AudioCapabilities Class

Capabilities of the audio device such as min/max sampling rate and number of channels available.

For a list of all members of this type, see AudioCapabilities Members.

System.Object AudioCapabilities

```
public class AudioCapabilities
```

Requirements

Namespace: DirectX.Capture Namespace

Assembly: DirectX.Capture.dll

See Also

AudioCapabilities Members | DirectX.Capture Namespace

© 2002 Brian Low. All rights reserved.

Generated from assembly DirectX.Capture [1.0.1122.23301]
DirectX.Capture Class Library
## AudioCapabilities Members

### Public Instance Fields

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>ChannelsGranularity</strong></td>
<td>Granularity of the channels. For example, channels 2 through 4, in steps of 2.</td>
</tr>
<tr>
<td><strong>MaximumChannels</strong></td>
<td>Maximum number of audio channels.</td>
</tr>
<tr>
<td><strong>MaximumSampleSize</strong></td>
<td>Maximum number of bits per sample.</td>
</tr>
<tr>
<td><strong>MaximumSamplingRate</strong></td>
<td>Maximum sample frequency.</td>
</tr>
<tr>
<td><strong>MinimumChannels</strong></td>
<td>Minimum number of audio channels.</td>
</tr>
<tr>
<td><strong>MinimumSampleSize</strong></td>
<td>Minimum number of bits per sample.</td>
</tr>
<tr>
<td><strong>MinimumSamplingRate</strong></td>
<td>Minimum sample frequency.</td>
</tr>
<tr>
<td><strong>SampleSizeGranularity</strong></td>
<td>Granularity of the bits per sample. For example, 8 bits per sample through 32 bits per sample, in steps of 8.</td>
</tr>
<tr>
<td><strong>SamplingRateGranularity</strong></td>
<td>Granularity of the frequency. For example, 11025 Hz to 44100 Hz, in steps of 11025 Hz.</td>
</tr>
</tbody>
</table>

### Public Instance Methods

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Equals</strong> (inherited from Object)</td>
<td>Determines whether the specified Object is equal to the current Object.</td>
</tr>
<tr>
<td><strong>GetHashCode</strong> (inherited from Object)</td>
<td>Serves as a hash function for a particular type, suitable for use in hashing algorithms and data</td>
</tr>
</tbody>
</table>
structures like a hash table.

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>GetType</strong> (inherited from Object)</td>
<td>Gets the <strong>Type</strong> of the current instance.</td>
</tr>
<tr>
<td><strong>ToString</strong> (inherited from Object)</td>
<td>Returns a <strong>String</strong> that represents the current <strong>Object</strong>.</td>
</tr>
</tbody>
</table>

See Also

AudioCapabilities Class | DirectX.Capture Namespace

© 2002 Brian Low. All rights reserved.

Generated from assembly DirectX.Capture [1.0.1122.23301]
DirectX.Capture Class Library
AudioCapabilities Fields

The fields of the AudioCapabilities class are listed below. For a complete list of AudioCapabilities class members, see the AudioCapabilities Members topic.

Public Instance Fields

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ChannelsGranularity</td>
<td>Granularity of the channels. For example, channels 2 through 4, in steps of 2.</td>
</tr>
<tr>
<td>MaximumChannels</td>
<td>Maximum number of audio channels.</td>
</tr>
<tr>
<td>MaximumSampleSize</td>
<td>Maximum number of bits per sample.</td>
</tr>
<tr>
<td>MaximumSamplingRate</td>
<td>Maximum sample frequency.</td>
</tr>
<tr>
<td>MinimumChannels</td>
<td>Minimum number of audio channels.</td>
</tr>
<tr>
<td>MinimumSampleSize</td>
<td>Minimum number of bits per sample.</td>
</tr>
<tr>
<td>MinimumSamplingRate</td>
<td>Minimum sample frequency.</td>
</tr>
<tr>
<td>SampleSizeGranularity</td>
<td>Granularity of the bits per sample. For example, 8 bits per sample through 32 bits per sample, in steps of 8.</td>
</tr>
<tr>
<td>SamplingRateGranularity</td>
<td>Granularity of the frequency. For example, 11025 Hz to 44100 Hz, in steps of 11025 Hz.</td>
</tr>
</tbody>
</table>

See Also

AudioCapabilities Class | DirectX.Capture Namespace

© 2002 Brian Low. All rights reserved.
DirectX.Capture Class Library
AudioCapabilities.ChannelsGranularity Field

Granularity of the channels. For example, channels 2 through 4, in steps of 2.

```
public int ChannelsGranularity;
```

See Also

AudioCapabilities Class | AudioCapabilities Members | DirectX.Capture Namespace
-------------------------------

© 2002 Brian Low. All rights reserved.
Generated from assembly DirectX.Capture [1.0.1122.23301]
DirectX.Capture Class Library
AudioCapabilities.MaximumChannels Field

Maximum number of audio channels.

```csharp
public int MaximumChannels;
```

See Also

AudioCapabilities Class | AudioCapabilities Members | DirectX.Capture Namespace

© 2002 Brian Low. All rights reserved.

Generated from assembly DirectX.Capture [1.0.1122.23301]
DirectX.Capture Class Library
AudioCapabilities.MaximumSampleSize Field

Maximum number of bits per sample.

```csharp
public int MaximumSampleSize;
```

See Also

AudioCapabilities Class | AudioCapabilities Members | DirectX.Capture Namespace

© 2002 Brian Low. All rights reserved.

Generated from assembly DirectX.Capture [1.0.1122.23301]
DirectX.Capture Class Library
AudioCapabilities.MaximumSamplingRate Field

Maximum sample frequency.

```csharp
public int MaximumSamplingRate;
```

See Also

AudioCapabilities Class | AudioCapabilities Members | DirectX.Capture Namespace

© 2002 Brian Low. All rights reserved.

Generated from assembly DirectX.Capture [1.0.1122.23301]
DirectX.Capture Class Library
AudioCapabilities.MinimumChannels Field

Minimum number of audio channels.


global int MinimumChannels;

See Also

AudioCapabilities Class | AudioCapabilities Members | DirectX.Capture Namespace

© 2002 Brian Low. All rights reserved.

Generated from assembly DirectX.Capture [1.0.1122.23301]
DirectX.Capture Class Library
AudioCapabilities.MinimumSampleSize Field

Minimum number of bits per sample.

```
public int MinimumSampleSize;
```

See Also

AudioCapabilities Class | AudioCapabilities Members | DirectX.Capture Namespace

© 2002 Brian Low. All rights reserved.

Generated from assembly DirectX.Capture [1.0.1122.23301]
DirectX.Capture Class Library
**AudioCapabilities.MinimumSamplingRate Field**

Minimum sample frequency.

```csharp
public int MinimumSamplingRate;
```

See Also

[AudioCapabilities Class] | [AudioCapabilities Members] | [DirectX.Capture Namespace]

© 2002 Brian Low. All rights reserved.

Generated from assembly DirectX.Capture [1.0.1122.23301]
DirectX.Capture Class Library
AudioCapabilities.SampleSizeGranularity Field

Granularity of the bits per sample. For example, 8 bits per sample through 32 bits per sample, in steps of 8.

```csharp
public int SampleSizeGranularity;
```

See Also

- [AudioCapabilities Class](#)
- [AudioCapabilities Members](#)
- [DirectX.Capture Namespace](#)

© 2002 Brian Low. All rights reserved.

Generated from assembly DirectX.Capture [1.0.1122.23301]
DirectX.Capture Class Library
Granularity of the frequency. For example, 11025 Hz to 44100 Hz, in steps of 11025 Hz.

```csharp
public int SamplingRateGranularity;
```

See Also

- [AudioCapabilities Class](#) | [AudioCapabilities Members](#) | [DirectX.Capture Namespace](#)

© 2002 Brian Low. All rights reserved.

Generated from assembly DirectX.Capture [1.0.1122.23301]
DirectX.Capture Class Library
AudioSource Class

Represents a physical connector or source on an audio device. This class is used on filters that support the IAMAudioInputMixer interface such as source cards.

For a list of all members of this type, see AudioSource Members.

System.Object Source AudioSource

Requirements

Namespace: DirectX.Capture Namespace

Assembly: DirectX.Capture.dll

See Also

AudioSource Members | DirectX.Capture Namespace

© 2002 Brian Low. All rights reserved.

Generated from assembly DirectX.Capture [1.0.1122.23301]
DirectX.Capture Class Library
### AudioSource Members

#### Public Instance Properties

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>Enabled</code></td>
<td>Enable or disable this source. For audio sources it is usually possible to enable several sources. When setting <code>Enabled=true</code>, set <code>Enabled=false</code> on all other audio sources.</td>
</tr>
<tr>
<td><code>Name</code> (inherited from <code>Source</code>)</td>
<td>The name of the source. Read-only.</td>
</tr>
</tbody>
</table>

#### Public Instance Methods

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>Dispose</code></td>
<td>Release unmanaged resources.</td>
</tr>
<tr>
<td><code>Equals</code> (inherited from <code>Object</code>)</td>
<td>Determines whether the specified <code>Object</code> is equal to the current <code>Object</code>.</td>
</tr>
<tr>
<td><code>GetHashCode</code> (inherited from <code>Object</code>)</td>
<td>Serves as a hash function for a particular type, suitable for use in hashing algorithms and data structures like a hash table.</td>
</tr>
<tr>
<td><code>GetType</code> (inherited from <code>Object</code>)</td>
<td>Gets the <code>Type</code> of the current instance.</td>
</tr>
<tr>
<td><code>ToString</code> (inherited from <code>Source</code>)</td>
<td>Obtains the String representation of this instance.</td>
</tr>
</tbody>
</table>

### See Also

[AudioSource Class] | [DirectX.Capture Namespace]

© 2002 Brian Low. All rights reserved.

Generated from assembly DirectX.Capture [1.0.1122.23301]
 DirectX.Capture Class Library
AudioSource Properties

The properties of the **AudioSource** class are listed below. For a complete list of **AudioSource** class members, see the **AudioSource Members** topic.

Public Instance Properties

| **Enabled** | Enable or disable this source. For audio sources it is usually possible to enable several sources. When setting Enabled=true, set Enabled=false on all other audio sources. |
| **Name** (inherited from **Source**) | The name of the source. Read-only. |

See Also

[AudioSource Class](#) | [DirectX.Capture Namespace](#)

© 2002 Brian Low. All rights reserved.

Generated from assembly DirectX.Capture [1.0.1122.23301]
| DirectX.Capture Class Library |
**AudioSource.Enabled Property**

Enable or disable this source. For audio sources it is usually possible to enable several sources. When setting Enabled=true, set Enabled=false on all other audio sources.

```csharp
public override bool Enabled {get; set;}
```

See Also

- [AudioSource Class](#)
- [AudioSource Members](#)
- [DirectX.Capture Namespace](#)

© 2002 Brian Low. All rights reserved.

Generated from assembly DirectX.Capture [1.0.1122.23301]
DirectX.Capture Class Library
## AudioSource Methods

The methods of the **AudioSource** class are listed below. For a complete list of **AudioSource** class members, see the [AudioSource Members](#) topic.

### Public Instance Methods

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>✷ <strong>Dispose</strong></td>
<td>Release unmanaged resources.</td>
</tr>
<tr>
<td>✷ <strong>Equals</strong> (inherited from <strong>Object</strong>)</td>
<td>Determines whether the specified <strong>Object</strong> is equal to the current <strong>Object</strong>.</td>
</tr>
<tr>
<td>✷ <strong>GetHashCode</strong> (inherited from <strong>Object</strong>)</td>
<td>Serves as a hash function for a particular type, suitable for use in hashing algorithms and data structures like a hash table.</td>
</tr>
<tr>
<td>✷ <strong>GetType</strong> (inherited from <strong>Object</strong>)</td>
<td>Gets the <strong>Type</strong> of the current instance.</td>
</tr>
<tr>
<td>✷ <strong>ToString</strong> (inherited from <strong>Source</strong>)</td>
<td>Obtains the String representation of this instance.</td>
</tr>
</tbody>
</table>

### See Also

[AudioSource Class](#) | [DirectX.Capture Namespace](#)

© 2002 Brian Low. All rights reserved.

Generated from assembly DirectX.Capture [1.0.1122.23301]
DirectX.Capture Class Library
**AudioSource.Dispose Method**

Release unmanaged resources.

```
public override void Dispose();
```

**Implements**

`IDisposable.Dispose`

**See Also**

[AudioSource Class](#) | [AudioSource Members](#) | [DirectX.Capture Namespace](#)

© 2002 Brian Low. All rights reserved.

Generated from assembly DirectX.Capture [1.0.1122.23301]
DirectX.Capture Class Library
Capture Class

Use the Capture class to capture audio and video to AVI files.
For a list of all members of this type, see Capture Members.

System.Object  Capture

public class Capture

Remarks
This is the core class of the Capture Class Library. The following sections introduce the Capture class and how to use this library.

Basic Usage
The Capture class only requires a video device and/or audio device to begin capturing. The Filters class provides lists of the installed video and audio devices.

```csharp
using DirectX.Capture
...
Capture capture = new Capture( Filters.\VideoInputDevices[0],
Filters.\AudioInputDevices[0] )
capture.Start();
...
capture.Stop();
```

This will capture video and audio using the first video and audio devices installed on the system. To capture video only, pass a null as the second parameter of the constructor.

The class is initialized to a valid temporary file in the Windows temp folder. To capture to a different file, set the Filename property before you begin capturing. Remember to add DirectX.Capture.dll to your project references.
Setting Common Properties

The example below shows how to change video and audio settings. Properties such as **FrameRate** and **AudioSampleSize** allow you to programmatically adjust the capture. Use **VideoCaps** and **AudioCaps** to determine valid values for these properties.

```csharp
Capture capture = new Capture( Filters.VideoInputDevices[0], Filters.AudioInputDevices[1] );
capture.VideoCompressor = Filters.VideoCompressors[0];
capture.AudioCompressor = Filters.AudioCompressors[0];
capture.FrameRate = 29.997;
capture.FrameSize = new Size( 640, 480 );
capture.AudioSamplingRate = 44100;
capture.AudioSampleSize = 16;
capture.Filename = "C:\MyVideo.avi";
capture.Start();
...
capture.Stop();
```

The example above also shows the use of video and audio compressors. In most cases you will want to use compressors. Uncompressed video can easily consume over a 1GB of disk space per minute. Whenever possible, set the **VideoCompressor** and **AudioCompressor** properties as early as possible. Changing them requires the internal filter graph to be rebuilt which often causes most of the other properties to be reset to default values.

