

VideoLab Custom Control

Order Form

Software License and Limited Warranty

Properties

Methods

Events

VideoLab™ is a trademark of Atrius Technologies, Inc.
ATRIUS® is a registered trademark of Atrius Technologies, Inc.
All rights reserved

Events

OnError

OnWaveStream

OnStatus

OnVideoStream

OnFrame

Methods

<u>AVIFileAllocate</u>	<u>CaptureAbort</u>	<u>CaptureStop</u>	<u>CompressionDialog</u>
<u>DisplayDialog</u>	<u>DriverDisconnect</u>	<u>EditCopy</u>	<u>FileSaveDIB</u>
<u>FormatDialog</u>	<u>GetAudioFormat</u>	<u>GetAudioFormatSize</u>	<u>GetCapHandle</u>
<u>GetDriverCaps</u>	<u>GetDriverName</u>	<u>GetDriverVersion</u>	<u>GetPlayHandle</u>
<u>GrabFrame</u>	<u>PlaybackLoad</u>	<u>PlaybackPause</u>	<u>PlaybackResume</u>
<u>PlaybackSeek</u>	<u>PlaybackStart</u>	<u>PlaybackStop</u>	<u>PlaybackUnload</u>
<u>Reset</u>	<u>SaveToBitmap</u>	<u>Sequence</u>	<u>SetAudioFormat</u>
<u>SetSequenceSetup</u>	<u>SingleFrame</u>	<u>SingleFrameClose</u>	<u>SingleFrameOpen</u>
<u>SourceDialog</u>			

Order Form

VideoLab™ Order Form

Name: _____
Address: _____

City/ST/Zip: _____
Phone: _____
Fax: _____
Email: _____

Send me:
____ Free Demo
(or GO DELPHI on CompuServe and search for VideoLab.Zip)
____ ID Key
(turns current Demo into complete product, \$129 per copy)
____ VideoLab™
(Demo with ID Key in one package, \$129 per copy)

Send my order via:
____ Fax ____ Email ____ Mail

Regularly priced at \$169. Buy direct today for only \$129! Our goal is to provide premier Delphi tools that will assist you in meeting your development goals quickly and easily. We look forward to having you as a valued client!

Copies ordered _____ x \$129 = _____

Tax 8.25% (tax for one copy is \$10.64) = _____

Shipping & Handling (\$8.00) = _____
(only if ordering on diskette by mail, \$0 otherwise)

TOTAL _____

Method of payment: VISA ___ MC ___ AMEX ___ DISC ___

Card Number: _____

Expiration Date: _____

Signature: _____

(Signature is needed only if faxing or mailing)

Order by using any one of the following options:

*Fax..... Fax your order to (214) 466-0266

*Email..... Email your order to atrius@why.net
*Phone..... Call in your order at (800) 928-7487
*Mail..... Mail the order form to:
Atrius Technologies, Inc.
1445 MacArthur Drive, Suite 200
Carrollton, TX 75007-4458

Call (800) 9-ATRIUS today to place your order or to request a Free Demo!
(all orders are processed on the date they are received)

Properties

Active

CaptureParms

Preview

AudioParms

Key

PreviewRate

AVIAllocate

KeyOwner

Scale

AVIFileName

Overlay

VideoDriver

Software License and Limited Warranty

PLEASE READ THIS LICENSE CAREFULLY BEFORE USING THE SOFTWARE, VIDEOLAB™, OWNED BY ATRIUS TECHNOLOGIES INC. BY USING THE SOFTWARE, YOU AGREE TO BECOME BOUND BY THE TERMS OF THIS AGREEMENT, WHICH INCLUDES THE SOFTWARE LICENSE AND WARRANTY DISCLAIMER (collectively referred to herein as the "Agreement"). THIS AGREEMENT CONSTITUTES THE COMPLETE AGREEMENT BETWEEN YOU AND ATRIUS TECHNOLOGIES, INC. IF YOU DO NOT AGREE TO THE TERMS OF THIS AGREEMENT, DO NOT USE THE SOFTWARE AND PROMPTLY RETURN IT FOR A FULL REFUND.

