Microsoft Media Foundation

Documentation of used methods

Ahmad Rashed

2024

Contents

[1. Description 1](#_Toc180848863)

[2. Implementation 2](#_Toc180848864)

[2.1. Prereq. 2](#_Toc180848865)

[2.2. Global pointers 2](#_Toc180848866)

[2.3. MFStartup function (mfapi.h) 3](#_Toc180848867)

[2.4. MFCreateAttributes function (mfapi.h) 3](#_Toc180848868)

[2.5. IMFAttributes interface (mfobjects.h) 4](#_Toc180848869)

[2.6. IMFAttributes::SetGUID method (mfobjects.h) 4](#_Toc180848870)

[2.7. IMFActivate interface (mfobjects.h) 5](#_Toc180848871)

[2.8. MFEnumDeviceSources function (mfidl.h) 5](#_Toc180848872)

[3. Source code of full example: 7](#_Toc180848873)

[References 9](#_Toc180848874)

# Description

This document explains the use of Windows Media Foundation in this project to list all connected video capture devices (cameras) and capture raw frames from a selected device.

# Implementation

## Prereq.



## Global pointers



pReader:

The IMFSourceReader interface in Microsoft Media Foundation is used to:

* **Read media samples** (video, audio, metadata) from a source like files or streams.
* **Control stream formats** by setting or getting media types (resolution, color format, sample rate).
* **Manage streams** by enabling/disabling specific streams and handling decoding as needed.

It simplifies media extraction and processing for applications that handle multimedia content.

[1]

pSource:

The IMFMediaSource interface, part of the Media Foundation framework in mfidl.h, represents a media source object, such as a file, network stream, or capture device. It provides key functionality for working with media sources in Windows applications:

1. **Media Stream Management**: Controls and manages multiple media streams (e.g., video and audio) within a media source.
2. **Media Source Control**: Supports starting, stopping, pausing, and managing playback for the media source.
3. **Event Generation**: Generates media events, like MESourceStarted, MESourcePaused, and MESourceStopped, which applications use to handle state changes and respond to playback events.
4. **Initialization and Configuration**: Enables configuration of the source, including format and properties, for compatibility with the application.

The IMFMediaSource is essential for applications needing to manage complex media sources, making it a core part of the Media Foundation pipeline.

[2]

## MFStartup function (mfapi.h)

MFStartup(MF\_VERSION);

The MFStartup function, defined in mfapi.h, initializes the Microsoft Media Foundation platform. This function must be called before using any other Media Foundation functions.

**Key Points:**

* **Purpose**: Initializes internal resources for the Media Foundation API.
* **Parameters**:
  + **Version**: Specifies the version of Media Foundation to load (use MF\_VERSION).
  + **Flags**: Defines initialization options, such as MFSTARTUP\_LITE (for lightweight startup with limited functionality) or MFSTARTUP\_FULL (for full functionality).
* **Usage**: Typically called once at the beginning of a Media Foundation application. Correspondingly, MFShutdown should be called to release resources before the application exits.

**Importance**: Ensures that Media Foundation components are ready for media processing tasks like playback, encoding, and decoding.

## MFCreateAttributes function (mfapi.h)

MFCreateAttributes(&pAttributes, 1);

The MFCreateAttributes function, defined in mfapi.h, creates a new IMFAttributes object, which is used to manage attribute-value pairs for various Media Foundation components.

**Key Points:**

* **Purpose:** To create an attribute store for managing various properties and settings associated with media objects.
* **Parameters:**
  + **ppMFAttributes:** A pointer to an IMFAttributes interface pointer, which receives the newly created attribute object. Provides a generic way to store key/value pairs on an object.
  + **InitialSize:** Specifies the initial size (number of attributes) for the attribute store. This helps optimize memory allocation.
* **Return Value:** The function returns an HRESULT value. On success, it returns S\_OK. If it fails, it returns an appropriate error code.
* **Usage:** Typically used when initializing components that require a set of attributes, such as media sources, sinks, or transforms.

**Importance:** This function is crucial for applications that need to set or retrieve media attributes, ensuring flexibility and configurability in media processing tasks.

[3]

## IMFAttributes interface (mfobjects.h)

IMFAttributes\* pAttributes = nullptr;

The IMFAttributes interface, defined in mfobjects.h, is used in the Microsoft Media Foundation framework to manage collections of attribute-value pairs. This interface provides methods to add, retrieve, and manipulate attributes associated with various media objects.