Listing Devices

Use the **VideoInputDevices** collection to list video capture devices installed on the system.

```csharp
foreach ( Filter f in Filters.VideoInputDevices ) {
}
The **Filters** class also provides collections for audio capture devices, video compressors and audio compressors.

**Preview**

Video preview is controled with the **PreviewWindow** property. Setting this property to a visible control will immediately begin preview. Set to null to stop the preview.

```csharp
    // Enable preview
    capture.PreviewWindow = myPanel;
    // Disable preview
    capture.PreviewWindow = null;
```

The control used must have a window handle (HWND), good controls to use are the Panel or the form itself.

Retrieving or changing video/audio settings such as FrameRate, FrameSize, AudioSamplingRate, and AudioSampleSize will cause the preview window to flash. This is because the preview must be temporarily stopped. Disable the preview if you need to access several properties at the same time.

**Property Pages**

Property pages exposed by the devices and compressors are available through the **PropertyPages** collection.

```csharp
    // Display the first property page
    capture.PropertyPages[0].Show();
```

The property pages will often expose more settings than the Capture class does directly. Some examples are brightness, color space,
audio balance and bass boost. The disadvantage to using the property pages is the user's choices cannot be saved and later restored. The exception to this is the video and audio compressor property pages. Most compressors support the saving and restoring state, see the `State` property for more information.

Changes made in the property page will be reflected immediately in the Capture class properties (e.g. `Capture.FrameSize`). However, the reverse is not always true. A change made directly to `FrameSize`, for example, may not be reflected in the associated property page. Fortunately, the filter will use requested `FrameSize` even though the property page shows otherwise.

**Saving and Restoring Settings**

To save the user's choice of devices and compressors, save `MonikerString` and user it later to recreate the Filter object.

To save a user's choices from a property page use `State`. However, only the audio and video compressor property pages support this.

The last items to save are the video and audio settings such as `FrameSize` and `AudioSamplingRate`. When restoring, remember to restore these properties after setting the video and audio compressors.

```csharp
// Disable preview
capture.PreviewWindow = null;

// Save settings
string videoDevice = capture.VideoDevice.MonikerString;
string audioDevice = capture.AudioDevice.MonikerString;
string videoCompressor = capture.VideoCompressor.MonikerString;
string audioCompressor = capture.AudioCompressor.MonikerString;
double frameRate = capture.FrameRate;
Size frameSize = capture.FrameSize;
short audioChannels = capture.AudioChannels;
short audioSampleSize = capture.AudioSampleSize;
```
int audioSamplingRate = capture.AudioSamplingRate;
ArrayList pages = new ArrayList();
foreach ( PropertyPage p in capture.PropertyPages )
{
    if ( p.SupportsPersisting )
        pages.Add( p.State );
}

// Restore settings
Capture capture = new Capture( new Filter( videoDevice),
    new Filter( audioDevice ),
    capture.VideoCompressor = new Filter( videoCompressor );
capture.AudioCompressor = new Filter( audioCompressor );
capture.FrameRate = frameRate;
capture.FrameSize = frameSize;
capture.AudioChannels = audioChannels;
capture.AudioSampleSize = audioSampleSize;
capture.AudioSamplingRate = audioSamplingRate;
foreach ( PropertyPage p in capture.PropertyPages )
{
    if ( p.SupportsPersisting )
    {
        p.State = (byte[]) pages[0];
        pages.RemoveAt( 0 );
    }
}
// Enable preview
capture.PreviewWindow = myPanel;

**TV Tuner**

To access the TV Tuner, use the **Tuner** property. If the device does not have a TV tuner, this property will be null. See **Channel**, **InputType** and **SignalPresent** for more information.
```csharp
// Change to channel 5
capture.Tuner.Channel = 5;
```

**Troubleshooting**

This class library uses COM Interop to access the full capabilities of DirectShow, so if there is another application that can successfully use a hardware device then it should be possible to modify this class library to use the device.

Try the **AMCap** sample from the DirectX SDK (DX9\Samples\C++\DirectShow\Bin\AMCap.exe) or **Virtual VCR** from [http://www.DigTV.ws](http://www.DigTV.ws)

**Credits**

This class library would not be possible without the DShowNET project by NETMaster:

Documentation is generated by nDoc available at

**Feedback**

Feel free to send comments and questions to me at mportobello@hotmail.com. If the topic may be of interest to others, post your question on the www.codeproject.com page for DirectX.Capture.

**Requirements**

**Namespace:** DirectX.Capture Namespace

**Assembly:** DirectX.Capture.dll

**See Also**
**DirectX.Capture Class Library**
### Capture Members

#### Public Instance Constructors

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Capture Constructor</strong></td>
<td>Create a new Capture object. videoDevice and audioDevice can be null if you do not wish to capture both audio and video. However at least one must be a valid device. Use the Filters class to list available devices.</td>
</tr>
</tbody>
</table>

#### Public Instance Properties

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>AudioCaps</strong></td>
<td>The capabilities of the audio device.</td>
</tr>
<tr>
<td><strong>AudioChannels</strong></td>
<td>Get or set the number of channels in the waveform-audio data.</td>
</tr>
<tr>
<td><strong>AudioCompressor</strong></td>
<td>The audio compression filter.</td>
</tr>
<tr>
<td><strong>AudioDevice</strong></td>
<td>The audio capture device filter. Read-only. To use a different device, dispose of the current Capture instance and create a new instance with the desired device.</td>
</tr>
<tr>
<td><strong>AudioSampleSize</strong></td>
<td>Get or set the number of bits recorded per sample.</td>
</tr>
<tr>
<td><strong>AudioSamplingRate</strong></td>
<td>Get or set the number of audio samples taken per second.</td>
</tr>
<tr>
<td><strong>AudioSource</strong></td>
<td>The current audio source. Use Capture.AudioSources to list available sources. Set to null to disable all sources (mute).</td>
</tr>
<tr>
<td>Property</td>
<td>Description</td>
</tr>
<tr>
<td>------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td><strong>AudioSources</strong></td>
<td>Collection of available audio sources/physical connectors on the current audio device.</td>
</tr>
<tr>
<td><strong>Capturing</strong></td>
<td>Is the class currently capturing. Read-only.</td>
</tr>
<tr>
<td><strong>Cued</strong></td>
<td>Has the class been cued to begin capturing. Read-only.</td>
</tr>
<tr>
<td><strong>Filename</strong></td>
<td>Name of file to capture to. Initially set to a valid temporary file.</td>
</tr>
<tr>
<td><strong>FrameRate</strong></td>
<td>Gets and sets the frame rate used to capture video.</td>
</tr>
<tr>
<td><strong>FrameSize</strong></td>
<td>Gets and sets the frame size used to capture video.</td>
</tr>
<tr>
<td><strong>PreviewWindow</strong></td>
<td>The control that will host the preview window.</td>
</tr>
<tr>
<td><strong>PropertyPages</strong></td>
<td>Available property pages.</td>
</tr>
<tr>
<td><strong>Stopped</strong></td>
<td>Is the class currently stopped. Read-only.</td>
</tr>
<tr>
<td><strong>Tuner</strong></td>
<td>The TV Tuner or null if the current video device does not have a TV Tuner.</td>
</tr>
<tr>
<td><strong>VideoCaps</strong></td>
<td>The capabilities of the video device.</td>
</tr>
<tr>
<td><strong>VideoCompressor</strong></td>
<td>The video compression filter. When this property is changed the internal filter graph is rebuilt. This means that some properties will be reset. Set this property as early as possible to avoid losing changes. This property cannot be changed while capturing.</td>
</tr>
<tr>
<td>Property</td>
<td>Description</td>
</tr>
<tr>
<td>--------------</td>
<td>-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td><strong>VideoDevice</strong></td>
<td>The video capture device filter. Read-only. To use a different device, dispose of the current Capture instance and create a new instance with the desired device.</td>
</tr>
<tr>
<td><strong>VideoSource</strong></td>
<td>The current video source. Use Capture.VideoSources to list available sources. Set to null to disable all sources (mute).</td>
</tr>
<tr>
<td><strong>VideoSources</strong></td>
<td>Collection of available video sources/physical connectors on the current video device.</td>
</tr>
</tbody>
</table>

**Public Instance Methods**

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Cue</strong></td>
<td>Prepare for capturing. Use this method when capturing must begin as quickly as possible.</td>
</tr>
<tr>
<td><strong>Dispose</strong></td>
<td>Calls Stop, releases all references. If a capture is in progress it will be stopped, but the CaptureComplete event will NOT fire.</td>
</tr>
<tr>
<td><strong>Equals</strong> (inherited from <strong>Object</strong>)</td>
<td>Determines whether the specified <strong>Object</strong> is equal to the current <strong>Object</strong>.</td>
</tr>
<tr>
<td><strong>GetHashCode</strong> (inherited from <strong>Object</strong>)</td>
<td>Serves as a hash function for a particular type, suitable for use in hashing algorithms and data structures like a hash table.</td>
</tr>
<tr>
<td><strong>GetType</strong> (inherited from <strong>Object</strong>)</td>
<td>Gets the <strong>Type</strong> of the current instance.</td>
</tr>
<tr>
<td><strong>Start</strong></td>
<td>Begin capturing.</td>
</tr>
<tr>
<td>Method</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td><strong>Stop</strong></td>
<td>Stop the current capture capture. If there is no current capture, this method will succeed.</td>
</tr>
<tr>
<td><strong>ToString</strong> (inherited from Object)</td>
<td>Returns a <strong>String</strong> that represents the current <strong>Object</strong>.</td>
</tr>
</tbody>
</table>

**Public Instance Events**

<table>
<thead>
<tr>
<th>Event</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>CaptureComplete</strong></td>
<td>Fired when a capture is completed (manually or automatically).</td>
</tr>
</tbody>
</table>

**See Also**

[Capture Class] [DirectX.Capture Namespace]

© 2002 Brian Low. All rights reserved.

Generated from assembly DirectX.Capture [1.0.1122.23301]
DirectX.Capture Class Library
Capture Constructor

Create a new Capture object. videoDevice and audioDevice can be null if you do not wish to capture both audio and video. However at least one must be a valid device. Use the Filters class to list available devices.

```csharp
public Capture(
    Filter videoDevice,
    Filter audioDevice
);```

See Also

Capture Class | Capture Members | DirectX.Capture Namespace

© 2002 Brian Low. All rights reserved.

Generated from assembly DirectX.Capture [1.0.1122.23301]
DirectX.Capture Class Library
Capture Properties

The properties of the **Capture** class are listed below. For a complete list of **Capture** class members, see the [Capture Members](#) topic.

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AudioCaps</td>
<td>The capabilities of the audio device.</td>
</tr>
<tr>
<td>AudioChannels</td>
<td>Get or set the number of channels in the waveform-audio data.</td>
</tr>
<tr>
<td>AudioCompressor</td>
<td>The audio compression filter.</td>
</tr>
<tr>
<td>AudioDevice</td>
<td>The audio capture device filter. Read-only. To use a different device, dispose of the current Capture instance and create a new instance with the desired device.</td>
</tr>
<tr>
<td>AudioSampleSize</td>
<td>Get or set the number of bits recorded per sample.</td>
</tr>
<tr>
<td>AudioSamplingRate</td>
<td>Get or set the number of audio samples taken per second.</td>
</tr>
<tr>
<td>AudioSource</td>
<td>The current audio source. Use Capture.AudioSources to list available sources. Set to null to disable all sources (mute).</td>
</tr>
<tr>
<td>AudioSources</td>
<td>Collection of available audio sources/physical connectors on the current audio device.</td>
</tr>
<tr>
<td>Capturing</td>
<td>Is the class currently capturing. Read-only.</td>
</tr>
<tr>
<td>Cued</td>
<td>Has the class been cued to begin capturing. Read-only.</td>
</tr>
<tr>
<td>Property</td>
<td>Description</td>
</tr>
<tr>
<td>------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td><strong>Filename</strong></td>
<td>Name of file to capture to. Initially set to a valid temporary file.</td>
</tr>
<tr>
<td><strong>FrameRate</strong></td>
<td>Gets and sets the frame rate used to capture video.</td>
</tr>
<tr>
<td><strong>FrameSize</strong></td>
<td>Gets and sets the frame size used to capture video.</td>
</tr>
<tr>
<td><strong>PreviewWindow</strong></td>
<td>The control that will host the preview window.</td>
</tr>
<tr>
<td><strong>PropertyPages</strong></td>
<td>Available property pages.</td>
</tr>
<tr>
<td><strong>Stopped</strong></td>
<td>Is the class currently stopped. Read-only.</td>
</tr>
<tr>
<td><strong>Tuner</strong></td>
<td>The TV Tuner or null if the current video device does not have a TV Tuner.</td>
</tr>
<tr>
<td><strong>VideoCaps</strong></td>
<td>The capabilities of the video device.</td>
</tr>
<tr>
<td><strong>VideoCompressor</strong></td>
<td>The video compression filter. When this property is changed the internal filter graph is rebuilt. This means that some properties will be reset. Set this property as early as possible to avoid losing changes. This property cannot be changed while capturing.</td>
</tr>
<tr>
<td><strong>VideoDevice</strong></td>
<td>The video capture device filter. Read-only. To use a different device, dispose of the current Capture instance and create a new instance with the desired device.</td>
</tr>
<tr>
<td><strong>VideoSource</strong></td>
<td>The current video source. Use</td>
</tr>
</tbody>
</table>
Capture.VideoSources to list available sources. Set to null to disable all sources (mute).

**VideoSources**

Collection of available video sources/physical connectors on the current video device.

See Also

[Capture Class] [DirectX.Capture Namespace]

© 2002 Brian Low. All rights reserved.

Generated from assembly DirectX.Capture [1.0.1122.23301]
DirectX.Capture Class Library
Capture.AudioCaps Property

The capabilities of the audio device.

```csharp
public AudioCapabilities AudioCaps {get;}
```

Remarks

It may be required to cue the capture (see Cue) before all capabilities are correctly reported. If you have such a device, the developer would be interested to hear from you.

The information contained in this property is retrieved and cached the first time this property is accessed. Future calls to this property use the cached results. This was done for performance.

However, this means you may get different results depending on when you access this property first. If you are experiencing problems, try accessing the property immediately after creating the Capture class or immediately after setting the video and audio compressors. Also, inform the developer.

See Also

Capture Class | Capture Members | DirectX.Capture Namespace

© 2002 Brian Low. All rights reserved.

Generated from assembly DirectX.Capture [1.0.1122.23301]
DirectX.Capture Class Library
**Capture.AudioChannels Property**

Get or set the number of channels in the waveform-audio data.

```csharp
public short AudioChannels {get; set;}
```

**Remarks**

Monaural data uses one channel and stereo data uses two channels.

Not all devices support getting/setting this property. If this property is not supported, accessing it will throw an exception.

This property cannot be changed while capturing. Changing this property while preview is enabled will cause some flickering while the internal filter graph is partially rebuilt. Changing this property while cued will cancel the cue. Call Cue() again to re-cue the capture.

