

Convision Systems GmbH

Security Solutions der innovativen Art

Programmers Manual

```
<html>
<HEAD>
<TITLE>Convision Systems GmbH</TITLE>
<link rel=stylesheet type="text/css" href="Files\text.css">
<script language="JavaScript">
<!--
var Seconds = 5;

function JS_Page1() {
  parent.main.location.href = "http://83.236.156.42/view/activex.htm?picsize=3"
function JS_Page2() {
  parent.main.location.href = "http://83.236.156.42:8080/view/activex.htm?"
}
function JS_Page3() {
  parent.main.location.href = "http://192.168.0.165/view/activex.htm?picsize=3"
function JS_Page4() {
  parent.main.location.href = "http://192.168.0.164/view/activex.htm?picsize=3"
}

var Mode = 1;
var Counter = 1;
function JS_Random()
{
  if(Counter == 4) {JS_Page4(); Counter = 1;}
  if(Counter == 3) {JS_Page3(); Counter = 4;}
  if(Counter == 2) {JS_Page2(); Counter = 3;}
  if(Counter == 1) {JS_Page1(); Counter = 2;}
  if(Mode == 2) window.setTimeout('JS_Random()', Seconds * 1000);
}
}
-->
</script>
```

INTERNET VIDEO SOLUTIONS

1.	Transmission of JPEG Images	2
	Example of the transmission of a JPEG image:	2
	Examples of the transmission of a JPEG image in different sizes for integration into a HTML page:	2
2.	Transmission of Image Sequences (Motion-JPEG-Streams).....	2
	Example Motion-JPEG Server Push.....	2
3.	Example – ActiveX-Control	3
4.	Example of a viewing application with HTML.....	3
	Definition of a frame set:.....	3
	Multiview HTML page with 4 devices.....	4
5.	HTML- Programming of Pan / Tilt / Zoom - Cameras.....	5
	URL Functions	5
	Examples	7
	Example for Programming Pan/Tilt/Zoom Cameras with "move" and "zoom" function for the transmission of single images	8
	Example for Programming Pan/Tilt/Zoom Cameras for the transmission of image sequence.....	9
6.	Programming the Outputs.....	10
7.	Status der Eingänge abfragen	10
8.	URL- calls to force actions	11
9.	ConvisionVideoStream ActiveX Control.....	11
	1. Interface Description	11
	1.1. Properties.....	11
	1.1.1. Atc.....	11
	1.1.2. AutoStart	12
	1.1.3. IpAddress.....	12
	1.1.4. Port.....	12
	1.1.5. MouseClickUrl	12
	1.1.6. ProxyIpAddress.....	12
	1.1.7. ProxyPort	12
	1.1.8. ShowCaption.....	13
	1.1.9. ShowTime	13
	1.1.10. Url.....	13
	1.1.11. UserName	13
	1.1.12. UserPassword.....	13
	1.1.13. DisplayMode	13
	1.1.14. MotionDisplayThreshold	14
	1.1.15. MotionAreaEditMode	14
	1.1.16. ShowMotionValue	14
	1.2. Methods	14
	1.2.1. StartStream.....	14
	1.2.2. StopStream	14
	1.2.3. ReadMotionArea	14
	1.2.4. WriteMotionArea	15
	1.2.5. ClearMotionArea	15
	1.2.6. FillMotionArea	15
	2. Examples	15
	2.1. HTML	15

1. Transmission of JPEG Images

The URL program for the transmission of an image is designated after the size of image. In addition parameters can be transmitted to obtain certain image properties. If several cameras are attached to the IP video server a parameter must be entered for the required camera (default is camera 1).

Image size		command
halfsize		halfsize.jpg
fullsize	352 x 288 Pixel	fullsize.jpg
hugesize	704 x 576 Pixel	hugesize.jpg

Example of the transmission of a JPEG image:

```
http://IP-Adresse/fullsize.jpg?camera=1
```

Examples of the transmission of a JPEG image in different sizes for integration into a HTML page:

```
<IMG SRC="http://IP-Adresse/halfsize.jpg?camera=2" border=0 WIDTH=192 HEIGHT=144>
<IMG SRC="http://IP-Adresse/fullsize.jpg?camera=2" border=0 WIDTH=384 HEIGHT=288>
<IMG SRC="http://IP-Adresse/hugesize.jpg?camera=2" border=0 WIDTH=768 HEIGHT=576>
```

The command URL is entered as image source in the HTML page.

