDOWS

Create and start the service with the following command: LicenseServer_x64.exe -inst

Once the service is created and started, it will be visible in Task Manager on the Services tab as 'licserver.' If you open Services from this tab, you will be able to stop and start the License Server service from here as well.

If you would like to stop and remove the License Server service, hit the following command: LicenseServer_x64.exe -uninst

If you hit the above command, the License Server service will no longer be visible in Windows Services.

2.2.2. ON LINUX

It is possible to use License Server as a service on Linux if 'systemd' is available on the system.

- Enable: systemctl enable licserverd.service
- Start: systemctl start licserverd.service
- Get the current status about the service: systemctl status licserverd.service
- Stop: systemctl stop licserverd.service
- Disable: systemctl disable licserverd.service

Important

CARMEN® has to be installed on the computer where the Server application is running.


SERVER APPLICATION FOR MORE THAN ONE CLIENT 2.3.

This solution is available for free for one client only. To use it for more clients, a License Server license is necessary. This license allows the application to share any Adaptive Recognition licenses for more clients simultaneously.

If your License Server license expires, the application will STOP working. After you restart, only one client will be able to connect to the Server application.

If you run the License Server application as a service, license expiration will do the following:

- On Windows: the service is stopped. If you restart it, it will share the licenses for one client only.
- _ On Linux: the service restarts automatically as a service for one client only.

If you would like to connect more than one client to the server, please contact your Sales Manager to check the possibilities.



3. CONFIGLSCLIENT

Once the Server application is running — either as a Console application or as a service — the ConfigLSClient application's configuration must be done as follows.

This application can be found in the following folder:

- On Windows: "C:\Program Files\Adaptive Recognition\Common Utils\LicenseServer\Client\
- On Linux: "/opt/gx64/LicenseServer/Client/"

Important

If you want to take effect **ConfigLSClient** application, you must run it as an **administrator** or **root**, otherwise the application will not be able to save the changes into **GXSD.DAT** file!

- Set the Server address where the licenses should search for:
 - Only IP address:
 Windows: ConfigLSClient_x64.exe SET 192.168.0.1
 Linux: ./ConfigLSClient_x86_64.out SET 192.168.0.1
 - IP address with port (in cases when, for example, the Server application port was changed from the default 8998 to 7683)
 Windows: ConfigLSClient_x64.exe SET 192.168.0.1:7683
 Linux: ./ConfigLSClient_x86_64.out SET 192.168.0.1:7683

🛛 Note

The SET command will automatically ENABLE the usage of the License Server.

 Enable License Server on the client computer (if it is configured, but DISABLED): Windows: ConfigLSClient_x64.exe ENABLE Linux: ./ConfigLSClient_x86_64.out ENABLE

🛛 Note

If License Server is enabled, licenses will always be looked for over the network, even if a HW Key with licenses is plugged into the client computer.

 Get the status of the License Server on the client computer: Windows: ConfigLSClient_x64.exe STATUS Linux: ./ConfigLSClient_x86_64.out STATUS

The STATUS messages could be the follows:

- If the usage of License Server is ENABLED on the client computer where the STATUS request is sent:
 - The connection is OK. More clients can connect.
 - The connection is OK. No more clients can connect.
 - The connection cannot be established. It's switching off the using of the License Server. (It means the STATUS request will DISABLE the License Server usage in this case)

- If the usage of License Server is DISABLED on the client computer where the STATUS request is sent:
 - The server is running. More clients can connect. 0
 - The server is running. NO more clients can connect! 0
 - The connection cannot be established. 0
- Set the client timeout from the default 10000 to, for example, 20000: Windows: ConfigLSClient_x64.exe TIMEOUT 20000 Linux: ./ConfigLSClient_x86_64.out TIMEOUT 20000
- Disable License Server on the client computer: Windows: ConfigLSClient_x64.exe DISABLE Linux: ./ConfigLSClient_x86_64.out DISABLE

Important I

CARMEN[®] must be installed on the computer where the Client application is configured.



VIDEO SDK

1. INTRODUCTION

This is a CARMEN® Video SDK library for the CARMEN® ANPR and MMR engines. This library offers interfaces in C, C++ and C# for getting ANPR and MMR results from passing vehicles in camera stream or video file.

Note

The latest available version from Video SDK: 1.2.0

ENGINE AND LICENSE REQUIREMENTS 2.

This SDK utilizes CARMEN® engines either locally or through the CARMEN® Cloud, and these engines require licenses to operate.

There are four use cases available to suit different needs:

- Local usage: Engines are installed and run on the machine. Hardware key is connected to the same machine with the proper licences.
- License Server usage: Engines are installed and run on the local machine but obtain licensing from a License Server on another computer, which has the hardware key with the appropriate licenses connected.
- Cloud licensing: Engines are installed and run on the local machine but obtain licensing from an LSC (like in the case of the License Server). LSC can run locally or remotely, and connects to the CARMEN® Cloud using an API Key to access the required licenses.
- Cloud processing: Only the video decoding and image preselection components run on the machine with the Video SDK. Licensing for preselection is obtained directly from the Cloud. The ANPR and MMR engines run in the Cloud

For system design details, see the chapter "Carmen Licensing". For instructions on setting the licensing mode, refer to the chapter "Licensing Modes".



3. BUILD REQUIREMENTS

CARMEN® Video SDK 1.2 version supports both Linux and Windows. Binary compatibility between different versions is not guaranteed.

3.1. LINUX

At the moment, we have released only one translation, in which we compile the core library with GCC 9.4.0 in Release for x86_64 ARM architecture with 64-bit version. We provide C, C++ and C# interfaces for this.

CARMEN[®] Video SDK requires a minimum of 7.3.1.28 CARMEN[®] ANPR package installed. This version of CARMEN[®] can be found in the package. The user must install this before installing the CARMEN[®] Video SDK, otherwise the installer will warn the user and exit with an error.

When you install the Carmen Video SDK, the files are placed in the following directories:

- Header files:
 - o C headers: /usr/include/carmen_video_sdk/c
 - C++ headers: /usr/include/carmen_video_sdk/cpp
- Shared object files:
 - Located in: /usr/lib/carmen_video_sdk
- NuGet Package (for C#):
 - o Located in: /usr/include/carmen_video_sdk/csharp

To run the samples, you need the CARMEN® ANPR engine and you may need the MMR engine.

The sample programs are provided to the user in a separate package with all supported languages.

Additional Information for Linux_x86_64 Users:

Some engines require the Intel "icclib" package. Therefore, the installer places its shared library files separately in /usr/lib. These files are not removed when you uninstall the Video SDK because they might be used by other applications on your system. If you wish to remove them manually, use the following commands:

sudo rm /usr/lib/libintcl.so.5 sudo rm /usr/lib/libsvml.so sudo rm /usr/lib/libimf.so

3.1.1. C

Compiler	Target architecture	Target platform
Any Linux C compiler	x86_64, arm64	64 bit

To build it, you need to set the following:

- 1. Import directory is: "usr/include/carmen_video_sdk/c"
- 2. Link directory is: "/usr/lib/carmen_video_sdk"
- 3. Link library name is: carmenVideoSDK_c

The C sample project's directory contains CMake files to be able to open the project with a wide range of IDEs.

3.1.2. C++

Compiler	Target architecture	Target platform	C++ standard
Any Linux C++ compiler, tested with GCC 9.4.0	x86_64, arm64	64 bit	C++17

This is a header-only interface over the C interface.

To build it, you need to set the following things:

- 1. Import directory is: "usr/include/carmen_video_sdk/c", "usr/include/carmen_video_sdk/cpp"
- 2. Link directory is: "/usr/lib/carmen_video_sdk"
- 3. Link library name is: carmenVideoSDK_c

The C++ sample project's directory contains CMake files to be able to open the project with a wide range of IDEs.

3.1.3. C#

Target architecture	Target platform	Target framework		
x86_64, arm64	64 bit	.NET 6.0		

In order to use this library, you need to add a reference to the local NuGet package located in the /usr/include/carmen_video_sdk/csharp folder. There is a nuget.linux.config file in the solution folder; by renaming it to NuGet.Config, you can add the package source to your NuGet configuration.

The C# sample project's directory contains a VisualStudio 2019 solution file.

() Important!

Important Note: Ensuring the Built Application Finds the Video SDK .so Files

In order for the built application to locate the Video SDK's shared object (.so) files, we have defined a method in the sample .csproj file. If you set the Runtimeldentifier to *linux-x64* or *linux-arm64*, the build process will perform the following actions:

- **Copy the Necessary .so File:** After the build, it will automatically copy the required shared object file to the output directory.
- Set the RPATH Using patchelf: It will modify the runtime search path (rpath) of the copied shared object file using patchelf to ensure it can find the .so files at runtime.

(E) Note

Prerequisite: Please make sure that patchelf is installed on your system. You can install it using your distribution's package manager.

3.2. WINDOWS

The MSI contains the software components required for running, except for CARMEN® ANPR and MMR engines.

At the moment, we release only one translation, in which we compile the core library with VS 2019 in Release for x86_64 architecture with 64-bit version. We provide C, C++ and C# interfaces for this.

The install path is: "C:\Program Files\Adaptive Recognition\CARMEN Video SDK". This Windows MSI installer sets this directory path to a system environment variable called "CMV_INSTALL_DIR". The built-in programs can use the necessary dynamic libraries from the SDK binary directory, which the installer also adds to the system PATH.

The sample programs are provided to the user in a separate package with all supported languages.

3.2.1. C

Compiler	Target architecture	Target platform
Any Windows C compiler	x86_64	64 bit

To build it, you need to set the following:

To access the installation folder, you can use this environment variable in CMake: "\$ENV{CMV_INSTALL_DIR}", in VisualStudio: "\$(CMV_INSTALL_DIR)".

- 1. Import directory is: "\$ENV{CMV_INSTALL_DIR}\\sdk\\c\\include"
- 2. Link directory is: "\$E NV{CMV_INSTALL_DIR}\\sdk\\c\\lib"
- 3. Link library name is: carmenVideoSDK_c

The C sample project's directory contains a VisualStudio 2019 solution file, and CMake files to be able to open the project with a wide range of IDEs.

3.2.2. C++

Compiler	Target architecture	Target platform	C++ standard
Any Windows C++	x86_64	64 bit	C++17
complier			

This is a header-only interface over the C interface.

To build it, you need to set the following things:

To access the installation folder, you can use this environment variable in CMake: "\$ENV{CMV_INSTALL_DIR}", in VisualStudio: "\$(CMV_INSTALL_DIR)".

1. Import directory is: "\$ENV{CMV_INSTALL_DIR}\\sdk\\C\\include ;\$ENV{CMV_INSTALL_DIR}\\sdk\\C++\\include"

- Link directory is: "\$ENV{CMV_INSTALL_DIR}\\sdk\\C\\lib " З.
- Link library name is: carmenVideoSDK_c 4.

The C++ sample project's directory contains a VisualStudio 2019 solution file, and CMake files to be able to open the project with a wide range of IDEs.

To use experimental features in C and C++, CMV_ENABLE_EXPERIMENTAL_FEATURES needs to be added as a compiler definition.

3.2.3. C#

Target architecture	Target platform	Target framework
x86_64	64 bit	.NET 6.0
		.NET Framework 4.6

In order to use this library, you need to add a reference to the local NuGet package located in the sdk/CSharp folder. There is a nuget.windows.config file in the solution folder; by renaming it to NuGet.Config, you can add the package source to your NuGet configuration.

The C# sample project's directory contains a VisualStudio 2019 solution file.



4. VIDEO INPUT

In this chapter, we explain the types of acceptable video inputs. The most important input parameter of the CARMEN[®] Video SDK is the video file or network stream that it must process and return the passing vehicles as an event.

In order for CARMEN[®] Video SDK to work properly on the video/stream, it is important that the license plates of the vehicles passing by on the video meet the specifications in the <u>Imaging for Carmen[®]</u>.

CARMEN[®] Video SDK is equipped with an advanced vehicle detection algorithm. This means, your camera does not need any hardware triggering for selecting the images from the stream to do number plate recognition. In turn, it needs a moving vehicle (car, bus, truck, etc..) on the stream to work.

The processing also works on recordings from fixed-installation cameras and moving vehicles. On fixed-installation, processing takes more power.

It is important that the vehicle detector is partly based on text recognition, so the ROI must be in a way that no other non-license plate characters can be seen such as posters, signs or camera watermarks.

There are two main types of videos inputs: video file and network stream. Their processing is partially different.