**See Also**

[Capture Class](#) | [Capture Members](#) | [DirectX.Capture Namespace](#)

© 2002 Brian Low. All rights reserved.

Generated from assembly DirectX.Capture [1.0.1122.23301]
DirectX.Capture Class Library
Capture.AudioCompressor Property

The audio compression filter.

```csharp
public Filter AudioCompressor {get; set;}
```

Remarks

When this property is changed the internal filter graph is rebuilt. This means that some properties will be reset. Set this property as early as possible to avoid losing changes. This property cannot be changed while capturing.

See Also

Capture Class | Capture Members | DirectX.Capture Namespace

© 2002 Brian Low. All rights reserved.

Generated from assembly DirectX.Capture [1.0.1122.23301]
DirectX.Capture Class Library
Capture.AudioDevice Property

The audio capture device filter. Read-only. To use a different device, dispose of the current Capture instance and create a new instance with the desired device.

```csharp
public Filter AudioDevice {get;}
```

See Also

- [Capture Class](#)
- [Capture Members](#)
- [DirectX.Capture Namespace](#)

© 2002 Brian Low. All rights reserved.

Generated from assembly DirectX.Capture [1.0.1122.23301]
Capture.AudioSampleSize Property

Get or set the number of bits recorded per sample.

```csharp
public short AudioSampleSize {get; set;}
```

Remarks

Common sample sizes are 8 bit and 16 bit. Not all samples sizes are supported.

Not all devices support getting/setting this property. If this property is not supported, accessing it will throw an exception.

This property cannot be changed while capturing. Changing this property while preview is enabled will cause some flickering while the internal filter graph is partially rebuilt. Changing this property while cued will cancel the cue. Call Cue() again to re-cue the capture.

See Also

- [Capture Class](#) | [Capture Members](#) | [DirectX.Capture Namespace](#)

© 2002 Brian Low. All rights reserved.

Generated from assembly DirectX.Capture [1.0.1122.23301]
DirectX.Capture Class Library
Capture.AudioSamplingRate Property

Get or set the number of audio samples taken per second.

```csharp
public int AudioSamplingRate {get; set;}
```

Remarks

Common sampling rates are 8.0 kHz, 11.025 kHz, 22.05 kHz, and 44.1 kHz. Not all sampling rates are supported.

Not all devices support getting/setting this property. If this property is not supported, accessing it will throw an exception.

This property cannot be changed while capturing. Changing this property while preview is enabled will cause some flickering while the internal filter graph is partially rebuilt. Changing this property while cued will cancel the cue. Call Cue() again to re-cue the capture.

See Also

Capture Class | Capture Members | DirectX.Capture Namespace

© 2002 Brian Low. All rights reserved.

Generated from assembly DirectX.Capture [1.0.1122.23301]
DirectX.Capture Class Library
Capture.AudioSource Property

The current audio source. Use Capture.AudioSources to list available sources. Set to null to disable all sources (mute).

```
public Source AudioSource {get; set;}
```

See Also

Capture Class | Capture Members | DirectX.Capture Namespace

© 2002 Brian Low. All rights reserved.

Generated from assembly DirectX.Capture [1.0.1122.23301]
DirectX.Capture Class Library
Collection of available audio sources/physical connectors on the current audio device.

```csharp
public SourceCollection AudioSources {get;}
```

Remarks

In most cases, if the device has only one source, this collection will be empty. For audio there are 2 different methods for enumerating audio sources an audio crossbar (usually TV tuners?) or an audio mixer (usually sound cards?). This class will first look for an audio crossbar. If no sources or only one source is available on the crossbar, this class will then look for an audio mixer. This class does not support both methods.

The information contained in this property is retrieved and cached the first time this property is accessed. Future calls to this property use the cached results. This was done for performance.

However, this means **you may get different results depending on when you access this property first**. If you are experiencing problems, try accessing the property immediately after creating the Capture class or immediately after setting the video and audio compressors. Also, inform the developer.

See Also

- [Capture Class](#)
- [Capture Members](#)
- [DirectX.Capture Namespace](#)

© 2002 Brian Low. All rights reserved.

Generated from assembly DirectX.Capture [1.0.1122.23301]
DirectX.Capture Class Library
Capture.Capturing Property

Is the class currently capturing. Read-only.

```csharp
public bool Capturing {get;}
```

See Also

Capture Class | Capture Members | DirectX.Capture Namespace

© 2002 Brian Low. All rights reserved.
Generated from assembly DirectX.Capture [1.0.1122.23301]
DirectX.Capture Class Library
Capture.Cued Property

Has the class been cued to begin capturing. Read-only.

```
public bool Cued {get;}
```

See Also

Capture Class | Capture Members | DirectX.Capture Namespace

© 2002 Brian Low. All rights reserved.

Generated from assembly DirectX.Capture [1.0.1122.23301]
DirectX.Capture Class Library
Capture.Filename Property

Name of file to capture to. Initially set to a valid temporary file.

public string Filename {get; set;}

Remarks

If the file does not exist, it will be created. If it does exist, it will be overwritten. An overwritten file will not be shortened if the captured data is smaller than the original file. The file will be valid, it will just contain extra, unused, data after the audio/video data.

A future version of this class will provide a method to copy only the valid audio/video data to a new file.

This property cannot be changed while capturing or cued.

See Also

Capture Class | Capture Members | DirectX.Capture Namespace

© 2002 Brian Low. All rights reserved.

Generated from assembly DirectX.Capture [1.0.1122.23301]
DirectX.Capture Class Library
Capture.FrameRate Property

Gets and sets the frame rate used to capture video.

```csharp
public double FrameRate {get; set;}
```

Remarks

Common frame rates: 24 fps for film, 25 for PAL, 29.997 for NTSC. Not all NTSC capture cards can capture at exactly 29.997 fps. Not all frame rates are supported. When changing the frame rate, the closest supported frame rate will be used.

Not all devices support getting/setting this property. If this property is not supported, accessing it will throw an exception.

This property cannot be changed while capturing. Changing this property while preview is enabled will cause some flickering while the internal filter graph is partially rebuilt. Changing this property while cued will cancel the cue. Call Cue() again to re-cue the capture.

See Also

Capture Class | Capture Members | DirectX.Capture Namespace

© 2002 Brian Low. All rights reserved.

Generated from assembly DirectX.Capture [1.0.1122.23301]
## DirectX.Capture Class Library
Capture.FrameSize Property

Gets and sets the frame size used to capture video.

```csharp
public System.Drawing.Size FrameSize {get; set;}
```

Remarks

To change the frame size, assign a new Size object to this property:
```csharp
capture.Size = new Size( w, h );
```
rather than modifying the size in place (capture.Size.Width = w;). Not all frame rates are supported.

Not all devices support getting/setting this property. If this property is not supported, accessing it will throw an exception.

This property cannot be changed while capturing. Changing this property while preview is enabled will cause some flickering while the internal filter graph is partially rebuilt. Changing this property while cued will cancel the cue. Call Cue() again to re-cue the capture.

See Also

Capture Class | Capture Members | DirectX.Capture Namespace

© 2002 Brian Low. All rights reserved.

Generated from assembly DirectX.Capture [1.0.1122.23301]
DirectX.Capture Class Library
Capture.PreviewWindow Property

The control that will host the preview window.

```csharp
public System.Windows.Forms.Control PreviewWindow {get; set;}
```

Remarks

Setting this property will begin video preview immediately. Set this property after setting all other properties to avoid unnecessary changes to the internal filter graph (some properties like FrameSize require the internal filter graph to be stopped and disconnected before the property can be retrieved or set).

To stop video preview, set this property to null.

See Also

Capture Class | Capture Members | DirectX.Capture Namespace

© 2002 Brian Low. All rights reserved.

Generated from assembly DirectX.Capture [1.0.1122.23301]
DirectX.Capture Class Library
Capture.PropertyPages Property

Available property pages.

```csharp
public PropertyPageCollection PropertyPages { get; }
```

Remarks

These are property pages exposed by the DirectShow filters. These property pages allow users modify settings on the filters directly.

The information contained in this property is retrieved and cached the first time this property is accessed. Future calls to this property use the cached results. This was done for performance.

However, this means you may get different results depending on when you access this property first. If you are experiencing problems, try accessing the property immediately after creating the Capture class or immediately after setting the video and audio compressors. Also, inform the developer.

See Also

Capture Class | Capture Members | DirectX.Capture Namespace

© 2002 Brian Low. All rights reserved.

Generated from assembly DirectX.Capture [1.0.1122.23301]
DirectX.Capture Class Library
Capture.Stopped Property

Is the class currently stopped. Read-only.

```csharp
public bool Stopped {get;}
```

See Also

Capture Class | Capture Members | DirectX.Capture Namespace

© 2002 Brian Low. All rights reserved.

Generated from assembly DirectX.Capture [1.0.1122.23301]
DirectX.Capture Class Library
Capture.Tuner Property

The TV Tuner or null if the current video device does not have a TV Tuner.

```csharp
public Tuner Tuner {get;}
```

See Also

Capture Class | Capture Members | DirectX.Capture Namespace

© 2002 Brian Low. All rights reserved.

Generated from assembly DirectX.Capture [1.0.1122.23301]
DirectX.Capture Class Library
The capabilities of the video device.

```csharp
public VideoCapabilities VideoCaps {get;}
```

**Remarks**

It may be required to cue the capture (see `Cue`) before all capabilities are correctly reported. If you have such a device, the developer would be interested to hear from you.

The information contained in this property is retrieved and cached the first time this property is accessed. Future calls to this property use the cached results. This was done for performance.

However, this means you may get different results depending on when you access this property first. If you are experiencing problems, try accessing the property immediately after creating the Capture class or immediately after setting the video and audio compressors. Also, inform the developer.

**See Also**

[Capture Class](#) | [Capture Members](#) | [DirectX.Capture Namespace](#)

© 2002 Brian Low. All rights reserved.

Generated from assembly DirectX.Capture [1.0.1122.23301]
DirectX.Capture Class Library
The video compression filter. When this property is changed the internal filter graph is rebuilt. This means that some properties will be reset. Set this property as early as possible to avoid losing changes. This property cannot be changed while capturing.

```csharp
public Filter VideoCompressor {get; set;}
```

See Also

[Capture Class](#) | [Capture Members](#) | [DirectX.Capture Namespace](#)

© 2002 Brian Low. All rights reserved.

Generated from assembly DirectX.Capture [1.0.1122.23301]
DirectX.Capture Class Library
Capture.VideoDevice Property

The video capture device filter. Read-only. To use a different device, dispose of the current Capture instance and create a new instance with the desired device.

```csharp
public Filter VideoDevice {get;}
```

See Also

Capture Class | Capture Members | DirectX.Capture Namespace

© 2002 Brian Low. All rights reserved.

Generated from assembly DirectX.Capture [1.0.1122.23301]
DirectX.Capture Class Library
**Capture.VideoSource Property**

The current video source. Use Capture.VideoSources to list available sources. Set to null to disable all sources (mute).

```
public Source VideoSource {get; set;}
```

See Also

- [Capture Class](#)
- [Capture Members](#)
- [DirectX.Capture Namespace](#)

© 2002 Brian Low. All rights reserved.

Generated from assembly DirectX.Capture [1.0.1122.23301]
DirectX.Capture Class Library
Capture.VideoSources Property

Collection of available video sources/physical connectors on the current video device.

```csharp
public SourceCollection VideoSources {get;}
```

Remarks
In most cases, if the device has only one source, this collection will be empty.

The information contained in this property is retrieved and cached the first time this property is accessed. Future calls to this property use the cached results. This was done for performance.

However, this means you may get different results depending on when you access this property first. If you are experiencing problems, try accessing the property immediately after creating the Capture class or immediately after setting the video and audio compressors. Also, inform the developer.

See Also

Capture Class | Capture Members | DirectX.Capture Namespace

© 2002 Brian Low. All rights reserved.

Generated from assembly DirectX.Capture [1.0.1122.23301]
| DirectX.Capture Class Library |
Capture Methods

The methods of the **Capture** class are listed below. For a complete list of **Capture** class members, see the **Capture Members** topic.

Public Instance Methods

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Cue</strong></td>
<td>Prepare for capturing. Use this method when capturing must begin as quickly as possible.</td>
</tr>
<tr>
<td><strong>Dispose</strong></td>
<td>Calls Stop, releases all references. If a capture is in progress it will be stopped, but the CaptureComplete event will NOT fire.</td>
</tr>
<tr>
<td><strong>Equals</strong> (inherited from <strong>Object</strong>)</td>
<td>Determines whether the specified <strong>Object</strong> is equal to the current <strong>Object</strong>.</td>
</tr>
<tr>
<td><strong>GetHashCode</strong> (inherited from <strong>Object</strong>)</td>
<td>Serves as a hash function for a particular type, suitable for use in hashing algorithms and data structures like a hash table.</td>
</tr>
<tr>
<td><strong>GetType</strong> (inherited from <strong>Object</strong>)</td>
<td>Gets the <strong>Type</strong> of the current instance.</td>
</tr>
<tr>
<td><strong>Start</strong></td>
<td>Begin capturing.</td>
</tr>
<tr>
<td><strong>Stop</strong></td>
<td>Stop the current capture. If there is no current capture, this method will succeed.</td>
</tr>
<tr>
<td><strong>ToString</strong> (inherited from <strong>Object</strong>)</td>
<td>Returns a <strong>String</strong> that represents the current <strong>Object</strong>.</td>
</tr>
</tbody>
</table>

See Also

- [Capture Class](#)
- [DirectX.Capture Namespace](#)
DirectX.Capture Class Library
Capture.Cue Method

Prepare for capturing. Use this method when capturing must begin as quickly as possible.

```csharp
public void Cue();
```

Remarks

This will create/overwrite a zero byte file with the name set in the Filename property.

This will disable preview. Preview will resume once capture begins. This problem can be fixed if someone is willing to make the change.

This method is optional. If Cue() is not called, Start() will call it before capturing. This method cannot be called while capturing.

See Also

[Capture Class](#) | [Capture Members](#) | [DirectX.Capture Namespace](#)

© 2002 Brian Low. All rights reserved.

Generated from assembly DirectX.Capture [1.0.1122.23301]
**DirectX.Capture Class Library**
Capture.Dispose Method

Calls Stop, releases all references. If a capture is in progress it will be stopped, but the CaptureComplete event will NOT fire.

```csharp
public void Dispose();
```

See Also

- [Capture Class](#)
- [Capture Members](#)
- [DirectX.Capture Namespace](#)

© 2002 Brian Low. All rights reserved.

