Introduction

IVC is pleased to furnish an ActiveX® software component for use with IVC's line of controllable video products. If you create integrated end-user solution software and you need to integrate video monitoring or surveillance, this software component is for you. We have designed it for use in a variety of solution development environments on the Microsoft® Windows® platform, including:

- Web applications based on Microsoft Internet Explorer (version 5 and higher)
- Microsoft Visual Basic (version 6 and higher)
- Microsoft Visual C++ (version 6 and higher)
- Wonderware® InTouch WindowMaker and WindowViewer
- Iconics® Genesis-32 GraphWorx32 (version 6 and higher) and related software

If you develop customized solution software and you wish to build live and controllable video windows into your solutions, IVC's ActiveX component makes it easy.

IVC's ActiveX component must be installed on each computer used for solution development, and each computer used for solution deployment. It is furnished to developers and end-users without charge for use with other IVC products. If you are a solution developer for IVC products you may freely redistribute the component to any and all end-users of IVC products.

This document describes the IVC ActiveX component. For best results you should have experience with one of the Windows-based solution development environments which incorporate Active X.

Features and Functions

The IVC ActiveX component offers a simple but powerful set of features to IVC solution developers and users.

- Live Streaming Video: By placing the component into a software solution you may build video into that solution. For example, an alarm screen in an industrial control solution may show details of the alarm and a video of the device issuing the alarm.
- Controllable Video Streaming: The solution developer may enable and disable video streaming, and choose an appropriate frame rate, for each application.
- Steerable Video: IVC's steerable video allows the end-user to point to objects in the video and take a closer look. The IVC ActiveX component supports point-and-click to look and zooming. The point-and-click to look is configurable.

- Full ActiveX standards compatibility: The component meets Microsoft requirements for a safe, scriptable, and data-bound ActiveX control.
- Configurability: A solution developer may configure the component using a built-in dialog box, and also programmatically via simple scripting interfaces.
- Simplicity for end-users: End-users of custom solutions have as little or as much control over the video system as the solution-developer wishes.
- Small size and simplicity: The IVC ActiveX component consists of a single software file approximately 150K in size.
- Efficiency and scalability: The IVC ActiveX component is suitable for creating video display applications showing the images from dozens of video cameras on a single computer monitor.
- Compatibility with the IVC product line: The IVC ActiveX component integrates seamlessly over intranets, extranets, and the public Internet with IVC's relay server products.

Distribution and Installation

IVC's ActiveX component, when it is installed on a Windows computer, consists of a single file – a self-contained dynamic-link-library for Windows – about 150k in size. It is very simple to use for development and deployment: the same exact software is used for development and deployment.

To install the component, simply download and run the installer. It will place the component on your computer and make it available for use.

Or, you can visit the web page http://www.ivcco.com/activex/ using Internet Explorer. This web page installs the component automatically.

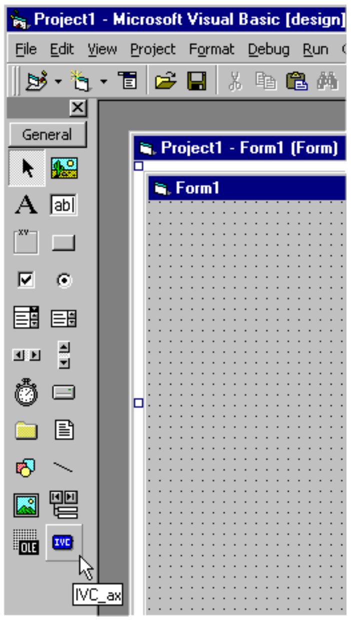
Developing a Live Controllable Video Application in Visual Basic

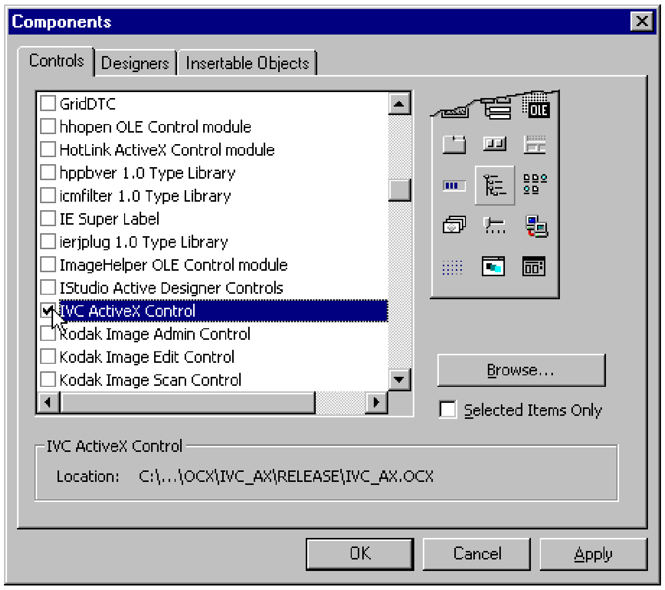
This is a simple example of the use of IVC's ActiveX component.

