A Managed DirectShow Toolkit
DirectShowLib_Utils Namespace

Namespace hierarchy

Classes

<table>
<thead>
<tr>
<th>Class</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>FilterGraphTools</td>
<td>A collection of methods to do common DirectShow tasks.</td>
</tr>
</tbody>
</table>
A Managed DirectShow Toolkit
**FilterGraphTools Class**

A collection of methods to do common DirectShow tasks.

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

**System.Object** DirectShowLib.Utils.FilterGraphTools

```csharp
public sealed class FilterGraphTools
```

**Thread Safety**

Public static *(Shared in Visual Basic)* members of this type are safe for multithreaded operations. Instance members are **not** guaranteed to be thread-safe.

**Requirements**

**Namespace:** DirectShowLib.Utils

**Assembly:** DirectShowLib.Utils (in DirectShowLib.Utils.dll)

**See Also**

[FilterGraphTools Members](#) | [DirectShowLib.Utils Namespace](#)
A Managed DirectShow Toolkit
## FilterGraphTools Members

### FilterGraphTools overview

### Public Static Methods

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AddFilterByDevicePath</td>
<td>Add a filter to a DirectShow Graph using its Moniker's device path</td>
</tr>
<tr>
<td>AddFilterByName</td>
<td>Add a filter to a DirectShow Graph using its name</td>
</tr>
<tr>
<td>AddFilterFromClsid</td>
<td>Add a filter to a DirectShow Graph using its CLSID</td>
</tr>
<tr>
<td>ConnectFilters</td>
<td>Overloaded. Connect pins from two filters</td>
</tr>
<tr>
<td>DisconnectAllPins</td>
<td>Disconnect pins of all the filters in a DirectShow Graph</td>
</tr>
<tr>
<td>DisconnectPins</td>
<td>Disconnect all pins on a given filter</td>
</tr>
<tr>
<td>FindFilterByClsid</td>
<td>Find a filter in a DirectShow Graph using its CLSID</td>
</tr>
<tr>
<td>FindFilterByName</td>
<td>Find a filter in a DirectShow Graph using its name</td>
</tr>
<tr>
<td>HasPropertyPages</td>
<td>Check if a DirectShow filter can display Property Pages</td>
</tr>
<tr>
<td>IsThisComObjectInstalled</td>
<td>Check if a COM Object is available</td>
</tr>
<tr>
<td>IsVMR7Present</td>
<td>Check if the Video Mixing Renderer 7 Filter is available</td>
</tr>
<tr>
<td>IsVMR9Present</td>
<td>Check if the Video Mixing Renderer 9 Filter is available</td>
</tr>
<tr>
<td>LoadGraphFile</td>
<td>Load a DirectShow Graph from</td>
</tr>
<tr>
<td>Method</td>
<td>Description</td>
</tr>
<tr>
<td>------------------------</td>
<td>-------------------------------------------------------</td>
</tr>
<tr>
<td><code>RemoveAllFilters</code></td>
<td>Remove and release all filters from a DirectShow Graph</td>
</tr>
<tr>
<td><code>RenderPin</code></td>
<td>Render a filter's pin in a DirectShow Graph</td>
</tr>
<tr>
<td><code>SaveGraphFile</code></td>
<td>Save a DirectShow Graph to a GRF file</td>
</tr>
<tr>
<td><code>ShowFilterPropertyPage</code></td>
<td>Display Property Pages of a given DirectShow filter</td>
</tr>
</tbody>
</table>

**Public Instance Methods**

- `Equals` (inherited from `Object`)
- `GetHashCode` (inherited from `Object`)
- `GetType` (inherited from `Object`)
- `ToString` (inherited from `Object`)

**See Also**

- [FilterGraphTools Class](#) | [DirectShowLib.Utils Namespace](#)
A Managed DirectShow Toolkit
FilterGraphTools Methods