1. **Ownership of Software.** The enclosed manual and computer programs ("Software") were developed and are copyrighted by Atrius Technologies, Inc. ("Atrius") and are licensed, not sold, to you by Atrius for use under the following terms, and Atrius reserves any rights not expressly granted to you. You own the disk(s) on which any software is recorded, but Atrius retains ownership of all copies of the Software itself. Neither the manual nor the Software may be copied in whole or in part except as explicitly stated below.
2. **License.** Atrius, as Licensor, grants to you, the LICENSEE, a non-exclusive, non-transferable right to use this Software subject to the terms of the license as described below:
 - a. You may make backup copies of the Software for your use provided they bear the Atrius copyright notice.
 - b. You may use this Software in an unlimited number of custom or commercial databases or applications created by the original licensee. No additional product license or royalty is required.
3. **Restrictions.** You may not distribute copies of the Software to others (except as an integral part of a database or application within the terms of this License) or electronically transfer the Software from one computer to another over a network. You may not distribute copies of the Software as an integral part of a development shell or non-compiled commercial data base. The Software contains trade secrets and to protect them you may not decompile, reverse engineer, disassemble, or otherwise reduce the Software to a human perceivable form. **YOU MAY NOT MODIFY, ADAPT, TRANSLATE, RENT, LEASE, LOAN OR RESELL FOR PROFIT THE SOFTWARE OR ANY PART THEREOF.**
4. **Termination.** This license is effective until terminated. This license will terminate immediately without notice from Atrius if you fail to comply with any of its provisions. Upon termination you must destroy the Software and all copies thereof, and you may terminate this license at any time by doing so.
5. **Update Policy.** Atrius may create, from time to time, updated versions of the Software. At its option, Atrius will make such updates available to the Licensee.
6. **Warranty Disclaimer.** THE SOFTWARE IS PROVIDED "AS IS" WITHOUT

WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. ATRIUS DOES NOT WARRANT, GUARANTEE, OR MAKE ANY REPRESENTATIONS REGARDING THE USE, OR THE RESULTS OF THE USE, OF THE SOFTWARE OR WRITTEN MATERIALS IN THE TERMS OF CORRECTNESS, ACCURACY, RELIABILITY, CURRENTNESS OR OTHERWISE. THE ENTIRE RISK AS TO THE RESULTS AND PERFORMANCE OF THE SOFTWARE IS ASSUMED BY YOU. IF THE SOFTWARE OR WRITTEN MATERIALS ARE DEFECTIVE YOU, AND NOT ATRIUS OR IT'S DEALERS, DISTRIBUTORS, AGENTS, OR EMPLOYEES, ASSUME THE ENTIRE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION. However, Atrius warrants to the original Licensee that the disk(s) on which the Software is recorded is free from defects in materials and workmanship under normal use and service for a period of thirty (30) days from the date of delivery as evidenced by a copy of the receipt.

THIS IS THE ONLY WARRANT OF ANY KIND, EITHER EXPRESS OR IMPLIED, THAT IS MADE BY ATRIUS ON THIS SOFTWARE PRODUCT. NO ORAL OR WRITTEN INFORMATION OR ADVICE GIVEN BY ATRIUS, IT'S DEALERS, DISTRIBUTORS, AGENTS, OR EMPLOYEES SHALL CREATE A WARRANTY OR IN ANY WAY INCREASE THE SCOPE OF THIS WARRANTY, AND YOU MAY NOT RELY ON SUCH INFORMATION OR ADVICE. THIS WARRANTY GIVES YOU SPECIFIC LEGAL RIGHTS. YOU MAY HAVE OTHER RIGHTS, WHICH VARY FROM STATE TO STATE.