**Key Points:**

* **Purpose:** To manage a set of attributes that describe properties and settings of media components (e.g., media sources, transforms).
* **Methods:**
  + **GetItem:** Retrieves the value of a specified attribute.
  + **SetItem:** Sets the value of a specified attribute.
  + **DeleteItem:** Removes an attribute from the collection.
  + **GetCount:** Retrieves the number of attributes in the collection.
  + **GetAllItems:** Retrieves all attributes and their values.
  + **…** etc.(check website for the full list.)
* **Usage:** Often used in conjunction with other Media Foundation components to configure settings, such as video format, audio settings, or stream attributes.

**Importance:** The IMFAttributes interface allows applications to be flexible and dynamic in managing media properties, facilitating better control over media processing and playback.

[4]

## IMFAttributes::SetGUID method (mfobjects.h)

pAttributes->SetGUID(MF\_DEVSOURCE\_ATTRIBUTE\_SOURCE\_TYPE, MF\_DEVSOURCE\_ATTRIBUTE\_SOURCE\_TYPE\_VIDCAP\_GUID);

The IMFAttributes::SetGUID method is a member of the IMFAttributes interface in the Microsoft Media Foundation framework. This method is used to set the value of an attribute as a globally unique identifier (GUID).

**Key Points:**

* **Purpose:** To store a GUID value in the attribute collection associated with a media object.
* **Parameters:**
  + **guidKey:** The unique identifier (key) for the attribute you want to set.
  + **guidValue:** The GUID value to associate with the specified key.
* **Return Value:** The method returns an HRESULT value. On success, it returns S\_OK. If it fails, it returns an appropriate error code.
* **Usage:** This method is typically used to define attributes related to media formats, configurations, or settings where GUIDs are needed (e.g., for format identifiers).

**Importance:** The SetGUID method allows applications to define and manage attributes effectively, enabling precise control over media component configurations and behaviors.

[5]

## IMFActivate interface (mfobjects.h)

IMFActivate\*\* ppDevices = nullptr;

The IMFActivate interface in the Microsoft Media Foundation framework is used to represent an object that can be activated, such as a media source, sink, or transform. It provides methods for initializing and configuring these objects before they are used.

**Key Points:**

* **Purpose:** To provide a way to create and configure media objects dynamically at runtime.
* **Methods:**
  + **Activate:** Initializes and creates the media object represented by the IMFActivate instance.
  + **GetAttributes:** Retrieves the attributes associated with the activated object, which can include configuration settings and other properties.
  + **GetStreamCount:** Returns the number of media streams that the activated object can support.
* **Usage:** Typically used in scenarios where media components need to be instantiated based on specific configurations or runtime requirements, such as selecting a media source or transform based on user input or media characteristics.

**Importance:** The IMFActivate interface is essential for managing the lifecycle of media objects in a flexible manner, allowing applications to adapt to different media processing scenarios dynamically.

[6]

## MFEnumDeviceSources function (mfidl.h)

MFEnumDeviceSources(pAttributes, &ppDevices, &deviceCount);

The MFEnumDeviceSources function is part of the Microsoft Media Foundation framework and is used to enumerate available media sources, such as cameras and microphones, that can be used for capturing media.

**Key Points:**

* **Purpose:** To retrieve a list of media source devices (like video capture devices and audio input devices) available on the system.
* **Parameters:**
  + **ppDevices:** A pointer to an array of IMFActivate pointers, which will receive the activated media sources.
  + **pCount:** A pointer to a variable that receives the number of devices in the array.
* **Return Value:** The function returns an HRESULT value. On success, it returns S\_OK. If it fails, it returns an appropriate error code.
* **Usage:** This function is typically called to populate a list of devices that can be used for capturing audio or video, allowing applications to present options to users for selecting a source.

**Importance:** MFEnumDeviceSources is crucial for applications that require media capture capabilities, enabling dynamic selection of input devices based on user preferences or system availability.

[7]

## IMFAttributes::GetAllocatedString method (mfobjects.h)

## WCHAR\* deviceName = nullptr;

UINT32 nameLength = 0;

ppDevices[i]->GetAllocatedString(MF\_DEVSOURCE\_ATTRIBUTE\_FRIENDLY\_NAME, &deviceName, &nameLength);

The IMFAttributes::GetAllocatedString method, part of the IMFAttributes interface in the Microsoft Media Foundation framework, retrieves the string value of a specified attribute and allocates memory for that string. The caller is responsible for freeing the allocated memory.