4.1. NETWORK STREAM

Typically, RTSP or H264 streams coming from cameras are considered network streams. In case of such an input, the processor continuously tries to request the images from the stream, and selects the received images for ANPR. If too many images are selected and the system cannot perform ANPR at a sufficient speed, the buffer will fill up (which with high RAM usage) which may result in some frames being dropped. In such cases, unfortunately, it may happen that the license plate of a vehicle only appears in images that are discarded by the system, thus losing an event.

It may happen that the network stream is interrupted, in this case if autoreconnect is false, (which is the default), the processing stops. If, in the event of a disconnection, we want the processor to continuously try to reconnect until the connection is restored, this parameter must be set to true when building it.

For events and frames, the timestamp that can be extracted from the stream, or the time when the frame arrives for processing in the StreamProcessor.

Accepted protocols: HTTP, HTTPS and RTSP

Examples: "rtsp://[USER]:[PASSWD]@[IP]:[PORT]" -> rtsp://192.168.0.50/stream/jpeg http://[URL] -> http://192.168.0.50:9901/video.mjpeg



4.2. VIDEO FILE

In case of video file input, the operation is similar to the network stream, with the difference, that since the file is continuously available, if the buffer is full, we do not discard the images, but wait until the ANPR makes room for the image in the buffer. In this case, the processing will slow down, but we will not lose the event because of this.

It is important to buffer the display of frames in this mode and synchronize them based on the timestamp, so that when the buffer is full, the display is not interrupted.

In this case, the timestamp is read from the video file by the codec, which will start from 0 in most cases. The advantage of this is that in case of two runs, the same frame will receive the same timestamp. The disadvantage is that it is more difficult to put it in real time, if this is important, then it is worth adding the video's production time or its playback time to the timestamps of all frames and events.

Accepted formats: KV (H.264), MP4 (H.264), ASF (MPEG4), MJPEG, AVI (H.264)

Examples: file://[path] vagy "file:[path]"

In case of relative path: file://videos/cars.mp4

Absolute path on Linux: file:///home/user/videos/cars.mp4

Absolute path on Windows: "file://C:\\Program Files\\videos\\cars.mp4"



5. A MINIMAL APPLICATION EXAMPLE

This minimal application performs recognition locally and serves as a basic demonstration of how to use the Carmen Video SDK. It is similar to sample code "01_minimal" included with the SDK. For more detailed results, advanced settings, and additional use cases-including different engines and licensing modes-please refer to the other sample applications provided in the SDK package and discussed in later chapters.

5.1. CREATE ANPR

In order to run ANPR on a stream, you have to build an Anpr object, which should be added to the StreamBuilder:

5.1.1. C

```
CM ANPR BUILDER anprBuilder;
```

You will have to free this Anpr object later with:

cm anpr free(anpr);

5.1.2. C++

cm::anpr::Anpr anpr = cm::anpr::AnprBuilder()

5.1.3. C#

Anpr anpr = Anpr.Builder() .Build();



Adaptive Recognition **Nordic** Adaptive Recognition Singapore REQUESTINFO@ADAPTIVERECOGNITION.COM WWW.ADAPTIVERECOGNITION.COM

5.2. ONEVENTCALLBACK

This is a callback function, which is triggered when an Event is generated during stream processing. These examples show a few properties that are available in an Event object, find more details there: RESULT CLASSES.

5.2.1. C

The callback parameter has to be of type Event* and at the end of the function has to free the event.

```
void onEventCallback(CM_EVENT* e, void* userdata) {
    printf("------\n");
    printf("Plate: %s\n", e->vehicle->plate->text);
    printf("Country: %s\n", e->vehicle->plate->country);
    printf("\n");
    fflush(stdout);
    cm_event_free(e);
}
```

5.2.2. C++

In C++ the callback function parameter is a const cm::Event&. You should use *try-catch* block around your code.

```
void onEventCallback(const cm::Event& event) {
    std::cout << "------" <<
std::endl;
    std::cout << "Plate: " << event.vehicle().plate().text() << std::endl;
    std::cout << "Country: " << event.vehicle().plate().country() << std::endl;
    std::cout << std::endl;</pre>
```

5.2.3. C#

In C# callback the function parameter is a Carmen.Event, you should use *try-catch* block around your code.





REQUESTINFO@ADAPTIVERECOGNITION.COM WWW.ADAPTIVERECOGNITION.COM

5.3. CREATE STREAMPROCESSOR OBJECT

You also need to use builder pattern to create a StreamProcessor object. In these examples will be only a minimal implementation, other options are STREAMPROCESSOR BUILDER.

5.3.1. C

```
cm_streamprocessorbuilder_set_source(builder, streamUrl);
```

5.3.2. C++

5.3.3. C#

```
StreamProcessor stream = Carmen.Video.StreamProcessor.Builder()
   .Region(region)
   .EventCallback(EventHandlerCallback)
```



START STREAMPROCESSING 5.4.

After you create the StreamProcessor object, you can Start and Stop it. Start function starts the stream processing asynchronously. You have to keep your main thread alive. In these examples we wait for a 'g' character input from the user. When you want to end processing, you have to call the Stop function.

5.4.1. C

In C you have to free the stream processor object.

```
cm streamprocessor start(stream);
```

5.4.2. C++

5.4.3. C#





6. CONSTRUCTING COMPONENTS

The following builders are present in the sample codes:

- AnprBuilder -> Anpr
- MmrBuilder -> Mmr
- AdaptiveRecognitionCloudBuilder -> AdaptiveRecognitionCloud
- StreamProcessorBuilder -> StreamProcessor

Please refer sample codes "01_minimal" and "02_basic" for simple implementation examples to get started, while sample "03_result_details" has more a detailed structure.

6.1. ANPR BUILDER

6.1.1. TYPE

It sets the type of ANPR. Currently there are two options:

- LOCAL: local ANPR processing with CARMEN® FreeFlow engines
- LOCAL_GO: local ANPR processing with CARMEN® Go engines

Default value is LOCAL.

6.1.2. LOCAL CONCURRENCY LIMIT

This allows you to set how many ANPR threads can be run via the given ANPR class. It is recommended to set it to the same value as the number of ANPR Core Licenses, but this can also be used to distribute resources in the case of multiple streams. Default value: 1

6.1.3. Registering ANPR Profile to an already built Anpr object

After built an Anpr object with an Anpr Builder *Build* function, it provides a registerProfile method, which allows you to define custom ANPR profiles. An ANPR profile determines how license plate recognition is performed, including regions, engine versions, locations, and custom properties. **Options for Registering Profiles:**

1. AnprStage

- **Purpose**: Allows you to add a single ANPR stage with specific settings.
- Configuration Options:
 - ANPR Region: Set the region from Region List for the recognition process.
 - Engine Version: Specify the version of the ANPR engine to use.
 - Location: Define the engine *location* property.
 - Custom Properties: Set additional engine properties as needed.
- **Usage**: Ideal for simple recognition tasks requiring a single stage.
- 2. AnprProfile
 - **Purpose**: Allows you to create a complex ANPR profile containing multiple stages and custom logic.
 - Components:
 - **Stages**: One or more AnprStage objects defining each recognition step.

- Callback Function: A user-defined function that dictates the logic for transitioning between stages. This function can be used to refine results or apply conditional processing.
- **Usage**: Suitable for advanced recognition workflows that require multiple stages or custom logic.

Return Value:

• The registerProfile method returns an **AnprProfileId**, a unique identifier for the registered profile. This ID can be used later to assign the profile to a Stream Processor.

Example Usage:

• **Sample "07_multistage_anpr"** demonstrates how to register an ANPR profile with multiple stages and custom logic. Refer to this sample for practical implementation details.

6.2. MMR BUILDER

6.2.1. TYPE

It sets the type of MMR, currently it can only be LOCAL. Default value is LOCAL.

6.2.2. Registering MMR Profile to an already built Anpr object

It is similar to the Anpr object. After built an Mmr object with an Mmr Builder *Build* function, it provides a registerProfile method, which allows you to define custom MMR profiles. **Profile Configuration:**

- mmrGroupName: Specifies the MMR group name, such as mmr-eur, to choose the MMR engine.
- Engine Version: Specifies the version of the MMR engine to use.

Return Value:

• The registerProfile method returns an **MmrProfileId**, a unique identifier for the registered profile. This ID can be used later to assign the profile to a Stream Processor.

Example Usage:

• **Sample "07_multistage_anpr"** also includes examples of registering MMR profiles. Refer to this sample to see how ANPR and MMR profiles can be used.

6.3. ADAPTIVE RECOGNITION CLOUD BUILDER

This solution allows you to run ANPR (Automatic Number Plate Recognition) and MMR (Make & Model Recognition) engines in the cloud instead of on your local device. This solution reduces local resource consumption and eliminates the need for a hardware key. To use this service, you must register with the CARMEN® Cloud to obtain an API Key.

6.3.1. API Key

An API Key is required to authenticate your application with the CARMEN® Cloud services. You must set the API Key in the Cloud Builder.

6.3.2. The built Cloud object

Once you have configured the Cloud Builder, you can build the Cloud object. This object must be set in the StreamProcessorBuilder using the *cloud setter* to enable cloud-based ANPR and MMR processing instead of local processing.

The Cloud object provides getCountries method that returns the possible options for the region and location parameters of the StreamProcessorBuilder.

6.4. STREAMPROCESSOR BUILDER

6.4.1. SOURCE

The source setter determines the input video stream or file that needs to be processed by the CARMEN® Video SDK. There are two options for specifying the source:

Using a Source URL

The string variable determines the input video stream or file which needs to be processed. For files, use: "<u>file:[path]</u>". For stream, use: "rtsp://[USER]:[PASSWD]@[IP]:[PORT]" or "<u>http://[URL]</u>". No default value, has to be set.

About CARMEN® Video SDK input source types there is a detailed description in VIDEO INPUT.

Using a StreamFactory for Custom Image Callback

This mode allows you to provide a custom StreamFactory object to the SDK. By implementing specific functions, you can control how the SDK retrieves images, which is useful for custom or non-standard input sources.

Purpose:

- Allows integration with custom video capture hardware.
- Enables processing of pre-captured frames or images from memory.
- Facilitates integration with other media frameworks or custom pipelines.

For an example implementation, please refer to the "06_custom_stream" sample in C++ and C# languages.

It's important to set *processingMode* when using StreamFactory as *source*.

This is an *experimental feature*.

6.4.2. NAME

This variable determines the stream name. Default value is: "Untitled".

6.4.3. ANPR

It determines which ANPR resource is used for license plate recognition. An ANPR object is built with ANPR BUILDER.

Stream has to have an ANPR object or a Cloud object (see below) in order to be able to work.

6.4.4. MMR

It determines which MMR resource is used for Make&Model recognition. An MMR object is built with ${\sf MMR}$.

MMR parameter is optional. If not set, the stream processor will not recognize make, model and color.

6.4.5. Cloud

It determines which Cloud resource is used for license plate recognition. A Cloud object is built with ANPR BUILDER.

Stream has to have a Cloud object or an ANPR object (see above) in order to be able to work.

6.4.6. REGION

The region setter in the Stream Processor Builder allows you to specify a region code that determines which ANPR and MMR engines are used for processing. This provides a quick and convenient way to configure the recognition engines without the need to set up profiles explicitly. The region code directly determines the ANPR and MMR engines used. It simplifies the setup by automatically selecting the appropriate engines based on the specified region.

In the 1.1.0 versions of the SDK, setting the region was mandatory.

For MMR, if the engine for the specified region was not installed, it defaulted to using the generic (GEN) engine.

In the 1.2.0 version with the introduction of **AnprProfile**, you now have other option for configuring recognition engines.

Using Adaptive Recognition Cloud, setting the region is mandatory.

6.4.7. LOCATION

Sets the ANPR engine location parameter from the ANPR reference manual.

Do not use the location parameter if you are using an AnprProfileId. When using profiles, the location should be specified within the profile itself.

6.4.8. ANPR Profile Id and MMR Profile Id

The AnprProfileId and MmrProfileId setters allow for more complex configurations, such as multiple ANPR stages, custom engine parameters, and specific engine versions.

You should assign to the Stream Processor Builder the **AnprProfileId** that you obtained from the same ANPR object's registerProfile method as set for Anpr resource.

These settings are ideal when you need fine-grained control over the recognition process.

When you assign ANPR Profile Id to the Stream Processor, do not set region and location paramter.

6.4.9. AUTO RECONNECTION TO STREAM

If this parameter is set to true, then the stream automatically restarts in case of connection interruptions. Default value is *false*.