Generated from assembly DirectX.Capture [1.0.1122.23301]
Capture.Start Method

Begin capturing.

```csharp
public void Start();
```

See Also

- [Capture Class](#)
- [Capture Members](#)
- [DirectX.Capture Namespace](#)

© 2002 Brian Low. All rights reserved.

Generated from assembly DirectX.Capture [1.0.1122.23301]
DirectX.Capture Class Library
Capture.Stop Method

Stop the current capture. If there is no current capture, this method will succeed.

```csharp
public void Stop();
```

See Also

[Capture Class](#) | [Capture Members](#) | [DirectX.Capture Namespace](#)
DirectX.Capture Class Library
The events of the **Capture** class are listed below. For a complete list of **Capture** class members, see the **Capture Members** topic.

### Public Instance Events

<table>
<thead>
<tr>
<th>Event</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="#" alt="CaptureComplete" /></td>
<td>Fired when a capture is completed (manually or automatically).</td>
</tr>
</tbody>
</table>

**See Also**

[Capture Class](#) | [DirectX.Capture Namespace](#)

© 2002 Brian Low. All rights reserved.

Generated from assembly DirectX.Capture [1.0.1122.23301]
DirectX.Capture Class Library
Capture.CaptureComplete Event

Fired when a capture is completed (manually or automatically).

```csharp
public event EventHandler CaptureComplete;
```

See Also

Capture Class | Capture Members | DirectX.Capture Namespace

© 2002 Brian Low. All rights reserved.

Generated from assembly DirectX.Capture [1.0.1122.23301]
CrossbarSource Class

Represents a physical connector or source on an audio/video device. This class is used on filters that support the IAMCrossbar interface such as TV Tuners.

For a list of all members of this type, see CrossbarSource Members.

System.Object   Source
    CrossbarSource

public class CrossbarSource : Source

Requirements

Namespace: DirectX.Capture Namespace

Assembly: DirectX.Capture.dll

See Also

CrossbarSource Members | DirectX.Capture Namespace

© 2002 Brian Low. All rights reserved.

Generated from assembly DirectX.Capture [1.0.1122.23301]
DirectX.Capture Class Library
## CrossbarSource Members

### Public Instance Properties

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>Enabled</code></td>
<td>Enabled or disable this source.</td>
</tr>
<tr>
<td><code>Name</code></td>
<td>The name of the source. Read-only.</td>
</tr>
</tbody>
</table>

### Public Instance Methods

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>Dispose</code></td>
<td>Release unmanaged resources.</td>
</tr>
<tr>
<td><code>Equals</code></td>
<td>Determines whether the specified <code>Object</code> is equal to the current <code>Object</code>.</td>
</tr>
<tr>
<td><code>GetHashCode</code></td>
<td>Serves as a hash function for a particular type, suitable for use in hashing algorithms and data structures like a hash table.</td>
</tr>
<tr>
<td><code>GetType</code></td>
<td>Gets the <code>Type</code> of the current instance.</td>
</tr>
<tr>
<td><code>ToString</code></td>
<td>Obtains the String representation of this instance.</td>
</tr>
</tbody>
</table>

### See Also

- [CrossbarSource Class](#)  
- [DirectX.Capture Namespace](#)  

© 2002 Brian Low. All rights reserved.

Generated from assembly DirectX.Capture [1.0.1122.23301]
DirectX.Capture Class Library
## CrossbarSource Properties

The properties of the `CrossbarSource` class are listed below. For a complete list of `CrossbarSource` class members, see the [CrossbarSource Members](#) topic.

**Public Instance Properties**

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Enabled</td>
<td>Enabled or disable this source.</td>
</tr>
<tr>
<td>Name</td>
<td>The name of the source. Read-only.</td>
</tr>
</tbody>
</table>

**See Also**

[CrossbarSource Class](#) | [DirectX.Capture Namespace](#)

© 2002 Brian Low. All rights reserved.

Generated from assembly DirectX.Capture [1.0.1122.23301]
DirectX.Capture Class Library
CrossbarSource.Enabled Property

Enabled or disable this source.

```csharp
public override bool Enabled {get; set;}
```

See Also

CrossbarSource Class | CrossbarSource Members | DirectX.Capture Namespace

© 2002 Brian Low. All rights reserved.

Generated from assembly DirectX.Capture [1.0.1122.23301]
DirectX.Capture Class Library
The methods of the **CrossbarSource** class are listed below. For a complete list of **CrossbarSource** class members, see the **CrossbarSource Members** topic.

**Public Instance Methods**

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Dispose</strong></td>
<td>Release unmanaged resources.</td>
</tr>
<tr>
<td><strong>Equals</strong> (inherited from <strong>Object</strong>)</td>
<td>Determines whether the specified <strong>Object</strong> is equal to the current <strong>Object</strong>.</td>
</tr>
<tr>
<td><strong>GetHashCode</strong> (inherited from <strong>Object</strong>)</td>
<td>Serves as a hash function for a particular type, suitable for use in hashing algorithms and data structures like a hash table.</td>
</tr>
<tr>
<td><strong>GetType</strong> (inherited from <strong>Object</strong>)</td>
<td>Gets the <strong>Type</strong> of the current instance.</td>
</tr>
<tr>
<td><strong>ToString</strong> (inherited from <strong>Source</strong>)</td>
<td>Obtains the String representation of this instance.</td>
</tr>
</tbody>
</table>

**See Also**

[CrossbarSource Class] | [DirectX.Capture Namespace]

© 2002 Brian Low. All rights reserved.

Generated from assembly DirectX.Capture [1.0.1122.23301]
DirectX.Capture Class Library
CrossbarSource.Dispose Method

Release unmanaged resources.

```csharp
public override void Dispose();
```

Implements

`IDisposable.Dispose`

See Also

CrossbarSource Class | CrossbarSource Members | DirectX.Capture Namespace

© 2002 Brian Low. All rights reserved.

Generated from assembly DirectX.Capture [1.0.1122.23301]
DirectX.Capture Class Library
**DeviceInUseException Class**

Exception thrown when the device cannot be rendered or started.

For a list of all members of this type, see [DeviceInUseException Members](#).

```
System.Object  Exception
             SystemException
            DeviceInUseException
```

**public class DeviceInUseException : SystemException**

**Requirements**

- **Namespace:** [DirecX.Capture Namespace](#)
- **Assembly:** DirectX.Capture.dll

**See Also**

- [DeviceInUseException Members](#)  |  [DirecX.Capture Namespace](#)

© 2002 Brian Low. All rights reserved.

Generated from assembly DirectX.Capture [1.0.1122.23301]
## DeviceInUseException Members

### Public Instance Constructors

<table>
<thead>
<tr>
<th>Constructor</th>
</tr>
</thead>
<tbody>
<tr>
<td>DeviceInUseException Constructor</td>
</tr>
</tbody>
</table>

### Public Instance Properties

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>HelpLink (inherited from Exception)</td>
<td>Gets or sets a link to the help file associated with this exception.</td>
</tr>
<tr>
<td>InnerException (inherited from Exception)</td>
<td>Gets the Exception instance that caused the current exception.</td>
</tr>
<tr>
<td>Message (inherited from Exception)</td>
<td>Gets a message that describes the current exception.</td>
</tr>
<tr>
<td>Source (inherited from Exception)</td>
<td>Gets or sets the name of the application or the object that causes the error.</td>
</tr>
<tr>
<td>StackTrace (inherited from Exception)</td>
<td>Gets a string representation of the frames on the call stack at the time the current exception was thrown.</td>
</tr>
<tr>
<td>TargetSite (inherited from Exception)</td>
<td>Gets the method that throws the current exception.</td>
</tr>
</tbody>
</table>

### Public Instance Methods

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Equals (inherited from Object)</td>
<td>Determines whether the specified Object is equal to the current Object.</td>
</tr>
<tr>
<td>GetBaseException (inherited from Exception)</td>
<td>When overridden in a derived class, returns the Exception that is the root cause of one or more subsequent exceptions.</td>
</tr>
<tr>
<td>GetHashCode (inherited from</td>
<td>Serves as a hash function for a</td>
</tr>
<tr>
<td><strong>Object</strong></td>
<td>particular type, suitable for use in hashing algorithms and data structures like a hash table.</td>
</tr>
<tr>
<td>---</td>
<td>---</td>
</tr>
<tr>
<td><strong>GetObjectData</strong> (inherited from Exception)</td>
<td>When overridden in a derived class, sets the SerializationInfo with information about the exception.</td>
</tr>
<tr>
<td><strong>GetType</strong> (inherited from Object)</td>
<td>Gets the Type of the current instance.</td>
</tr>
<tr>
<td><strong>ToString</strong> (inherited from Exception)</td>
<td>Creates and returns a string representation of the current exception.</td>
</tr>
</tbody>
</table>

**See Also**

[DeviceInUseException Class] | [DirectX.Capture Namespace]  

© 2002 Brian Low. All rights reserved.  
Generated from assembly DirectX.Capture [1.0.1122.23301]
DirectX.Capture Class Library
DeviceInUseException Constructor

```csharp
public DeviceInUseException(
    string deviceName,
    int hResult
);
```

See Also

DeviceInUseException Class | DeviceInUseException Members | DirectX.Capture Namespace

© 2002 Brian Low. All rights reserved.

Generated from assembly DirectX.Capture [1.0.1122.23301]
DirectX.Capture Class Library
Property pages for a DirectShow filter (e.g. hardware device). These property pages do not support persisting their settings.

For a list of all members of this type, see DirectShowPropertyPage Members.

System.Object  PropertyPage

DirectShowPropertyPage

public class DirectShowPropertyPage : PropertyPage

Requirements

Namespace: DirectX.Capture Namespace
Assembly: DirectX.Capture.dll

See Also

DirectShowPropertyPage Members  |  DirectX.Capture Namespace

© 2002 Brian Low. All rights reserved.
Generated from assembly DirectX.Capture [1.0.1122.23301]
# DirectX.Capture Class Library
## DirectShowPropertyPage Members

### Public Instance Constructors

<table>
<thead>
<tr>
<th>Constructor</th>
</tr>
</thead>
<tbody>
<tr>
<td>DirectShowPropertyPage</td>
</tr>
</tbody>
</table>

### Public Instance Fields

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Name of property page. This name may not be unique.</td>
</tr>
<tr>
<td>SupportsPersisting</td>
<td>Does this property page support saving and loading the user's choices.</td>
</tr>
</tbody>
</table>

### Public Instance Properties

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>State</td>
<td>Get or set the state of the property page. This is used to save and restore the user's choices without redisplaying the property page.</td>
</tr>
</tbody>
</table>

### Public Instance Methods

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dispose</td>
<td>Release unmanaged resources.</td>
</tr>
<tr>
<td>Equals</td>
<td>Determines whether the specified Object is equal to the current Object.</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Serves as a hash function for a particular type, suitable for use in hashing algorithms and data structures like a hash table.</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the Type of the current instance.</td>
</tr>
<tr>
<td>Show</td>
<td>Show the property page. Some</td>
</tr>
<tr>
<td>Property</td>
<td>Description</td>
</tr>
<tr>
<td>----------</td>
<td>-------------</td>
</tr>
<tr>
<td><strong>ToString (inherited from Object)</strong></td>
<td>Returns a <strong>String</strong> that represents the current <strong>Object</strong>.</td>
</tr>
</tbody>
</table>

See Also

[DirectShowPropertyPage Class] | [DirectX.Capture Namespace]

© 2002 Brian Low. All rights reserved.

Generated from assembly DirectX.Capture [1.0.1122.23301]
DirectX.Capture Class Library
DirectShowPropertyPage Constructor

Constructor

```csharp
public DirectShowPropertyPage(
    string name,
    ISpecifyPropertyPages specifyPropertyPages
);
```

See Also

[DirectShowPropertyPage Class] | [DirectShowPropertyPage Members] | [DirectX.Capture Namespace]

© 2002 Brian Low. All rights reserved.

Generated from assembly DirectX.Capture [1.0.1122.23301]
DirectX.Capture Class Library
### DirectShowPropertyPage Methods

The methods of the `DirectShowPropertyPage` class are listed below. For a complete list of `DirectShowPropertyPage` class members, see the [DirectShowPropertyPage Members](#) topic.

#### Public Instance Methods

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>Dispose</code></td>
<td>Release unmanaged resources</td>
</tr>
<tr>
<td><code>Equals</code> (inherited from <code>Object</code>)</td>
<td>Determines whether the specified <code>Object</code> is equal to the current <code>Object</code>.</td>
</tr>
<tr>
<td><code>GetHashCode</code> (inherited from <code>Object</code>)</td>
<td>Serves as a hash function for a particular type, suitable for use in hashing algorithms and data structures like a hash table.</td>
</tr>
<tr>
<td><code>GetType</code> (inherited from <code>Object</code>)</td>
<td>Gets the <code>Type</code> of the current instance.</td>
</tr>
<tr>
<td><code>Show</code></td>
<td>Show the property page. Some property pages cannot be displayed while previewing and/or capturing.</td>
</tr>
<tr>
<td><code>ToString</code> (inherited from <code>Object</code>)</td>
<td>Returns a <code>String</code> that represents the current <code>Object</code>.</td>
</tr>
</tbody>
</table>

#### See Also

- [DirectShowPropertyPage Class](#) | [DirectX.Capture Namespace](#)
| DirectX.Capture Class Library |
DirectShowPropertyPage.Dispose Method

Release unmanaged resources

```csharp
new public void Dispose();
```

Implements

- IDisposable.Dispose

See Also

- DirectShowPropertyPage Class
- DirectShowPropertyPage Members
- DirectX.Capture Namespace

© 2002 Brian Low. All rights reserved.

Generated from assembly DirectX.Capture [1.0.1122.23301]
DirectX.Capture Class Library
### DirectShowPropertyPage.Show Method

Show the property page. Some property pages cannot be displayed while previewing and/or capturing.

```csharp
public override void Show(
    Control owner
);
```

See Also

- [DirectShowPropertyPage Class](#)
- [DirectShowPropertyPage Members](#)
- [DirectX.Capture Namespace](#)

© 2002 Brian Low. All rights reserved.

Generated from assembly DirectX.Capture [1.0.1122.23301]
DirectX.Capture Class Library
Filter Class

Represents a DirectShow filter (e.g. video capture device, compression codec).

For a list of all members of this type, see Filter Members.