To build a simple Visual Basic controllable video solution, use the following steps.

Ensure the IVC ActiveX Component is installed on your computer.

1. Start Visual Basic and choose a new "Standard EXE" project.
2. Add the IVC Component to your project. Choose Components… from the Project menu.
3. In the Components dialog box, locate the IVC ActiveX Control, select it, and press the OK button. The IVC icon appears at the bottom of the Visual Basic designer palette.

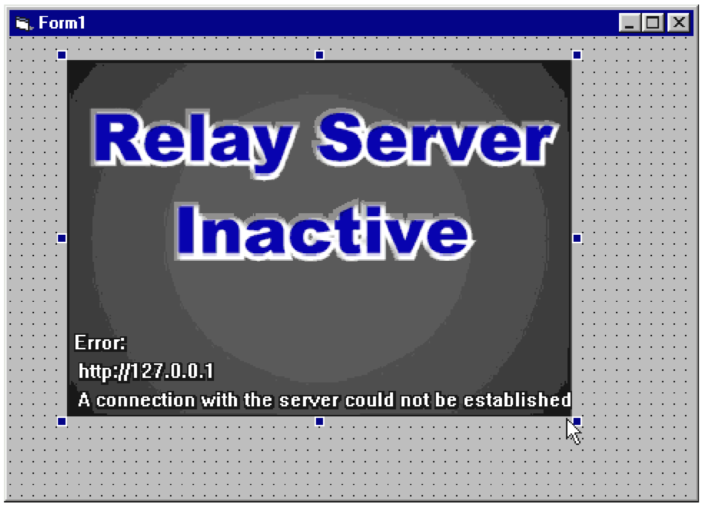

Project1 - Microsoft Visual Basic [design]
File Edit View Project Format Debug Run
General
Project1 - Form1 (Form)
Form1
IVC
IVC_ax



GridDTC
hhopen OLE Control module
HotLink ActiveX Control module
hppbver 1.0 Type Library
icmfilter 1.0 Type Library
IE Super Label
ierjplug 1.0 Type Library
ImageHelper OLE Control module
IStudio Active Designer Controls
IVC ActiveX Control
Kodak Image Admin Control
Kodak Image Edit Control
Kodak Image Scan Control

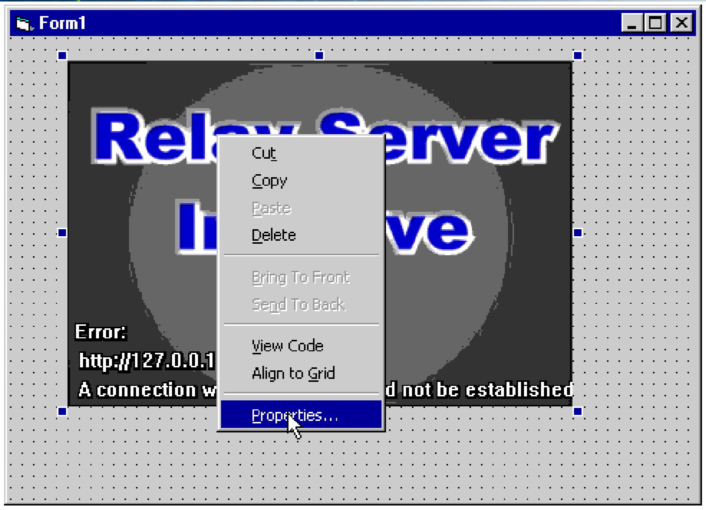
IVC ActiveX Control
Location: C:\...\OCX\IVC_AX\RELEASE\IVC_AX.OCX

4. Choose the IVC icon, then draw a control on the Form in your Visual Basic window. The video window sizes itself to an appropriate size, and displays itself on your screen. If you are running an IVC Relay Server on your development computer, a video image will appear. However, ordinarily you will see an error frame indicating that no Relay Server is responding. This is normal.

