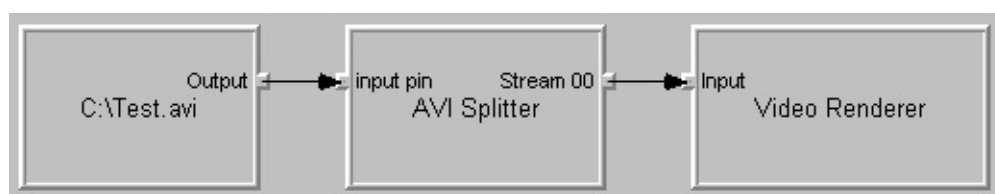


Introduction to GraphEdit

GraphEdit is a visual tool for building and testing filter graphs. It is provided as an executable with the DirectX SDK. With GraphEdit, you can quickly build and test filter graphs to see if they function as you expect. You can even view a filter graph created by an application running in another process.

The following illustration shows a simple filter graph as it appears in GraphEdit.



Filters are displayed as boxes, with a text caption showing the name of the filter. Pins appear as small squares along the edge of the filter. Input pins are on the left, output pins on the right. A pin connection appears as an arrow connecting the output pin to the input pin.

GraphEdit uses a custom persistence format. This format is not supported for application use, but it is helpful for testing and debugging your application.

Note The GraphEdit source code is not provided, and you may not redistribute the executable.

Build a File Playback Graph

GraphEdit can automatically build a filter graph that plays a file. The file can be a local file, or a file located at a URL.

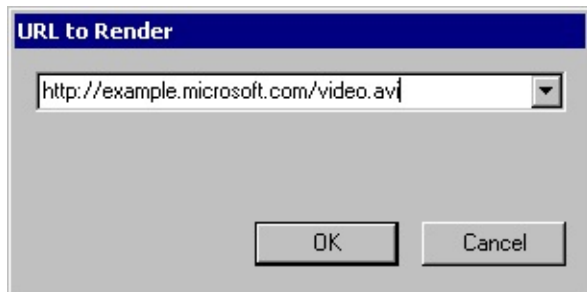
To build a graph that plays a local file:

1. On the **File** menu, click **Render Media File**.
2. Select the file.
3. Click **Open**.

Note You can also drag a file icon to the GraphEdit application window, or to the GraphEdit program icon.

To build a graph the plays a file at a URL:

1. On the **File** menu, click **Render URL**.
2. In the dialog box that appears, type or select the URL.
3. Click **Open**.



These two commands are equivalent to calling the **IGraphBuilder::Render** file method.

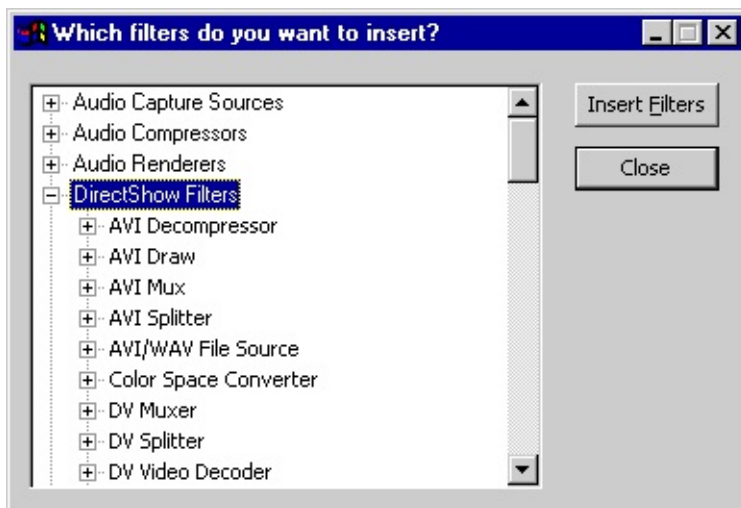
Build a Graph Manually

To build a filter graph manually, perform the following steps.

- [Add Filters](#)
- [Connect Filters](#)
- [Render Pins](#)

Add Filters

1. On the **Graph** menu, click **Insert Filters**.
2. GraphEdit displays a dialog box with a tree view. The tree view contains entries for all the filters registered on your system, enumerated by filter category. (Refer to the following diagram.)
3. In the tree view, click the plus (+) symbol to expand a category.
4. Select a filter name and click **Insert Filters**, or double-click a filter name.
5. Click **Close**.



See Also

[View Filter Registry Data](#)

Connect Filters

Connect two filters by dragging the mouse from an output pin to an input pin. If the pins reject the connection, GraphEdit displays a dialog box with an error message.

If the [Intelligent Connect](#) option is turned on, GraphEdit automatically adds any intermediate filters needed to complete the connection. This is equivalent to calling the **IGraphBuiler::Connect** method. If Intelligent Connect is turned off, GraphEdit attempts to connect the pins directly. This is equivalent to calling the **IFilterGraph::ConnectDirect** method.