**Key Points:**

* **Purpose:** To retrieve a string attribute value and allocate memory for it, enabling flexible memory management for string data.
* **Parameters:**
  + **guidKey:** The unique identifier (GUID) of the attribute to retrieve.
  + **ppwszValue:** A pointer that receives the allocated string value. The caller must free this memory using CoTaskMemFree.
  + **pcchLength:** A pointer to receive the length of the string, in characters, excluding the null terminator.
* **Return Value:** Returns an HRESULT value. On success, it returns S\_OK. If the attribute is not a string or is not present, it returns an appropriate error code.
* **Usage:** Often used when reading descriptive or configuration information, such as device names or format descriptions, stored as attributes.

**Importance:** The GetAllocatedString method is essential for safely retrieving variable-length string attributes, helping applications manage memory effectively when working with text-based attribute data.

[8]

## IMFActivate::ActivateObject method (mfobjects.h)

ppDevices[0]->ActivateObject(IID\_PPV\_ARGS(&pSource));

The IMFActivate::ActivateObject method, part of the IMFActivate interface in Microsoft Media Foundation, creates and initializes the media object represented by an IMFActivate instance. This method is commonly used to instantiate objects like media sources, media sinks, or transforms, which are configured by the IMFActivate instance.

**Key Points:**

* **Purpose:** To activate and retrieve a pointer to the media object that the IMFActivate instance represents.
* **Parameters:**
  + **riid:** The interface identifier (IID) of the interface requested on the activated object.
  + **ppv:** Receives the pointer to the requested interface on the activated object.
* **Return Value:** Returns an HRESULT value. On success, it returns S\_OK, and ppv points to the activated object. If activation fails, it returns an error code.
* **Usage:** Typically used to create instances of media components, such as capturing sources or rendering sinks, based on the settings previously configured in the IMFActivate instance.

**Importance:** ActivateObject is crucial for dynamically creating and initializing media components, allowing applications to manage media objects as needed based on user input or runtime configuration.

[9]

### IID\_PPV\_ARGS macro (combaseapi.h)

ppDevices[0]->ActivateObject(IID\_PPV\_ARGS(&pSource));

The IID\_PPV\_ARGS macro in combaseapi.h ensures that the interface identifier (IID) matches the type of interface pointer provided. By automatically deriving the IID from the pointer type, it reduces type mismatch errors in COM methods like QueryInterface and CoCreateInstance.

**Key Points:**

* **Purpose:** Derives the correct IID from the pointer type to prevent mismatches.
* **Parameters:**
  + **ppType:** The interface pointer type.
* **Usage:** Commonly used in COM interface retrieval to improve code reliability.

**Importance:** IID\_PPV\_ARGS enhances code safety and readability in COM programming.

[10]

## MFCreateSourceReaderFromMediaSource function (mfreadwrite.h)

MFCreateSourceReaderFromMediaSource(pSource, pAttributes, &pReader);

The MFCreateSourceReaderFromMediaSource function in Microsoft Media Foundation creates a source reader from an existing media source, enabling simplified access to media streams for reading. It’s commonly used to retrieve samples, control playback, and manage stream formats.

**Key Points:**

* **Purpose:** Simplifies reading samples from an existing IMFMediaSource object.
* **Parameters:**
  + **pMediaSource:** Pointer to the IMFMediaSource object.
  + **pAttributes:** Optional, specifies attributes for the source reader.
  + **ppSourceReader:** Receives the pointer to the IMFSourceReader.
* **Return Value:** Returns an HRESULT, with S\_OK indicating success.

**Importance:** Ideal for applications needing efficient, customizable access to media data for playback or processing.

[11]

## IMFSample interface (mfobjects.h)

IMFSample\* pSample = nullptr;

The IMFSample interface in Microsoft Media Foundation represents a single media sample, such as an audio or video frame, used for processing media data in the pipeline.

**Key Points:**

* **Purpose:** Holds media data (buffers) along with metadata (attributes) for individual samples.
* **Methods:**
  + **AddBuffer** and **RemoveBuffer:** Manage media data buffers in the sample.
  + **GetSampleTime** and **SetSampleTime:** Access and set the timestamp for synchronization.
  + **GetSampleDuration:** Retrieves the sample's duration.
  + … etc.(check website for the full list.)

**Importance:** IMFSample enables fine-grained control over media frames, essential for processing, transforming, and rendering multimedia content accurately.

[12]

## IMFSourceReader::ReadSample method (mfreadwrite.h)

pReader->ReadSample(MF\_SOURCE\_READER\_ANY\_STREAM, 0, &streamIndex, &flags, &llTimeStamp, &pSample);

The IMFSourceReader::ReadSample method retrieves the next sample from a media stream, enabling applications to read and process audio or video data sequentially.