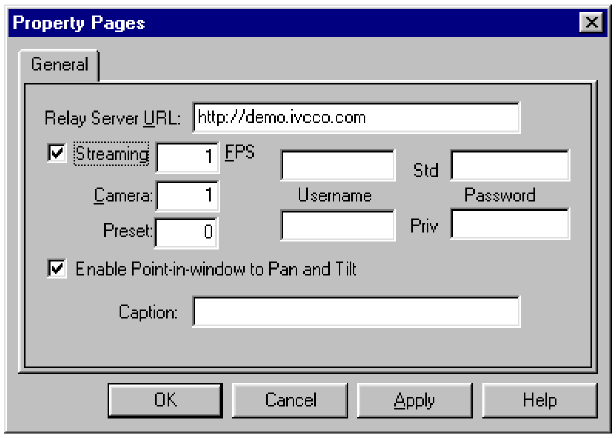


Error:
http://127.0.0.1
A connection with the server could not be established

5. Press the right mouse button on top of the video window (anywhere in the frame) and choose the Properties item.



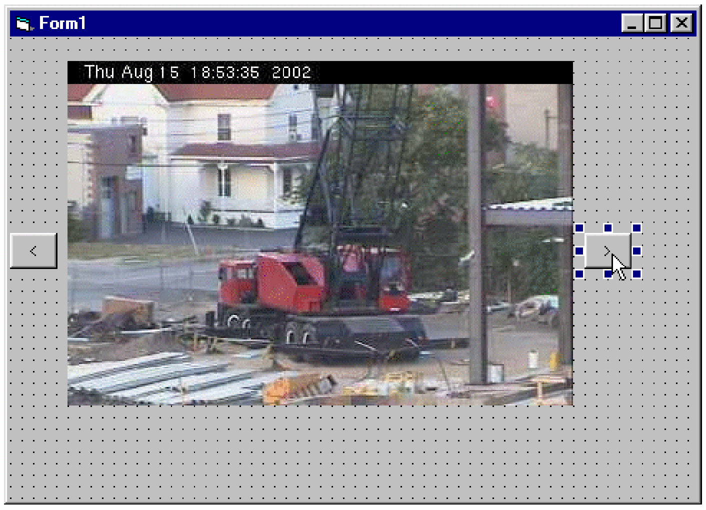
6. Enter the URL of a valid IVC Relay Server into the Properties dialog box and press OK. You may also choose an appropriate streaming frame rate, camera number, and preset if you wish. Use a Relay Server in your organization, or if you wish you may use http://demo.ivcco.com for testing purposes.



Relay Server URL: http://demo.ivcco.com
[✓] Streaming 1 FPS
Camera: 1
Preset: 0
Std Username Password
Priv
[✓] Enable Point-in-window to Pan and Tilt
Caption:

7. At this point the error frame should disappear and a video picture should appear.

8. Add Visual Basic buttons to zoom in and out. Choose the button icon from the Visual Basic palette, place a button to the left of the video frame, and type a left angle bracket <. Repeat for a button to the right. The buttons should appear on the screen next to the video frame.



9. Next hook up the buttons to the IVC ActiveX video component, using two lines of Basic. To do this double-click on the left button and type the following line of code:

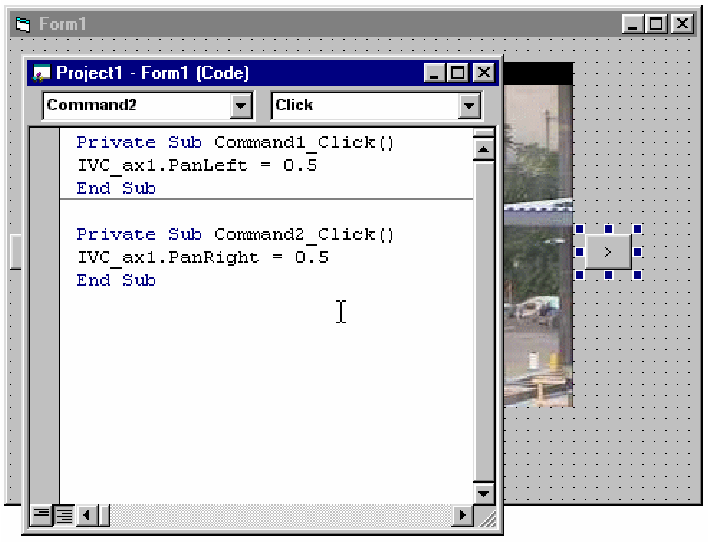
```
ivc_ax1.PanLeft = 0.5
```

Repeat this step for the right button, and type the following line of code:

```
ivc_ax1.PanRight = 0.5
```