System.Object  Filter

public class Filter : IComparable

Remarks

To save a chosen filter for later recall save the MonikerString property on the filter:

```
string savedMonikerString = myFilter.MonikerString;
```

To recall the filter create a new Filter class and pass the string to the constructor:

```
Filter mySelectedFilter = new Filter(savedMonikerString);
```

Requirements

Namespace: DirectX.Capture Namespace
Assembly: DirectX.Capture.dll

See Also

Filter Members  |  DirectX.Capture Namespace

© 2002 Brian Low. All rights reserved.
Generated from assembly DirectX.Capture [1.0.1122.23301]
DirectX.Capture Class Library
# Filter Members

## Public Instance Constructors

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Filter Constructor</td>
<td>Create a new filter from its moniker string.</td>
</tr>
</tbody>
</table>

## Public Instance Fields

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MonikerString</td>
<td>Unique string referencing this filter. This string can be used to recreate this filter.</td>
</tr>
<tr>
<td>Name</td>
<td>Human-readable name of the filter</td>
</tr>
</tbody>
</table>

## Public Instance Methods

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CompareTo</td>
<td>Compares the current instance with another object of the same type.</td>
</tr>
<tr>
<td>Equals (inherited from Object)</td>
<td>Determines whether the specified Object is equal to the current Object.</td>
</tr>
<tr>
<td>GetHashCode (inherited from Object)</td>
<td>Serves as a hash function for a particular type, suitable for use in hashing algorithms and data structures like a hash table.</td>
</tr>
<tr>
<td>GetType (inherited from Object)</td>
<td>Gets the Type of the current instance.</td>
</tr>
<tr>
<td>ToString (inherited from Object)</td>
<td>Returns a String that represents the current Object.</td>
</tr>
</tbody>
</table>

## See Also

[Filter Class](#) | [DirectX.Capture Namespace](#)
DirectX.Capture Class Library
Filter Constructor

Create a new filter from its moniker string.

```csharp
public Filter(
    string monikerString
);
```

See Also

Filter Class | Filter Members | DirectX.Capture Namespace

© 2002 Brian Low. All rights reserved.

Generated from assembly DirectX.Capture [1.0.1122.23301]
DirectX.Capture Class Library
Filter Fields

The fields of the Filter class are listed below. For a complete list of Filter class members, see the Filter Members topic.

Public Instance Fields

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MonikerString</td>
<td>Unique string referencing this filter. This string can be used to recreate this filter.</td>
</tr>
<tr>
<td>Name</td>
<td>Human-readable name of the filter</td>
</tr>
</tbody>
</table>

See Also

Filter Class | DirectX.Capture Namespace

© 2002 Brian Low. All rights reserved.
Generated from assembly DirectX.Capture [1.0.1122.23301]
DirectX.Capture Class Library
Filter.MonikerString Field

Unique string referencing this filter. This string can be used to recreate this filter.

```csharp
public string MonikerString;
```

See Also

Filter Class | Filter Members | DirectX.Capture Namespace

© 2002 Brian Low. All rights reserved.

Generated from assembly DirectX.Capture [1.0.1122.23301]
DirectX.Capture Class Library
Filter.Name Field

Human-readable name of the filter

```csharp
public string Name;
```

See Also

Filter Class | Filter Members | DirectX.Capture Namespace

© 2002 Brian Low. All rights reserved.

Generated from assembly DirectX.Capture [1.0.1122.23301]
DirectX.Capture Class Library
Filter Methods

The methods of the **Filter** class are listed below. For a complete list of **Filter** class members, see the [Filter Members](#) topic.

Public Instance Methods

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>CompareTo</code></td>
<td>Compares the current instance with another object of the same type.</td>
</tr>
<tr>
<td><code>Equals</code></td>
<td>Determines whether the specified <a href="#">Object</a> is equal to the current <a href="#">Object</a>.</td>
</tr>
<tr>
<td><code>GetHashCode</code></td>
<td>Serves as a hash function for a particular type, suitable for use in hashing algorithms and data structures like a hash table.</td>
</tr>
<tr>
<td><code>GetType</code></td>
<td>Gets the <a href="#">Type</a> of the current instance.</td>
</tr>
<tr>
<td><code>ToString</code></td>
<td>Returns a <a href="#">String</a> that represents the current <a href="#">Object</a>.</td>
</tr>
</tbody>
</table>

See Also

[Filter Class](#) | [DirectX.Capture Namespace](#)

© 2002 Brian Low. All rights reserved.

Generated from assembly DirectX.Capture [1.0.1122.23301]
DirectorX.Capture Class Library
Filter.CompareTo Method

Compares the current instance with another object of the same type.

```csharp
public int CompareTo(object obj);
```

Implements

IComparable.CompareTo

See Also

Filter Class | Filter Members | DirectX.Capture Namespace

© 2002 Brian Low. All rights reserved.

Generated from assembly DirectX.Capture [1.0.1122.23301]
DirectX.Capture Class Library
FilterCollection Class

A collection of Filter objects (DirectShow filters). This is used by the Capture class to provide lists of capture devices and compression filters. This class cannot be created directly.

For a list of all members of this type, see FilterCollection Members.

System.Object  CollectionBase
  FilterCollection

public class FilterCollection : CollectionBase

Requirements

Namespace: DirectX.Capture Namespace
Assembly: DirectX.Capture.dll

See Also

FilterCollection Members  |  DirectX.Capture Namespace

© 2002 Brian Low. All rights reserved.
Generated from assembly DirectX.Capture [1.0.1122.23301]
DirectX.Capture Class Library
## FilterCollection Members

### Public Instance Properties

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>Count</code> (inherited from <code>CollectionBase</code>)</td>
<td>Gets the number of elements contained in the <code>CollectionBase</code> instance.</td>
</tr>
<tr>
<td><code>Item</code></td>
<td>Get the filter at the specified index.</td>
</tr>
</tbody>
</table>

### Public Instance Methods

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>Clear</code> (inherited from <code>CollectionBase</code>)</td>
<td>Removes all objects from the <code>CollectionBase</code> instance.</td>
</tr>
<tr>
<td><code>Equals</code> (inherited from <code>Object</code>)</td>
<td>Determines whether the specified <code>Object</code> is equal to the current <code>Object</code>.</td>
</tr>
<tr>
<td><code>GetEnumerator</code> (inherited from <code>CollectionBase</code>)</td>
<td>Returns an enumerator that can iterate through the <code>CollectionBase</code> instance.</td>
</tr>
<tr>
<td><code>GetHashCode</code> (inherited from <code>Object</code>)</td>
<td>Serves as a hash function for a particular type, suitable for use in hashing algorithms and data structures like a hash table.</td>
</tr>
<tr>
<td><code>GetType</code> (inherited from <code>Object</code>)</td>
<td>Gets the <code>Type</code> of the current instance.</td>
</tr>
<tr>
<td><code>RemoveAt</code> (inherited from <code>CollectionBase</code>)</td>
<td>Removes the element at the specified index of the <code>CollectionBase</code> instance.</td>
</tr>
<tr>
<td><code>ToString</code> (inherited from <code>Object</code>)</td>
<td>Returns a <code>String</code> that represents the current <code>Object</code>.</td>
</tr>
</tbody>
</table>

### Explicit Interface Implementations

<table>
<thead>
<tr>
<th>Interface</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>ICollection.CopyTo</code> (inherited)</td>
<td></td>
</tr>
<tr>
<td>Method</td>
<td>Description</td>
</tr>
<tr>
<td>--------------------</td>
<td>----------------------------------</td>
</tr>
<tr>
<td><code>IList.Add</code></td>
<td>(inherited from CollectionBase)</td>
</tr>
<tr>
<td><code>IList.Contains</code></td>
<td>(inherited from CollectionBase)</td>
</tr>
<tr>
<td><code>IList.IndexOf</code></td>
<td>(inherited from CollectionBase)</td>
</tr>
<tr>
<td><code>IList.Insert</code></td>
<td>(inherited from CollectionBase)</td>
</tr>
<tr>
<td><code>IList.Remove</code></td>
<td>(inherited from CollectionBase)</td>
</tr>
</tbody>
</table>

See Also

[FilterCollection Class](/FilterCollection Class) | [DirectX.Capture Namespace](/DirectX.Capture Namespace)

© 2002 Brian Low. All rights reserved.
Generated from assembly DirectX.Capture [1.0.1122.23301]
FilterCollection Properties

The properties of the FilterCollection class are listed below. For a complete list of FilterCollection class members, see the FilterCollection Members topic.

Public Instance Properties

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Count</td>
<td>(inherited from CollectionBase) Gets the number of elements contained in the CollectionBase instance.</td>
</tr>
<tr>
<td>Item</td>
<td>Get the filter at the specified index.</td>
</tr>
</tbody>
</table>

See Also

FilterCollection Class | DirectX.Capture Namespace

© 2002 Brian Low. All rights reserved.
Generated from assembly DirectX.Capture [1.0.1122.23301]
DirectX.Capture Class Library
FilterCollection.Item Property

Get the filter at the specified index.

```csharp
public Filter this[int index] {get;}
```

See Also

FilterCollection Class | FilterCollection Members | DirectX.Capture Namespace

© 2002 Brian Low. All rights reserved.

Generated from assembly DirectX.Capture [1.0.1122.23301]
DirectX.Capture Class Library
Filters Class

Provides collections of devices and compression codecs installed on the system.

For a list of all members of this type, see Filters Members.

Example

Devices and compression codecs are implemented in DirectShow as filters, see the Filter class for more information. To list the available video devices:

```csharp
Filters filters = new Filters();
foreach (Filter f in filters.VideoInputDevices)
{
    Debug.WriteLine(f.Name);
}
```

Requirements

Namespace: DirectX.Capture Namespace

Assembly: DirectX.Capture.dll

See Also

Filters Members | DirectX.Capture Namespace | Filter

© 2002 Brian Low. All rights reserved.

Generated from assembly DirectX.Capture [1.0.1122.23301]
DirectX.Capture Class Library
# Filters Members

## Public Instance Constructors

| ❯ Filters Constructor | Initializes a new instance of the Filters class. |

## Public Instance Fields

| ❯ AudioCompressors | Collection of available audio compressors. |
| ❯ AudioInputDevices | Collection of available audio capture devices. |
| ❯ VideoCompressors | Collection of available video compressors. |
| ❯ VideoInputDevices | Collection of available video capture devices. |

## Public Instance Methods

| ❯ Equals (inherited from Object) | Determines whether the specified Object is equal to the current Object. |
| ❯ GetHashCode (inherited from Object) | Serves as a hash function for a particular type, suitable for use in hashing algorithms and data structures like a hash table. |
| ❯ GetType (inherited from Object) | Gets the Type of the current instance. |
| ❯ ToString (inherited from Object) | Returns a String that represents the current Object. |

## See Also

Filters Class | DirectX.Capture Namespace | Filter
DirectX.Capture Class Library
Initializes a new instance of the Filters class.

```csharp
public Filters();
```

See Also

Filters Class | Filters Members | DirectX.Capture Namespace

© 2002 Brian Low. All rights reserved.

Generated from assembly DirectX.Capture [1.0.1122.23301]
DirectX.Capture Class Library
The fields of the **Filters** class are listed below. For a complete list of **Filters** class members, see the **Filters Members** topic.

### Public Instance Fields

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>AudioCompressors</strong></td>
<td>Collection of available audio compressors.</td>
</tr>
<tr>
<td><strong>AudioInputDevices</strong></td>
<td>Collection of available audio capture devices.</td>
</tr>
<tr>
<td><strong>VideoCompressors</strong></td>
<td>Collection of available video compressors.</td>
</tr>
<tr>
<td><strong>VideoInputDevices</strong></td>
<td>Collection of available video capture devices.</td>
</tr>
</tbody>
</table>

**See Also**

- [Filters Class](#) | [DirectX.Capture Namespace](#) | [Filter](#)

© 2002 Brian Low. All rights reserved.

Generated from assembly DirectX.Capture [1.0.1122.23301]
DirectX.Capture Class Library
Filters.AudioCompressors Field

Collection of available audio compressors.

```csharp
public FilterCollection AudioCompressors;
```

See Also

[Filters Class](#) | [Filters Members](#) | [DirectX.Capture Namespace](#)

© 2002 Brian Low. All rights reserved.

Generated from assembly DirectX.Capture [1.0.1122.23301]
DirectX.Capture Class Library
Filters.AudioInputDevices Field

Collection of available audio capture devices.

```csharp
public FilterCollection AudioInputDevices;
```

See Also

- [Filters Class](#)
- [Filters Members](#)
- [DirectX.Capture Namespace](#)

© 2002 Brian Low. All rights reserved.

Generated from assembly DirectX.Capture [1.0.1122.23301]
DirectX.Capture Class Library
Filters.VideoCompressors Field

Collection of available video compressors.

```
public FilterCollection VideoCompressors;
```

See Also

Filters Class | Filters Members | DirectX.Capture Namespace

© 2002 Brian Low. All rights reserved.

Generated from assembly DirectX.Capture [1.0.1122.23301]
DirectX.Capture Class Library
Filters.VideoInputDevices Field

Collection of available video capture devices.

```
public FilterCollection VideoInputDevices;
```

See Also

[Filters Class] | [Filters Members] | [DirectX.Capture Namespace]

© 2002 Brian Low. All rights reserved.

Generated from assembly DirectX.Capture [1.0.1122.23301]
DirectX.Capture Class Library
PropertyPage Class

A base class for representing property pages exposed by filters. For a list of all members of this type, see PropertyPage Members.

System.Object  PropertyPage

public class PropertyPage : IDisposable

Requirements

Namespace: DirectX.Capture Namespace

Assembly: DirectX.Capture.dll

See Also

PropertyPage Members | DirectX.Capture Namespace

© 2002 Brian Low. All rights reserved.

Generated from assembly DirectX.Capture [1.0.1122.23301]
DirectX.Capture Class Library
### PropertyPage Members

#### Public Instance Constructors

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PropertyPage Constructor</td>
<td>Constructor</td>
</tr>
</tbody>
</table>

#### Public Instance Fields

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Name of property page. This name may not be unique</td>
</tr>
<tr>
<td>SupportsPersisting</td>
<td>Does this property page support saving and loading the user's choices.</td>
</tr>
</tbody>
</table>

#### Public Instance Properties

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>State</td>
<td>Get or set the state of the property page. This is used to save and restore the user's choices without redisplaying the property page.</td>
</tr>
</tbody>
</table>

#### Public Instance Methods

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dispose</td>
<td>Release unmanaged resources</td>
</tr>
<tr>
<td>Equals (inherited from Object)</td>
<td>Determines whether the specified Object is equal to the current Object.</td>
</tr>
<tr>
<td>GetHashCode (inherited from Object)</td>
<td>Serves as a hash function for a particular type, suitable for use in hashing algorithms and data structures like a hash table.</td>
</tr>
<tr>
<td>GetType (inherited from Object)</td>
<td>Gets the Type of the current instance.</td>
</tr>
<tr>
<td>Show</td>
<td>Show the property page. Some property pages cannot be</td>
</tr>
</tbody>
</table>
displayed while previewing and/or capturing. This method will block until the property page is closed by the user.