7. Governing Law. This agreement shall be governed by the laws of the State of Texas.

VideoLab™ is a trademark of Atrius Technologies, Inc.
ATRIUS® is a registered trademark of Atrius Technologies, Inc.
All rights reserved

TVideoAudioParms

Description

Defines the parameters to setup for the audio capture.

Properties

Channels: Number of audio channels. (1-Mono, 2-Stereo).

FormatTag: Audio format type. (use 1: WAVE_FORMAT_PCM).

SamplesPerSec: The sample rate. Common values are 8.0 kHz, 11.025 kHz, 22.05 kHz, and 44.1 kHz.

BitsPerSample: The number of bits per sample. Should be 8 or 16.

TVideoCaptureParms

Description

Defines the parameters to setup for the video capture.

Properties

RequestMicroSecPerFrame:	Requested capture rate.
MakeUserHitOKToCapture:	Show "Hit OK to cap" dlg?
PercentDropForError:	Give error message if greater than percentage
Yield:	Capture via background task?
IndexSize:	Max index size in frames.
ChunkGranularity:	Junk chunk granularity.
NumVideoRequested:	Number of video buffers to use.
CaptureAudio:	Capture audio?
NumAudioRequested:	Number of audio buffers to use.
KeyAbort:	Virtual key code to use as an abort.
AbortLeftMouse:	Abort on left mouse button?
AbortRightMouse:	Abort on right mouse button?
LimitEnabled:	Use a time limit? (TimeLimit)
TimeLimit:	Number of seconds to capture.
MCIControl:	Use MCI video source?
StepMCIDevice:	Step MCI device?
MCIStartTime:	Time to start in milliseconds.
MCIStopTime:	Time to stop in milliseconds.
StepCaptureAt2x:	Perform spatial averaging 2x.
StepCaptureAverageFrames:	Temporal average n Frames.
AudioBufferSize:	Size of audio buffers.
DisableWriteCache:	Attempt to disable write cache.
AVStreamMaster:	Which stream controls length?

TVideoDriverCaps

Description

Video driver capabilities..

Properties

wDeviceIndex:	WORD;	Driver index in system.ini. Same as the number used in <u>VideoDriver</u> .
fHasOverlay:	BOOL;	Can device overlay? <u>Overlay</u> .
fHasDlgVideoSource:	BOOL;	Has Video source dlg? <u>SourceDialog</u> .
fHasDlgVideoFormat:	BOOL;	Has Format dlg? <u>FormatDialog</u> .
fHasDlgVideoDisplay:	BOOL;	Has External out dlg? <u>DisplayDialog</u> .
fCaptureInitialized:	BOOL;	Driver ready to capture?
fDriverSuppliesPalettes:	BOOL;	Can driver make palettes?
hVideoIn:	Integer;	Driver In channel. (Not used in Win32 applications)
hVideoOut:	Integer;	Driver Out channel. (Not used in Win32 applications)
hVideoExtIn:	Integer;	Driver Ext In channel. (Not used in Win32 applications)
hVideoExtOut:	Integer;	Driver Ext Out channel. (Not used in Win32 applications)

TWaveFormatX

Description

Audio format structure.

Properties

wFormatTag:	WORD;	Audio format type. (use 1: WAVE_FORMAT_PCM).
nChannels:	WORD;	Number of audio channels. (1-Mono, 2-Stereo).
nSamplesPerSec:	DWORD;	The sample rate. Common values are 8.0 kHz, 11.025 kHz, 22.05 kHz, and 44.1 kHz.
nAvgBytesPerSec:	DWORD;	Required average data-transfer rate, in bytes per second, for the format tag. Equal to $nSamplesPerSec * nBlockAlign$.
nBlockAlign:	WORD;	The block alignment is the minimum atomic unit of data for the wFormatTag format type. For WAVE_FORMAT_PCM: $nBlockAlign = nChannels * (wBitsPerSample \div 8)$;
wBitsPerSample:	WORD;	The number of bits per sample. Should be 8 or 16.
cbSize:	WORD;	Size of extra format information appended to the end of the structure. For WAVE_FORMAT_PCM, this is set to 0.