**Key Points:**

* **Purpose:** Reads samples from a specific stream, handling events like end-of-stream and format changes.
* **Parameters:**
  + **dwStreamIndex:** Specifies the stream index.
  + **pdwStreamFlags:** Receives flags for stream status.
  + **ppSample:** Receives a pointer to the IMFSample interface for the sample.
* **Return Value:** Returns an HRESULT to indicate success or an error.

**Importance:** ReadSample is essential for real-time media processing, making it easier to manage sample-based playback or analysis.

[13]

# Source code of full example:





# References

|  |  |
| --- | --- |
| [1] | Microsoft, "IMFSourceReader interface (mfreadwrite.h)," Microsoft, 27 07 2022. [Online]. Available: https://learn.microsoft.com/en-us/windows/win32/api/mfreadwrite/nn-mfreadwrite-imfsourcereader. [Accessed 26 10 2024]. |
| [2] | Microsoft, "IMFMediaSource interface (mfidl.h)," Microsoft, 22 02 2024. [Online]. Available: https://learn.microsoft.com/en-us/windows/win32/api/mfidl/nn-mfidl-imfmediasource. [Accessed 26 10 2024]. |
| [3] | Microsoft, "MFCreateAttributes function (mfapi.h)," Microsoft, 22 02 2024. [Online]. Available: https://learn.microsoft.com/en-us/windows/win32/api/mfapi/nf-mfapi-mfcreateattributes#return-value. [Accessed 26 10 2024]. |
| [4] | Microsoft, „IMFAttributes interface (mfobjects.h),“ Microsoft, 22 07 2021. [Online]. Available: https://learn.microsoft.com/en-us/windows/win32/api/mfobjects/nn-mfobjects-imfattributes. [Zugriff am 26 10 2024]. |
| [5] | Microsoft, "IMFAttributes::SetGUID method (mfobjects.h)," Microsoft, 22 02 2024. [Online]. Available: https://learn.microsoft.com/en-us/windows/win32/api/mfobjects/nf-mfobjects-imfattributes-setguid. [Accessed 26 10 2024]. |
| [6] | Microsoft, „IMFActivate interface (mfobjects.h),“ Microsoft, 22 02 2024. [Online]. Available: https://learn.microsoft.com/en-us/windows/win32/api/mfobjects/nn-mfobjects-imfactivate. [Zugriff am 26 10 2024]. |
| [7] | Microsoft, „MFEnumDeviceSources function (mfidl.h),“ Microsoft, 13 10 2021. [Online]. Available: https://learn.microsoft.com/en-us/windows/win32/api/mfidl/nf-mfidl-mfenumdevicesources. [Zugriff am 26 10 2024]. |
| [8] | Microsoft, "IMFAttributes::GetAllocatedString method (mfobjects.h)," Microsoft, 22 02 2024. [Online]. Available: https://learn.microsoft.com/en-us/windows/win32/api/mfobjects/nf-mfobjects-imfattributes-getallocatedstring. [Accessed 26 10 2024]. |
| [9] | Microsoft, „IMFActivate::ActivateObject method (mfobjects.h),“ Microsoft, 22 02 2024. [Online]. Available: https://learn.microsoft.com/en-us/windows/win32/api/mfobjects/nf-mfobjects-imfactivate-activateobject. [Zugriff am 26 10 2024]. |
| [10] | Microsoft, "IID\_PPV\_ARGS macro (combaseapi.h)," Microsoft, 02 04 2021. [Online]. Available: https://learn.microsoft.com/en-us/windows/win32/api/combaseapi/nf-combaseapi-iid\_ppv\_args. [Accessed 26 10 2024]. |
| [11] | Microsoft, "MFCreateSourceReaderFromMediaSource function (mfreadwrite.h)," Microsoft, 22 02 2024. [Online]. Available: https://learn.microsoft.com/en-us/windows/win32/api/mfreadwrite/nf-mfreadwrite-mfcreatesourcereaderfrommediasource. [Accessed 26 10 2024]. |
| [12] | Microsoft, „IMFSample interface (mfobjects.h),“ Microsoft, 22 07 2021. [Online]. Available: https://learn.microsoft.com/en-us/windows/win32/api/mfobjects/nn-mfobjects-imfsample. [Zugriff am 26 10 2024]. |
| [13] | Microsoft, „IMFSourceReader::ReadSample method (mfreadwrite.h),“ Microsoft, 13 10 2021. [Online]. Available: https://learn.microsoft.com/en-us/windows/win32/api/mfreadwrite/nf-mfreadwrite-imfsourcereader-readsample. [Zugriff am 26 10 2024]. |