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>ToString</code> (inherited from <code>Object</code>)</td>
<td>Returns a <code>String</code> that represents the current <code>Object</code>.</td>
</tr>
</tbody>
</table>

See Also

[PropertyPage Class] | [DirectX.Capture Namespace]

© 2002 Brian Low. All rights reserved.

Generated from assembly DirectX.Capture [1.0.1122.23301]
DirectX.Capture Class Library
PropertyPage Constructor

Constructor

```public PropertyPage();```

See Also

[PropertyPage Class](#) | [PropertyPage Members](#) | [DirectX.Capture Namespace](#)

© 2002 Brian Low. All rights reserved.

Generated from assembly DirectX.Capture [1.0.1122.23301]
DirectX.Capture Class Library
PropertyPage Fields

The fields of the **PropertyPage** class are listed below. For a complete list of **PropertyPage** class members, see the [PropertyPage Members](#) topic.

**Public Instance Fields**

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>✷ Name</td>
<td>Name of property page. This name may not be unique</td>
</tr>
<tr>
<td>✷ SupportsPersisting</td>
<td>Does this property page support saving and loading the user's choices.</td>
</tr>
</tbody>
</table>

**See Also**

- [PropertyPage Class](#) | [DirectX.Capture Namespace](#)

© 2002 Brian Low. All rights reserved.

Generated from assembly DirectX.Capture [1.0.1122.23301]
DirectX.Capture Class Library
**PropertyPage.Name Field**

Name of property page. This name may not be unique

```csharp
public string Name;
```

See Also

[PropertyPage Class] [PropertyPage Members] [DirectX.Capture Namespace]

© 2002 Brian Low. All rights reserved.

Generated from assembly DirectX.Capture [1.0.1122.23301]
DirectX.Capture Class Library
PropertyPage SupportsPersisting Field

Does this property page support saving and loading the user's choices.

```csharp
public bool SupportsPersisting;
```

See Also

- [PropertyPage Class](#)
- [PropertyPage Members](#)
- [DirectX.Capture Namespace](#)

© 2002 Brian Low. All rights reserved.

Generated from assembly DirectX.Capture [1.0.1122.23301]
DirectX.Capture Class Library
PropertyPage Properties

The properties of the PropertyPage class are listed below. For a complete list of PropertyPage class members, see the PropertyPage Members topic.

Public Instance Properties

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>State</td>
<td>Get or set the state of the property page. This is used to save and restore the user's choices without redisplaying the property page.</td>
</tr>
</tbody>
</table>

See Also

PropertyPage Class | DirectX.Capture Namespace

© 2002 Brian Low. All rights reserved.
Generated from assembly DirectX.Capture [1.0.1122.23301]
DirectX.Capture Class Library
**PropertyPage.State Property**

Get or set the state of the property page. This is used to save and restore the user's choices without redisplaying the property page.

```csharp
public virtual byte[] State {get; set;}
```

**Remarks**

After showing this property page, read and store the value of this property. At a later time, the user's choices can be reloaded by setting this property with the value stored earlier. Note that some property pages, after setting this property, will not reflect the new state. However, the filter will use the new settings.

When reading this property, copy the entire array at once then manipulate your local copy (e.g. `byte[] myState = propertyPage.State`). When setting this property set the entire array at once (e.g. `propertyPage = myState`).

Not all property pages support saving/loading state. Check the [SupportsPersisting](#) property to determine if this property page supports it.

**See Also**

- [PropertyPage Class](#) | [PropertyPage Members](#) | [DirectX.Capture Namespace](#)
DirectX.Capture Class Library
The methods of the PropertyPage class are listed below. For a complete list of PropertyPage class members, see the PropertyPage Members topic.

### Public Instance Methods

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dispose</td>
<td>Release unmanaged resources</td>
</tr>
<tr>
<td>Equals (inherited from Object)</td>
<td>Determines whether the specified Object is equal to the current Object.</td>
</tr>
<tr>
<td>GetHashCode (inherited from Object)</td>
<td>Serves as a hash function for a particular type, suitable for use in hashing algorithms and data structures like a hash table.</td>
</tr>
<tr>
<td>GetType (inherited from Object)</td>
<td>Gets the Type of the current instance.</td>
</tr>
<tr>
<td>Show</td>
<td>Show the property page. Some property pages cannot be displayed while previewing and/or capturing. This method will block until the property page is closed by the user.</td>
</tr>
<tr>
<td>ToString (inherited from Object)</td>
<td>Returns a String that represents the current Object.</td>
</tr>
</tbody>
</table>

See Also

PropertyPage Class | DirectX.Capture Namespace

© 2002 Brian Low. All rights reserved.

Generated from assembly DirectX.Capture [1.0.1122.23301]
DirectX.Capture Class Library
PropertyPage.Dispose Method

Release unmanaged resources

```csharp
public void Dispose();
```

Implements

`IDisposable.Dispose`

See Also

PropertyPage Class | PropertyPage Members | DirectX.Capture Namespace

© 2002 Brian Low. All rights reserved.

Generated from assembly DirectX.Capture [1.0.1122.23301]
DirectX.Capture Class Library
**PropertyPage.Show Method**

Show the property page. Some property pages cannot be displayed while previewing and/or capturing. This method will block until the property page is closed by the user.

```csharp
public virtual void Show(
    Control owner
);
```

**See Also**

- [PropertyPage Class](#)
- [PropertyPage Members](#)
- [DirectX.Capture Namespace](#)

© 2002 Brian Low. All rights reserved.

Generated from assembly DirectX.Capture [1.0.1122.23301]
DirectX.Capture Class Library
PropertyPageCollection Class

A collection of available PropertyPages in a DirectShow filter graph. It is up to the driver manufacturer to implement a property pages on their drivers. The list of supported property pages will vary from driver to driver.

For a list of all members of this type, see PropertyPageCollection Members.

System.Object  CollectionBase  PropertyPageCollection

public class PropertyPageCollection : CollectionBase, IDisposable

Requirements

Namespace: DirectX.Capture Namespace
Assembly: DirectX.Capture.dll

See Also

PropertyPageCollection Members  DirectX.Capture Namespace

© 2002 Brian Low. All rights reserved.
Generated from assembly DirectX.Capture [1.0.1122.23301]
DirectX.Capture Class Library
### PropertyPageCollection Members

#### Public Instance Properties

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Count</strong> (inherited from CollectionBase)</td>
<td>Gets the number of elements contained in the CollectionBase instance.</td>
</tr>
<tr>
<td><strong>Item</strong></td>
<td>Get the filter at the specified index.</td>
</tr>
</tbody>
</table>

#### Public Instance Methods

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Clear</strong></td>
<td>Empty the collection.</td>
</tr>
<tr>
<td><strong>Dispose</strong></td>
<td>Release unmanaged resources</td>
</tr>
<tr>
<td><strong>Equals</strong> (inherited from Object)</td>
<td>Determines whether the specified Object is equal to the current Object.</td>
</tr>
<tr>
<td>** GetEnumerator** (inherited from CollectionBase)</td>
<td>Returns an enumerator that can iterate through the CollectionBase instance.</td>
</tr>
<tr>
<td><strong>GetHashCode</strong> (inherited from Object)</td>
<td>Serves as a hash function for a particular type, suitable for use in hashing algorithms and data structures like a hash table.</td>
</tr>
<tr>
<td><strong>GetType</strong> (inherited from Object)</td>
<td>Gets the Type of the current instance.</td>
</tr>
<tr>
<td><strong>RemoveAt</strong> (inherited from CollectionBase)</td>
<td>Removes the element at the specified index of the CollectionBase instance.</td>
</tr>
<tr>
<td><strong>ToString</strong> (inherited from Object)</td>
<td>Returns a String that represents the current Object.</td>
</tr>
</tbody>
</table>

#### Explicit Interface Implementations
ICollection.CopyTo (inherited from CollectionBase)

IList.Add (inherited from CollectionBase)

IList.Contains (inherited from CollectionBase)

IList.IndexOf (inherited from CollectionBase)

IList.Insert (inherited from CollectionBase)

IList.Remove (inherited from CollectionBase)

See Also

PropertyPageCollection Class | DirectX.Capture Namespace

© 2002 Brian Low. All rights reserved.

Generated from assembly DirectX.Capture [1.0.1122.23301]
DirectX.Capture Class Library
The properties of the `PropertyPageCollection` class are listed below. For a complete list of `PropertyPageCollection` class members, see the `PropertyPageCollection Members` topic.

### Public Instance Properties

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Count</strong> (inherited from <code>CollectionBase</code>)</td>
<td>Gets the number of elements contained in the <code>CollectionBase</code> instance.</td>
</tr>
<tr>
<td><strong>Item</strong></td>
<td>Get the filter at the specified index.</td>
</tr>
</tbody>
</table>

See Also

- [PropertyPageCollection Class](#) | [DirectX.Capture Namespace](#)
DirectX.Capture Class Library
PropertyPageCollection.Item Property

Get the filter at the specified index.

```csharp
public PropertyPage this[int index] {get;}
```

See Also

PropertyPageCollection Class | PropertyPageCollection Members | DirectX.Capture Namespace

© 2002 Brian Low. All rights reserved.

Generated from assembly DirectX.Capture [1.0.1122.23301]
DirectX.Capture Class Library
The methods of the **PropertyPageCollection** class are listed below. For a complete list of **PropertyPageCollection** class members, see the [PropertyPageCollection Members](#) topic.

### Public Instance Methods

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Clear</strong></td>
<td>Empty the collection.</td>
</tr>
<tr>
<td><strong>Dispose</strong></td>
<td>Release unmanaged resources</td>
</tr>
<tr>
<td><strong>Equals</strong> (inherited from <strong>Object</strong>)</td>
<td>Determines whether the specified <strong>Object</strong> is equal to the current <strong>Object</strong>.</td>
</tr>
<tr>
<td><strong>GetEnumerator</strong> (inherited from <strong>CollectionBase</strong>)</td>
<td>Returns an enumerator that can iterate through the <strong>CollectionBase</strong> instance.</td>
</tr>
<tr>
<td><strong>GetHashCode</strong> (inherited from <strong>Object</strong>)</td>
<td>Serves as a hash function for a particular type, suitable for use in hashing algorithms and data structures like a hash table.</td>
</tr>
<tr>
<td><strong>GetType</strong> (inherited from <strong>Object</strong>)</td>
<td>Gets the <strong>Type</strong> of the current instance.</td>
</tr>
<tr>
<td><strong>RemoveAt</strong> (inherited from <strong>CollectionBase</strong>)</td>
<td>Removes the element at the specified index of the <strong>CollectionBase</strong> instance.</td>
</tr>
<tr>
<td><strong>ToString</strong> (inherited from <strong>Object</strong>)</td>
<td>Returns a <strong>String</strong> that represents the current <strong>Object</strong>.</td>
</tr>
</tbody>
</table>

### Explicit Interface Implementations

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>ICollection.CopyTo</strong> (inherited from <strong>CollectionBase</strong>)</td>
<td></td>
</tr>
<tr>
<td><strong>IList.Add</strong> (inherited from <strong>CollectionBase</strong>)</td>
<td></td>
</tr>
</tbody>
</table>
See Also

PropertyPageCollection Class | DirectX.Capture Namespace

© 2002 Brian Low. All rights reserved.
Generated from assembly DirectX.Capture [1.0.1122.23301]
DirectX.Capture Class Library
PropertyPageCollection.Clear Method

Empty the collection.

```
new public void Clear();
```

Implements

```
IList.Clear
```

See Also

```
PropertyPageCollection Class | PropertyPageCollection Members | DirectX.Capture Namespace
```

© 2002 Brian Low. All rights reserved.

Generated from assembly DirectX.Capture [1.0.1122.23301]
DirectX.Capture Class Library
## PropertyPageCollection.Dispose Method

Release unmanaged resources

```csharp
public void Dispose();
```

**Implements**

- `IDisposable.Dispose`

**See Also**

- [PropertyPageCollection Class](#)
- [PropertyPageCollection Members](#)
- [DirectX.Capture Namespace](#)

---

© 2002 Brian Low. All rights reserved.

Generated from assembly DirectX.Capture [1.0.1122.23301]
DirectX.Capture Class Library
Source Class

Represents a physical connector or source on an audio/video device.
For a list of all members of this type, see Source Members.

System.Object  Source

public class Source : IDisposable

Requirements

Namespace: DirectX.Capture Namespace
Assembly: DirectX.Capture.dll

See Also

Source Members | DirectX.Capture Namespace

© 2002 Brian Low. All rights reserved.
Generated from assembly DirectX.Capture [1.0.1122.23301]
DirectX.Capture Class Library
### Public Instance Constructors

| Source Constructor | Initializes a new instance of the Source class. |

### Public Instance Properties

<table>
<thead>
<tr>
<th>Enabled</th>
<th>Is this source enabled.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>The name of the source. Read-only.</td>
</tr>
</tbody>
</table>

### Public Instance Methods

<table>
<thead>
<tr>
<th>Dispose</th>
<th>Release unmanaged resources.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Equals (inherited from Object)</td>
<td>Determines whether the specified Object is equal to the current Object.</td>
</tr>
<tr>
<td>GetHashCode (inherited from Object)</td>
<td>Serves as a hash function for a particular type, suitable for use in hashing algorithms and data structures like a hash table.</td>
</tr>
<tr>
<td>GetType (inherited from Object)</td>
<td>Gets the Type of the current instance.</td>
</tr>
<tr>
<td>ToString</td>
<td>Obtains the String representation of this instance.</td>
</tr>
</tbody>
</table>

### See Also

- [Source Class](#) | [DirectX.Capture Namespace](#)

© 2002 Brian Low. All rights reserved.

Generated from assembly DirectX.Capture [1.0.1122.23301]
DirectX.Capture Class Library
Source Constructor

Initializes a new instance of the Source class.

```java
public Source();
```

See Also

- [Source Class](#) | [Source Members](#) | [DirectX.Capture Namespace](#)

© 2002 Brian Low. All rights reserved.

Generated from assembly DirectX.Capture [1.0.1122.23301]
DirectX.Capture Class Library
Source Properties

The properties of the **Source** class are listed below. For a complete list of **Source** class members, see the [Source Members] topic.

### Public Instance Properties

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Enabled</strong></td>
<td>Is this source enabled.</td>
</tr>
<tr>
<td><strong>Name</strong></td>
<td>The name of the source. Read-only.</td>
</tr>
</tbody>
</table>

See Also

[Source Class] | [DirectX.Capture Namespace]

© 2002 Brian Low. All rights reserved.

Generated from assembly DirectX.Capture [1.0.1122.23301]
DirectX.Capture Class Library
Source.Enabled Property

Is this source enabled.