6.4.10. ANPR COLOR RECOGNITION

This parameter controls the number plate color (text color, strip color, dedicated area color) recognition feature of the ANPR engine. If set to false, there will be no color information in the result. Default value is true.

6.4.11. MMR COLOR RECOGNITION

This parameter enables or disables the vehicle color recognition feature of the MMR engine. If set to false, there will be no color information in the result. Default parameter is true.

6.4.12. EVENT DUPLICATION TIMEOUT

This parameter determines how much time is needed to pass between two identical results to be considered separate events. (Identical results generated within this timeframe will be disregarded. (Think of a slowly moving vehicle, with the LP partially getting covered in the flow of traffic sometimes. This LP can show up as separate events, however it is just one event in reality. On the other hand, if the vehicle takes a turn and passes again, that is really a separate event, which should be recorded.)

Default parameter is 600 000 (600 000ms = 600s = 10min).



6.4.13. ROI (REGION OF INTEREST)

This parameter determines a convex quadrilateral shape. Triggering will only happen inside this area, so if the number plate is out of the ROI, that is not going to generate an event. MMR is always working on the full image, but it is based on the number plate caught. Less area in ROI means less processing time. This function needs an array of 4 points. The points' order can be clockwise or counterclockwise and coordinates are normalized to [0.0, 1.0]. (0.0, 0.0) is equal to (0, 0) px of the image (top left corner) and (1.0, 1.0) is equal to (image.width - 1, image.height - 1) px of the image (bottom right corner).

Default parameter is *same as full frame ([(0, 0), (1, 0), (1, 1), (0, 1)])*. Example of a setting:



ROI is: [(0.39271, 0.174946), (0.653949, 0.157667), (0.701337, 0.99568), (0.242041, 0.965443)]

6.4.14. EVENT CALLBACK

This function is called when an event was generated in video processing. This function is called in asynchronous mode.

Examples are in 0 section.

If you don't set an event callback, the program won't call any function when an event occurs.

6.4.15. ON FRAME CALLBACK

This function is called after a frame from the source is decoded. This function is called in asynchronous mode. If you don't set an "onFrame" callback, the program won't call any function when a frame is decoded.

6.4.16. STATUS CHANGE CALLBACK

This function is called when the video processor status is changing. If you don't set a status change callback, the program won't call any function.

Possible stream status parameters are:



Callback function parameter is the changing Stream object and the new status of it.

6.4.17. Processing Mode

The CARMEN® Video SDK provides a processingMode setting that allows you to control how the stream processor handles incoming video streams or files. The StreamProcessorMode enum class defines the available modes:



- Auto: (*Default*) The SDK automatically determines the processing mode based on the source URL. If the URL starts with "file:", it will use NonLiveStream mode; otherwise, it will use LiveStream mode.
- LiveStream: Intended for real-time streaming sources (e.g., live camera feeds). In this mode:
 - Buffers can drop images if they overflow, which helps maintain real-time processing by discarding frames when the system is under heavy load.
 - Suitable for applications where staying current is more important than processing every single frame.
- NonLiveStream: Designed for processing video files. In this mode:
 - If buffers are full, the processor will not drop frames, ensuring that all frames are processed.
 - The video processing speed is limited by the hardware capabilities, not by the video's FPS.
 - With Go licenses, this mode is limited to 30 FPS.
 - Ideal for offline processing where completeness is required.

6.5. LOGGER

The inner library of the CARMEN[®] Video SDK makes logs. There are 5 log levels: *Debug, Info,* Warning, Error, Critical. The default log level is Warning. The default logging method is to logging to the standard output: "[\${LOG_LEVEL}]: \${MESSAGE}".

6.5.1. CHANGING LOGGING LEVEL

There is example of setting log levels in the "2_basic" and "3_result_details" sample codes in each program language.

6.5.2. SET LOGGING CALLBACK

There is example of changing log callback in the 3rd sample codes in each program language. Set logger callback only before initializing any Video SDK class!

In this callback there is option to write logs to file, or turn off the logger.



6.6. LICENSING

The CARMEN® Video SDK utilizes a flexible licensing system that can be configured according to your deployment needs. By default, GX reads the licensing mode from the gxsd.dat file. However, you can modify the licensing mode directly within your application code to suit different scenarios. For example implementations, refer to sample "05_cloud" to see how cloud licensing is set up. You can also adjust any of the other sample applications to use License Server mode if you wish to connect to a License Server or LSC while experimenting with these samples.

6.6.1. Licensing Modes

The SDK supports multiple licensing types, and you can retrieve the current licensing mode using the getCurrentLicensingType() function. This function returns an enum value representing the licensing type:

```
enum class LicensingType {
    Unknown,
    Local,
    LicenseServer,
    CloudNNC
...
```

- Unknown: The licensing type could not be determined.
- Local: Licensing is managed locally with a hardware key connected to the machine.
- LicenseServer: Licensing is managed through a License Server or LSC within your LAN or WAN.
- **CloudNNC**: Licensing for the PlateFinder preselector is managed via the CARMEN® Cloud using an API key.

6.6.2. Setting the Licensing Mode

You can set the licensing mode programmatically using the provided setter functions. This allows you to override the default settings from gxsd.dat and configure the licensing mode that best fits your application.

- Set Local Licensing
 - Use this function to configure the SDK to use local licensing with a hardware key connected to the machine:

void setLocalLicensing()

```
    Set License Server
```

Use this function to connect to a License Server or LSC within your network:

void setLicenseServer(const std::string& host, uint16_t

- host: The IP address or hostname of the License Server.
- port: The port number on which the License Server is running.
- Set Local Licensing
 - Use this function to configure cloud-based licensing using an API key:
- void setCloudNNC(const std::string& apiKey)
 - apiKey: Your CARMEN® Cloud API key for authentication.

7. RESULT CLASSES

The CARMEN® Video SDK provides several result classes to represent data extracted from video processing, including events for each vehicle captured. Event includes related information, such as license plate details, vehicle attributes, and relevant images.

7.1. IMAGE CLASSES

7.1.1. Image

The **Image** class represents a bitmap image in memory. In the C and C++ SDKs, it is provided as a class. In C#, users should utilize the corresponding image handling classes provided by the .NET framework or the SDK's specific implementation.

7.1.2. ImageProxy

The **ImageProxy** class is an image wrapper used in the SDK's callback parameters. It provides access to image metadata and allows the retrieval of image data through cloning.

7.2. ANPR (AUTOMATIC NUMBER PLATE RECOGNITION) CLASSES

7.2.1. Plate

The Plate class represents the result of a license plate recognition.

7.2.2. PlateDetection

The **PlateDetection** class contains information related to the license plate recognition process on an image.

7.3. MMR (MAKE & MODEL RECOGNITION) CLASSES

7.3.1. MmrData

The **MmrData** class contains attributes of a vehicle as recognized by the Make & Model Recognition process.

7.3.2. MmrDetection

The **MmrDetection** class contains information related to the Make & Model Recognition process on an image.

7.4. EVENT CLASSES

7.4.1. PlateOnImage

The **PlateOnImage** class contains information related to the license plate recognition process on an image, along with a reference to that image.

- Fields:
 - **detection**: *(PlateDetection)* The properties of the recognized license plate.
 - **image**: *(ImageProxy)* A reference to the image on which the license plate was detected.

7.4.2. MmrOnImage

The **MmrOnImage** class contains information related to the Make & Model Recognition process on an image, along with a reference to that image. **Fields:**

- detection: (MmrDetection)
 The Make & Model recognition result.
- **image**: *(ImageProxy)* A reference to the image on which the vehicle attributes were detected.

7.4.3. Vehicle

The **Vehicle** class represents a vehicle descriptor, which may include license plate information and vehicle attributes.

Fields:

- plate: *(optional, Plate)* The license plate of the vehicle. This may be null if no license plate was detected.
- mmrData: *(optional, MmrData)* The attributes of the vehicle as determined by the Make & Model Recognition process. This may be null if MMR was not performed.

7.4.4. Event

The **Event** class represents the result of processing a single vehicle passage. It aggregates all relevant information, including detected license plates, vehicle attributes, and associated images. An event may include multiple detections if the vehicle was captured in multiple frames or from multiple viewpoints.

8. REGION LIST

Region name	Region code
ARABIC	ARAB
Australia	AUS
Bangladesh	BGD
Caribbean	CAR
Central America	CAM
Central Asia	CAS
East Asia	EAS
Europe	EUR
General Latin	GEN
India	IND
Indonesia-Timor-Papua	ITP
Inside Asia	IAS
Iran	IRN
Iraq	IRQ
Israel	ISR
Japan	JPN
Nepal	NPL
New Zealand	NZL
North Africa	NAF
North America	NAM
Pacific	PAC
Pakistan	РАК
Philippines	PHL
South America	SAM
South Asia	SAS
Southern Africa	SAF
Taiwan	TWN
Turkey	TUR
Vietnam	VNM



KNOWN ISSUES 9.

- All C functions return with error code -1 if error occurs inside. Later there will be more dedicated • error codes
- Memory Usage Increase During Overload Conditions: When the system is overloaded, the • process's memory usage may grow beyond expectations and not decrease afterward due to FFmpeg's buffer handling, where internal buffers are not freed until the StreamProcessor object is destroyed.



ADI DEMO

1. INTRODUCTION

The ANPR Demo for Images (ADI) is a program that was developed by Adaptive Recognition o serve as a simple and versatile tool for evaluating our core LPR technology before having to first develop an application.

The goal of the SDK is to allow you to develop a similar or even a more complicated application based on the Carmen API in order to meet the exact requirements of the particular project where it will be deployed.

The demo can handle both a single image as well as an image directory as an input source. Supported image formats: .jpg, .jpeg, .png, .bmp, .jp2

This application can be found in the following folder:

- On Windows: "c:\Program Files\Adaptive Recognition\CARMEN softwares\Demos\ADI\"
- On Linux: "\opt\gx64\ADI_Demo\"

The latest available version from ANPR Demo for Images: 7.4.1.0

Not all images are adequate for ANPR, the input images must meet a specific set of criteria for the engine to be able to recognize them. Please study the following document to learn more about these requirements: Imaging for Carmen[®].

Important

Please note, that this application is not available for ARM and CentOS6 packages on Linux.



2. MAIN SCREEN

After starting the demo, you will be presented with the main screen of the application. This window is split into three separate sections; the **Result Image**, the **Result Tree** and the **Log** sections. All three sections provide information in connection with the scanned image.

🕽 Note

The application puts a red frame around the image area, where it has located a license plate and turquoise frame around the vehicle (only in case of running MMR).