OnError Event

Declaration

ECallbackError = function (Sender: TObject; nId: Integer; pMsg: PAnsiChar): Longint of object;
property OnError: ECallbackError;

Description

The video driver calls this function when ever an error occurs. pMsg is the error text. Your function should return 1.

OnFrame Event

Declaration

ECallbackFrame = function (Sender: TObject; pVideo: PVideoHdr): Longint of object;
property OnFrame: ECallbackFrame;

Description

The capture window calls this function before displaying preview frames. Your function should return 1.

OnStatus Event

Declaration

ECallbackStatus = function (Sender: TObject; nId: Integer; pMsg: PAnsiChar): Longint of object;
property OnStatus: ECallbackStatus;

Description

The video driver calls this function when ever the status of the driver has changed. Your function should return 1.

OnVideoStream Event

Declaration

ECallbackVideo = function (Sender: TObject): Longint of object;
property OnVideoStream: ECallbackVideo;

Description

The capture window calls this function for each frame in a capture session before writing to disk. Your function should return 1.

OnWaveStream Event

Declaration

ECallbackWave = function (Sender: TObject): Longint of object;
property OnWaveStream: ECallbackWave;

Description

The capture window calls this function for each frame in a capture session before writing to disk. Your function should return 1.

AVIFileAllocate Function

Declaration

function AVIFileAllocate(ISize: Longint): Longint;

Description

This function will preallocate disk space for the file specified in AVIFileName property. Having the file preallocated will save time during a video capture.

Return

True on success, False on error.

CaptureAbort Function

Declaration

function CaptureAbort: Longint;

Description

Aborts the current video capture session started with Sequence.

Return

True on success, False on error.

CaptureStop Function

Declaration

function CaptureStop: Longint;

Description

Stops the current video capture session started with Sequence.

Return

True on success, False on error.

CompressionDialog Function

Declaration

function CompressionDialog: Longint;

Description

Displays a dialog box in which the user can select a compressor to use during the capture process.

Return

True on success, False on error.

DisplayDialog Function

Declaration

function DisplayDialog: Longint;

Description

Displays a dialog box in which the user can set or adjust the video output. This dialog box usually contains controls that affect the hue, contrast, brightness, and color alignment.

Return

True on success, False on error.

DriverDisconnect Function

Declaration

function DriverDisconnect: Longint;

Description

This function will disconnect the currently connected video driver. DriverDisconnect is called when Active is changed from True to False.

Return

True on success, False on error.

EditCopy Function

Declaration

function EditCopy: Longint;

Description

EditCopy will ask the video driver to save the current frame to the clipboard. Saving this DIB requires a lot of memory and will fail if there is not enough resources available.

Return

True on success, False on error.

FileSaveDIB Function

Declaration

function FileSaveDIB(pFileName: PAnsiChar): Longint;

Description

FileSaveDIB will request the video driver to save the current frame to a file. Some video cards require a lot of resources to perform this task. [SaveToBitmap](#) has been included for this reason.

Return

True on success, False on error.

FormatDialog Function

Declaration

function FormatDialog: Longint;

Description

Displays a dialog box in which the user can select the video format. The Video Format dialog box usually controls image dimensions, bit depth, and hardware compression options.

Return

True on success, False on error.

GetAudioFormat Function

Declaration

function GetAudioFormat(pWave: PWaveFormatX; nSize: Integer): Longint;

Description

Retrieves the current audio format. You should call GetAudioFormatSize to retrieve the size of buffer before calling this function.

Return

Returns the size of the audio format.

GetAudioFormatSize Function

Declaration

function GetAudioFormatSize: Longint;

Description

GetAudioFormatSize returns the size of PWaveFormatX for the current compression method. This size can be used with GetAudioFormat function.

Return

Returns the size of the audio format buffer.

