

# **SmartBuilder**

## **Program control by LAN**

Version: 2.6.037  
Date: 2014\_02\_20

# Contents

Overview .....	3
Description of the protocol .....	3
Table of existing commands .....	3
001 Check Program change #001 .....	3
002 Operation mode „Start“ #002.....	4
003 Operation mode „Stop“ #003 .....	4
004 Operation mode „Stop immediately“ #004.....	4
005 Command „Restart program“ #005.....	4
006 Command „Reset Counters“ #006 .....	4
007 Command „Trigger simulation“ #007.....	5
008 Command „Ping“ #008 .....	5
009 Command „RAW-Message“ #009 .....	5
010 Command “Set System Time” #010 .....	6
014 Command “DeleteAllStaticVars” #014.....	6
016 Command “SelectProgram” #016.....	6
017 Command “AddProgram” #017.....	7
018 Command “UnloadProgram” .....	7
020 Command “SetParameter for command on position” #020.....	7
021 Command “SetParameter for command with command name” #021 .....	7
023 Command “GetParameter for command with command name” #023 .....	8
024 Command “Test Program Store” #024.....	8
998 Command „Reset Camera“ #998 .....	8
999 Command “ExitEyeVision” #999 .....	8
Command “Softreset” .....	8
Stoploops.....	9
GetAllInfo: .....	9
Addition for all LAN commands: .....	10
Acknowledgement of the successful execution .....	11
Using the implementation of the protocol in the SmartBuilderControl dll.....	12

## **Overview**

For the program control by LAN two different options are at ones command. On the one hand the communication protocol between the SmartBuilder software and the camera can be implemented afterwards. This is by comparison quite complicated and the usage is subjected to restrictions or rather characteristics of this connection (e.g. immediate stop of the inspection program as the link is connected). On the other hand there is a very simple and proprietary protocol for the inspection program control by UDP. This is described below.

The communication port number is 5952 in the standard case and should only be changed in an emergency, because the SmartControl tool or rather the network browsing service finds the VizCam cameras also by this port number. The changing of this port number can be carried out in the EVHD.CFG-File in the section [RemoteProfile] in the entry PConLanPort.

## **Description of the protocol**

**All control telegrams have to be sent as UDP packets to the camera (normally) on port 5952.**

### **Control protocol:**

A control telegram always consists of a start character („#“), a command number of three characters (for example „001“), and an optional data array of maximum 200 characters. Every telegram is terminated by a terminator („#“). In the special case of „RAW – Message“ (see below), the message length is limited to 65.000 bytes. Only the commands „STOPLOOPS“, „GETALLINFO“ and „RESET“ are an exception of this telegram convention.

## **Table of existing commands**

The following table describes the existing commands:

### **001 Check Program change #001**

Telegram: **#001Std.ckp#**

Function: The telegram switches to the check program Std.ckp. In the operation mode „Start“ the program is started immediately in a continuous operation. In the operation mode „Stop“ the program is only loaded and activated. The specification of the file extension „.ckp“ is compulsory. Upper and lower case is relevant.

To check if the request was successful one can readout the current inspection programs for example by the GETALLINFO-command as well as by the explicit „acknowledgement of the successful execution“. More details are available in the corresponding section on page 10 of this manual.

## **002 Operation mode „Start“ #002**

Telegram: **#002#**

Function: The active check program will be started. The operation mode start will stay active even if no program is loaded or rather a no readable program is created. After check program switch the loaded program will be started immediately.

By request an explicit “acknowledgement of the successful execution” is carried out. More details are available in the corresponding section on page 10 of this manual.

## **003 Operation mode „Stop“ #003**

Telegram: **#003#**

Function: The active check program will be stopped after the next complete run. After a program change the new check program will stay stopped until the operation mode „Start“ has been selected.

By request an explicit “acknowledgement of the successful execution” is carried out. More details are available in the corresponding section on page 10 of this manual.

## **004 Operation mode „Stop immediately“ #004**

Telegram: **#004#**

Function: The check program will be stopped immediately. All running operations will be stopped. After a program switch the new program will stay stopped until the operation mode „Start“ has been chosen. Use this command to abort „endless loops“ in the script editor and waiting for trigger signal.