🗿 ADI 🖸	emo 7.4.1.0	(64 bit) [ANPR: cm	anpr-7.3.16.127 : world]	- [MMR: mmr-7.3.5.9	9 : mmr-eur] - [Licens	e: Local NNC]											-	ð X
File Ed	t Process	Help																
										ANPR Result Tree								
										■ P412GTV	V							
	1 10	340		1, 764	NY OF			1000		Text(A	SCII): P4	12GTW						
in a	Y			Hit	the	1 1				Plate	ategory:							
14							THE OWNER			Confid	lence: 48							
	1 -	1				NEFIL				Typol	ry/state: Gu	latemala						
		in ha					1/100		11 mars	Color	RGB) (0)	2012						
1 E	- P72				300 / A				Lin	BkCole	or(RGB): (25	5,255,255)						
题 9	54.00	A 5.5								Frame	: ((2	028,616),(2151,613),(2156,670),(2	034,673))				
							10			Make	kmodel: Kia	Picanto						
T			Sac Harris				10	Contraction of the second		Catego	ory: Ca	r						
				$\mathcal{F}^{a}\mathcal{F}$	CARLE.					Color:	W	HITE						
				A Trans						ViewP	oint: rea	ar						
	2							102			time: 11	2						
										Chara	rters (2						
					It Im	000				• Tips			Docu	lt Tra				
	Path	Plate Text	Plate Text (ASCII)	Plate category	Country/State	Contidence	TypeID	Frame	Character Size	Text Background	Text Color	Dedicated Area Color	ANPR Time(ms)	Make&Model	Category	Color	ViewPoint	MMR confid *
E	- 04							((· • • • · · · · · ·		(01010)	(ennienniem				-			
5	C:/	P564JWV	P564JWV		GTM	42	602012	((659,6	29	(255,255,255)	(0,0,0)	No color	112	Mazda CX-5	Car	RED	front	99
6	C./	P412GTW	P412GTW		GIM	48	602012	((2028,	30	(255,255,255)	(0,0,0)	No color	158	Kia Picanto	Car	WHITE	rear	100
/	C./	M356139	M356139		NIC	55	605002	((629,9	51	(255,255,255)	(0,0,0)	No color	92	Chauma lat	Pick-up	WHITE	front	99
8	- C·/	01M002402	01M002402		UZB	//	221008	((/42,1	67	(0,0,0)	(255,255,2	No color	169	Cnevrolet	Car	WHITE	tront	85
9	C./	AA164YN	AA164YN		ALB	89	11/011	((493,6	29	(255,255,255)	(0,0,0)	No color	70	Audi A4	Car	GRAY	rear_side	99
10	C./	3483FPK	3483FPK	MIXED	ESP	83	128060	((305,3	18	(255,255,255)	(0,0,0)	No color	56	BIMIN 3	Car	GRAY	front	99
11	C./	ZN10840	ZN10840		SJM	23	193004	((1291,	53	(0,0,0)	(255,255,2	No color	268	Mitsubishi	Car	GRAY	tront	56
12		No result	No result	-		-	-	-	-	-	-	-	469	loyota	Car	WHITE	rear_side	99
13	C./	02/2/101	02/2/10143		DZA	31	402028	((253,5	34	(255,255,255)	(0,0,0)	No color	130	Peugeot 406	Car	BLUE	front	100
14	C:/	3xxA2597	3(12a0)		ETH	36	407006	((786,8	49	(255,255,255)	(0,0,0)	No color	286	Toyota HiAce	Van	WHITE	front_side	99
			All a second at		-			-	-	-	1 1 1		396	Ford	Pick-up	WHITE	front	
15	C:/	No result	ino result				31002212400444				0474204200	6.00 52	015,024		and the second s		(10,0740)	99
15 16	C:/	No result CFK9291	CFK9291	TEMPORARY	USA / FL	34	550083	((934,8	34	(255,255,255)	(0,0,0)	No color	119	Hyundai Sant	Car	GRAY	rear	99 100
15 16 17	C:/ C:/	No result CFK9291 ST86UA	CFK9291 ST86UA	TEMPORARY	USA / FL CPV	34 68	550083 446002	((934,8 ((620,5	34 54	(255,255,255) (255,255,255)	(0,0,0)	No color No color	119 85	Hyundai Sant Toyota Coaster	Car Bus	GRAY WHITE	rear front	99 100 99

Quick overview of the main screen

In the title you can see the following information:

ADI Demo 7.4.1.0 (64 bit) [ANPR: cmanpr-7.3.16.127 : world] - [MMR: mmr-7.3.5.9 : mmr-eur] - [License: Local NNC]

ANPR: the currently used ANPR engine region and version MMR: the currently used MMR engine region and version License: showing where CARMEN[®] is looking for the licenses (local/network)



🛛 Note

In case of missing hardware key, the following error message will appear:



3. FILE MENU

The ADI menu bar consists of four menu buttons, located in the upper left corner of the program window as follows:

File, Edit, Process, and Help.



The File menu contains the following menu items:

3.1. OPEN IMAGE(S)

Click to select specific images for LPR processing.

3.2. OPEN DIRECTORY

Click to specify a directory containing images for LPR processing.

3.3. SAVE LOG

Click to save the results log manually. The log can also be saved automatically by adjusting the Demo Preferences under the Edit menu.

4. EDIT MENU

The Edit menu contains the following menu items:

O ADI Demo 7.4.0.16 (64 bit) [ANPR: cmanpr-7.3.12.254 : bra] - [MMR: mmr-7.3.1.24 : mmr-gen] - [License: local]



4.1. ANPR / MMR SETTINGS

Contains various configuration options related to the ANPR / MMR process. The engine used for processing is also selected here. By clicking this menu item, the following pop-up window will appear:

Engine: Click the drop-down menu to select an engine from the list of installed engines.

Properties: Lists the property values for the selected engine. To adjust the pre-set values, click on one of the value fields and then input a desired value. A full description of the engine properties can be found in the <u>CARMEN® ANPR Reference Manual</u>.

Click on the **[Close]** button to apply your changes. These settings will remain active until the application is closed or until the engine in use is replaced with another one from the drop-down list. When you exit the application, the changes will be lost and the values will revert to the ones saved in the gxsd.dat configuration file.

Clicking on the **[Save Properties]** button will write the current values of each property into the gxsd.dat configuration file. This feature requires write permissions to the gxsd.dat file. The default values of an engine can be reset by reinstalling (uninstalling and installing) it, check it <u>here</u> how to do that. This function works only with ANPR properties as the MMR properties are read only.

0	ANPR Settings			×		
Curr	Current engine: cmanpr-7.3.13.5:gen					
Prop	erties:					
	Nam	e	Value	^		
1	adapt_environme	ent	0			
2	analyzecolors		0			
3	autotypemodification		0			
4	colortype		0			
5	confidencemode		7			
6	contrast_min		10			
7	convert0to0		0			
8	cyrillic_style		0			
9	datafile		cmanpr-13.5-gen.dat			
Sav	Save Properties Close					

ANPR Settings

0	MMR Settings	×
Cur	rrent engine: mmr-7.3.1.40:mm	r-cam 🔻
Pro	perties:	
1	Name	Value mmr-1.40-cam.dat
2	heading	frontal
3	tf_version_check	0
Sa	ve Properties	Close

MMR Settings

4.2. DEMO PREFERENCES

Contains various configuration options related to the demo software.

By clicking this menu item, the following pop-up window will appear:

Auto-Log:

If checked, the results will be logged automatically.

Find All Plates:

If checked, the engine will search for all plates in the image that conform to the set parameters.

Find Empty ADR

If checked, the engine will search for the Empty ADR plates as well.

Loop (toggle:L):

If checked, the program continuously repeats processing of the specified image sequence/folder.



If checked, the engine will process the specified sample image right after application start-up.

Real plate frame drawing

If checked, the real plate will be framed.

Show ROI

If checked, the set ROI will be visible on the image.

Show ROU

If checked, the set ROU will be visible on the image.

Sample Image:

This field shows the file that is loaded into ADI upon starting the application.

Log Directory:

Displays the directory, where the log file will be saved. Click [...] button to specify another directory. The default folder is: "*installation folder / logs*"

Demo Preferences	×
Settings Auto-Log Find Empty ADR Process Sample Image At Start-Up Show ROI Sample Image: Log Directory: Auto-Process (toggle: Space)	 ➢ Find All Plates □ Loop (toggle: L) □ Real plate frame drawing ☑ Show ROU demo_images/EUR_01.jpg …
Auto-Process Delay(ms):	1000 🛨
Font Family: Read MMR automatically MMR engines: mmr-7.3.5.9 ; mr	Segoe UI
Show MMR frames	/ehicle frame
	OK Cancel

MMR engine is installed

Demo Preferences	×
Settings	
Auto-Log	Find All Plates
Find Empty ADR	Loop (toggle: L)
Process Sample Image At Start-Up	Real plate frame drwaing
Show ROI	Show ROU
Sample Image:	demo_images/hun_MGJ292.jpg
Log Directory:	
Auto-Process (toggle: Space)	
Auto-Process Delay(ms):	10
Font Family:	Segoe UI V
Read MMR automatically	
MMR engines:	¥
	OK Cancel

MMR engine is not installed

Auto-Process (toggle: Space):

If checked, the application runs through the images and do ANPR on them automatically.

If 'Auto-Process' is on, you will not be able to examine the individual results, even if there is no more image to processing. You must set it back to 'Manual'. You can do it by clicking on the image and then pressing the 'space' on the keyboard.

Auto-Process Delay(ms):

Type or use the arrows to adjust the amount of delay (in milliseconds) before the next image is loaded during Auto-Process.

Font Family: Select the font style of the Result Tree (LPR details in the upper right corner of the ADI screen).

Read MMR automatically: (can be checked only in case if you have any MMR engine installed). If checked, ADI demo will automatically return the MMR results.

MMR frame:

- vehicle frame: the frame enclosing the entire vehicle.
- extended plate frame: the frame enclosing a broader area around the license plate. •
- processed frame: This frame marks the vehicle, processed by the engine. (extended pate • frame or in its absence a vehicle frame).

🛛 Note

MMR engines need separate MMR licenses and must be the same region as ANPR license to be able to work together. GEN MMR engine can work with every ANPR engines (except GEN ANPR engine, which is not capable to run with any MMR engine)

5. **PROCESS MENU**

The Process menu contains the following menu items:

O A	DIDe	emo 7.4.1.	0 (64 bit) <mark>[ANPR: cma</mark> r	npr-7.3.16.127 : world] - [MMR: OFF] - [License: Local NNC]
File	Edit	Process	Help	
Next Image (toggle: N) Previous Image (toggle: P)			mage (toggle: N) us Image (toggle: P)	

Click Next Image (N) or Previous Image (P) to navigate between the images. Alternatively, you may also use the N/P keys on the keyboard or simply click once on the image.





6. **HELP MENU**

The Help menu contains the following menu items:



User Manual: Opens this User Manual.

About: Provides the license, version, and copyright information for the installed application.

About ADI Demo					
	ADI Demo (ANPR Demo for Images)				
	ADI Demo is a sample program that shows how you can use CARMEN ANPR software for recognizing number plates on images loaded from the file sytem.				
	Version: 7.4.1.0				
	Copyright © 2012-2024, Adaptive Recognition				
	If you need help from our support, please follow this link				
	For more information please visit our site: https://adaptiverecognition.com				
	ОК				



7. SET ROI/ROU ON THE IMAGE

From CARMEN® 7.3.1.27 there is a possibility to set more ROI/ROU polygons on the image which is

loaded to ADI Demo application.

Steps:

- ROI: Press CTRL+I
- Start clicking the points of the polygon, if you are ready, press CTRL + R to save ROI



In written form: 22,731;1468,893;922,2047;29,2047 If you would like to add more polygons, press CTRL+I again



In written form: 22,731;1468,893;922,2047;29,2047 + 2036,915;2546,2079;3783,2105;3671,900



REQUESTINFO@ADAPTIVERECOGNITION.COM WWW.ADAPTIVERECOGNITION.COM
- ROU: Press CTRL+U
- Start clicking the points of the polygon, if you are ready, press CTRL + R to save ROI



In written form: 7,7;11,1345;3826,1447;3823,40 If you would like to add more polygons, press CTRL+U again



In written form: 7,7;11,1345;3826,1447;3823,40 + 22,2014;3815,2177;3823,2622;18,2622

73/114

Adaptive Recognition AmericaAdaptive Recognition Hungary

Adaptive Recognition **Nordic** Adaptive Recognition **Singapore**

This is how it looks like if you are using ROI and ROU at the same time:



🛛 Note

ROU is the stronger property so if there is ROI and ROU on the same part of the image. The engine will not search on that area!

The property will be saved into the set property, but if you would like to save them to GXSD.DAT as well, do not forget to press Save Properties button in the Edit -> Demo Preferences menu

If you would like to delete the set ROI/ROU in ADI Demo you can do that by pressing ALT + "I" for ROI or ALT + "U" for ROU.



8. MAGNIFIER

If the characters are very small on the visible processed image in ADI Demo, but there is a result for that, for example:



But are not able to decide if it is correct or not, you can hover your mouse over the license plate which you would like to check and hold down the **right button** and a magnifier will highlight the surrounded area.



9. ANPR RESULT TREE

The ANPR Result Tree is located to the right of the input image display area. It is divided into three main sections.

1. License plate result:

 Provides a detailed summary of the recognized image:
 * EUI9755

 Text(ASCII): alphanumeric license plate text in ASCII characters
 Text(ASCII)

 Plate type: category of the license plate
 Confidence:

 Confidence: overall confidence level (%) of the plate.
 Color(RGE)

 Country/State: Country full name / state full name
 Dedicated

 TypeID: code containing country/state ID of the plate.
 Make&mode

 Color: color of the dedicated area of the license plate.
 Make&mode

 BkColor: background color of license plate text.
 MMR com

 Dedicated Area Color: color of dedicated area in RGB format
 * Characters

 Frame: pixel coordinates of the plate corners.
 * Code

Only available if MMR is enabled!

- Make&Model: the recognized vehicle Make and Model data
- Category: the recognized vehicle category
- Color: the color of the recognized vehicle
- ViewPoint: the viewpoint of the recognized vehicle
- MMR Confidence: overall MMR confidence
- MMR Time: MMR processing time in milliseconds

2. Characters:

Individual details for each recognized character. Ordered from left to right than up to down.