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

Public Static Methods

<table>
<thead>
<tr>
<th>Method Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AddFilterByDevicePath</td>
<td>Add a filter to a DirectShow Graph using its Moniker's device path</td>
</tr>
<tr>
<td>AddFilterByName</td>
<td>Add a filter to a DirectShow Graph using its name</td>
</tr>
<tr>
<td>AddFilterFromClsid</td>
<td>Add a filter to a DirectShow Graph using its CLSID</td>
</tr>
<tr>
<td>ConnectFilters</td>
<td>Overloaded. Connect pins from two filters</td>
</tr>
<tr>
<td>DisconnectAllPins</td>
<td>Disconnect pins of all the filters in a DirectShow Graph</td>
</tr>
<tr>
<td>DisconnectPins</td>
<td>Disconnect all pins on a given filter</td>
</tr>
<tr>
<td>FindFilterByClsid</td>
<td>Find a filter in a DirectShow Graph using its CLSID</td>
</tr>
<tr>
<td>FindFilterByName</td>
<td>Find a filter in a DirectShow Graph using its name</td>
</tr>
<tr>
<td>HasPropertyPages</td>
<td>Check if a DirectShow filter can display Property Pages</td>
</tr>
<tr>
<td>IsThisComObjectInstalled</td>
<td>Check if a COM Object is available</td>
</tr>
<tr>
<td>IsVMR7Present</td>
<td>Check if the Video Mixing Renderer 7 Filter is available</td>
</tr>
<tr>
<td>IsVMR9Present</td>
<td>Check if the Video Mixing Renderer 9 Filter is available</td>
</tr>
<tr>
<td>Method</td>
<td>Description</td>
</tr>
<tr>
<td>------------------------</td>
<td>-------------------------------------------------------</td>
</tr>
<tr>
<td>LoadGraphFile</td>
<td>Load a DirectShow Graph from a file</td>
</tr>
<tr>
<td>RemoveAllFilters</td>
<td>Remove and release all filters from a DirectShow Graph</td>
</tr>
<tr>
<td>RenderPin</td>
<td>Render a filter's pin in a DirectShow Graph</td>
</tr>
<tr>
<td>SaveGraphFile</td>
<td>Save a DirectShow Graph to a GRF file</td>
</tr>
<tr>
<td>ShowFilterPropertyPage</td>
<td>Display Property Pages of a given DirectShow filter</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></td>
</tr>
<tr>
<td>GetHashCode (inherited from Object)</td>
<td></td>
</tr>
<tr>
<td>GetType (inherited from Object)</td>
<td></td>
</tr>
<tr>
<td>ToString (inherited from Object)</td>
<td></td>
</tr>
</tbody>
</table>

**See Also**

FilterGraphTools Class | DirectShowLib.Utils Namespace
A Managed DirectShow Toolkit
Add a filter to a DirectShow Graph using its Moniker's device path

```csharp
public static IBaseFilter AddFilterByDevicePath(
    IGraphBuilder graphBuilder,
    string devicePath,
    string name
);
```

Parameters

- `graphBuilder`  
  the IGraphBuilder interface of the graph

- `devicePath`  
  a moniker path

- `name`  
  the name to use for the filter in the graph

Return Value

an instance of the filter if the method successfully creates it, null if not

Exceptions

<table>
<thead>
<tr>
<th>Exception Type</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>ArgumentNullException</td>
<td>Thrown if <code>graphBuilder</code> is null</td>
</tr>
<tr>
<td>COMException</td>
<td>Thrown if errors occur when the filter is add to the graph</td>
</tr>
</tbody>
</table>

Example

This sample shows how to programmatically add a NVIDIA Video decoder filter to a graph

```csharp
string devicePath = @"@device:sw:{083863F1-70DE-11D0-BD40-00A0C911CE86}\{71E4616A-DB5E-452B-8CA5-71D9CC7805E9}";
filter = FilterGraphTools.AddFilterByDevicePath(graphBuilder, devicePath, "NVIDIA Video Decoder");
```
See Also

FilterGraphTools Class | DirectShowLib.Utils Namespace
A Managed DirectShow Toolkit
FilterGraphTools.AddFilterByName Method