By request an explicit “acknowledgement of the successful execution” is carried out. More details are available in the corresponding section on page 10 of this manual.

## **005 Command „Restart program“ #005**

Telegram: **#005#**

Function: The check program will be stopped immediately and started again. This command matches to the „Stop immediately“ command followed by „Start“.

By request an explicit “acknowledgement of the successful execution” is carried out. More details are available in the corresponding section on page 10 of this manual.

## **006 Command „Reset Counters“ #006**

Telegram: **#006#**

Function: The good- and bad counter as well as the counter of the number of program loops will be reset to zero. The function is done asynchronous to any running check program and is therefore also possible if the camera e.g. is waiting for a trigger signal.

## 007 Command „Trigger simulation“ #007

Telegram: **#007#**

Function: Simulates a trigger signal. Even if a hardware signal does not exist, a triggered image capture can be simulated.

The trigger simulation is not supported by the line scan camera.

An explicit “acknowledgement of the successful execution” is not sent, because the command is carried out directly. The trigger simulation influences the active or subsequently triggered image capture.

## 008 Command „Ping“ #008

Telegram: **#008#**

Function: This command always returns „OK“ (even if acknowledgements are disabled) as soon as the camera received it. The addressability of the camera can be tested with this command. An explicit “acknowledgement of the successful execution” is not sent, because the command is carried out directly.

## 009 Command „RAW-Message“ #009

Telegram: **#009CCLM<sub>0</sub>...M<sub>n-1</sub>#**

Function: By means of this telegram it is possible to send a message M by LAN to the camera. The camera accepts this message with its RS232 receive buffer. Therefore one can test the reaction of the inspection program to serial messages, even without actually having a serial connection. Unfortunately it is currently not possible to route back return messages, which were sent serially by the camera, also by LAN telegram.

The command code #009 is followed by two configuration bytes **CC**, that should be filled with 0 (=0x00).

The configuration bytes will be used for a specification of check sums or other extensions in later versions. The next byte **L** describes the length **M** in bytes.

Should the cubic capacity **M** have more than 255 characters, **L** has to be set to 0 (=0x00) and the following 2 bytes (interpreted as 16 bit value (LSB first)) describe the length. In this case the telegram has the form: #009**CC**0**LH**M#, where **L** and **H** stand for low and high byte of the message length.

## 010 Command “Set System Time” #010

Telegram: **#010;**<Sec>;<Minute>;<Hour>;<Day>;<Month>;<Year>#

Function: The system time is set according the parameters. After the telegram number 010, 5 parameters are awaited, each separated by a space character.

<Sec>	0 - 59
<Minute>	0 - 59
<Hour>	0 - 24
<Day>	1 - 31
<Month>	1 - 12
<Year>	1900 –

Especially for VizCam cameras:

The system time is the amount of seconds since 00:00 January 1<sup>st</sup>, 1900. Some camera models are resetting the system time after each time the camera is restarted, and the system time has to be newly adjusted. An explicit “Acknowledgement of the successful execution” is not sent, because the command is carried out directly.

**Telegram numbers #011# - #013#: Reserved for the image transfer.** Details for this see also the document “supplement for the document program control by LAN.pdf”

## 014 Command “DeleteAllStaticVars” #014

Telegram: **#014#**

Function: All static variables are deleted.

By request an explicit “acknowledgement of the successful execution” is carried out. More details are available in the corresponding section on page 10 of this manual.

**Telegram number #015#: Reserved for the image transfer**

## 016 Command “SelectProgram” #016

Telegram: **#016** <programname>#

Function: The program <programname> is selected and ready to use. <programname> is the complete name of the inspection program (e.g. “Std.ckp”). If the inspection program is already loaded in the memory (e.g. as subprogram) or explicit by the command AddProgramm (#017), the program can be started shortly without having to load it into the memory or activate it again from the Flash or the SD-card.

It is recommended to execute this command only when the camera is set to the operation modus “Stop” (see also commands #003# and #004#).

## 017 Command “AddProgram” #017

Telegram: **#017 <programname>#**

Function: The program <programname> is loaded into the memory and activated. <programname> is the complete name of the inspection program (e.g. “Std.ckp”). All other programs remain in the memory. If one wants to start the program <programname> one has to select first (see also command #016).