- Code: unicode character ID
- Confidence: confidence level (%) of the overall plate
- Color: color of the character
- BkColor: background color of character
- Frame: pixel coordinates of the character corners

ANPR Result Tree	
 EUI9755 	
Text(ASCII):	EU19755
Plate catego	COMMON
Confidence:	59
Country/State:	Brazil
TypeID:	657027
Color(RGB):	(0,0,0)
BkColor(RGB):	(255,255,255)
Dedicated A	(255,255,255)
Frame:	((317, 1754), (485, 1749), (484, 1799), (315, 1803))
Make&model:	VW Polo
Category:	Car
Color:	GRAY
ViewPoint:	front
MMR confid	99
MMR time:	163
✓ Characters (
~ E	
Code:	0x0045
Confide	99
Color(R	(0,0,0)
BkColor	(255,255,255)
Frame:	$((324,1771),\!(342,1771),\!(341,1797),\!(323,1798))$
> U	
>	
> 9	
> 7	
> 5	
> 5	
> E	

3. Tips:

List of preliminary results from which the engine assembled the result.

- _ Code: unicode character ID
- Confidence: confidence level (%) of the overall plate
- Color: color of the character _
- BkColor: background color of character
- Frame: pixel coordinates of the character corner



When ADI is not in automatic mode, and it encounters an individual image with multiple plates, it will display all results from that image in the same result tree (if Find-All-Plates is checked in Edit -> Demo Preferences).





10. ANPR DATA

Results of the ANPR process are displayed in the **log** section in the bottom part of the screen. The log contains the following data:

- Path: location of the image file.
- Plate Text: alphanumeric license plate text.
- Plate Text (ASCII): alphanumeric license plate text in ASCII characters
- Plate category: category of the license plate.
- **Country/State** *ISO 3166-1 alpha-3* country code / State. If you hover your mouse over the Country/State column in the Log area, it will pop up the full name of the country / state.
- **Confidence:** overall confidence level of the plate.
- TypeID: our internal ID of the currently recognized plate type
- Frame: pixel coordinates of the corners of the license plate.
- Character Size: average height of the characters in pixels.
- Text Background: background color of text in RGB format.
- Text Color: character color in RGB format.
- Dedicated Color: color of dedicated area in RGB format.
- ANPR Time(ms): ANPR processing time in milliseconds.

Only in case if MMR reading is enabled in Demo Preferences and there is an ANPR result.

- Make&Model: the recognized vehicle Make and Model data
- Category: the recognized vehicle category
- Color: the color of the recognized vehicle
- ViewPoint: the viewpoint of the recognized vehicle
- MMR Confidence: overall MMR confidence
- MMR Time(ms): MMR processing time in milliseconds

If you need more information about MMR please check this document: MMR Brief Description.

#	Path	Plate Text	e Text (AS	Plate category	Country/State	Confidence	Typel D	:rame	racter !	fext Background	Text Color	Dedicated Area Color	ANPR Time(ms)	Make&Model	Category	Color	ViewPoint	MMR confidence	MMR Time(ms)
	C:/I	FOD0121	FOD0121	COMMON	BRA	44	657027	((1	29	(255,255,255)	(0,0,0)	(255,255,255)	226	Chevrolet Camaro	Car	YELLOW	front	79	219
2	C:/I	EU19755	EUI9755	COMMON	BRA	59	657027	((3	27	(255,255,255)	(0,0,0)	(255,255,255)	37	VW Polo	Car	GRAY	front	99	163
3	C:/I	EZJ0074	EZJ0074	COMMON	BRA	60	657027	((3	30	(255,255,255)	(0,0,0)	(255,255,255)	38	Mitsubishi Outl	Car	BLUE	front_side	99	89

🛛 Note

The table above can be saved in a semi-colon delimited log file by clicking File/Save Log.

By clicking a data row, the corresponding image and its result tree will appear above the log.

DATA LOGGING 11.

The application saves each ANPR process in a log file.

Naming format of the log file:

ADI_yyyy-mm-dd_hhmm_sss.log (ADI_Year-Month-Day_HourMinute_Second.log)

E.g.: ADI_2012-07-09_1712_001

If the date changes, the process of the automatic logging continues in a different log file.

When saving a log file, a separate anpr file with the same name will also be created in the same directory that contains the property settings of the currently used engine.

Format of the log file:

UTF16 (LE) encoding

Semicolon separated values

First Line: header with information in the following order:

#;Path;Plate Text;Plate Text (ASCII);Plate category;Country/State;Confidence;TypeID;Frame;Character Size;Text Background;Text Color;Dedicated Area Color;ANPR Time(ms);Make&Model;Category;Color;ViewPoint;MMR confidence;MMR Time(ms);EADR type;EADR frame



ADV DEMO

🛛 Note

From Carmen 7.3.1.28 this application is not part of the package any more.

1. INTRODUCTION

The ANPR Demo for Videos (ADI) is an application that was developed by Adaptive Recognition to serve as a simple, yet versatile tool for testing, evaluating, and familiarizing oneself with the core features the CARMEN[®] engine has to offer. This document will describe in detail the features and functionalities available within the demo software.

ADV can operate live (processing a live camera stream) or as a backend process (processing a recorded video stream). The demo supports the following video formats:

- .mjpg
- .mjpeg
- .avi
- .mpg
- .mpeg
- .mp4
- .mkv

Not all images are adequate for ANPR processes, the input images (in this case the video frames) have to meet a specific set of criteria in order for the engine to be able to recognize the licenses. Please study the following document to learn more about these requirements: <u>Imaging for Carmen[®]</u>. This application can be found in the following folder:

- On Windows: "C:\Program Files\Adaptive Recognition\CARMEN softwares\Demos\ADV\"
- On Linux: "\opt\gx64\ADV_Demo\"

🛛 Note

The latest available version from ANPR Demo for Videos: 7.4.0.22

Important

Please note, that this application is not available for ARM and CenOS6 packages on Linux.

🕻 Note

The application was tested only with MJPEG streams from ParkIT and FreewayCAM cameras. Using third-party cameras as a source may lead to performance issues.

a. .

2. MAIN SCREEN

After starting ADV, you will be presented with the main screen of the application. This window is split into four separate sections; the Active Stream, the Result Image, the Result Tree and the Log sections. All four sections provide information in connection with the scanned image. The ADV menu bar consists of seven menu buttons, located in the upper left corner of the program window. The following menus are available:

- File
- Edit
- View
- Camera
- Video Player
- Process
- Help

A	ctive St	ream		Re	sult Im	age	Partie - Some - Some	meth) 1900 Conlance Two Generation Dataset two Constitution Dataset two Constitution Dataset two Constitution Dataset two Conference Conference Conference Conference Conference Conference Conference Conference Conference Conference Conference Conference Conference Conference Conference Conference Conference Conference Dataset the Dataset the Dataset Dataset the Dataset the		ult Tre
	France Time	Pate leat	Country State	Customer	liane	Ownerfor Size	last Beckground	Teast Caller	Destinated Calor	ANPS Tendinal
96.0.0	1975-01-01 01dA3*288	(AC CRORAL	6N (21628)	34	INFRIDAKING AND AND	10	858	100,00,000	655	
98.1.2	1875-01-01 01 03 7 228	\$2'0944	614-270225	17	INUE OF BUILDING	10	0.1.0	0010109	458	34
9623	1975-01-01 01 03 37 380	40.03944	010270035		ISHLICE/PROCESUREME.	-10	41.0	000,000,000	6.5.0	
108	1975-01-01 01 (0-40-80)	8725444	EN CTUBEI	-8	DOLUGIOUS DE ST. DALESSI.	-10	10.00	010110	8,5,8	
98.1.0	1976-01-01 0103-45.025	1075444E	IDA CTIERO	21	GREET, MATHINGSON,	10	0.55	-1275,275,275	655	31
96.2.0	1976-01-01 01 01 04 0 MB	8755444	EN 07006	*	167325.04408.04398.	- 19	610	0201201209	43.0	
	1815-01-01 01:03-43.788	(#1852546	KN4 (270028)		348.275,738.075,796.495.	14	4.5.0	620,201,276	83.0	
100.00					hand which they want they also			citere inch static	810	
106.0.0 106.1.4	1975-01-01 0143-41-000	818802554	EN CREDE		Designed on the round of the		4.6.9	-60'07'08		
100.2.0 100.1.0 100.2.0	1975-01-01 0140-41.000 1975-01-01 0140-41.046	#1002554 #1002554	EN CREUS EN CREUS		Decard Art. Mt. Mt. Ort are.	10	610	100,00,00	610	
106.0.0 106.1.0 106.0.0 106.0.0	1975-01-01 1120-02.00 1975-01-01 1120-02.00 1975-01-01 1120-02.00	FINGSSI FINGSSI (FINGSSI	EN CTEUR EN CTEUR	LOG	CHESTICAL STRUCTURE	10	4.55	271,271,276 (271,271,276)	610	e a
1963.0 1963.0 1963.0 1963.0 1963.0 1963.0	1975-01-01 2123-41.00 1975-01-01 2123-41.00 1975-01-01 2123-41.00 1975-01-01 2123-41.00	100554 200554 200554	EN CTEUR EN CTEUR EN CTEUR EN CTEUR	LOG	DREAM AND ADDREAM AND ADDREAM AND ADDREAM ADDR	15 14 16	418 418 418	201201200 201201200 201201200 201201200	6.5.0 42.0 42.0	6 17 24
19630 19618 19638 19638 19650 19658 19658	1975-01-01 21428-4148 1975-01-01 21428-4148 1975-01-01 21428-4148 1975-01-01 21428-4148	Entropy Entropy Entropy Entropy	EN CTEUR EN CTEUR EN CTEUR EN CTEUR EN CTEUR	LOG	INFORMATION OF COMPANY INFORMATION OF ANY INFORMATION OF ANY INFORMATION OF ANY INFORMATION OF ANY INFORMATION OF ANY	1	400 400 400 400	201201200 201201200 201201200 201201200 201201200	6.50 4558 6558 6568	е 7 3
1962-0 1962-0 1962-0 1962-0 1962-0 1962-0 1962-0	1955 51 of 120041.000 1955 51 of 120041.00 1955 51 of 120040.00 1955 51 of 120040.00 1955 51 of 120040.00 1955 51 of 120040.00 1955 51 of 120040.00	FIRCON FIRCON SHIFTER FIRCON FIRCON	EN-CTEUD DN-CTEUD EN-CTEUD EN-CTEUD EN-CTEUD EN-CTEUD	LOG	International Action Control Action International Action International Action International Action International Action International Action		4.00 4.00 4.00 4.00 4.00 4.00	271,272,275 271,271,275 271,272,275 271,272,275 271,272,275 271,272,275	6.58 6.28 6.28 6.28 6.28	6 7 94 40
19620 19618 19628 19628 19628 19628 19628		8180554 8180554 818058 818058 818058 818051 818051	EN-CTEUD EN-CTEUD EN-CTEUD EN-CTEUD EN-CTEUD EN-CTEUD EN-CTEUD EN-CTEUD	LOG	LAND, THE CALL AND THE THE AND		4.00 4.00 4.00 4.00 4.00 4.00 4.00	20120120 20120120 20120120 20120120 20120120 20120120 20120120	42.00 4220 4230 4230 4230 4230 4230	8 7 9 4 4 4 4

Quick overview of the main screen

In the title you can see the following information:

0	ADV Demo 7.4.0.21 (64 bit) [ANPR: cmanpr-7.3.14.205 : eur] - [License: local]								
File	Edit	View	Camera	Video Player	Process	Help			

ANPR: the currently used ANPR engine region and version MMR: the currently used MMR engine region and version License: showing where CARMEN[®] is looking for the licenses (local/network)

🛛 Note

In case of missing hardware key, the following error message will appear:

💿 Warn	ing X	<
\bigotimes	There is a problem with the engine: cmanpr-7.3.12.238 : eur -> No license for the engine	
	ОК	

The upper part of the screen is divided into three sections:

- Active Stream: Section on the left that displays the active video stream.
- **Result Image:** Section in the centre that displays the image corresponding to the row selected in the log.
- **Result Tree:** Section on the right that lists the details for each frame processed. ADV saves all engine parameters for each frame processed. In addition, it also displays an ANPR result tree where it recognized a license plate.

The lower part of the screen contains only one section.

• Log: Shows the license plate data for each frame. See a detailed description in the <u>ANPR Data</u> chapter.

3. FILE MENU

Open: displays two options:

 Video from file: Select this option to add a single video file (see supported formats above) for ANPR processing.