Each of these lines of code indicates that whenever each button is pressed, the video object (named “ivc_ax1”) should pan half the visible field.

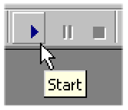
10. When you have finished typing these lines Visual Basic's code display should look something like this.

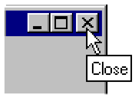


```
Private Sub Command1_Click()
IVC_ax1.PanLeft = 0.5
End Sub

Private Sub Command2_Click()
IVC_ax1.PanRight = 0.5
End Sub
```

11. You now have a ready-to-run custom video solution (a very simple one to be sure). Run it: Press the Start button in the Visual Basic toolbar. Try clicking the buttons – the image pans left and right. Try clicking in the video frame – the image centers where you click. When you're done trying out your new video application, click the X icon at the upper right of your window to close it and return to Visual Basic.





Properties

When you decide to develop a more sophisticated video application, you'll need to know the names and functions of the IVC ActiveX Component Properties. Each of the properties allows you to control some aspect of the component from your program. The table describes each of the properties. It explains each property, and tells whether you can set it from your Integrated Development Environment (IDE), from the property sheet, and at runtime.

Property Name	Meaning when set	Meaning when queried	Type of data	Settable
Active	True (1) causes the video frame to be displayed and	True (1) means video is being received, and False (0) means it is	True or False (1 or 0)	From IDE and at runtime

Property Name	**Meaning when set**	**Meaning when queried**	**Type of data**	**Settable**
	updated. False (0) displays a gray background and shuts off video reception	shut off.		
Camera	Chooses a particular camera (video feed) on the relay server.	No reliable meaning.	Small number from 1 up	From property sheet, from IDE, and at runtime
FramesPerSecond (deprecated)	Chooses a rate for displaying frames.	Indicates the chosen frame rate.	Number between 1 and 30	From property sheet, from IDE, and at runtime
FramesPerSecond2	Chooses a rate for displaying frames.	Indicates the chosen frame rate.	Number between 0.01 and 30.0	From property sheet, from IDE, and at runtime
Image		The JPEG data for the image currently being displayed	String, potentially very long, with embedded null characters.	Not settable
PanLeft	Causes the camera to pan to the left by the chosen amount. Setting this to 0.5, for example, makes the camera pan to the left by half a	No reliable meaning.	Decimal number, for example 0.5 or 1.5.	At runtime only

Property Name	Meaning when set	Meaning when queried	Type of data	Settable
	video field.			
PanRight	Causes the camera to pan to the right by the chosen amount.	No reliable meaning.	Decimal number	At runtime only
TiltUp	Causes the camera to tilt up by the chosen amount.	No reliable meaning.	Decimal number	At runtime only
TiltDown	Causes the camera to tilt down the by the chosen amount.	No reliable meaning.	Decimal number	At runtime only
Timeout	The number of seconds to wait before giving up on a network connection request	Indicates the timeout setting	Number of seconds	From IDE and at runtime
SocketReadWriteTimeout	The number of seconds to wait before giving up on a network read/write request	Indicates the timeout setting	Number of seconds	From IDE and at runtime
Preset	Chooses a particular preset position	No reliable meaning	Small Number from 1 up	From property sheet, from IDE, and at runtime
RelayServerURL	Specifies the network address of	Indicates the chosen Relay Server's	Text, for example, http://demo.ivcco	From property sheet, from

Property Name	Meaning when set	Meaning when queried	Type of data	Settable
	the Relay Server to display	network address	.com	IDE, and at runtime
StreamVideo	True (1) causes video to be streamed. False (0) displays a still image, updated whenever Active is set to true or any other Property is set.	Indicates whether streaming video is being displayed.	True or False	From property sheet, from IDE, and at runtime
EnablePoint	True(1) enables the user to point to something in the window and click it to cause the camera to pan and tilt to center it.	Indicates whether the point-in-window feature is activated.	True or False	From property sheet, from IDE, and at runtime
ZoomLevel	Setting this to 0 zooms out the camera one step. Setting it to 11 zooms in one step. Setting it to a value from 1-10 sets the zoom to the chosen	No reliable meaning.	Number between –1 and 11	From IDE and at runtime, to an absolute zoom value between 1 and 10