```
public virtual bool Enabled {get; set;}
```

See Also

[Source Class](#) | [Source Members](#) | [DirectX.Capture Namespace](#)

© 2002 Brian Low. All rights reserved.

Generated from assembly DirectX.Capture [1.0.1122.23301]
DirectX.Capture Class Library
Source.Name Property

The name of the source. Read-only.

```csharp
public string Name {get;}
```

See Also

[Source Class](#) | [Source Members](#) | [DirectX.Capture Namespace](#)

© 2002 Brian Low. All rights reserved.

Generated from assembly DirectX.Capture [1.0.1122.23301]
DirectX.Capture Class Library
Source Methods

The methods of the **Source** class are listed below. For a complete list of **Source** class members, see the **Source Members** topic.

### Public Instance Methods

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>✧ Dispose</td>
<td>Release unmanaged resources.</td>
</tr>
<tr>
<td>✧ Equals (inherited from <strong>Object</strong>)</td>
<td>Determines whether the specified <strong>Object</strong> is equal to the current <strong>Object</strong>.</td>
</tr>
<tr>
<td>✧ GetHashCode (inherited from <strong>Object</strong>)</td>
<td>Serves as a hash function for a particular type, suitable for use in hashing algorithms and data structures like a hash table.</td>
</tr>
<tr>
<td>✧ GetType (inherited from <strong>Object</strong>)</td>
<td>Gets the <strong>Type</strong> of the current instance.</td>
</tr>
<tr>
<td>✧ ToString</td>
<td>Obtains the String representation of this instance.</td>
</tr>
</tbody>
</table>

**See Also**

- [Source Class](#) | [DirectX.Capture Namespace](#)

© 2002 Brian Low. All rights reserved.

Generated from assembly DirectX.Capture [1.0.1122.23301]
DirectX.Capture Class Library
Source.Dispose Method

Release unmanaged resources.

```
public virtual void Dispose();
```

Implements

IDisposable.Dispose

See Also

Source Class | Source Members | DirectX.Capture Namespace

© 2002 Brian Low. All rights reserved.

Generated from assembly DirectX.Capture [1.0.1122.23301]
DirectX.Capture Class Library
Obtains the String representation of this instance.

```csharp
public override string ToString();
```

See Also

- [Source Class](#)
- [Source Members](#)
- [DirectX.Capture Namespace](#)

© 2002 Brian Low. All rights reserved.

Generated from assembly DirectX.Capture [1.0.1122.23301]
DirectX.Capture Class Library
SourceCollection Class

A collection of sources (or physical connectors) on an audio or video device. This is used by the Capture class to provide a list of available sources on the currently selected audio and video devices. This class cannot be created directly. This class assumes there is only 1 video and 1 audio crossbar and all input pins route to a single output pin on each crossbar.

For a list of all members of this type, see SourceCollection Members.

**System.Object**  **CollectionBase**

**SourceCollection**

```csharp
public class SourceCollection : CollectionBase, IDisposable
```

Requirements

**Namespace:** DirectX.Capture Namespace

**Assembly:** DirectX.Capture.dll

See Also

SourceCollection Members | DirectX.Capture Namespace

© 2002 Brian Low. All rights reserved.

Generated from assembly DirectX.Capture [1.0.1122.23301]
DirectX.Capture Class Library
# SourceCollection Members

## Public Instance Properties

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Count</strong> (inherited from CollectionBase)</td>
<td>Gets the number of elements contained in the CollectionBase instance.</td>
</tr>
<tr>
<td><strong>Item</strong></td>
<td>Get the source at the specified index.</td>
</tr>
</tbody>
</table>

## Public Instance Methods

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Clear</strong></td>
<td>Empty the collection.</td>
</tr>
<tr>
<td><strong>Dispose</strong></td>
<td>Release unmanaged resources.</td>
</tr>
<tr>
<td><strong>Equals</strong> (inherited from Object)</td>
<td>Determines whether the specified Object is equal to the current Object.</td>
</tr>
<tr>
<td><strong>GetEnumerator</strong> (inherited from CollectionBase)</td>
<td>Returns an enumerator that can iterate through the CollectionBase instance.</td>
</tr>
<tr>
<td><strong>GetHashCode</strong> (inherited from Object)</td>
<td>Serves as a hash function for a particular type, suitable for use in hashing algorithms and data structures like a hash table.</td>
</tr>
<tr>
<td><strong>GetType</strong> (inherited from Object)</td>
<td>Gets the Type of the current instance.</td>
</tr>
<tr>
<td><strong>RemoveAt</strong> (inherited from CollectionBase)</td>
<td>Removes the element at the specified index of the CollectionBase instance.</td>
</tr>
<tr>
<td><strong>ToString</strong> (inherited from Object)</td>
<td>Returns a String that represents the current Object.</td>
</tr>
</tbody>
</table>

## Explicit Interface Implementations
<table>
<thead>
<tr>
<th>Method</th>
<th>Inherited From</th>
</tr>
</thead>
<tbody>
<tr>
<td>ICollection.CopyTo</td>
<td>CollectionBase</td>
</tr>
<tr>
<td>IList.Add</td>
<td>CollectionBase</td>
</tr>
<tr>
<td>IList.Contains</td>
<td>CollectionBase</td>
</tr>
<tr>
<td>IList.IndexOf</td>
<td>CollectionBase</td>
</tr>
<tr>
<td>IList.Insert</td>
<td>CollectionBase</td>
</tr>
<tr>
<td>IList.Remove</td>
<td>CollectionBase</td>
</tr>
</tbody>
</table>

See Also

SourceCollection Class | DirectX.Capture Namespace

© 2002 Brian Low. All rights reserved.

Generated from assembly DirectX.Capture [1.0.1122.23301]
DirectX.Capture Class Library
SourceCollection Properties

The properties of the **SourceCollection** class are listed below. For a complete list of **SourceCollection** class members, see the **SourceCollection Members** topic.

Public Instance Properties

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Count</strong> (inherited from <strong>CollectionBase</strong>)</td>
<td>Gets the number of elements contained in the <strong>CollectionBase</strong> instance.</td>
</tr>
<tr>
<td><strong>Item</strong></td>
<td>Get the source at the specified index.</td>
</tr>
</tbody>
</table>

See Also

[SourceCollection Class](#) | [DirectX.Capture Namespace](#)  

© 2002 Brian Low. All rights reserved.

Generated from assembly DirectX.Capture [1.0.1122.23301]
DirectX.Capture Class Library
SourceCollection.Item Property

Get the source at the specified index.

```csharp
public Source this[int index] {get;}
```

See Also

SourceCollection Class | SourceCollection Members | DirectX.Capture Namespace

© 2002 Brian Low. All rights reserved.

Generated from assembly DirectX.Capture [1.0.1122.23301]
DirectX.Capture Class Library
# SourceCollection Methods

The methods of the `SourceCollection` class are listed below. For a complete list of `SourceCollection` class members, see the `SourceCollection Members` topic.

## Public Instance Methods

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Clear</strong></td>
<td>Empty the collection.</td>
</tr>
<tr>
<td><strong>Dispose</strong></td>
<td>Release unmanaged resources.</td>
</tr>
<tr>
<td><strong>Equals</strong> (inherited from <code>Object</code>)</td>
<td>Determines whether the specified <code>Object</code> is equal to the current <code>Object</code>.</td>
</tr>
<tr>
<td><strong>GetEnumerator</strong> (inherited from <code>CollectionBase</code>)</td>
<td>Returns an enumerator that can iterate through the <code>CollectionBase</code> instance.</td>
</tr>
<tr>
<td><strong>GetHashCode</strong> (inherited from <code>Object</code>)</td>
<td>Serves as a hash function for a particular type, suitable for use in hashing algorithms and data structures like a hash table.</td>
</tr>
<tr>
<td><strong>GetType</strong> (inherited from <code>Object</code>)</td>
<td>Gets the <code>Type</code> of the current instance.</td>
</tr>
<tr>
<td><strong>RemoveAt</strong> (inherited from <code>CollectionBase</code>)</td>
<td>Removes the element at the specified index of the <code>CollectionBase</code> instance.</td>
</tr>
<tr>
<td><strong>ToString</strong> (inherited from <code>Object</code>)</td>
<td>Returns a <code>String</code> that represents the current <code>Object</code>.</td>
</tr>
</tbody>
</table>

## Explicit Interface Implementations

<table>
<thead>
<tr>
<th>Interface</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>ICollection.CopyTo</strong> (inherited from <code>CollectionBase</code>)</td>
<td></td>
</tr>
<tr>
<td><strong>IList.Add</strong> (inherited from <code>CollectionBase</code>)</td>
<td></td>
</tr>
<tr>
<td>Method</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------</td>
<td>--------------------------------------------------</td>
</tr>
<tr>
<td><code>IList.Contains</code></td>
<td>(inherited from <code>CollectionBase</code>)</td>
</tr>
<tr>
<td><code>IList.IndexOf</code></td>
<td>(inherited from <code>CollectionBase</code>)</td>
</tr>
<tr>
<td><code>IList.Insert</code></td>
<td>(inherited from <code>CollectionBase</code>)</td>
</tr>
<tr>
<td><code>IList.Remove</code></td>
<td>(inherited from <code>CollectionBase</code>)</td>
</tr>
</tbody>
</table>

**See Also**

- [SourceCollection Class](#) | [DirectX>Capture Namespace](#)

© 2002 Brian Low. All rights reserved.

Generated from assembly DirectX>Capture [1.0.1122.23301]
DirectX.Capture Class Library
SourceCollection.Clear Method

Empty the collection.

new public void Clear();

Implements

IList.Clear

See Also

SourceCollection Class | SourceCollection Members | DirectX.Capture Namespace

© 2002 Brian Low. All rights reserved.

Generated from assembly DirectX.Capture [1.0.1122.23301]
DirectX.Capture Class Library
SourceCollection.Dispose Method

Release unmanaged resources.

```csharp
public void Dispose();
```

Implements

IDisposable.Dispose

See Also

SourceCollection Class | SourceCollection Members | DirectX.Capture Namespace

© 2002 Brian Low. All rights reserved.

Generated from assembly DirectX.Capture [1.0.1122.23301]
DirectX.Capture Class Library
Control and query a hardware TV Tuner.
For a list of all members of this type, see Tuner Members.

See Also
Tuner Members | DirectX.Capture Namespace

© 2002 Brian Low. All rights reserved.
Generated from assembly DirectX.Capture [1.0.1122.23301]
DirectX.Capture Class Library
## Tuner Members

### Public Instance Constructors

<table>
<thead>
<tr>
<th>Tuner Constructor</th>
<th>Initialize this object with a DirectShow tuner</th>
</tr>
</thead>
</table>

### Public Instance Properties

<table>
<thead>
<tr>
<th>Channel</th>
<th>Get or set the TV Tuner channel.</th>
</tr>
</thead>
<tbody>
<tr>
<td>InputType</td>
<td>Get or set the tuner frequency (cable or antenna).</td>
</tr>
<tr>
<td>SignalPresent</td>
<td>Indicates whether a signal is present on the current channel. If the signal strength cannot be determined, a NotSupportedException is thrown.</td>
</tr>
</tbody>
</table>

### Public Instance Methods

<table>
<thead>
<tr>
<th>Dispose</th>
<th>Determines whether the specified Object is equal to the current Object.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Equals (inherited from Object)</td>
<td>Serves as a hash function for a particular type, suitable for use in hashing algorithms and data structures like a hash table.</td>
</tr>
<tr>
<td>GetHashCode (inherited from Object)</td>
<td>Gets the Type of the current instance.</td>
</tr>
<tr>
<td>GetType (inherited from Object)</td>
<td>Returns a String that represents the current Object.</td>
</tr>
<tr>
<td>ToString (inherited from Object)</td>
<td></td>
</tr>
</tbody>
</table>

### See Also
DirectX.Capture Class Library
Initialize this object with a DirectShow tuner

```csharp
public Tuner(IAMTVTuner tuner);
```

See Also

- Tuner Class
- Tuner Members
- DirectX.Capture Namespace

© 2002 Brian Low. All rights reserved.

Generated from assembly DirectX.Capture [1.0.1122.23301]
Tuner Properties

The properties of the Tuner class are listed below. For a complete list of Tuner class members, see the Tuner Members topic.

Public Instance Properties

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Channel</td>
<td>Get or set the TV Tuner channel.</td>
</tr>
<tr>
<td>InputType</td>
<td>Get or set the tuner frequency (cable or antenna).</td>
</tr>
<tr>
<td>SignalPresent</td>
<td>Indicates whether a signal is present on the current channel. If the signal strength cannot be determined, a NotSupportedException is thrown.</td>
</tr>
</tbody>
</table>

See Also

Tuner Class | DirectX.Capture Namespace

© 2002 Brian Low. All rights reserved.

Generated from assembly DirectX.Capture [1.0.1122.23301]
DirectX.Capture Class Library
Get or set the TV Tuner channel.

```csharp
public int Channel {get; set;}
```

See Also

[Tuner Class] | [Tuner Members] | [DirectX.Capture Namespace]

© 2002 Brian Low. All rights reserved.

Generated from assembly DirectX.Capture [1.0.1122.23301]
DirectX.Capture Class Library
**Tuner.InputType Property**

Get or set the tuner frequency (cable or antenna).

```csharp
public TunerInputType InputType {get; set;}
```

See Also

[Tuner Class] | [Tuner Members] | [DirectX.Capture Namespace]

© 2002 Brian Low. All rights reserved.

Generated from assembly DirectX.Capture [1.0.1122.23301]
DirectX.Capture Class Library
Tuner.SignalPresent Property

Indicates whether a signal is present on the current channel. If the signal strength cannot be determined, a NotSupportedException is thrown.

```csharp
public bool SignalPresent {get;}
```

See Also

Tuner Class | Tuner Members | DirectX.Capture Namespace

© 2002 Brian Low. All rights reserved.
Generated from assembly DirectX.Capture [1.0.1122.23301]
DirectX.Capture Class Library
The methods of the **Tuner** class are listed below. For a complete list of **Tuner** class members, see the [Tuner Members](#) topic.

### Public Instance Methods

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Dispose</strong></td>
<td>Determines whether the specified <strong>Object</strong> is equal to the current <strong>Object</strong>.</td>
</tr>
<tr>
<td><strong>Equals</strong> (inherited from <strong>Object</strong>)</td>
<td>Serves as a hash function for a particular type, suitable for use in hashing algorithms and data structures like a hash table.</td>
</tr>
<tr>
<td><strong>GetHashCode</strong> (inherited from <strong>Object</strong>)</td>
<td>Gets the <strong>Type</strong> of the current instance.</td>
</tr>
<tr>
<td><strong>GetType</strong> (inherited from <strong>Object</strong>)</td>
<td>Returns a <strong>String</strong> that represents the current <strong>Object</strong>.</td>
</tr>
</tbody>
</table>

**See Also**

[Tuner Class](#) | [DirectX.Capture Namespace](#)

© 2002 Brian Low. All rights reserved.

Generated from assembly DirectX.Capture [1.0.1122.23301]
DirectX.Capture Class Library
Tuner.Dispose Method

```csharp
public void Dispose();
```

Implements

- `IDisposable.Dispose`

See Also

- Tuner Class | Tuner Members | DirectX.Capture Namespace

© 2002 Brian Low. All rights reserved.

Generated from assembly DirectX.Capture [1.0.1122.23301]
DirectX.Capture Class Library
**TunerInputType Enumeration**

Specify the frequency of the TV tuner.