• Video(s) From Directory: Select to add a directory with multiple video files (see supported formats above) for ANPR processing.

Exit: Closes the application.

EDIT MENU 4.

4.1. TRIGGER CONFIGURATION

Important

Trigger configuration largely depends or monitors front or rear license plates ar configuration will always be custom for eve

The trigger signal can automatically identify an observed event (the time interval during which the system



encounters a vehicle for ANPR processing). Typically, one event represents a sequence of frames/images of only one vehicle at a time.

Trigger configuration allows the user to determine a time interval when the system anticipates only those frames from the continuous video frame sequence that are relevant for ANPR. From these relevant images, the engineer must decide what to select as triggered images. Triggered images should be those that were captured at an optimal time (when all of the optical characteristics of the target license plate are in their ideal ranges).



The diagrams below demonstrate how Pre-Trigger Time and Post-Trigger Time are used to define what the triggered images are within an observed event. The application will request the triggered images from the camera for ANPR processing. Since the camera buffers the images internally, even negative offset values of a few seconds (depending on the frame rate) can be set. The following examples demonstrate some possible trigger scenarios:



1. Trigger on the rising edge

2. Trigger on the falling edge



Trigger Settings				×
Trigger Type No Trigger Keyboard Timed Motion Detector	Settings			
		[OK	Cancel

No Trigger: The engine processes all incoming frames that are not dropped from the buffer.

Trigger Settings			×
Trigger Type No Trigger Keyboard Timed Motion Detector	Settings Pre-Trigger Time: Post-Trigger Time:	200 ms 100 ms	•
		ОК	Cancel

Keyboard: Generates a manual trigger event when the [F10] button is pressed. The engine processes all images within a specified time interval of the manual trigger (defined in milliseconds by the **Pre-Trigger Time** and **PostTrigger Time** values).

Trigger Settings		×
Trigger Type No Trigger Keyboard Timed Motion Detector	Settings Period: 1000 ms Image/trigger: 1	* *
	OK	Cancel

Timed:	The engine	processe	es ir	mages accordi	ng to	о а
preset	frequency	(defined	in	milliseconds,	up	to
60,000r	ms).					

Trigger Settings		×
Trigger Type No Trigger Keyboard Timed Motion Detector	Settings On Rising Edge (beginning of motion) On Falling Edge (end of motion) Pre-Trigger Time: 200 ms \$\$ Post-Trigger Time: 100 ms \$\$ Image: Setting Former \$\$ \$\$ Setting Former \$\$ \$\$	
	OK Cancel	

Motion Detector (Only applicable to Adaptive Recognition cameras): The application uses the built-in motion detection feature of the camera to create a trigger event. Select either the On Rising Edge (beginning of motion) or the On Falling Edge (end of motion) option. The first one will capture the images at the beginning of the motion, the second option at the

end of motion. **Pre-Trigger Time** and **Post-Trigger Time** allows you to set a time interval before and after the trigger signal. The engine processes all images captured within this specified interval.

In No Trigger and Timed modes, the system sends individual images for processing, while in Keyboard and Motion Detector modes, it sends image packages based on the pre-set time intervals.

When developing a final application with an efficient system architecture in mind, it is highly recommended to use a hardware trigger (e.g., inductive loop, infrared gate) in order to identify events. This way the events will only contain relevant frames with useful LPR information.

4.2. ANPR THREADS

Select the number of simultaneous ANPR processing threads. In order to run multiple processing threads parallel to each other, each thread must have a dedicated CPU core and a CARMEN[®] software license (NNC hardware key). Multi-core software licenses for parallel processing, are available in dual and quad versions.

Possible values: 1, 2, 4, 8





Adaptive Recognition **Nordic** Adaptive Recognition Singapore

ANPR SETTINGS 4.3.

Contains various configuration options related to the ANPR process. The engine used for processing is also selected here.

By clicking this menu item, the following pop-up window will appear:

Engine: Click the drop-down menu to select an engine from the list of installed engines.

Properties: Lists the property values for the selected engine. To adjust the pre-set values, click on one of the value fields and then input a desired value. A full description of the engine properties can be found in the CARMEN® ANPR Reference Manual.

ing	ine: cmanpr-7.3.1	3.5:gen	
rop	perties:		
	Name	Value	
1	adapt_environment	0	
2	analyzecolors	0	
3	autotypemodification	0	
4	colortype	0	
5	confidencemode	7	
6	contrast_min	10	
7	convert0toO	0	
8	cyrillic_style	0	
9	datafile	cmanpr-13.5-gen.dat	

Click on the [Close] button to apply your changes. These settings will remain active until the application is closed or until the engine in use is replaced with another one from the drop-down list. When you exit the application, the changes will be lost and the values will revert to the ones saved in the gxsd.dat configuration file.

Clicking on the [Save Properties] button will write the current values of each property into the gxsd.dat configuration file. This feature requires write permissions to the gxsd.dat file. The default values of an engine can be reset by reinstalling (uninstalling and installing) it, check it here how to do that.



ANPR PROCESSING PREFERENCES 4.4.

The ANPR Processing Preferences menu item contains three sub-menu items.

OA	DV De	emo 7.4.().20 (64 bit)	[cmanpr-7.	3.12	.238 : eur]	
File	Edit	View	Camera	Video Play	er	Process	Help
		Trigger (Configurati	on			
		ANPR T	hreads		•		
	ANPR Settings						
		ANPR P	rocessing F	references	•	lma	age Buffer
		Log Sett	ings			Bes	t Plate
						Fin	d All
						Dro	op Same Text

Image Buffer:

Max Bytes: Maximum size of the image buffer, specified in kilobytes or megabytes. The size of the image buffer is equal to the sum of all the image sizes in the buffer. When the image buffer reaches the pre-set value, ADV will start to overwrite the oldest images in the buffer.

Image Buffer		×					
Max Bytes:	0 🖨 MB 🔻						
Number of packages:	0						
Number of frames:	0						
Number of data bytes: 0 kB							
ОК	Cancel Apply						



Best Plate:

This is an algorithm that aids the selection of the best possible results.

ADV anticipates plates in the Plate Area marked with green on the sample frame. You can adjust the Plate Area by configuring the **Top Margin, Left Margin, Bottom Margin, and Right Margin** fields.

The Target Area (rectangle marked with an X) is used to identify the part of the frame that is most likely to contain an optimal license plate image.

Best Plate	×
Top Margin: 🛛 þ% 🚔	
Left Margin: 0% 🖨	
Bottom Margin: 0% 🖨	
Right Margin: 0% 🖨	
	L
Minimum Frequency: 1 🖨	Horizontal Preference: Center
Sufficient Frequency: 4 ≑	Vertical Preference: Center
	OK Cancel

Minimum Frequency: The minimum number of identical license plate recognitions within the image package of a single event.

Sufficient Frequency: The number of identical license plate recognitions required to stop additional image processing within the image package of a single event.

Horizontal Preference: Moves the target area horizontally.

Vertical Preference: Moves the Target Area vertically.

ADV will not consider a result for final selection if any of the following applies:

- 4. The plate is not within the margins
- 5. Minimum Frequency criteria is not fulfilled

From the remaining results, the application will select a final result by finding the frame that is closest to the Target Area (frame centre point closest to "X").

Find All:

If checked, the system searches for all license plates on the image.

Drop Same Text:

If checked, the system will not provide a result when there was an identical license plate text in the previous 20 seconds.

LOG SETTINGS 4.5.

Allows you to specify the contents, size, and path of the log as well as the corresponding images.

Log Settings	×
Log Contents All Triggered Frames All Processed Frames All Frames With ANPR Result Selected Results Only	Log Size Limit Max. Size/log file: 1024 kB Max. Size of Image Log Directory: 64 MB Log Mode O Disabled O Synchronous (Not Recommended) () Asynchronous (logger thread)
Log Directory Log Directory: Image Log Directory:	OK Cancel

5. VIEW

5.1. FONTS

Set Tree Font...

Change the font used to display the Result Tree (the detailed ANPR data found in the upper right corner of the main screen).

Set Table Fonts...

Change the font used to display the Log (the table of results in the lower part of the ADV screen). Set Image Font...

Select the font used to display the license plate text displayed in the image.





RESULT IMAGE 5.2.

Set Text Color:

Select the color of text displayed in the image.

Text position:

Adjust the position of the text displayed in the image.



Possible values:

Result text will not be displayed in the image: 🗹 No text Result text will be displayed as indicated by sub-menu item:

- 🗵 Above Plate Frame
- 🗵 Top Left Corner
- 🗵 Top Right Corner
- Bottom Left Corner
- 🗵 Bottom Right Corner

Display Character Frames:

Display a frame around each license plate character found in the image. Possible values: on, off



TABLE COLUMNS: 5.3.

Select the columns that should be displayed in the header of the Log.

- # (sequence number of frame)
- Frame Time
- Plate Text
- 🗵 Country/State
- Confidence
- Frame
- Char.Size
- Text Background
- Text Color
- 🗵 Dedicated Area Color
- ANPR Time

5.4 **TABLE ROW HEIGHT**

Adjusts the row height of the Log by clicking on SADV Demo 7.4.0.20 (64 bit) [cmanpr-7.3.12.238 : eur] Increase or Decrease.

Hotkeys for Increase: Ctrl + +

Hotkeys for Decrease: Ctrl + -







6. CAMERA MENU

By clicking this menu item, the following pop-up window will appear:

Connect To New Adaptive Recognition Camera:

The default factory IP address of new Adaptive Recognition cameras is set to 192.0.2.3 with the 255.255.255.0 netmask. Clicking on this menu item will start a scanning process that will look for this type of IP cameras in the default subnet.

will will)V Den	no 7.4.0.	20 (64 bit)	[cmanpr-7.3.12.	238 : eur]				
*****	File	Edit	View	Camera	Video Player	Process	Help			
				Con	Connect To Factory Default ARH Camera					
				Con	nect To IP Cam	era				
				Disc	onnect/Close					
				Sho	w Histogram					
				Rec	ord					
				Stop	0					
Co	nnec	ting					:	×		
Searchi Pinging	ng fa 192.	or fre 0.2.	e/un 4	used IP		Ca	ncel]		

Important

You must set up the network adapter with the same subnet range as the factory default IP subnet of the ARH cameras. Please avoid IP address conflicts by selecting a number different from the camera's default "3" in the last section of the IP address.

Connect To IP Camera:

Allows user to connect to a camera on the network.

🛛 Note

The stream URL of the cameras are different by manufacturers! Please visit your camera's manual to get more information about its stream link (only "mjpeg" stream is allowed). Authentication on the cameras may also change this URL address.

	×	Connect To Camera	
Camera Type		Camera Type	
ARH Camera O MJPEG stream		O ARH Camera	MJPEG stream
Address Address: 192.168.3.37 Control Port: 80 User Name: Password:	÷	Address URL: 192.168.3.37:9901/1 Note: The default port is "http://". So if the MJPEG stream is o 80" in the URL after the IP a Example URL of the MJPEG stream of 192.168.2.247:80/axis-cg// camera=8showlength=18re	video.mjpeg s 9901 and the address must not con n the default HTTP port you have to inse address/domain name. fan Axis camera: mjog/video.cgi? solution=640x480
Saved		Saved	
Alkotas		Aikotas	
Alkotas	Save	Allector	C
Alkotas Alkotas	Save	Alkotas	Save
Alkotas Alkotas	Save	Alkotas	Save
Alkotas Alkotas	Save Load Delete	Alkotas	Save Load Delet

Disconnect/Close:

Disconnects the camera and closes the session.

Show Histogram:

Displays the image histogram on the live view image of the camera (Feature only available with Adaptive Recognition cameras).

Record:

Allows user to save the MJPEG stream downloaded from the camera (or other URL).

Stop:

Stops the recording of the MJPEG stream.

7. VIDEO PLAYER MENU

Allows the user to control playback and SADV Demo 7.4.0.20 (64 bit) [cmanpr-7.3.12.238 ; eur] switch between multiple video streams (if more than one video stream has been loaded).

Play One: Playback will stop at the end of

the video stream.

- Play All: Playback continues with the next video stream until the end of the last one.
- Repeat One: Plays the selected video in an infinite loop.
- Repeat All: Plays all selected videos in an infinite loop.