Add a filter to a DirectShow Graph using its name

```csharp
public static IBaseFilter AddFilterByName(
    IGraphBuilder graphBuilder,
    Guid deviceCategory,
    string friendlyName
);
```

Parameters

- `graphBuilder`:
  the IGraphBuilder interface of the graph

- `deviceCategory`:
  the filter category (see DirectShowLib.FilterCategory)

- `friendlyName`:
  the filter name (case-sensitive)

Return Value

- an instance of the filter if the method successfully created it, null if not

Exceptions

<table>
<thead>
<tr>
<th>Exception Type</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>ArgumentNullException</td>
<td>Thrown if graphBuilder is null</td>
</tr>
<tr>
<td>COMException</td>
<td>Thrown if errors occur when the filter is add to the graph</td>
</tr>
</tbody>
</table>

Example

This sample shows how to programmatically add a NVIDIA Video decoder filter to a graph

```csharp
filter = FilterGraphTools.AddFilterByName(graphBuilder,
    FilterCategory.LegacyAmFilterCategory,
    "NVIDIA Video Decoder");
```
See Also

FilterGraphTools Class | DirectShowLib.Utils Namespace
A Managed DirectShow Toolkit
**FilterGraphTools.AddFilterFromClsid Method**

Add a filter to a DirectShow Graph using its CLSID

```csharp
public static IBaseFilter AddFilterFromClsid(
    IGraphBuilder graphBuilder,
    Guid clsid,
    string name
);
```

**Parameters**

- `graphBuilder`
  - the IGraphBuilder interface of the graph
- `clsid`
  - a valid CLSID. This object must implement IBaseFilter
- `name`
  - the name used in the graph (may be null)

**Return Value**

- an instance of the filter if the method successfully created it, null if not

**Remarks**

You can use `IsThisComObjectInstalled` to check if the CLSID is valid before calling this method

**Exceptions**

<table>
<thead>
<tr>
<th>Exception Type</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>ArgumentNullException</td>
<td>Thrown if <code>graphBuilder</code> is null</td>
</tr>
<tr>
<td>COMException</td>
<td>Thrown if errors occur when the filter is added to the graph</td>
</tr>
</tbody>
</table>

**Example**
This sample shows how to programmatically add a NVIDIA Video decoder filter to a graph

```csharp
Guid nvidiaVideoDecClsid = new Guid("71E4616A-DB5E-452B-8CA5-71D9CC7805E9");

if (FilterGraphTools.IsThisComObjectInstalled(nvidiaVideoDecClsid))
{
    filter = FilterGraphTools.AddFilterFromClsid(graphBuilder, nvidiaVideoDecClsid, "NVIDIA Video Decoder");
}
else
{
    // use another filter...
}
```

See Also

- FilterGraphTools Class
- DirectShowLib.Utils Namespace
- IsThisComObjectInstalled
A Managed DirectShow Toolkit
FilterGraphTools.ConnectFilters Method

Connect pins from two filters

Overload List

Connect pins from two filters

public static void ConnectFilters(IGraphBuilder, IBaseFilter, string, IBaseFilter, string, bool);

Connect pins from two filters

public static void ConnectFilters(IGraphBuilder, IPin, IPin, bool);

See Also

FilterGraphTools Class | DirectShowLib_Utils Namespace
A Managed DirectShow Toolkit
FilterGraphTools.ConnectFilters Method (IGraphBuilder, IBaseFilter, String, IBaseFilter, String, Boolean)

Connect pins from two filters

```csharp
public static void ConnectFilters(
    IGraphBuilder graphBuilder,
    IBaseFilter upFilter,
    string sourcePinName,
    IBaseFilter downFilter,
    string destPinName,
    bool useIntelligentConnect
);
```

**Parameters**

- `graphBuilder`  
  the IGraphBuilder interface of the graph

- `upFilter`  
  the upstream filter

- `sourcePinName`  
  the upstream filter pin name

- `downFilter`  
  the downstream filter

- `destPinName`  
  the downstream filter pin name

- `useIntelligentConnect`  
  indicate if the method should use DirectShow's Intelligent Connect