2. Transmission of Image Sequences (Motion-JPEG-Streams)

Die Dateiendung nach der Bildgröße entscheidet darüber, welches Programm im Webserver gestartet wird:

Parameter	Definition
bildgröße.jpg	an individual jpg picture is retrieved
bildgröße.push	a push image stream started, functional only with Netscape Browser and IE Macintosh version and within an HTML file
fullsize_activex.htm	ActiveXControl

Example Motion-JPEG Server Push

```
<!-- Kommentar: Beispiel fuer Bildfolgen-->
```

```
<TABLE WIDTH=100%>
<TR>
<TD><IMG SRC="http://IP-Adresse/fullsize.push?camera=1" border=0></TD>
</TR>
</TABLE>
```

3. Example – ActiveX-Control

The following example shows the minimum set of parameters for HTML use. The streaming starts instantly without any scripting.

Please make sure that you have the file "ConvisionVideo.cab" in your HTML folder or already installed locally on your computer.

```
<object classid="clsid:99C4B3C5-166D-4B6B-8305-C59FB5AF5BF6"
codebase="/ConvisionVideo.cab#version=1,0,0,32" width="352" height="288" border="0" >
  <param name="Url" value="/stream.live">
  <param name="DisplayMode" value="1">
  <param name="AutoStart" value="1">
  <param name="ShowTime" value="1">
  <param name="ShowCaption" value="1">
  ActiveX is just supported by Microsoft Internet Explorer!
</object>
```

4. Example of a viewing application with HTML

Definition of a frame set:

```
<html>
<HEAD>
<TITLE>Convision Systems GmbH</TITLE>
<link rel="stylesheet" type="text/css" href="Files\text.css">
</HEAD>
<Frameset rows="50,*" border="0" framespacing="0" frameborder="NO" cols="*">
  <Frame src="Files\top.htm" name="topframe" scrolling=no frameborder=0 framespacing=0 marginwidth=0
marginheight=0 noresize scrolling="no">
  <Frameset cols="150,*" border="0" framespacing="0" frameborder="NO" rows="*">
    <Frame src="navi.htm" name="navi" frameborder=0 framespacing=0 marginwidth=0 marginheight=0 noresize
scrolling="no">
    <Frame src="Files\leer.htm" name="main" frameborder=0 framespacing=0 marginwidth=0 marginheight=0>
  </frameset>
</frameset>
</frameset>
<BODY>
<noframes>
Your Browser does not support Frames! Please Update!
</noframes>
</BODY>
</HTML>
```

Multiview HTML page with 4 devices

```

<html>
<HEAD>
<TITLE>Convision Systems GmbH</TITLE>
<link rel=stylesheet type="text/css" href="Files\text.css">
<script language="JavaScript">
<!--
var Seconds = 5;

function JS_Page1() {
  parent.main.location.href =
"http://83.236.156.42/view/activex.htm?picsize=3&displaymode=65551&camera=1";
}
function JS_Page2() {
  parent.main.location.href =
"http://83.236.156.42:8080/view/activex.htm?picsize=3&displaymode=65551&camera=1";
}
function JS_Page3() {
  parent.main.location.href =
"http://192.168.0.165/view/activex.htm?picsize=3&displaymode=65551&camera=1";
}
function JS_Page4() {
  parent.main.location.href =
"http://192.168.0.164/view/activex.htm?picsize=3&displaymode=65551&camera=1";
}

var Mode = 1;
var Counter = 1;
function JS_Random()
{
  if(Counter == 4) {JS_Page4(); Counter = 1;}
  if(Counter == 3) {JS_Page3(); Counter = 4;}
  if(Counter == 2) {JS_Page2(); Counter = 3;}
  if(Counter == 1) {JS_Page1(); Counter = 2;}
  if(Mode == 2) window.setTimeout('JS_Random()', Seconds * 1000);
}

!-->
</script>
</HEAD>
<body class="main2" onload="JS_Page1()">
<table border="0" cellpadding="2" cellspacing="2" height="20%">
<tr><td><a href="javascript:Mode=1;JS_Page1()" class="type1">System 1</a></td>
<tr><td><a href="javascript:Mode=1;JS_Page2()" class="type1">System 2</a></td>
<tr><td><a href="javascript:Mode=1;JS_Page3()" class="type1">System 3</a></td>
<tr><td><a href="javascript:Mode=1;JS_Page4()" class="type1">System 4</a></td>
</table>
<table border="0" cellpadding="2" cellspacing="2" height="20%">
<tr><td><a href="javascript:Mode=2; JS_Random()" class="type1" align="left">Random - 5s</a></td>
</table>
</BODY>
</HTML>