Render Pins

Instead of connecting two pins, you can *render* an output pin. When you render a pin, GraphEdit adds the necessary downstream filters and connects them. This is equivalent to calling the **IGraphBuilder::Render** method.

To render an output pin, perform the following steps.

1. Right-click an output pin.
2. On the pop-up menu, click **Render**.

Intelligent Connect

In Microsoft® DirectShow®, *Intelligent Connect* is the mechanism by which the filter graph manager adds intermediate filters when it connects two pins.

For a detailed description of Intelligent Connect, see the DirectShow documentation. Briefly, the filter graph manager searches for filters with matching media types. If more than one filter matches, the filter graph manager tries them in the following order:

- Filters in the [filter cache](#).
- Unconnected filters in the graph.
- Registered filters, in order of merit.

In GraphEdit, you can turn Intelligent Connect on or off by clicking **Connect Intelligent** in the **Graph** menu. This option determines how GraphEdit connects two pins when you manually build a graph. For more information, see [Connect Filters](#).

Filter Cache

For dynamic graph building, the filter graph manager maintains a cache of filters. If an application performs a dynamic reconnection, the filter graph manager can cache filters that it removes from the graph. During [Intelligent Connect](#), the filter graph manager tries cached filters first.

In GraphEdit, you can simulate a dynamic reconnection by manually adding a filter to the cache, as follows:


1. Add a filter to the graph. (See [Build a Graph Manually](#).)
2. Select the filter.
3. In the **Graph** menu, click **Add Filter to the Cache**.

GraphEdit removes the filter from the graph and caches it. To see which filters are in the cache, click **Enum Cached Filters** in the **Graph** menu.

For more information about dynamic graph building in Microsoft® DirectShow®, see **IGraphConfig** in the DirectShow documentation.

Load a GraphEdit File

To load a GraphEdit file, perform the following steps.

1. On the **File** menu, click **Open**.
2. Or, you can click the **Open** icon. The icon is a small square button with a white background and a grey border. It contains a black outline of an open book with a small arrow pointing to the right, indicating an 'Open' action.
3. Select the .grf file you want to open.

Note You can load a GraphEdit file in a Microsoft® DirectShow® application. Call the **IGraphBuilder::RenderFile** method and pass the name of the .grf file. You should not do this in your final application, but it can be useful for application testing.

Load a Graph From an External Process

GraphEdit can load a filter graph created by an external process. With this feature, you can see exactly what filter graph your application builds, with only a minimal amount of additional code in your application.


The application must register the filter graph instance in the Running Object Table (ROT). The ROT is a globally accessible look-up table that keeps track of running objects. For information about how your application can add its filter graph to the ROT, see the topic Using the GraphEdit Utility in the Microsoft® DirectShow® documentation.

To use this feature, perform the following steps.

1. Compile your application with code that adds a ROT entry for the filter graph.
2. Run GraphEdit.
3. Run your application.
4. In the **File** menu, choose **Connect**.
5. In the **Connect To Graph** dialog box, select the process id (pid) of the second application.
6. Click **OK**.

Save a GraphEdit File

To save a GraphEdit file, perform the following steps.

1. On the **File** menu, click **Save**.
2. Or, you can click the **Save** icon. 
3. In the **Save As** dialog box, enter the name of the file.

GraphEdit files use the extension .grf.

Applications should not use GraphEdit files as a persistence format. The format is designed for testing and debugging purposes only.

Save a Filter Graph as XML

Microsoft® DirectX® 8 includes a preliminary version of this feature. This feature may change. Future releases are not guaranteed to support it.

To save a filter graph as an XML file, perform the following steps.

1. On the **File** menu, click **Save As XML...**
2. In the **Save As** dialog box, enter the name of the file.




XML graph files use the extension `.xgr`.

Applications should not use this format as a persistence format. It is designed for testing and debugging purposes only.

Run, Pause, or Stop a Filter Graph

After you build or load a filter graph in GraphEdit, you can run, pause, or stop the graph. You can do this through the menu or the toolbar.

The following table lists the menu command, toolbar button, and equivalent Microsoft® DirectShow® method for each operation.

Command	Menu	Toolbar	DirectShow Method
Run	On the Graph menu, click Play .		IMediaControl::Run
Pause	On the Graph menu, click Pause .		IMediaControl::Pause
Stop	On the Graph menu, click Stop .		IMediaControl::Stop

If the graph is seekable, you can seek using the slider bar that appears below the toolbar.

Set the Reference Clock

If a filter can provide a reference clock, a clock icon appears inside the filter:



If the filter graph manager is using this clock as the filter graph's reference clock, the clock icon is yellow. Otherwise, the icon is gray.