GetCapHanle Function

Declaration

function GetCapHandle: HWND;

Description

GetCapWindow will return the window handle to the capture window. This function will return zero if Active is False.

Return

0 on Error, otherwise the window handle to the current capture window.

GetDriverCaps Function

Declaration

function GetDriverCaps(pCap: PVideoDriverCaps): Longint;

Description

Call this function to retrieve video driver capabilities.

Return

True on success, False on error.

GetDriverName Function

Declaration

function GetDriverName(nIdx: Integer; pName: PAnsiChar; nLen: Integer): Longint;

Description

This function returns the video driver description. An application can find all of the installed video drivers by repeatedly calling this function and incrementing nIdx by one. nIdx is the driver number to query ([VideoDriver](#)), valid values are 0 - 9. pName will contain the driver discription if the call is successfull. nLen should be set to the maximum size of pName.

Example

```
var
    pDesc: array[0..80] of Char;
begin
    nIdx := 0;
    while (GetDriverName(nIdx, pDesc, SizeOf(pDesc) <> 0)
        DriverName.Caption := pDesc;
end;
```

Return

True on success, False on error.

GetDriverVersion Function

Declaration

function GetDriverVersion(nIdx: Integer; pName: PAnsiChar; nLen: Integer): Longint;

Description

This function returns the video driver version. nIdx is the driver number to query ([VideoDriver](#)), valid values are 0 - 9. pName will contain the driver version if the call is successful. nLen should be set to the maximum size of pName.

Return

True on success, False on error.

GetPlayHandle Function

Declaration

function GetPlayHandle: HWND;

Description

GetCapWindow will return the window handle to the capture window. This function will return zero if there is no Playback window active.

Return

0 on Error, otherwise the window handle to the current playback window.

GrabFrame Function

Declaration

function GrabFrame(bStop: Boolean): Longint;

Description

GrabFrame will store the current frame into the video driver's internal buffer. Setting **bStop** to True will pause the frame on the screen, **False** will store the image and continue in current Overlay or Preview mode.

Return

True on success, False on error.

PlaybackLoad Function

Declaration

function PlaybackLoad(bInFrame: Boolean; dwStyle: DWORD; pFileName: PAnsiChar): Longint;

Description

This function will create a video play back window from an existing video file.

bInFrame: **True** is create the control within the current control window. **False** will create a separate window.

dwStyle: The following styles or together.
MCIWINDF_NOAUTOSIZEWINDOW
MCIWINDF_NOPLAYBAR
MCIWINDF_NOAUTOSIZEMOVIE
MCIWINDF_NOMENU
MCIWINDF_SHOWNAME
MCIWINDF_SHOWPOS
MCIWINDF_SHOWMODE
MCIWINDF_SHOWALL
MCIWINDF_NOTIFYANSI
MCIWINDF_NOTIFYMODE
MCIWINDF_NOTIFYPOS
MCIWINDF_NOTIFYSIZE
MCIWINDF_NOTIFYERROR
MCIWINDF_NOTIFYALL

pFileName: Name of the AVI file to load.

Return

Non-Zero on success, False on error.

PlaybackPause Function

Declaration

function PlaybackPause: Longint;

Description

Pause the current playback window.

Return

True on success, False on error.

PlaybackResume Function

Declaration

function PlaybackResume: Longint;

Description

Resumes a Paused play back window.

Return

True on success, False on error.

PlaybackSeek Function

Declaration

function PlaybackSeek(nPos: Longint): Longint;

Description

Seek to **nPos** in video stream.

Return

True on success, False on error.

PlaybackStart Function

Declaration

function PlaybackStart: Longint;

Description

Starts a playback window that has been loaded.

Return

True on success, False on error.

PlaybackStop Function

Declaration

function PlaybackStop: Longint;

Description

Stops the current playback window.

Return

True on success, False on error.

PlaybackUnload Function

Declaration

function PlaybackUnload: Longint;

Description

Removes the current playback window. The playback window was loaded by calling [PlaybackLoad](#).