```

5. HTML- Programming of Pan / Tilt / Zoom – Cameras

You can attach different controllable cameras and dome cameras to the Convision IP video server. The cameras are controlled with the following URL parameters for the transmission of a image or an image sequence. The cameras have different minimum and maximum values or do not support all functions depending on design and manufacturing conditions.

URL Functions

The parameters can be utilized with the functions "fullsize.jpg" or "pantilt". In the first case the camera is controlled and an image transmitted, in the other case only the camera is controlled, e.g. during the transmission of a image sequence, which is separately called with "fullsize.push".

Parameter	Definition	Scope	Remarks
pan=<value>	moves the camera horizontal to the specified value in degrees in the indicated direction	- 180 ... +180	The max. area is dependent in each case on the assigned camera
tilt=<value>	moves the camera vertically to the specified value in degrees in the indicated direction	- 180 ... +180	The max. area is dependent in each case on the assigned camera
zoom=<value>	stops the indicated zoom shot factor, whereby the value 1 is for the fully complete zoom status	1 ... 999	The max. area is dependent in each case on the assigned camera
move=<value>	moves the camera in the indicated direction	up / down / right / left / home	The angle depends on the assigned camera
ppan=<value>	moves the camera horizontal to the proportional value, related to the zero position and the max. direction	-100...+100	Max. horizontal direction is equal to 100%
ptilt=<value>	moves the camera vertically to the proportional value, related to the zero position and the max. direction	-100...+100	Max. vertical direction is equal to 100%
pzoom=<value>	zoom shot of the camera to the proportional value, related to the zero position and that max. zoom shot	0...+100	Max. zoom is equal to 100%
rpan=<value>	moves the camera horizontal relative to the current position around the specified value in degrees in the indicated direction	- 180 ... +180	The max. area is dependent in each case on the assigned camera
rtilt=<value>	moves the camera vertically relative to the current position around the specified value in degrees in the indicated direction	- 180 ... +180	The max. area is dependent in each case on the assigned camera
zoomrel=<value>	reduces or increases the zoom factor	wide, tele	Changes the value amount, dependent on the assigned camera.
focusrel=<value>	modifies the focus of a camera without autofocus for a close-distant or remote-distant range	far, near	
irisrel=<value>	adjusts the light beam into the iris	open, close	
focusmode=<value>	Changes the operating mode of the focus	auto, manual	
setpreset=<value>	stores camera position and shows it again	1 ... 6	
showpreset=<value>			

The movement commands pan and tilt, rpan and rtilt as well as ppan and ptilt should not be mixed in the URL, e.g. ppan together only with ptilt, but not with tilt or rtilt.

The following table shows the supported cameras from the Convision IP video servers with their function range and the possible values:

Camera	RS485	pan	tilt	zoom	move	home
Canon VC-C4	-	-100 ... 100	-30 ... 30	0 ... 1070	yes *1	yes
Canon VC-C4R	-	-170 ... 170	-90 ... 10	0 ... 1070	yes *1	yes
Dennard Type 2050 (>= V4.0 Build 1883)	yesH	-180 ... 180	-90 ... 0	0 ... 100	yes *1	yes
Ernitec BDR510	-	-	-	-	yes	-
Ernitec BDR550	-	-	-	-	yes	-
Hitron-2404 (Fastrax II-9600 bps)	yesH	-180 ... 180	-90 ... 0	0 ... 100	yes *1	yes
Honeywell HSD-251P	yesH	-	-	-	yes	-
JVC TK-C675 (JCBP-S)	yesH	-180 ... 180	0 ... 166	0 ... 166	yes *1	yes
Lilin Dome PIH-7000 Series	yesH	-	-	-	yes	-
Panasonic WV-CS850 Series	yesH	-	-	-	yes	-
Pelco Spectra II	yesH	-	-	-	yes	-
Pelco Spectra II und III (D-Protocol)	yesH	-	-	-	yes	-
Philips G3 AutoDome	-	-	-	-	yes	-
SAE Dragon (4800 bps)	yesH	-	-	-	yes	-
Samsung SCC-641 (Vicon-Protocol)	yesH	-	-	-	yes	-
Santec VDC-300ID/VDC-400ID (9600 bps)	yesH	-180 ... 180	-90 ... 0	0 ... 100	yes *1	yes
Sensormatic SpeedDome Optima	yesF	-180 ... 180	-90 ... 20	0 ... 4096	yes *1	yes
Sensormatic SpeedDome Ultra	yesF	-180 ... 180	-90 ... 22	0 ... 100	yes *1	yes
Sony EVI-D30/D31 (Convision CC2)	-	-100 ... 100	-25 ... 25	0 ... 1023	yes *1	yes
VCL Microsphere Orbiter	yesH	-180 ... 180	-90 ... 0	0 ... 1023	yes *1	yes
VCL Microsphere OrbiterGold	yesH	-180 ... 180	-90 ... 0	0 ... 1023	yes *1	yes
Vicon Surveyor 2000-16 (19200 bps)	yesH	-180 ... 180	-90 ... 0	0 ... 128	yes *1	yes
Vicon Surveyor 2000-22 (19200 bps)	yesH	-180 ... 180	-90 ... 0	0 ... 176	yes *1	yes
Vicon Surveyor 99 (19200 bps)	yesH	-	-	-	yes	-

RS485 – H = Halbduplex, F = Fullduplex

Camera	ppan ptilt	pzoom	rpan rtilt	zoomrel	focusrel	irisrel	preset
Canon VC-C4	yes	yes	yes	-	-	-	yes
Canon VC-C4R	yes	yes	yes	-	-	-	yes
Dennard Type 2050 (>= V4.0 Build 1883)	yes	yes	yes	-	-	-	yes
Ernitec BDR510	-	-	-	wide, tele	far, near	open, close	-
Ernitec BDR550	-	-	-	wide, tele	far, near	open, close	yes
Hitron-2404 (Fastrax II-9600 bps)	yes	yes	yes	-	-	-	yes
Honeywell HSD-251P	-	-	-	wide, tele	far, near	open, close	yes
JVC TK-C675 (JCBP-S)	yes	yes	yes	-	-	-	yes
Lilin Dome PIH-7000 Series	-	-	-	wide, tele	far, near	open, close	yes
Panasonic WV-CS850 Series	-	-	-	wide, tele	far, near	open, close	yes
Pelco Spectra II	-	-	-	wide, tele	far, near	open, close	yes
Pelco Spectra II und III (D-Protocol)	-	-	-	wide, tele	far, near	open, close	yes
Philips G3 AutoDome	-	-	-	wide, tele	far, near	open, close	yes
SAE Dragon (4800 bps)	-	-	-	wide, tele	far, near	open, close	yes
Samsung SCC-641 (Vicon-Protocol)	-	-	-	wide, tele	far, near	open, close	yes
Santec VDC-300ID/VDC-400ID (9600 bps)	yes	yes	yes	-	-	-	yes
Sensormatic SpeedDome Optima	yes	yes	yes	-	-	-	yes
Sensormatic SpeedDome Ultra	yes	yes	yes	-	-	-	yes
Sony EVI-D30/D31 (Convision CC2)	yes	yes	yes	-	-	-	yes
VCL Microsphere Orbiter	yes	yes	yes	-	-	-	yes
VCL Microsphere OrbiterGold	yes	yes	yes	-	-	-	yes
Vicon Surveyor 2000-16 (19200 bps)	yes	yes	yes	-	-	-	yes
Vicon Surveyor 2000-22 (19200 bps)	yes	yes	yes	-	-	-	yes
Vicon Surveyor 99 (19200 bps)	-	-	-	wide, tele	far, near	open, close	yes

*1 – Camera angle moves exactly "actual apex angle/10 "

Examples

[HTTP://IP-Adresse/fullsize.jpg?camera=1&pzoom=50](http://IP-Adresse/fullsize.jpg?camera=1&pzoom=50)

This command adjusts camera 1 to Zoom=50% and then transmits an image

[HTTP://IP-Adresse/fullsize.jpg?camera=1&pan=30&tilt=-20](http://IP-Adresse/fullsize.jpg?camera=1&pan=30&tilt=-20)

This command moves the camera from the " home" position, horizontal 30° to the right and turns 20° downward and then transmits an image.