**Remarks**

If useIntelligentConnect is true, this method can add missing filters between the two pins.
If useIntelligentConnect is false, this method works only if the two media types are compatible.
Exceptions

<table>
<thead>
<tr>
<th>Exception Type</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>ArgumentNullException</td>
<td>Thrown if graphBuilder, upFilter or downFilter are null</td>
</tr>
<tr>
<td>ArgumentException</td>
<td>Thrown if pin names are not found in filters</td>
</tr>
<tr>
<td>COMException</td>
<td>Thrown if pins can't connect</td>
</tr>
</tbody>
</table>

See Also

FilterGraphTools Class | DirectShowLib.Utils Namespace | FilterGraphTools.ConnectFilters Overload List
A Managed DirectShow Toolkit
Connect pins from two filters

```csharp
public static void ConnectFilters(
    IGraphBuilder graphBuilder,
    IPin sourcePin,
    IPin destPin,
    bool useIntelligentConnect
);
```

**Parameters**

- `graphBuilder`  
  the IGraphBuilder interface of the graph
- `sourcePin`  
  the source (upstream / output) pin
- `destPin`  
  the destination (downstream / input) pin
- `useIntelligentConnect`  
  indicates if the method should use DirectShow's Intelligent Connect

**Remarks**

If `useIntelligentConnect` is true, this method can add missing filters between the two pins.  
If `useIntelligentConnect` is false, this method works only if the two media types are compatible.

**Exceptions**

<table>
<thead>
<tr>
<th>Exception Type</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>ArgumentNullException</code></td>
<td>Thrown if <code>graphBuilder</code>, <code>sourcePin</code> or <code>destPin</code> are null</td>
</tr>
<tr>
<td><code>COMException</code></td>
<td>Thrown if pins can't connect</td>
</tr>
</tbody>
</table>
See Also

FilterGraphTools Class | DirectShowLib.Utils Namespace | FilterGraphTools.ConnectFilters Overload List
A Managed DirectShow Toolkit
FilterGraphTools.DisconnectAllPins Method

Disconnect pins of all the filters in a DirectShow Graph

```csharp
public static void DisconnectAllPins(IGraphBuilder graphBuilder);
```

Parameters

- `graphBuilder`  
  the IGraphBuilder interface of the graph

Remarks

This method doesn't throw an exception if an error occurs during pin disconnections

Exceptions

<table>
<thead>
<tr>
<th>Exception Type</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>ArgumentException</td>
<td>Thrown if <code>graphBuilder</code> is null</td>
</tr>
<tr>
<td>COMException</td>
<td>Thrown if the method can't enumerate its filters</td>
</tr>
</tbody>
</table>

See Also

- [FilterGraphTools Class](#)  
- [DirectShowLib.Utils Namespace](#)
A Managed DirectShow Toolkit
FilterGraphTools.DisconnectPins Method

Disconnect all pins on a given filter

```csharp
public static void DisconnectPins(
    IBaseFilter filter
);
```

Parameters

- `filter`  
  the filter on which to disconnect all the pins

Remarks

Both input and output pins are disconnected

Exceptions

<table>
<thead>
<tr>
<th>Exception Type</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>ArgumentNullException</td>
<td>Thrown if <code>filter</code> is null</td>
</tr>
<tr>
<td>COMException</td>
<td>Thrown if errors occurred during the disconnection process</td>
</tr>
</tbody>
</table>

See Also

- [FilterGraphTools Class](#)  
- [DirectShowLib_Utils Namespace](#)
A Managed DirectShow Toolkit
FilterGraphTools.FindFilterByClsid Method

Find a filter in a DirectShow Graph using its CLSID

```csharp
public static IBaseFilter FindFilterByClsid(
    IGraphBuilder graphBuilder,
    Guid filterClsid
);
```

Parameters

- `graphBuilder` - the IGraphBuilder interface of the graph
- `filterClsid` - the CLSID to find