Return

True on success, False on error.

Reset Function

Declaration

function Reset: Longint;

Description

Resets all attached video drivers.

Return

True on success, False on error.

SaveToBitmap Function

Declaration

function SaveToBitmap(nBits: Integer; nCompress: Integer; pFileName: PAnsiChar): Longint;

Description

SaveToBitmap will save the current paused video window to a bitmap file. This routine is included as an alternative to FileSaveDIB, which requires a lot of resources to use.

Return

True on success, False on error.

Sequence Function

Declaration

function Sequence: Longint;

Description

This function will begin a video capture sequence. The sequence will use the current values set in CaptureParms and AudioParms. The video stream will be saved to AVIFileName.

Return

True on success, False on error.

SetAudioFormat Function

Declaration

function SetAudioFormat(pWave: PWaveFormatX; nSize: Integer): Longint;

Description

Sets the audio format to use during a video capture Sequence. If AudioParms is not **nil** whatever you set will be overwritten at Sequence startup.

Return

True on success, False on error.

SetSequenceSetup Function

Declaration

function SetSequenceSetup(pCapParms: PVideoCaptureParms): Longint;

Description

This function is called by Sequence if CaptureParms is not **nil**.

Return

True on success, False on error.

SingleFrame Function

Declaration

function SingleFrame: Longint;

Description

This function will save a single frame to the capture file. Use [SingleFrameOpen](#) to setup the capture file.

Return

True on success, False on error.

SingleFrameClose Function

Declaration

function SingleFrameClose: Longint;

Description

Closes the capture which was opened by calling SingleFrameOpen.

Return

True on success, False on error.

SingleFrameOpen Function

Declaration

function SingleFrameOpen: Longint;

Description

Prepare the capture file for adding calling SingleFrame to add frames to a capture file.

Return

True on success, False on error.

SourceDialog Function

Declaration

function

Description

Displays a dialog box in which the user can control the video source. The Video Source dialog box might contain controls that select input sources; alter the hue, contrast, brightness of the image; and modify the video quality before digitizing the images into the frame buffer.

Return

True on success, False on error.

AVIAllocate Property

Declaration

property AVIAllocate: Longint;

Description

This property will preallocate the size of the file specified in AVIFileName before beginning a video capture.

AVIFileName Property

Declaration

property AVIFileName: string;

Description

This is the file path and name of the file to capture video to. This file will be over-written on each call to [Sequence](#).

Active Property

Declaration

property Active: Boolean;

Description

Turns the video capture window on and off.

AudioParms Property

Declaration

property AudioParms: TVideoAudioParms;

Description

Property setups up the parameters for the audio.

CaptureParms Property

Declaration

property CaptureParms: TVideoCaptureParms;

Description

The property defines the parameters for recording video.

Key Property

Declaration

property Key: string;

Description

When you register **TVideoLab** you would receive a key. Enter the key value here along with your name in the KeyOwner field to register your copy.

Order Form

KeyOwner Property

Declaration

property KeyOwner: string;

Description

The name in which **TVideoLab** was registered under..

Order Form

Overlay Property

Declaration

property Overlay: Boolean;

Description

Allows live video to be previewed within a window. The video will be shown when Active is set to True.

Preview Property

Declaration

property Preview: Boolean;

Description

Allows live video frames to be previewed within a window. The video will be shown when Active is set to True.

PreviewRate Property

Declaration

property PreviewRate: Integer; (milliseconds)

Description

Rate at which new frames are captured and displayed. Preview mode takes up massive CPU resources. Applications should increase this value whenever possible, to allow other applications to run.

Scale Property

Declaration

property Scale: Boolean;

Description

When enabled, the captured video frame is stretched to the dimensions of the capture window. This is only valid for Preview mode.

VideoDriver Property

Declaration

property VideoDriver: Integer;

Description

This is the integer value of the video driver to connect to. A list of video drivers may retrieved by calling [GetDriverName](#) until the function returns a False value.