Example for Programming Pan/Tilt/Zoom Cameras with "move" and "zoom" function for the transmission of single images

```

<HTML>
<HEAD>
<TITLE>cameracontrol-JPEG</TITLE>

<SCRIPT LANGUAGE="JavaScript">
<!--
var nZoom = 200;
var IPAddress = "192.168.0.1"; // Here place the IP address of your IP video servers.

function JS_Move(x){
  var nTime;
  nTime = Math.random(); // sets a random number for the newURL value

  if (x == 1) document.image.src="http://" + IPAddress + "/fullsize.jpg?camera=1&move=up&newURL=" + nTime;
  if (x == 2) {
    nZoom = nZoom + 50;
    if (nZoom > 1023) nZoom = 1023; // magnification of the zoom factor
    document.image.src="http://" + IPAddress + "/fullsize.jpg?camera=1&zoom=" + nZoom + "&newURL=" + nTime;
  }
  if (x == 3) document.image.src="http://" + IPAddress + "/fullsize.jpg?camera=1&move=left&newURL=" + nTime;
  if (x == 4) document.image.src="http://" + IPAddress + "/fullsize.jpg?camera=1&move=home&newURL=" + nTime;
  if (x == 5) document.image.src="http://" + IPAddress + "/fullsize.jpg?camera=1&move=right&newURL=" + nTime;
  if (x == 6) {
    nZoom = nZoom - 50;
    if (nZoom < 1) nZoom = 1; // reduction of the zoom factor
    document.image.src="http://" + IPAddress + "/fullsize.jpg?camera=1&zoom=" + nZoom + "&newURL=" + nTime;
  }
  if (x == 7) document.image.src="http://" + IPAddress + "/fullsize.jpg?camera=1&move=down&newURL=" +
    nTime;
}
// -->
</SCRIPT>

</HEAD>
<BODY>
<table>
<tr><!-- PanTiltControl-Button (with map "control") -->
  <td></td>
  <td></td><!-- Calling Image sequence -->
</tr>
</table>
<map name="control">
  <area shape="rect" coords="33,2,66,28" HREF="javascript:JS_Move(1)">
  <area shape="rect" coords="66,2,95,27" HREF="javascript:JS_Move(2)">
  <area shape="rect" coords="1,29,32,54" HREF="javascript:JS_Move(3)">
  <area shape="rect" coords="34,28,65,54" HREF="javascript:JS_Move(4)">
  <area shape="rect" coords="66,28,97,55" HREF="javascript:JS_Move(5)">
  <area shape="rect" coords="1,57,33,83" HREF="javascript:JS_Move(6)">
  <area shape="rect" coords="33,55,67,82" HREF="javascript:JS_Move(7)">
</map>
</BODY>
</HTML>

```

This example HTML page contains a table with two columns, on the left the controls, on the right the image. The Javascript function JS_Move(x) is called through one click on the occupied fields of the control (map) Depending upon the transferred parameters the camera is moved or zoomed in a direction then an image is transmitted.

Example for programming Pan/Tilt/Zoom cameras for the transmission of image sequence

```
<HTML>
<HEAD>
<TITLE>Camera control Motion-JPEG</TITLE>
<meta http-equiv="cache-control" content="no-cache">
<meta http-equiv="pragma" content="no-cache">

</HEAD>
<BODY >

<table>
<tr>
<td>
<table>
<tr>
<td>
<!-- PanTiltControl-Button (with map "position") -->
<td></td>
</tr>
<tr>
<td>
<!-- Zoombuttons -->
<td><a href="http://IP-address/pantilt?pzoom=100&camera=1" target="vempty">
</a></td>
<td><a href="http://IP-address/pantilt?pzoom=50&camera=1" target="vempty">
</a></td>
<td><a href="http://IP-address/pantilt?pzoom=0&camera=1" target="vempty">
</a></td>
</tr>
<tr>
<td>
<!-- Preset display (one Position) -->
<td><a href="http://IP-address/pantilt?camera=1&showpreset=1" target="vempty">
</a></td>
</tr>
<tr>
<td>
<!-- Preset setting -->
<td><a href="http://IP-address/pantilt?camera=1&setpreset=1" target="vempty">
</a></td>
</tr>
</table>
</td>
<td>
<!-- Calling Image sequence -->
<td></td>
</tr>
</table>
```

```
<map name="position">
<area shape="rect" coords="29,57,52,77" href="http://IP-address/pantilt?camera=1&move=down" target="vcmpty">
<area shape="rect" coords=" 4,30,25,52" href="http://IP-address/pantilt? camera=1&move=left" target="vcmpty">
<area shape="rect" coords="30,30,52,52" href="http://IP-address/pantilt?camera=1&move=home"target="vcmpty">
<area shape="rect" coords="55,30,78,51" href="http://IP-address/pantilt?camera=1&move=right" target="vcmpty">
<area shape="rect" coords="29, 5,52,26" href="http://IP-address/pantilt? camera=1&move=up" target="vcmpty">
</map>

</BODY>
</HTML>
```