It is recommended to add programs only when the camera is set to the operation modus “Stop” (see also commands #003# and #004#). Otherwise the cycle time of a running program can not be guaranteed.

Use a combination of #016 and #017 to prevent long times when loading and activating at the inspection program change. The programs in the memory can only be deleted by a normal inspection program change (#001). Please note that the inspection program needs main memory. It depends on the camera type and the inspection program on the camera, how many programs can be loaded simultaneously.

By request an explicit “acknowledgement of the successful execution” is carried out. More details are available in the corresponding section on page 10 of this manual.

## 018 Command “UnloadProgram”

Telegram: **#018 <programname>#**

It is currently not used on VizCam cameras.

## 020 Command “SetParameter for command on position” #020

Telegram: **#20;<Position>;<ParType>;<ParName>;<Value>#**

<Position>: command position at the inspection program

<ParType>: type of parameters String, Double, Long

0 – String

1 – Double

2 – Long

<ParName>: Name of the parameters

<Value>: new value

By request an explicit “acknowledgement of the successful execution” is carried out. More details are available in the corresponding section on page 11 of this manual.

## 021 Command “SetParameter for command with command name” #021

Telegram: **#21;<CMDName>;<ParType>;<ParName>;<Value>#**

<CMDName>: command name in the comment column of the command

<ParType>: type of parameters String, Double, Long

0 – String

1 – Double

2 – Long

<ParName>: Name of the parameters

<Value>: new value

By request an explicit “acknowledgement of the successful execution” is carried out. More details are available in the corresponding section on page 11 of this manual.

## 023 Command “GetParameter for command with command name” #023

Telegram: #23;<CMDName>;<ParType>;<ParName>#

<CMDName>: command name in the comment column of the command

<ParType>: type of parameters String, Double, Long

0 – String

1 – Double

2 – Long

<ParName>: Name of the parameters

By request an explicit “acknowledgement of the successful execution” is carried out. More details are available in the corresponding section on page 11 of this manual.

(This command is available from version 2.6.032 onwards)

## 024 Command “Test Program Store” #024

Telegram: #24Save.ckp#

The telegram stated stores the parameters of the active test program under the name Save.ckp. The name can be chosen arbitrarily. If there is already a test program with this name, it is overwritten.

## 998 Command „Reset Camera“ #998

Telegram: #998#

Function: This command performs a soft reset of the camera. Besides the “reset” command there is also another possibility. One can trigger a soft-reset of the camera within the #-commands.

## 999 Command “ExitEyeVision” #999

Telegram: #999#

Function: The SmartBuilder software is stopped immediately. The active inspection program is terminated at once. To connection with the PC SmartBuilder software is not possible any more. Also a soft reset by LAN-command is no longer possible. Also to use SmartControl for a direct camera control is no longer possible.

The camera is still accessible via telnet. (Optionally for the operating system update) By starting the program “es.cex” in the telnet panel, the software can be restarted.

By request an explicit “acknowledgement of the successful execution” is carried out. More details are available in the corresponding section on page 11 of this manual.

## Command “Softreset”

Telegram: RESET

Function: A soft reset is executed. The camera is restarted. The command is replaced by #999# and only exists out of compatibility reasons.



## Stoploops

Telegram: STOPLOOPS

Function: The “Stoploops” command is sent always if continuing processes on the camera should be interrupted. This command is necessary for the SmartBuilder software to interrupt long processes, to get a fast connection to the camera. It is not recommended to use this command. If one wants to interrupt immediately, it is better to use the command “Stop immediately”.

## GetAllInfo:

Telegram: GETALLINFO

Function: After reception of this command the camera sends a string that contains information about the current state of the camera. Beyond information about the camera type, operating system and software version you will also find information about the current check program like name, cycle time and information about licence.

Answer about GETALLINFO. All parameters are separated by hex=00 !