Property Name	Meaning when set	Meaning when queried	Type of data	Settable
	level.			
Caption	Specifies a caption to display at the lower left of the video window.	Indicates the caption.	Text. When %u appears in the string, the Relay Server URL is displayed. If %c appears, the camera number is displayed. %w and %h cause the display of the width and height of the video window.	From property sheet, from IDE, and at runtime
Exclusive (deprecated)	Setting this to True (1) enables exclusive access to the camera control function. Setting it to False (0) disables exclusive access. .	Returns the best guess of current state of Exclusive mode. The property automatically is reset to False after the period of inactivity mentioned in ExclusiveTimeout.	True or False	From IDE and at runtime. The function of this property depends upon PrivUsername and PrivPassword values being set correctly. Application software must set this property to True regularly to keep Exclusive mode in effect.
PrivUsername (deprecated)	Specifies the System Manager username	Indicates the username.	Text. Set to blank if the relay server is not configured to	From property sheet, from IDE, and at

Property Name	Meaning when set	Meaning when queried	Type of data	Settable
	for access to the Exclusive function.		require a System Manager password.	runtime
PrivPassword (deprecated)	Specifies the System Manager password for access to the Exclusive function.	Indicates the password.	Text. Set to blank if the relay server is not configured to require a System Manager password.	From property sheet, from IDE, and at runtime
ExclusiveTimeout (deprecated)	The number of seconds for which an Exclusive-access request should persist.	Number of seconds most recently set.	Seconds	From IDE, and at runtime
ExclusivePriority (deprecated)	The priority to use for Exclusive access.	Priority most recently set.	A number: 0 is normal. 1,2,3 are above-normal. –1 and –2 are below-normal.	From IDE, and at runtime. The IVC Relay Server Software limits the actual priority to the value authorized for the user named in PrivUsername.
Username	Specifies a username for accessing to the relay server.	Indicates the username.	Text. Set to blank if the relay server is not configured to require a password.	From property sheet, from IDE, and at runtime
Password	Specifies a	Indicates the	Text. Set to blank	From

Property Name	Meaning when set	Meaning when queried	Type of data	Settable
	password for general (nonprivileged) access to the relay server.	password.	if the relay server is not configured to require a password.	property sheet, from IDE, and at runtime
Priority	The priority to use for controlling the camera.	Priority most recently set.	A number between 1 and 6. 1 is normal and 6 is high.	From IDE, and at runtime. The IVC Relay Server Software limits
OverrideTimeout	The number of seconds for which an user access request should persist.	Number of seconds most recently set.	Seconds	From IDE, and at runtime. The IVC Relay Server Software limits
ImageHeight	The height of the image.	Number of pixels of the image height.	Pixels	From IDE, and at runtime. The IVC Relay Server Software limits
ImageWidth	The width of the image.	Number of pixels of the image width.	Pixels	From IDE, and at runtime. The IVC Relay Server Software limits

Notice that the lines of code mentioned in Step 10 of the Visual Basic program set the PanLeft and PanRight properties. Various solution-development environments offer different ways for the developer to set and query the properties of components; consult your favorite environment's documentation for details.

Methods

The ActiveX control offers two Methods that may be called from other software.

> *CenterAt (x,y)*
> *SendCommand (command)*

CenterAt(x, y)

Ordinarily if the EnablePoint property is set to true, when the user clicks in the displayed image the camera moves to center the clicked-on point in the center of the image. The CenterAt method allows software to simulate clicking in the image. The x and y parameter indicate the position within the image at which the simulated click occurs.

For example, this Visual Basic command pans the camera up and to the left.

```
Call CenterAt (10,10)
```

This method does not return any value.

SendCommand (command)

This method sends an arbitrary command to the relay server (at the URL specified by the property RelayServerURL). Some substitutions are made in the command string before sending it, as follows:

- %c – the camera current number.
- %u – the relay server URL. This is not necessary to get the command sent to the right server.
- %f – the requested frame rate in frames per second.
- %m – either “manage” or “control” depending on whether Exclusive mode is current.
- %% – a single "%" character.