The example describes a HTML page which is part of a Frameset. On this page there is a double column table, again on the left is a table with the controls and on the right it contains the display of the image sequence. In the example the .gif image buttons, which utilize the control, are stored in a subdirectory (< img src="buttons/x.gif " >). They are here, up to the item "control.gif ", to which the map "position" refers directly, not described near, the creation is left up to your imagination. By clicking on the buttons the appropriate parameters and the PanTilt in the IP video server is utilized .

All commands for camera control are sent to an empty Frame (target="vcmpty "). This Frame is in the same Frameset as the Frame which contain the image display and control but it has no contents. Thus the command is sent to the IP video server, however it causes no reloading of the frames with the image sequence and the stream can run continuously.

6. Programming the outputs

The outputs can be switched on and off via the "dioswitch" function.

These can be switched through the following URL:

```
http://IP-Adresse/output/dioswitch?rl1=on
http://IP-Adresse/output/dioswitch?rl1=off
```

Parameter	Definition	Scope	Remarks
rl1, ..., rl4	schaltet den Relais-Ausgang, rl1: 1. Relaisausgang	on, off	on = ein off = aus

7. Get the status of the inputs

Through the following URL you query the state of the outputs:

```
http://IP-Adresse/output/dioswitch?getstate=1
```

8. URL- calls to force actions

With the following url commands you can force dedicated actions. The example starts or stops an action. You can control up to 8 actions.

On: <http://ipaddress'/config/urlevent?type=1&nr='1-8'>
 On: <http://ipaddress'/config/urlevent?type=0&nr='1-8'>

Action	
Recording start	on
Recording stop	on (stops recording)
Recording start/stop	on/off start and Stop of recording
Speed up recording for event period	on
Speed up recording period	on
Protect hard disk event	on
E-Mail	on
Ftp Upload	on
Time Client Update	on/off
Input	on/off
Output	on/off
Video Out	on

9. ConvisionVideoStream ActiveX Control

The ConvisionVideoStream ActiveX Control is a component for displaying video streams from the Convision V4000/V800/V1600 and V400+/V600+ series of video surveillance servers.

1. Interface Description

1.1. Properties

1.1.1. Atc

Type: *STRING*
 Default: ""

Atc is the authorization string which is being needed to get a stream from protected video servers. The string consists of a *BASE64* coded string of the following type: *name:password*.

Example:

Name: user
 Password: topsecret
 String to be *BASE64* encoded: user:topsecret

Resulting pseudo code (check it here: <http://makcoder.sourceforge.net/demo/base64.php>):

```
Atc = "dXNlcjpw0b3BzZWNYZXQ="
```

Also, the parameters **UserName** and **UserPassword** provide the same functionality.

1.1.2. AutoStart

Type: *BOOL*
Default: *FALSE*

If **AutoStart** is *TRUE* the streaming will be started immediately after the control is visible.

1.1.3. IpAddress

Type: *STRING*
Default: *""*

IpAddress is the IP address or the name of the video server from which the video will be streamed.

1.1.4. Port

Type: *LONG*
Default: *80 (HTTP standard)*

Port is the socket port of the video server to get the stream from.

1.1.5. MouseClickUrl

Type: *STRING*
Default: *""*

MouseClickUrl is the controlling command for a *PTZ* camera to move it to the desired position when the user clicked the left mouse inside the video area of the control.

1.1.6. ProxyIpAddress

Type: *LONG*
Default: *""*

The **ProxyIpAddress** is needed if the video server is only being reached via a proxy server. If there's no need of a proxy server you have to use the default value.