Description of the GETALLINFO Protocol (Version 1)

<b>Protocol Version</b>	1	
<b>Camera Name</b>		
<b>Camera Type</b>	ES <NR>00	
<b>Camera IP</b>		
<b>Camera Color Type</b>	0 = grey 10 = color	
<b>Connect-Port</b>		
<b>Control-Port</b>		
<b>Software-Version</b>		
<b>OS Version</b>		
<b>Valid Licence File</b>	1 = Licence File Error ( Demo Mode) 0 = Licence ok	
<b>Licence covers program</b>	0 = undetermined 1 = Licence covers program 2 = Additional Licence needed	
<b>Current Program</b>		
<b>Program Status</b>	0 = Stopped 1 = Running else = Error	
<b>Cycle Time</b>		
<b>Good Counter</b>		
<b>Bad Counter</b>		
<b>Serial Program Control Active</b>	0 = disabled 1 = enabled	
<b>Digital IO Program Control Active</b>	0 = disabled 1 = enabled	

### ***Addition for all LAN commands:***

The user receives an acknowledgement. This is "OK" if the telegram could be parsed by the camera or "NOK" if the telegram has the correct shape but the command code isn't known (for example #99#).

In case of any unknown error the camera returns the received telegram, so the telegram can be analysed on the senders site.

#### **IMPORTANT ADDITION:**

The SmartBuilder software also sends commands for the check program control to the camera.

The command of the SmartBuilder software is always higher prior than the commands of the LAN protocol.

If the camera is connected to the SmartBuilder software, all commands via LAN are ignored (except for RESET, STOPLOOPS and GETALLINFO).

The user receives an "IGNORED" instead of "OK" or "NOK".

## ***Acknowledgement of the successful execution***

Some commands can not be executed directly, if the inspection program is in a state, which does not allow execution. Therefore the commands are inserted in a queue. As soon as the command has been executed, an answer will be sent to a defined UDP receiver. To the command “completed” or “failed” is attached, depending on whether the execution was successful or not.

Concretely this concerns commands with the characteristics (the respective answer is attached):

```
#001 „#001<programname># completed“ or „#001<programname># failed“
#002 „#002# completed“
#003 „#003# completed“
#004 „#004# completed“
#005 „#004# completed“ gefolgt von „#002# completed“
#006 is executed directly, does not need an extended acknowledge
#007 is executed directly, does not need an extended acknowledge
#010 is executed directly, does not need an extended acknowledge
#014 „#014# completed“
#016 „#016<programname># completed“ or „#016<programname># failed“
#017 „#017<programname># completed“ or „#017<programname># failed“
#020 „#020;<Position>;<ParType>;<ParName>;<Value># completed“ or „#020
<Position>;<ParType>;<ParName>;<Value># failed“ Note: The number of positions after decimal
point can be different from the sent commands.
#021 „#021;<CMDName>;<ParType>;<ParName>;<Value># completed“ or „#021
<CMDName>;<ParType>;<ParName>;<Value># failed“ Note: The number of positions after
decimal point can be different from the sent commands.

#999 „#999 completed“
```

The receiver can be entered into the `evhd.cfg` file :

The entries in the section [RemoteProfile] are entered similar to the entry for the PconLanPort.

**UsePConLANResponse** = [0|1]

Here the entry 1 causes, that the answer is active.

By the means of **PConLANResponsePort** the port is defined, to which the answer is sent.  
Default is 4558.

Currently (2\_4\_031) the receiver IP-address has to be defined with **PConLANResponseIP**. In the versions from this time on the processor, from which the remote command has been sent, is used.  
Default is 192.168.0.1

By the means of **LocPConLANResponsePort** , the port on the camera is specified, from which the answer has been sent. Default is 5953

## ***Using the implementation of the protocol in the SmartBuilderControl dll***

Use the SmartBuilderControl.dll to send any program control request to VizCam cameras. The function ECSEndProgramControlRequest can be used send the requests to any camera with given IP and port number..

`long STDCALL ECSEndProgramControlRequest( char *sIPAddress, long IPortNumber, char *sRequest)`

Parameterdescription:

**sIPAddress**

is a NULL-terminated string that contains the IP-Address for example "192.168.0.65"

**IPortNumber**

is a long value containing the PConLAN port number. for example 5952

**sRequest**

is the NULL-terminated string contains the PConLAN request message for example  
"#001standard.ckp#"

Return value:

Return value	Description
0	Answer camera: OK
-1	No answer camera ( Timeout)
-2	Answer camera: NOK
-3	Answer camera: IGNORED