For example, this Visual Basic command sets the zoom level to 1 on the current camera.

```
Call SendCommand( “/control/%c/cmd=dir&dir=z&dist=1” )
```

Notice that valid command strings start with the “/” character. Command strings are described in the document entitled *IVC Relay Server HTTP API.* Note also that commands are queued and processed one at a time, and that commands that have aged more than three seconds in the queue are discarded.

This method does not return any value.

Using the Image property to take snapshots in Visual Basic

The Image property allows a Visual Basic or other program to retrieve the JPEG image data for the image currently displayed at runtime. This makes it easy for such a program to implement a “snapshot” function.

To use the Image property in Visual Basic, you might for example create a Snapshot button for a user to click. The Click handler for your button would look like this:

```
Private Sub Snapshot_Click()
   Dim Img As String
   Dim Imglen As Long
   Img = IVC_ax1.Image
   Imglen = Len(Img)
   If Imglen > 0 Then
      Imgfile = FreeFile()
      Open "C:\temp\snapshot.jpg" For Binary Access Write Lock Write As
#Imgfile
      Put #Imgfile, , Img
      Close #imgfile
   End If
End Sub
```

This handler retrieves the value of the Image property. It then examines its length (if the length is zero, no image is available). It then opens a file called “snapsnot.jpg” in the C:\TEMP folder, and stores the file. This file, a standard JPEG file, can be displayed using a web browser or any other suitable program.

Using the IVC ActiveX Component in Web Pages

The IVC ActiveX Component can be used in custom web pages to offer video capability. These web pages may be displayed by Microsoft Internet Explorer. Of course, IVC video can be displayed in web pages without the need for an ActiveX component as well. You would incorporate the ActiveX Component in a page if you wanted the ability to control the video with scripts.

To use the component in a web page, you use the OBJECT tag in the page’s HTML. The Object tag for the IVC component looks like this:

```
      <OBJECT CLASSID="clsid:AD7E110C-F0F3-4574-B4BE-E69F5193BFA7"
                  CODEBASE="http://www.ivcco.com/activex
/ivc_ax.cab#version=1,0,2,7"
                           ID="ivc_ax"
                  WIDTH="352"
                  HEIGHT="240"   >
         <PARAM name=RelayServerURL value="http://demo.ivcco.com">
         <PARAM name=Camera value=1>
```

```
		<PARAM name=StreamVideo value=0>
		<PARAM name=FramesPerSecond value=2>
	</OBJECT>
```

You must specify the CLASSID value *exactly* as shown here, without any deviations whatsoever.

You may omit the line with the CODEBASE on it if you are sure the visitors to your web site have the control installed. You may change the CODEBASE's URL if you are serving the ivc_ax.cab file from some other location.

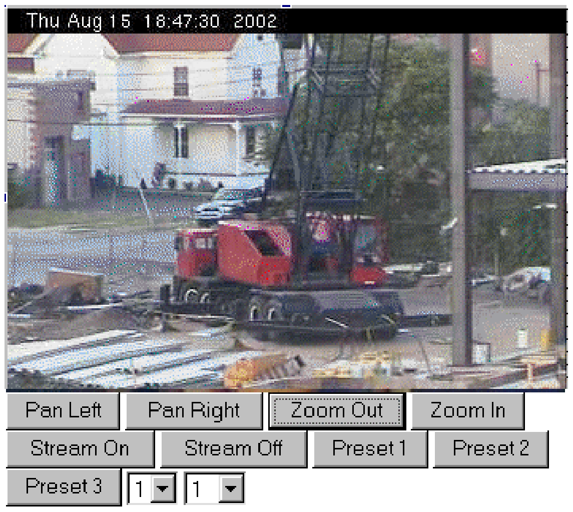
You may give the video frame a name using the ID value. This facilitates scripting.

The WIDTH and HEIGHT values describe the size of the video frame in the web page. 352x240 is normal North American video size. Large video frames are 704x480.

The remaining PARAM items allow you to set the properties of the component, as described in the table. Provide a PARAM item for each property you wish to set.