1.1.7. ProxyPort

Type: *LONG*
Default: *0*

The **ProxyPort** is needed if the video server is only being reached via a proxy server. This parameter is only used if the **ProxyIpAddress** is being set to a proper value.



1.1.8. ShowCaption

Type: *BOOL*
Default: *FALSE*

If **ShowCaption** is *TRUE* the text in property **Caption** is being displayed in the video stream.

1.1.9. ShowTime

Type: *BOOL*
Default: *FALSE*

If **ShowTime** is *TRUE* the time stamp of the current frame will be displayed in the stream (if any).

1.1.10. Url

Type: *STRING*
Default: ""

Url is the path name of the stream from the video server you want to display. Actually this parameter should always be `"/stream.live"`.

1.1.11. UserName

Type: *STRING*
Default: ""

UserName is the name of the user which is needed to get authenticated with the video web server. This parameter is used in combination with the **UserPassword** parameter.

1.1.12. UserPassword

Type: *STRING*
Default: ""

UserPassword is the password of the user which is needed to get authenticated with the video web server. This parameter is used in combination with the **UserName** parameter.

1.1.13. DisplayMode

Type: *LONG*
Default: *0x00000000*

The display mode of the area and the selection of the cameras to be streamed are being set by the parameter **DisplayMode**. The lower 16 bits define the cameras to be streamed whereas the upper 16 bits define the display mode.

Examples for a single display of camera 4:

```
DisplayMode = 0x00000008
```

Example for a quad display of cameras 5, 6, 7 and 8:

```
DisplayMode = 0x000100F0
```

1.1.14. MotionDisplayThreshold

Type: *LONG*

Default: *0xFFFFFFFF*

If motion is being detected in the current frame and the resulting motion intensity is greater than the value specified by **MotionDisplayThreshold** a red border will be displayed in the current frame. So the default value of *0xFFFFFFFF* suppresses the display of the border since motion values will never be that high.

1.1.15. MotionAreaEditMode

Type: *BOOL*

Default: *FALSE*

If **MotionAreaEditMode** is *TRUE* the control will display an editable grid of active motion areas for the specified camera. Note: this grid is only available in a single display mode.

See also the related methods **ReadMotionArea** and **WriteMotionArea**.

1.1.16. ShowMotionValue

Type: *BOOL*

Default: *FALSE*

The current motion value will be displayed if **ShowMotionValue** is *TRUE* in a bar like graph.

The displayed value, however, is the result of the following function:

```
display_value = min(ld(motion_value), 16)
```

1.2. Methods

The following methods listed here will return *TRUE* on success else *FALSE*.

1.2.1. StartStream

StartStream starts the streaming from the video server. See also the property **AutoStart**.

1.2.2. StopStream

StopStream stops the current streaming from the video server.

1.2.3. ReadMotionArea

ReadMotionArea reads the current active motion area used for motion detection. This method should be called when the control is the "motion area edit mode". See also the property **MotionAreaEditMode**.

1.2.4. WriteMotionArea

WriteMotionArea writes the user defined motion area back to the server. This method should be called when the control is the "motion area edit mode". See also the property **MotionAreaEditMode**.

1.2.5. ClearMotionArea

ClearMotionArea clears the user defined motion area but it doesn't affect the current active motion area inside the server. This method should be called when the control is the "motion area edit mode". See also the property **MotionAreaEditMode**.

1.2.6. FillMotionArea

FillMotionArea fills the whole user defined motion area but it doesn't affect the current active motion area inside the server. This method should be called when the control is the "motion area edit mode". See also the property **MotionAreaEditMode**.

2. Examples

2.1. HTML

The following example code snippet shows the minimum required parameters to be set inside a HTML page. The streaming from camera 1 will be started immediately so there's no need of a script to do so.

Please note that you need to have a copy of ConvisionVideo.cab in the same folder as the HTML page if the control has not been installed already!

```
<object classid="clsid:99C4B3C5-166D-4B6B-8305-C59FB5AF5BF6"  
codebase="ConvisionVideo.cab#version=1,0,0,33" border="4" width="704" height="576"  
name="Controll">  
<param name="AutoStart" value="1">  
<param name="DisplayMode" value="1">  
<param name="IpAddress" value="convision.dyndns.org">  
<param name="Url" value="/stream.live">  
</object>
```