```csharp
public enum TunerInputType
```

### Members

<table>
<thead>
<tr>
<th>Member Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cable</td>
<td>Cable frequency</td>
</tr>
<tr>
<td>Antenna</td>
<td>Antenna frequency</td>
</tr>
</tbody>
</table>

### Requirements

- **Namespace:** [DirectX.Capture Namespace](#)
- **Assembly:** DirectX.Capture.dll

### See Also

- [DirectX.Capture Namespace](#)

© 2002 Brian Low. All rights reserved.

Generated from assembly DirectX.Capture [1.0.1122.23301]
The property page to configure a Video for Windows compliant compression codec. Most compressors support this property page rather than a DirectShow property page. Also, most compressors do not support the IAMVideoCompression interface so this property page is the only method to configure a compressor.

For a list of all members of this type, see VfwCompressorPropertyPage Members.

```
public class VfwCompressorPropertyPage : PropertyPage
```

Requirements

**Namespace:** DirectX.Capture Namespace  
**Assembly:** DirectX.Capture.dll

See Also

VfwCompressorPropertyPage Members | DirectX.Capture Namespace

© 2002 Brian Low. All rights reserved.
Generated from assembly DirectX.Capture [1.0.1122.23301]
DirectX.Capture Class Library
## VfwCompressorPropertyPage Members

### Public Instance Constructors

<table>
<thead>
<tr>
<th>Constructor</th>
</tr>
</thead>
<tbody>
<tr>
<td>VfwCompressorPropertyPage Constructor</td>
</tr>
</tbody>
</table>

### Public Instance Fields

<table>
<thead>
<tr>
<th>Name (inherited from PropertyPage)</th>
<th>Name of property page. This name may not be unique</th>
</tr>
</thead>
<tbody>
<tr>
<td>SupportsPersisting (inherited from PropertyPage)</td>
<td>Does this property page support saving and loading the user's choices.</td>
</tr>
</tbody>
</table>

### Public Instance Properties

<table>
<thead>
<tr>
<th>State</th>
</tr>
</thead>
<tbody>
<tr>
<td>Get or set the state of the property page. This is used to save and restore the user's choices without redisplaying the property page. This property will be null if unable to retrieve the property page's state.</td>
</tr>
</tbody>
</table>

### Public Instance Methods

<table>
<thead>
<tr>
<th>Dispose (inherited from PropertyPage)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Release unmanaged resources</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Equals (inherited from Object)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Determines whether the specified Object is equal to the current Object.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>GetHashCode (inherited from Object)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Serves as a hash function for a particular type, suitable for use in hashing algorithms and data structures like a hash table.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>GetType (inherited from)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gets the Type of the current</td>
</tr>
</tbody>
</table>
**Object**

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Show</strong></td>
<td>Show the property page. Some property pages cannot be displayed while previewing and/or capturing.</td>
</tr>
<tr>
<td><strong>ToString</strong> (inherited from <strong>Object</strong>)</td>
<td>Returns a <strong>String</strong> that represents the current <strong>Object</strong>.</td>
</tr>
</tbody>
</table>

**See Also**

[VfwCompressorPropertyPage Class] | DirectX.Capture Namespace

© 2002 Brian Low. All rights reserved.

Generated from assembly DirectX.Capture [1.0.1122.23301]
DirectX.Capture Class Library
Constructor

```csharp
public VfwCompressorPropertyPage(
    string name,
    IAMVfwCompressDialogs compressDialogs
);
```

See Also

VfwCompressorPropertyPage Class | VfwCompressorPropertyPage Members | DirectX.Capture Namespace

© 2002 Brian Low. All rights reserved.
Generated from assembly DirectX.Capture [1.0.1122.23301]
DirectX.Capture Class Library
The properties of the `VfwCompressorPropertyPage` class are listed below. For a complete list of `VfwCompressorPropertyPage` class members, see the `VfwCompressorPropertyPage Members` topic.

### Public Instance Properties

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>State</code></td>
<td>Get or set the state of the property page. This is used to save and restore the user's choices without redisplaying the property page. This property will be null if unable to retrieve the property page's state.</td>
</tr>
</tbody>
</table>

**See Also**

[VfwCompressorPropertyPage Class](#) | [DirectX.Capture Namespace](#)

© 2002 Brian Low. All rights reserved.

Generated from assembly DirectX.Capture [1.0.1122.23301]
DirectX.Capture Class Library
VfwCompressorPropertyPage.State Property

Get or set the state of the property page. This is used to save and restore the user's choices without redisplaying the property page. This property will be null if unable to retrieve the property page's state.

```csharp
public override byte[] State {get; set;}
```

Remarks

After showing this property page, read and store the value of this property. At a later time, the user's choices can be reloaded by setting this property with the value stored earlier. Note that some property pages, after setting this property, will not reflect the new state. However, the filter will use the new settings.

See Also

VfwCompressorPropertyPage Class | VfwCompressorPropertyPage Members | DirectX.Capture Namespace

© 2002 Brian Low. All rights reserved.

Generated from assembly DirectX.Capture [1.0.1122.23301]
<table>
<thead>
<tr>
<th>DirectX.Capture Class Library</th>
</tr>
</thead>
</table>
The methods of the **VfwCompressorPropertyPage** class are listed below. For a complete list of **VfwCompressorPropertyPage** class members, see the [VfwCompressorPropertyPage Members](#) topic.

### Public Instance Methods

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Dispose</strong> (inherited from <strong>PropertyPage</strong>)</td>
<td>Release unmanaged resources</td>
</tr>
<tr>
<td><strong>Equals</strong> (inherited from <strong>Object</strong>)</td>
<td>Determines whether the specified <strong>Object</strong> is equal to the current <strong>Object</strong>.</td>
</tr>
<tr>
<td><strong>GetHashCode</strong> (inherited from <strong>Object</strong>)</td>
<td>Serves as a hash function for a particular type, suitable for use in hashing algorithms and data structures like a hash table.</td>
</tr>
<tr>
<td><strong>GetType</strong> (inherited from <strong>Object</strong>)</td>
<td>Gets the <strong>Type</strong> of the current instance.</td>
</tr>
<tr>
<td><strong>Show</strong></td>
<td>Show the property page. Some property pages cannot be displayed while previewing and/or capturing.</td>
</tr>
<tr>
<td><strong>ToString</strong> (inherited from <strong>Object</strong>)</td>
<td>Returns a <strong>String</strong> that represents the current <strong>Object</strong>.</td>
</tr>
</tbody>
</table>

---

**See Also**

[VfwCompressorPropertyPage Class](#) | [DirectX.Capture Namespace](#)

© 2002 Brian Low. All rights reserved.

Generated from assembly DirectX.Capture [1.0.1122.23301]
DirectX.Capture Class Library
VfwCompressorPropertyPage.Show Method

Show the property page. Some property pages cannot be displayed while previewing and/or capturing.

```csharp
public override void Show(
    Control owner
);
```

See Also

- [VfwCompressorPropertyPage Class](#) | [VfwCompressorPropertyPage Members](#) | [DirectX.Capture Namespace](#)

© 2002 Brian Low. All rights reserved.

Generated from assembly DirectX.Capture [1.0.1122.23301]
DirectX.Capture Class Library
**VideoCapabilities Class**

Capabilities of the video device such as min/max frame size and frame rate.

For a list of all members of this type, see VideoCapabilities Members.

System.Object VideoCapabilities

```csharp
public class VideoCapabilities
```

Requirements

**Namespace:** DirectX.Capture Namespace

**Assembly:** DirectX.Capture.dll

See Also

VideoCapabilities Members | DirectX.Capture Namespace

© 2002 Brian Low. All rights reserved.

Generated from assembly DirectX.Capture [1.0.1122.23301]
DirectX.Capture Class Library
## VideoCapabilities Members

### Public Instance Fields

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>FrameSizeGranularityX</strong></td>
<td>Granularity of the output width. This value specifies the increments that are valid between MinFrameSize and MaxFrameSize. Read-only.</td>
</tr>
<tr>
<td><strong>FrameSizeGranularityY</strong></td>
<td>Granularity of the output height. This value specifies the increments that are valid between MinFrameSize and MaxFrameSize. Read-only.</td>
</tr>
<tr>
<td><strong>InputSize</strong></td>
<td>Native size of the incoming video signal. This is the largest signal the filter can digitize with every pixel remaining unique. Read-only.</td>
</tr>
<tr>
<td><strong>MaxFrameRate</strong></td>
<td>Maximum supported frame rate. Read-only.</td>
</tr>
<tr>
<td><strong>MaxFrameSize</strong></td>
<td>Maximum supported frame size. Read-only.</td>
</tr>
<tr>
<td><strong>MinFrameRate</strong></td>
<td>Minimum supported frame rate. Read-only.</td>
</tr>
<tr>
<td><strong>MinFrameSize</strong></td>
<td>Minimum supported frame size. Read-only.</td>
</tr>
</tbody>
</table>

### Public Instance Methods

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Equals</strong> (inherited from Object)</td>
<td>Determines whether the specified Object is equal to the current Object.</td>
</tr>
<tr>
<td><strong>GetHashCode</strong> (inherited from Object)</td>
<td>Serves as a hash function for a</td>
</tr>
<tr>
<td>Object</td>
<td>particular type, suitable for use in hashing algorithms and data structures like a hash table.</td>
</tr>
<tr>
<td>---</td>
<td>---</td>
</tr>
<tr>
<td><strong>GetType</strong> (inherited from Object)</td>
<td>Gets the <strong>Type</strong> of the current instance.</td>
</tr>
<tr>
<td><strong>ToString</strong> (inherited from Object)</td>
<td>Returns a <strong>String</strong> that represents the current <strong>Object</strong>.</td>
</tr>
</tbody>
</table>

**See Also**

[VideoCapabilities Class](#) | [DirectX.Capture Namespace](#)
| DirectX.Capture Class Library |
VideoCapabilities Fields

The fields of the **VideoCapabilities** class are listed below. For a complete list of **VideoCapabilities** class members, see the **VideoCapabilities Members** topic.

Public Instance Fields

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>FrameSizeGranularityX</strong></td>
<td>Granularity of the output width. This value specifies the increments that are valid between MinFrameSize and MaxFrameSize. Read-only.</td>
</tr>
<tr>
<td><strong>FrameSizeGranularityY</strong></td>
<td>Granularity of the output height. This value specifies the increments that are valid between MinFrameSize and MaxFrameSize. Read-only.</td>
</tr>
<tr>
<td><strong>InputSize</strong></td>
<td>Native size of the incoming video signal. This is the largest signal the filter can digitize with every pixel remaining unique. Read-only.</td>
</tr>
<tr>
<td><strong>MaxFrameRate</strong></td>
<td>Maximum supported frame rate. Read-only.</td>
</tr>
<tr>
<td><strong>MaxFrameSize</strong></td>
<td>Maximum supported frame size. Read-only.</td>
</tr>
<tr>
<td><strong>MinFrameRate</strong></td>
<td>Minimum supported frame rate. Read-only.</td>
</tr>
<tr>
<td><strong>MinFrameSize</strong></td>
<td>Minimum supported frame size. Read-only.</td>
</tr>
</tbody>
</table>

See Also

[VideoCapabilities Class](#) | [DirectX.Capture Namespace](#)
DirectX.Capture Class Library
### VideoCapabilities.FrameSizeGranularityX Field

Granularity of the output width. This value specifies the increments that are valid between MinFrameSize and MaxFrameSize. Read-only.

```
public int FrameSizeGranularityX;
```

See Also

[VideoCapabilities Class](#) | [VideoCapabilities Members](#) | [DirectX.Capture Namespace](#)

© 2002 Brian Low. All rights reserved.

Generated from assembly DirectX.Capture [1.0.1122.23301]
DirectX.Capture Class Library
VideoCapabilities.FrameSizeGranularityY Field

Granularity of the output height. This value specifies the increments that are valid between MinFrameSize and MaxFrameSize. Read-only.

```csharp
public int FrameSizeGranularityY;
```

See Also

- VideoCapabilities Class
- VideoCapabilities Members
- DirectX.Capture Namespace

© 2002 Brian Low. All rights reserved.

Generated from assembly DirectX.Capture [1.0.1122.23301]
DirectX.Capture Class Library
Native size of the incoming video signal. This is the largest signal the filter can digitize with every pixel remaining unique. Read-only.

public Size InputSize;

See Also

VideoCapabilities Class | VideoCapabilities Members | DirectX.Capture Namespace

© 2002 Brian Low. All rights reserved.

Generated from assembly DirectX.Capture [1.0.1122.23301]
DirectX.Capture Class Library
**VideoCapabilities.MaxFrameRate Field**

Maximum supported frame rate. Read-only.

```java
public double MaxFrameRate;
```

See Also

- [VideoCapabilities Class](#) | [VideoCapabilities Members](#) | [DirectX.Capture Namespace](#)

© 2002 Brian Low. All rights reserved.

Generated from assembly DirectX.Capture [1.0.1122.23301]
DirectX.Capture Class Library
**VideoCapabilities.MaxFrameSize Field**

Maximum supported frame size. Read-only.

```csharp
public Size MaxFrameSize;
```

**See Also**

- [VideoCapabilities Class](#) | [VideoCapabilities Members](#) | [DirectX.Capture Namespace](#)

© 2002 Brian Low. All rights reserved.

Generated from assembly DirectX.Capture [1.0.1122.23301]
DirectX.Capture Class Library
**VideoCapabilities.MinFrameRate Field**

Minimum supported frame rate. Read-only.

```csharp
public double MinFrameRate;
```

See Also

[VideoCapabilities Class](#) | [VideoCapabilities Members](#) | [DirectX.Capture Namespace](#)

© 2002 Brian Low. All rights reserved.

Generated from assembly DirectX.Capture [1.0.1122.23301]
DirectX.Capture Class Library
VideoCapabilities.MinFrameSize Field

Minimum supported frame size. Read-only.

```csharp
public Size MinFrameSize;
```

See Also

- VideoCapabilities Class
- VideoCapabilities Members
- DirectX.Capture Namespace

© 2002 Brian Low. All rights reserved.

Generated from assembly DirectX.Capture [1.0.1122.23301]