_					-			
File	Edit	View	Camera	Video Player	Process	Help	_	
				Pause		Ctrl+Shift+Space	-	
				Play		Ctrl+Shift+Up		
				Previous		Ctrl+Shift+Left		
				Next		Ctrl+Shift+Right		
				Stop		Ctrl+Shift+Down		
				Play/Rep	eat Mode	•	~	Play One
								Play All
								Repeat One
								Repeat All



1 h

8. PROCESS MENU

Start:

Allows the user to initiate the ANPR process.

Stop:

Allows the user to manually stop the ANPR process.

Time Window:

The ANPR process can also be stopped by setting a timer in the Time Window sub-menu prior to starting the ANPR process.

O A	DV Den	no 7.4.0.	20 (64 bit)	[cmanpr-7.3.12.	238 : eur]				
File	Edit	View	Camera	Video Player	Process	Help			
					Sta	rt			
					Sto	р			
					Tir	ne Window	•	~	No Time Window
									1 sec
									10 sec
									1 min
									10 min

9. HELP MENU

By clicking this menu item, the following pop-up window will appear:

		.238 : eur]	[cmanpr-7.3.12.	.20 (64 bit)	no 7.4.0.	DV Den	Al
	Help	Process	Video Player	Camera	View	Edit	File
lelp Contents	H						
bout	Д						

Help Contents: Opens this User's Manual

About: Provides the license, version, and copyright information for the installed application.

About	it ADV Demo	×
(\mathbf{i})	ADV Demo (ANPR Demo for Videos)	
	ADV Demo is a sample program that shows how you can: - Handle the stream of ParkIT/Freeway cameras with the help of the - Process the images with the cmAnpr module - Save the stream to an mjpeg file.	E ARHIP API
	Version: 7.4.0.22	
	Copyright © 2012-2023, Adaptive Recognition	
	If you need help from our support, please follow this link	
	For more information please visit our site: https://adaptiverecognition.com	
		ОК

10. ANPR RESULT TREE

For each selected frame in the **log** section, the **Result Tree** displays all of the corresponding details available.

FRAME:

Shows date- and timestamp of the processed frame, and contains subheadings called **Params** and **ANPR Results**.

I. PARAMS:

Parameters contains a list of all the engine property settings.

- II. ANPR RESULT(S):
- 1. LICENSE PLATE RESULT:
- Confidence: Overall confidence level (%) of the plate.
- Type: Code containing country/state ID of the plate.
- Color: Color of the dedicated area on the license plate.
- BkColor: Background color of license plate text.
- Frame: Pixel coordinates of the plate corners.

2. CHARACTERS:

Individual details of each character of the final result. Ordered from left to right.

- Code: Unicode character ID.
- Confidence: Confidence level (%) of the overall plate.
- **Color:** Color of the character.
- BkColor: Background color of character.
- Frame: Pixel coordinates of the character corners.

3. TIPS:

List of preliminary results from which the engine assembled the final result.

- Code: Unicode character ID.
- **Confidence:** Confidence level (%) of the character tip.
- Color: Color of the character.
- **BkColor:** Background color of character.
- **Frame**: Pixel coordinates of the character corners



ANPR DATA 11.

Results of the ANPR process are displayed in a log found at the bottom part of the screen. The log contains the following data:

- Frame Time: Time and date when the image was capture
- Plate Text: Alphanumeric license plate text.
- Country/State ISO 3166-1 alpha-3 country code.
- Confidence: Overall confidence level of the plate.
- Frame: Pixel coordinates of the corners of the license plate.
- Character Size: Average height of the characters in pixels.
- Text Background: Background color of text in RGB t. •
- Text Color : Character color in RGB.
- Dedicated Color: Color of dedicated area in RGB.
- ANPR Time(ms): ANPR processing time in milliseconds.

#	Frame Time	Plate Text	Country/State	Confidence	Frame	Character Size	Text Background	Text Color	ANPR Time(ms)
24.0.0	2012-12-04 16:26:23.691	[ARH001]		92	((306,332),(424,329),(425,360),(306,362))	26	(255,255,255)	(0,0,0)	21
23.0.0	2012-12-04 16:26:23.642	[ARH001]		91	((307,332),(422,330),(423,359),(307,361))	25	(255,255,255)	(0,0,0)	18
22.0.0	2012-12-04 16:26:23.592	[ARH001]		92	((307,333),(424,331),(424,360),(308,361))	26	(255,255,255)	(0,0,0)	21
21.0.0	2012-12-04 16:26:23.541	[ARH001]		91	((307,332),(424,331),(424,359),(308,360))	25	(255,255,255)	(0,0,0)	22
29.0.0	2012-12-04 16:26:23.941	[ARH001]		91	((307,332),(424,329),(425,358),(308,361))	26	(255,255,255)	(0,0,0)	20
20.0.0	2012-12-04 16:26:23.491	[ARH001]		92	((307,332),(424,329),(425,360),(308,362))	26	(255,255,255)	(0,0,0)	22
1.0.0	2012-12-04 16:26:22.541	[ARH001]		91	((306,333),(424,329),(424,359),(306,362))	26	(255,255,255)	(0,0,0)	22
18.0.0	2012-12-04 16:26:23.391	[ARH001]		91	((306,333),(424,332),(424,359),(306,361))	26	(255,255,255)	(0,0,0)	17
17.0.0	2012-12-04 16:26:23.342	[ARH001]		93	((306,332),(424,331),(424,360),(306,360))	27	(255,255,255)	(0,0,0)	18
16.0.0	2012-12-04 16:26:23.292	[ARH001]		91	((306,331),(423,331),(423,360),(306,360))	25	(255,255,255)	(0,0,0)	19

The data present in the table above is included in the log file.



DATA LOGGING 12.

Log entries of the ANPR processes are logged by the application in the CARMEN® installation folder. Naming format of the log file:

ADV_yyyy-mm-dd_hhmm_sss.log ADV_Year-Month-Day_HourMinute_counter E.g.: ADV_2021-02-08_1726_000.log

If the date changes, then automatic logging continues in a different log file

When saving a log file, a separate anpr file with the same name is also created in the same directory that contains the property settings of the engine being used.

FILE FORMAT:

UTF16 (LE) encoding Semicolon separated values First Line: header as follows (see detailed descriptions under ANPR Data) #;Path; Plate Text;Country/State;Confidence;Frame;Character Size;Text Background;Text Color; Dedicated Color;ANPR Time(ms);

In case of missing hardware key, the application indicates it in the status bar at the bottom of the screen.





ODI DEMO

1. INTRODUCTION

The OCR Demo for Images (ODI) is a program that was developed by Adaptive Recognition to serve as a simple and versatile tool for evaluating our core OCR technology before having to first develop an application.

The goal of the SDK is to allow you to develop a similar or even a more complicated application based on the Carmen® API in order to meet the exact requirements of the particular project where it will be deployed.

The demo can handle both a single image as well as an image directory as an input source. Supported image formats: .jpg, .jpeg, .png, .bmp, .jp2

This application can be found in the following folder:

- On Windows: "C:\Program Files\Adaptive Recognition\CARMEN softwares\Demos\ODI\"
- On Linux: "\opt\gx64\ODI_Demo\"

The latest available version from OCR Demo for Images: 7.3.2.0

Not all images are adequate for OCR, the input images have to meet a specific set of criteria in order for the engine to be able to recognize them. Please study the following document to learn more about these requirements: Imaging for Carmen[®].

Important

Please note, that this application is not available for ARM and CenOS6 packages on Linux.



2. MAIN SCREEN

After starting the demo, you will be presented with the main screen of the application. This window is split into three separate sections; the **Result Image**, the **Result Tree**, the **Process Handler** and the **Log** sections. All four sections provide information in connection with the scanned image.



Quick overview of the main screen

In the title you can see the following information:

OCR Demo 7.3.2.0 (64 bit) [OCR: cmocr-7.3.17.68 : iso] - [License: local]

OCR: the currently used OCR engine type and version **License:** showing where CARMEN[®] is looking for the licenses (local/network)



3. **FILE MENU**

The ODI menu bar consists of four menu buttons, located in the upper left corner of the program window as follows:

File, Edit, Processing, and Help.



The File menu contains the following menu items:

3.1. **OPEN DIRECTORY**

Click to specify a directory containing images for OCR processing.

3.2. **OPEN IMAGES**

Click to select specific images for OCR processing.

3.3. **CLOSE**

Close the ODI Demo application.



4. EDIT MENU

The Edit menu contains the following menu items:



4.1. ENGINE SETTINGS

Contains various configuration options related to the OCR process. The engine used for processing is also selected here.

By clicking this menu item, the following pop-up window will appear:

Current engine: Click the drop-down menu to select an engine from the list of installed engines.

Properties: Lists the property values for the selected engine. To adjust the pre-set values, click on one of the value fields and then input a desired value. A full description of the engine properties can be found in the <u>CARMEN® OCR Reference</u> <u>Manual</u>.

Click on the **[Close]** button to apply your changes. These settings will remain active until the application is closed or until the engine in use is replaced with another one from the drop-down list. When you exit the application, the changes will be lost and the values will revert to the ones saved in the gxsd.dat configuration file.

Clicking on the **[Save Properties]** button will write the current values of each property into the gxsd.dat configuration file. This feature requires write permissions to the gxsd.dat file. The default values of an engine can be reset by reinstalling (uninstalling and installing) it, check it <u>here</u> how to do that. This function works only with ANPR properties as the MMR properties are read only.



Sav	ve properties		Close	
48	timeout	2000		•
47	symbolvertical	32		
46	symbolspace	32		
45	symbolnewrow	32		
44	symbollowcha	0		
43	symbollowcha	0		
42	symbolhyphen	32		
41	symbolarms	32		
40	slope_min	-100		
39	slope_max	100		
38	slope	0		
37	slant_min	-100		
36	slant_max	100		
35	slant	0		
34	size_min	10		
33	size_max	80		
32	size	36		
31	ROU			
30	ROI			
29	recognitionm	Fix_Normal		
28	read_without_t	1		
27	read_licence	1		
26	read_emptyad	0		
25	read_base_type	0		

DEMO PREFERENCES 4.2.

Contains various configuration options related to the demo software.

By clicking this menu item, the following pop-up window will appear:

Logging section:

Time Format:

Allow the user to customize the time format.

Log File:

Browse where you would like to save the log files.



Enable Logging:

If you check this box, ODI will save the log files as you set above.

Processing section:

Max. no. of container codes in a picture:

Set maximum how many codes are allowed to return from 1 image.

Max. no. of cached images:

Set maximum how many images can be stored in the queue.

Max. no. of images in group:

Set maximum how many images are allowed in one group of images (Group means that engine will return only 1 time for the same code from the number of images in the group).

Auto Process Delay:

Set how many milliseconds should wait between 2 OCR processes.

Rotate Image for Display:

Set if you would like to rotate the images if the codes are turned on the image.

Initial Directory:

Browse from which directory ODI should search for an image for processing on start-up.

Demo Preferences		×
Logging		
Time Format:		
yyyy-MM-dd hh:mm:ss		
Log File:		
Inc/CARMEN softwares/Demos/ODI/logs/log.csv	Browse	
Enable Logging		
Processing		
Max. no. of container codes in a picture:	10	•
Max. no. of cached pictures:	30	-
Max. no. of images in a group:	unlimited	•
Auto Process Delay:	2000 ms	-
Rotate Images for Display:	Yes	•
Initial Directory:		
on/CARMEN softwares/Demos/ODI/demo_images	Browse	
Close		

PROCESSING MENU 5.

The **Processing** menu contains the following menu items:

Click Next Group (Right Arrow) or Previous Group (Left Arrow) to navigate between the images/groups. Alternatively, you may also use the left/right arrow keys on the keyboard.

OCR Demo 7.3.1.6 (64 bit) [OCR: cmocr-7.3.2.77:iso] - [License: local]									
File	Edit	Processing Help							
		Next G	roup						
		Previou	is Group						
		ReScan							
		Auto P	rocessing						
		Loopin	g						
		Find Al	l l						

ReScan:

If you click on this item, ODI will do the OCR process again on the same image/group.

Auto-Process:

If checked, the application plays the images automatically as a slideshow. (You can find related settings in the Demo Preferences menu)

Looping:

If checked, the program continuously repeats processing of the specified image sequence/folder.

Find All:

If checked, the engine will search for all plates in the image that conform to the set parameters.



6. **HELP MENU**

The Help menu contains the following menu items:

OCR Demo 7.3.1.6 (64 bit) [OCR: cmocr-7.3.2.77:iso] - [License: local]								
File	Edit	Processing	Help					
Γ			L A	Jser Manual About				

User Manual: Opens this User Manual.

About: Provides the license, version, and copyright information for the installed application.