By default, the filter graph manager selects the reference clock. You can select another reference clock, as follows:

1. Right-click the filter.
2. On the pop-up menu that appears, click **Select Clock**.

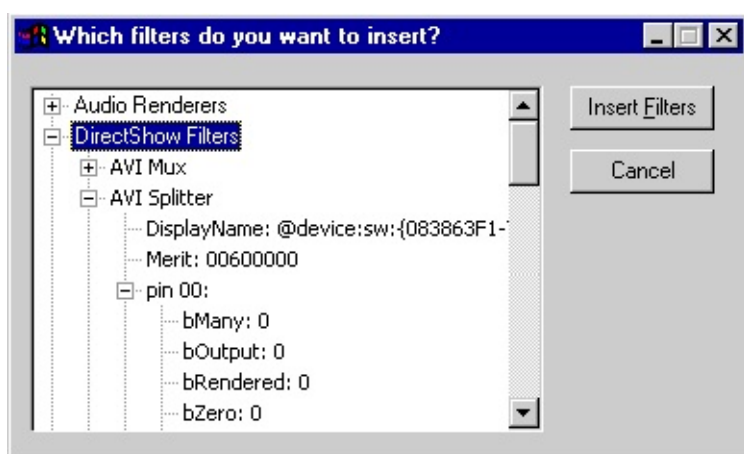
You can enable or disable the reference clock. On the **Graph** menu, click **Use Clock**. A check mark next to this menu item means that the reference clock is enabled. Otherwise, the reference clock is disabled. With no reference clock, samples are rendered as soon as they arrive at the renderer filter.

Note You cannot set the reference clock unless the graph is stopped. For more information, see [Run, Pause, or Stop a Filter Graph](#).

View Filter Registry Data

When you add filters, GraphEdit uses the system device enumerator to enumerate all of the filters registered on your system. They appear in a tree view, listed by category (see [Add Filters](#)).

To view the registry data for a particular filter, click the plus (+) symbol next to the name of the filter.



The following information is displayed:

Filter Information:

Item	Description
DisplayName	The display name provided by the device moniker, as returned by the IMoniker::GetDisplayName method.
Merit	The filter's merit value.
Version	The version of the registration format used to register this filter.

For more information, see REGFILTER2 in the Microsoft® DirectShow® documentation.

Pin Information:

Item	Description
bMany	If 1, the filter might have multiple instances of this pin.
bOutput	If 1, this pin is an output pin.
bRendered	If 1, the filter renders the data from this pin. (Applies only to input pins.)
bZero	If 1, the filter might have zero instances of this pin.
ClsPinCategory	The pin category, taken from the pin property set.

For more information, see REGFILTERPINS and REGFILTERPINS2 in the DirectShow documentation.

Media Type Information:

Item	Description
major type	The major type GUID.
subtype	The subtype GUID.

For more information, see AM_MEDIA_TYPE in the DirectShow documentation.

Medium Information:

Item	Description
medium clsid	The GUID that specifies the medium set.
Data 1	The ID number of the medium.
Data 2	Not used.

For more information, see REGPINMEDIUM in the DirectShow documentation, or KSPIN_MEDIUM in the Microsoft® Windows® 2000 DDK.

View Filter Property Pages

To view a filter's property page, perform the following steps.

1. Add the filter to the graph. (See [Build a Graph Manually.](#))
2. Right-click the filter.
3. On the pop-up menu, click **Properties**.

The property page contains any property sheets defined by the filter. In addition, it contains a property sheet for each pin on the filter. (These are defined by GraphEdit, not by the filter.)

If the pin is connected, the pin property sheet lists the media type for the connection. Otherwise, it lists the pin's preferred media types.

View Filter Graph Statistics

GraphEdit measures statistical information about the filter graph manager's performance.

To view the statistics:

1. On the **View** menu, click **Graph Stats**.
2. GraphEdit displays a dialog box titled **Graph Statistics**.

To refresh the statistics after a graph operation:

- In the **Graph Statistics** dialog box, click **Refresh**.

To reset the statistics to zero:

- In the **Graph Statistics** dialog box, click **Zero**.

Log Graph-Building Operations

GraphEdit can log the steps it takes when building a graph.

1. On the **File** menu, click **Set Log File...**
2. In the **Save As** dialog box, enter the name of the log file.

When you are done using GraphEdit, you can view the file with a text editor.

The following is an excerpt from a typical log file.

```
Render file C:\Example.avi
File has media type 0xe436eb83... Subtype 0xe436eb88...
Source filter clsid is 0xe436ebb5...
Source filter has loaded the source file
RenderFile: Source filter added: Address=87bb4
RenderFile: Render source filter pin 87cc4
Render: Pin 87cc4 has major type 0xe436eb83...
Render: Trying filter at address 87bb4
Render: Examining filter at address 87bb4 for input pin to connect o
Render: No more pins - FAILED to find a pin to use on filter 87bb4
[...]
```