Return Value

- an instance of the filter if found, null if not

Exceptions

<table>
<thead>
<tr>
<th>Exception Type</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>ArgumentNullException</td>
<td>Thrown if <code>graphBuilder</code> is null</td>
</tr>
</tbody>
</table>

See Also

- FilterGraphTools Class
- DirectShowLib_Utils Namespace
- FindFilterByName
A Managed DirectShow Toolkit
Find a filter in a DirectShow Graph using its name

```csharp
public static IBaseFilter FindFilterByName(IgraphBuilder graphBuilder, string filterName);
```

**Parameters**

- `graphBuilder` the IGraphBuilder interface of the graph
- `filterName` the filter name to find (case-sensitive)

**Return Value**

an instance of the filter if found, null if not

**Exceptions**

<table>
<thead>
<tr>
<th>Exception Type</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>ArgumentException</td>
<td>Thrown if graphBuilder is null</td>
</tr>
</tbody>
</table>

**See Also**

- FilterGraphTools Class
- DirectShowLib.Utils Namespace
- FindFilterByClsid
A Managed DirectShow Toolkit
FilterGraphTools.HasPropertyPages Method

Check if a DirectShow filter can display Property Pages

```csharp
public static bool HasPropertyPages(
    IBaseFilter filter
);
```

Parameters

- **filter**
  - A DirectShow Filter

Return Value

true if the filter has Property Pages, false if not

Remarks

This method is intended to be used with ShowFilterPropertyPage

Exceptions

<table>
<thead>
<tr>
<th>Exception Type</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>ArgumentNullException</td>
<td>Thrown if filter is null</td>
</tr>
</tbody>
</table>

See Also

FilterGraphTools Class | DirectShowLib.Use Namespace | ShowFilterPropertyPage
A Managed DirectShow Toolkit
FilterGraphTools.IsThisComObjectInstalled Method

Check if a COM Object is available

```csharp
public static bool IsThisComObjectInstalled(Guid clsid);
```

Parameters

clsid
The CLSID of this object

Return Value

true if the object is available, false if not

Example

This sample shows how to check if the MPEG-2 Demultiplexer filter is available

```csharp
if (FilterGraphTools.IsThisComObjectInstalled(typeof(MPEG2Demultiplexer).GUID)) {
    // Use it...
}
```

See Also

FilterGraphTools Class | DirectShowLib.Utils Namespace
A Managed DirectShow Toolkit
FilterGraphTools.IsVMR7Present Method

Check if the Video Mixing Renderer 7 Filter is available

```csharp
public static bool IsVMR7Present();
```

Return Value

true if VMR7 is present, false if not

Remarks

This method uses `IsThisComObjectInstalled` internally

See Also

[FilterGraphTools Class] | [DirectShowLib.Utils Namespace] | [IsThisComObjectInstalled]
A Managed DirectShow Toolkit
Check if the Video Mixing Renderer 9 Filter is available

```csharp
public static bool IsVMR9Present();
```

**Return Value**

true if VMR9 is present, false if not

**Remarks**

This method uses `IsThisComObjectInstalled` internally

**See Also**

FilterGraphTools Class | DirectShowLib.Utils Namespace | IsThisComObjectInstalled
A Managed DirectShow Toolkit
**FilterGraphTools.LoadGraphFile Method**

Load a DirectShow Graph from a file

```csharp
public static void LoadGraphFile(IGraphBuilder graphBuilder, string fileName);
```

**Parameters**

- `graphBuilder`  
  the IGraphBuilder interface of the graph

- `fileName`  
  the file to be loaded

**Exceptions**

<table>
<thead>
<tr>
<th>Exception Type</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>ArgumentNullException</strong></td>
<td>Thrown if <code>graphBuilder</code> is null</td>
</tr>
<tr>
<td><strong>ArgumentException</strong></td>
<td>Thrown if the given file is not a valid graph file</td>
</tr>
<tr>
<td><strong>COMException</strong></td>
<td>Thrown if errors occur during loading</td>
</tr>
</tbody>
</table>