About ODI Demo X									
	ODI Demo (OCR Demo for Images)								
	DI Demo is a sample program that shows how you can use CARMEN OCR oftware for recognizing ISO/UIC/ILU/DOT codes on images loaded from the file /tem.								
	Version: 7.3.2.0								
	Copyright © 2014-2024, Adaptive Recognition								
	If you need help from our support, please follow this <u>link</u>								
	For more information please visit our site: <u>https://adaptiverecognition.com</u>								
	ОК								



7. OCR RESULT TREE

The OCR Result Tree is located to the right of the input image display area. It is divided into three

main sections.

ECMU 9115841 45G1	
category	
checksum	0
checksum_validatio	n valid
codelen	15
confidence	97
🕀 images	
nimage	3
picture	C:/Program Files/Adaptive Recognition/CARMEN softwares/Demos/ODI/demo_images/4518258.jpg
🕀 pictures	
proctime	306 msec
success	true
type	995601 (OCR ISO_Type)

4. Code result:

Provides a detailed summary of the recognized image:

- **checksum:* the value of the check digit calculated based on the read code
- **checksum_validation:* the validity of the calculated checksum value based on comparison with the actual digit present in the code.
- codelen: length of the code
- confidence: the overall confidence of the result
- nimage: number of images in the group
- picture: path of the first image in the group
- proctime: the processing time in milliseconds
- success: true or false based on the results
- type: our internal ID of the currently recognized code

Important Important

* Starting from Q1 2024, the released OCR engines no longer have a separate **checksum** value, because its calculation and verification have been integrated into the algorithm. Therefore, for the new engines, the **checksum** and **checksum_validation** fields should be disregarded.

Ę	- ir	nages	
	E	characters	
		checksum	0
		checksum_validation	valid
		confidence	90
		imgindex	3
		ncharacter	15
		- picture	C:/Program Files/Adaptive Recognition/CARMEN softwares/Demos/ODI/demo_images/4518258.jpg
		text	ECMU911584145G1
	E	- characters	
		checksum	0
		checksum_validation	valid
		confidence	87
		imgindex	1
		ncharacter	15
		- picture	C:/Program Files/Adaptive Recognition/CARMEN softwares/Demos/ODI/demo_images/4518153.jpg
		text	ECMU911584145G1
	B	 characters 	
		checksum	0
		checksum_validation	valid
		confidence	R9

5. Results on the images:

Individual details for each recognized character on every image in the group. Ordered from left to right than up to down.

• characters:

- o code: ASCII code
- o code_char: ACSII character
- o confidence: confidence of this character
- o frame: pixel coordinates of the character corners
- *checksum: the value of the check digit calculated based on the read code
- **checksum_validation:* the validity of the calculated checksum value based on comparison with the actual digit present in the code.
- confidence: confidence level (%) of the overall plate
- imgindex: index of the image in the group
- ncharacter: number of characters on this image
- **picture:** resized image, with its path
- text: recognized text from the image
- type: our internal ID of the currently recognized code

Important

* Starting from Q1 2024, the released OCR engines no longer have a separate **checksum** value, because its calculation and verification have been integrated into the algorithm. Therefore, for the new engines, the **checksum** and **checksum_validation** fields should be disregarded.

images	
characters	
code	69
code_char	E
confidence	100
frame	(15,7)-(16,7)-(16,9)-(15,9)
code	67
code_char	С
confidence	100
frame	(17,7)-(18,7)-(18,9)-(17,9)
code	77
code_char	м
confidence	98
frame	(18,7)-(19,7)-(19,9)-(18,9)
code	85
code_char	U
confidence	100
frame	(19,7)-(20,7)-(20,9)-(19,9)
code	57



6. Pictures:

List of resized images in the group with their path.

8. OCR DATA

Results of the OCR process are displayed in the **log** section in the bottom part of the screen. The log contains the following data:

- Code: the read code from the image/group.
- Codelen: length of the code.
- Char height: average height of the characters
- Type: our internal ID of the currently recognized code
- *checksum_validation: the validity of the calculated checksum value based on comparison with the actual digit present in the code.
- Confidence: overall confidence level of the plate.
- Processing time: OCR processing time in milliseconds
- No. of Images: number of images in the group.
- Picture: path of the first image in the group.
- Success: true or false based on the result.

Important

* Starting from Q1 2024, the released OCR engines no longer have a separate **checksum** value, because its calculation and verification have been integrated into the algorithm. Therefore, for the new engines, the **checksum** and **checksum_validation** fields should be disregarded.

_													
Γ	Code	Codelen	Char height	Туре	Category	Checksum	Checksum validation	Confidence	Processing time	No. of Images	Picture	Success	Error
1	TC VU252 032 9	11	2	0 (No type result)		32765	valid	81	96 msec	1	C:/Program Files/Adaptive Recognition/CARMEN softwares/Demos/ODI/demo_images/	true	
2	ECMU 9115841 45G1	15	1	995601 (OCR ISO_Type)		0	valid	97	307 msec	3	C:/Program Files/Adaptive Recognition/CARMEN softwares/Demos/ODI/demo_images/	true	
3	CPIU 561789 4 45G1	15	2	995690 (OCR ISO_Type)		0	valid	69	630 msec	1	C:/Program Files/Adaptive Recognition/CARMEN softwares/Demos/ODI/demo_images/	true	
4	NOSU 2463 45 4 SG 2210	17	2	995606 (OCR ISO_Type)		0	valid	76	237 msec	1	C:/Program Files/Adaptive Recognition/CARMEN softwares/Demos/ODI/demo_images/	true	
5	CPIU 561789 4 45G1	15	2	995690 (OCR ISO_Type)		0	valid	69	631 msec	1	C:/Program Files/Adaptive Recognition/CARMEN softwares/Demos/ODI/demo_images/	true	
6	ECMU 9115841 45G1		1	995601 (OCR ISO Type)			valid	97	306 msec		C:/Program Files/Adaptive Recognition/CARMEN softwares/Demos/ODI/demo images/	true	


9. DATA LOGGING

The application saves each OCR process in a log file if you allow it in the Demo Preferences menu. Naming format of the log file: it is set in Demo Preferences menu

Format of the log file:

UTF16 (LE) encoding Semicolon separated values First Line: header with information in the following order: #time;code;codelen;checksum;checksum_validation;confidence;proctime;nimage;picture;success;err or



SDK DESCRIPTION

Example codes to show how to use CARMEN® for license plate and industrial code recognition.

1. ANPR SAMPLE CODES

CMANPR01

At the beginning of the execution the application checks the syntax of the gxsd.dat file with the new ispropertiesvalid() function. If the gxsd.dat is valid, the application loads the default engine check if is there any available license for it and use it for an ANPR process. The application loads an image and if the engine finds a license plate on it then the application will print out the result (Plate text, country code and internal type).

CMANPR02

Same as CMANPR01, but do the ANPR process, until new plate is not found on the same image, or timeout is not exceeded.

CMANPR03

Same as CMANPR02, just do the ANPR process on several images, which you can give to the application as argument.

CMANPR04

Same as CMANPR03, just do the ANPR process until MAXSTEPS (99) is not reached. The application measures the ANPR time for every recognized plate on every image and calculate the average of the first recognition time and the total recognition time for every steps.

CMANPR05

This is quite the same as CMANPR01, but this time in a multithread application. This application shows the difference between a single core license and a dual core license usage. If you have only a single core license the application runs the 2 times 100 readings one after another, but in case of dual core licenses it is done parallelly.

CMANPR06

If you have an engine which is capable to give you back the ADR/Empty ADR plates as well, you can check with this application how to read License Plates and ADR Plates on the same image.

CMANPR07

List the currently default ANPR engine and the licenses which are capable to run that engine. After this the application will show you all the installed engines and the licenses which are capable to run them.

110/114

CMANPR08

Shows the way how to use the 'K' licenses. This application simulates an entry point, each image represents a car in front of the barrier, after successful ANPR reading the application opens the barrier, if free credit still available. The images are processed from a folder which was given to the application as an argument.

CMANPR09

Shows the way how to use multi-step engines. Load more engines and use them in a sequence. If the first engine didn't give you back any result than the application calls the same image with the next engine and so on. The images are processed from a folder which was given to the application as an argument.

MMR01

Same as CMANPR02, but do MMR function after every successful ANPR reading automatically.

2. OCR SAMPLE CODES

CMOCR01

At the beginning of the execution the application checks the syntax of the gxsd.dat file with the new ispropertiesvalid() function. If the gxsd.dat is valid, the application asks the user which engine should be used for testing. The user can choose with an input from a keyboard / console. After everything is set the application choose the proper images (image series from the same code, from different view) for the engine and do the OCR reading with ReadCode() / cmocr_readcodea() function. The result contains the industrial code, the confidence and the checksum validation result (if possible).

CMOCR02

At the beginning the application asks the user which engine should be used for testing. The user can choose with an input from a keyboard / console. After everything is set the application choose the proper images (image series from the same code, from different view) for the engine and do the OCR reading with FindFirstContainerCode() / cmocr_getresult() function. The result contains the industrial code, the confidence and the checksum validation result (if possible)

CMOCR03

At the beginning the application asks the user which engine should be used for testing. The user can choose with an input from a keyboard / console. After everything is set the application choose the proper image for the engine and do the OCR reading with FindFirstContainerCode() and FindNextContainerCode() / cmocr_getresult() and the cmocr_nextresult() functions. The result contains the industrial code, the confidence and the checksum validation result (if possible). With this application is it possible to read more industrial codes from one image.

3. GX SAMPLE CODES

GXDEVICES01

This application list how many GX devices are connected to your computer and the following details about them: name, type, serial, priority.

GXDEVICES02

This application list how many and what kind of licenses are attached to your system.

GXLICENSES01

This application list all the GX devices (type and serial) attached to your system and the available licenses on them.

4. EXAMPLE CODES AVAILABILITY

ON WINDOWS

	С	C++	C#	Java	VB .NET
CMANPR EXAMPLES					
CMANPR01	+	+	+	+	+
CMANPR02	+	+	+	+	+
CMANPR03	+	+	+	+	+
CMANPR04	+	+	+	+	+
CMANPR05	+	+	-	+	-
CMANPR06	+	+	+	+	+
CMANPR07	+	+	NA	NA	NA
CMANPR08	+	+	+	+	+
CMANPR09	+	+	+	+	+
MMR01	+	+	+	+	+
CMOCR EXAMPLES					
CMOCR01	+	+	+	+	+
CMOCR02	+	+	+	+	+
CMOCR03	+	+	+	+	+
GX EXAMPLES					
GXDEVICES01	+	+	+	NA	+
GXDEVICES02	+	+	+	+	+
GXLICENSES01	+	+	+	NA	+

+	Available	
-	Not ready yet	
NA	Not possible	

Adaptive Recognition AmericaAdaptive Recognition Hungary

REQUESTINFO@ADAPTIVERECOGNITION.COM WWW.ADAPTIVERECOGNITION.COM

ON LINUX

	С	C++	Java		
CMANPR EXAMPLES					
CMANPR01	+	+	+		
CMANPR02	+	+	+		
CMANPR03	+	+	+		
CMANPR04	+	+	+		
CMANPR05	+	+	+		
CMANPR06	+	+	+		
CMANPR07	+	+	NA		
CMANPR08	+	+	+		
CMANPR09	+	+	+		
MMR01	+	+	+		
CMOCR EXAMPLES					
CMOCR01	+	+	+		
CMOCR02	+	+	+		
CMOCR03	+	+	+		
GX EXAMPLES					
GXDEVICES01	+	+	NA		
GXDEVICES02	+	+	+		
GXLICENSES01	+	+	NA		

+	Available	
-	Not ready yet	
NA	Not possible	





CONTACT INFORMATION

Headquarters: Adaptive Recognition, Hungary Inc. Alkotás utca 41 HU 1123 Budapest Hungary Web: adaptiverecognition.com

Service Address: Adaptive Recognition, Hungary Inc. Ipari Park HRSZ1113/1 HU 2074 Perbál Hungary Web: adaptiverecognition.com/support/

Adaptive Recognition Technical Support System (ATSS) is designed to provide you the fastest and most proficient assistance, so you can guickly get back to business.

Information regarding your hardware, latest software updates and manuals are easily accessible for customers via our Documents Site (www.adaptiverecognition.com/doc) after a quick registration.

New User

If this is your first online support request, please contact your sales representative to register you in our Support System. More help here (www.adaptiverecognition.com/support)!

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 <u>atsshelp@adaptiverecognition.com</u>.