A Sample Scripted Web Page

This example shows a web page including a video frame and some Javascript to control it. The page contains buttons to turn on and off streaming, to pan and zoom the camera, and to choose presets. It includes a couple of dropdown lists to choose the camera to view and the frame rate. The web page looks like this:



The page's HTML code is this:

```
<HTML>
<HEAD>
```

```
<title>IVC ActiveX Sample</title>
</HEAD>
<BODY>
        <OBJECT classid="clsid:AD7E110C-F0F3-4574-B4BE-E69F5193BFA7"
                id="ivc_ax"
                width="352"
                height="240" >
          <PARAM name=RelayServerURL value="http://demo.ivcco.com">
          <PARAM name=Camera value=1>
                    <PARAM name=StreamVideo value=0>
                    <PARAM name=FramesPerSecond value=2>
        </OBJECT>

        <INPUT type="button" value="Pan Left" name="BtnPanLeft"
onClick="PanLeft(.99)">
        <INPUT type="button" value="Pan Right" name="BtnPanRight"
onClick="PanRight(.99)">
        <INPUT type="button" value="Zoom Out" name="BtnZoomIn"
onClick="SetZoom(0)">
        <INPUT type="button" value="Zoom In" name="BtnZoomOut"
onClick="SetZoom(11)">
        <INPUT type="button" value="Stream On" name="VStreamOn"
onClick="VideoStreamOn()">
        <INPUT type="button" value="Stream Off" name="VStreamOff"
onClick="VideoStreamOff()">
        <INPUT type="button" value="Preset 1" name="GoToPreset1"
onClick="GoToPreset(1)">
        <INPUT type="button" value="Preset 2" name="GoToPreset2"
onClick="GoToPreset(2)">
        <INPUT type="button" value="Preset 3" name="GoToPreset3"
onClick="GoToPreset(3)">
        <SELECT name="cams" onChange="SelectCam(
this.options[this.selectedIndex].value )">
                    <OPTION value="1">1
                    <OPTION value="2">2
                    <OPTION value="3">3
        </SELECT>
        <SELECT name="FPS" onChange="SetFPS(
this.options[this.selectedIndex].value )">
                    <OPTION value="1">1
                    <OPTION value="2">2
                    <OPTION value="3">3
                    <OPTION value="4">4
                    <OPTION value="5">5
                    <OPTION value="6">6
                    <OPTION value="7">7
                    <OPTION value="8">8
                <OPTION value="9">9
                <OPTION value="10">10
        </SELECT>
</BODY>
 <SCRIPT LANGUAGE="JScript">
function SelectCam( val ) {ivc_ax.Camera = val}
function SetFPS( val ){ivc_ax.FramesPerSecond = val }
function PanLeft( val ) {ivc_ax.PanLeft = val}
function PanRight( val ) {ivc_ax.PanRight = val }
function SetZoom( val ) {ivc_ax.ZoomLevel = val }
function VideoStreamOn() { ivc_ax.StreamVideo = 1 }
function VideoStreamOff() {ivc_ax.StreamVideo = 0 }
function GoToPreset( val ) { ivc_ax.Preset = val  }
</SCRIPT>
</HTML>
```

Programming Tips

When your program is not displaying a video stream (for example if you set form.Visible to False in Visual Basic) it is a good idea also to set the IVC ActiveX control's Active property to False as this will disconnect the streaming video and reduce the consumption of network bandwidth. When making the video stream visible again set the Active property back to True.

You may use the Caption property to help you troubleshoot application and network problems. To do this, insert special strings (like %u) into the value you set into the Caption property. When the caption is displayed it substitutes values for the special strings. The special strings are:

String	Meaning
%u	Relay Server URL
%c	Camera number
%f	Frames per second (0 if Streaming is not set to True)
%w	Onscreen video window width
%h	Onscreen video window height
%q	Number of video frames retrieved from relay server by this component since it started running.
%r	Number of video frames displayed by this component since it started running. Note that this number may be smaller than the number of video frames retrieved if the computer cannot keep up with displaying video, or if the video window is obscured.
%s	Number of commands to relay server (e.g. to pan, tilt, zoom, choose a preset, etc) accepted by this component from the user or from the custom application it is built into.

%t	Number of commands to relay server completed and acknowleged. This number is smaller than the number of commands accepted when the relay server has delays in processing commands.
%%	A simple percent sign.

Contact IVC

If you have any questions regarding this Tech Note you may contact us at:

Industrial Video & Control
330 Nevada Street
Newton, MA 02460
617-467-3059
support@ivcco.com