**See Also**

- FilterGraphTools Class
- DirectShowLib.Utils Namespace
- SaveGraphFile
A Managed DirectShow Toolkit
**FilterGraphTools.RemoveAllFilters Method**

Remove and release all filters from a DirectShow Graph

```csharp
public static void RemoveAllFilters(
    IGraphBuilder graphBuilder
);
```

**Parameters**

`graphBuilder`  
the IGraphBuilder interface of the graph

**Exceptions**

<table>
<thead>
<tr>
<th>Exception Type</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>ArgumentNullException</td>
<td>Thrown if graphBuilder is null</td>
</tr>
<tr>
<td>COMException</td>
<td>Thrown if the method can't enumerate its filters</td>
</tr>
</tbody>
</table>

**See Also**

FilterGraphTools Class  | DirectShowLib.Utils Namespace
A Managed DirectShow Toolkit
FilterGraphTools.RenderPin Method

Render a filter's pin in a DirectShow Graph

```csharp
public static bool RenderPin(
    IGraphBuilder graphBuilder,
    IBaseFilter source,
    string pinName
);
```

Parameters

- `graphBuilder` the IGraphBuilder interface of the graph
- `source` the filter containing the pin to render
- `pinName` the pin name

Return Value

true if rendering is a success, false if not

Remarks

This method assumes that the filter is part of the given graph

Exceptions

<table>
<thead>
<tr>
<th>Exception Type</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>ArgumentNullException</td>
<td>Thrown if graphBuilder or source is null</td>
</tr>
</tbody>
</table>

Example

```csharp
hr = graphBuilder.AddSourceFilter(@"foo.avi", "Source Filter", out filter);
DsError.ThrowExceptionForHR(hr);
if (!FilterGraphTools.RenderPin(graphBuilder, filter,
```
{  
  // Something went wrong...
}

See Also

FilterGraphTools Class | DirectShowLib.Utils Namespace
A Managed DirectShow Toolkit
FilterGraphTools.SaveGraphFile Method

Save a DirectShow Graph to a GRF file

```csharp
public static void SaveGraphFile(
    IGraphBuilder graphBuilder,
    string fileName
);
```

Parameters

- `graphBuilder`  
  the IGraphBuilder interface of the graph

- `fileName`  
  the file to be saved

Remarks

This method overwrites any existing file

Exceptions

<table>
<thead>
<tr>
<th>Exception Type</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>ArgumentException</td>
<td>Thrown if <code>graphBuilder</code> is null</td>
</tr>
<tr>
<td>COMException</td>
<td>Thrown if errors occur during the file creation</td>
</tr>
</tbody>
</table>

See Also

- [FilterGraphTools Class](#) | [DirectShowLib.Utils Namespace](#) | [LoadGraphFile](#)
A Managed DirectShow Toolkit
FilterGraphTools.ShowFilterPropertyPage Method

Display Property Pages of a given DirectShow filter

```csharp
public static void ShowFilterPropertyPage(
    IBaseFilter filter,
    IntPtr parent
);
```

Parameters

- `filter` A DirectShow Filter
- `parent` A hwnd handle of a window to contain the pages

Remarks

You can check if a filter supports Property Pages with the `HasPropertyPages` method.

**Warning** : This method is blocking. It only returns when the Property Pages are closed.

Exceptions

<table>
<thead>
<tr>
<th>Exception Type</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>ArgumentNullException</td>
<td>Thrown if filter is null</td>
</tr>
</tbody>
</table>

Example

This sample shows how to check if a filter supports Property Pages and displays them

```csharp
if (FilterGraphTools.HasPropertyPages(myFilter))
{
    FilterGraphTools.ShowFilterPropertyPage(myFilter,
}
```

See Also
FilterGraphTools Class | DirectShowLib.Utils Namespace | HasPropertyPages
A Managed DirectShow Toolkit
DirectShowLib.Utils Hierarchy

**System.Object**

**DirectShowLib.Utils.FilterGraphTools**

**See Also**

**DirectShowLib.Utils Namespace**