MICROSOFT MEDIA FOUNDATION

BY ALEX TRYLYSENKO

Introduction

What is Microsoft Media Foundation?

MICROSOFT MEDIA FOUNDATION

a COM-based multimedia framework pipeline
and infrastructure platform for digital media
in Windows Vista, Windows 7 and Windows 8

- COM (Component Object Model) – is a component interoperability technology
  - Introduced by Microsoft in 1993
  - Provides a way for various objects to interoperate with each other without knowing each other's internal structure or detail

MICROSOFT MEDIA FOUNDATION

- Designed to meet the challenges posed by high-definition content
- Intended to replace Microsoft DirectShow, Windows MediaSDK, DirectX Media Objects (DMOs) and all other “legacy” multimedia APIs such as Audio Compression Manager (ACM) and Video for Windows (VfW)
- The existing DirectShow technology is intended to be replaced by Media Foundation step-by-step, starting with a few features
- For some time there will be co-existence of Media Foundation and DirectShow
- MF will not be available for previous versions of Windows

FEATURES AND IMPROVEMENTS

- DirectX Video Accelerator (DXVA) 2.0 offers a more efficient video acceleration, compared with DXVA 1.0
  - More robust and streamlined video decoding and extended use of hardware in video processing
  - With DXVA 2.0, Windows can handle some of the most demanding HD content with high quality and glitch-resilience

- Color-space information is preserved throughout the video pipeline
  - Color information and interlaced images are passed to hardware for single-pass compositions
  - Preserving color-space information also reduces unnecessary color space conversions, which frees more cycles to process demanding HD content

FEATURES AND IMPROVEMENTS

- The Enhanced Video Renderer (EVR) offers better timing support, enhanced video processing, and improved glitch-resilience
  - Full-screen playback support has been enhanced, and video tearing in windowed mode has been minimized

- MF uses the Multimedia Class Scheduler (MMCSS), a new system service in Windows Vista
  - MMCSS enables multimedia applications to ensure that their time-sensitive processing receives prioritized access to CPU resources

- MF also offers content protection

HISTORY

WHY MICROSOFT MEDIA FOUNDATION?

HISTORY

- 1996-1997 – Together with Internet Explorer 3.0, Microsoft released API to work with media content (i.e. movies) called Quartz
    - This was very convenient set of interfaces and was widely used by the industry
    - Now it is called DirectShow
- Years passed, but DirectShow remains the same.
    - It worked and worked very well
- In 2009 Microsoft decided that change is required and started to design Microsoft Media Foundation
    - Much more generic and extensible
    - But, also much more complicated and detailed

HISTORY

- 1st Release – present in Vista, focused on audio and video playback quality, high-definition content (i.e. HDTV), content protection and a more unified approach for digital data access control for digital rights management (DRM) and its interoperability
  - *DXVA 2.0*
  - Videos are processed in the colorspace they were encoded in
  - *Enhanced Video Renderer (EVR)* and *Multimedia Class Scheduler Service (MMCSS)*

- 2nd Release – included in Windows 7, introduced expanded media format support and DXVA HD for acceleration of HD content if WDDM 1.1 drivers are used

DEVELOPMENT

HOW TO DEVELOP MEDIA FOUNDATION APPLICATIONS?

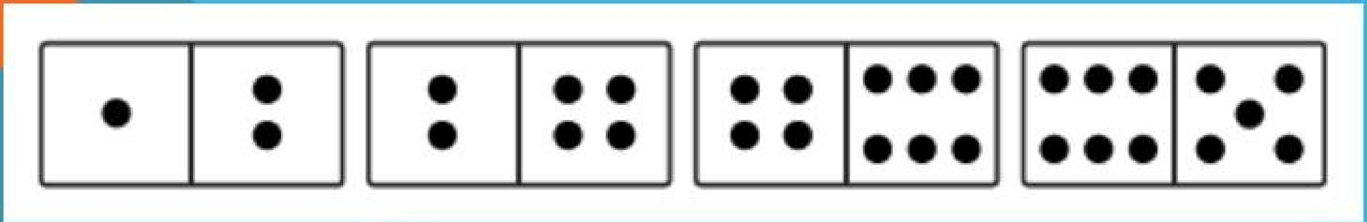
APPLICATIONS

- MMF applications are programs that load and use MF components and modules to process various media data streams

- Some MF applications are designed to simply play back video or audio files

- Others convert the media streams between different formats, store them in different files, and even send and receive media over the internet

- MF applications break up the tasks necessary to process media data streams into multiple steps.
    - Each step is performed by a separate MF component that is loaded into an MF application
    - Different MF components link up together to process the data and do the work in the application just like domino chains



APPLICATIONS

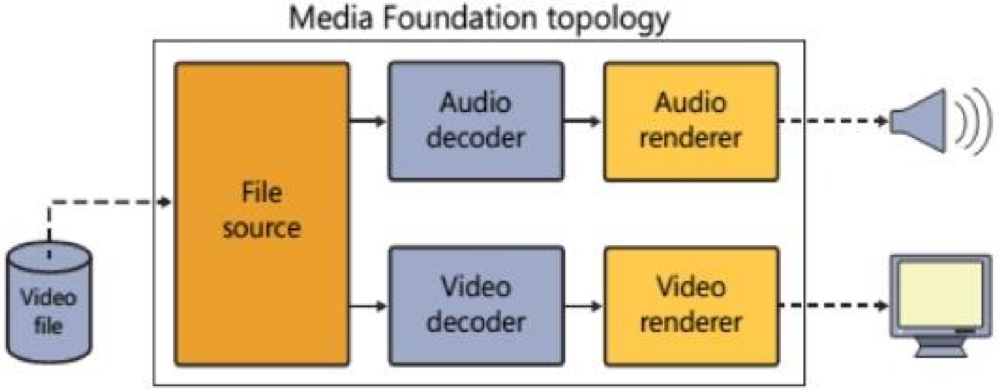
- MF applications are containers for these collections of domino chains, processing data flowing through them

- Each MF application can contain any number of separate chains, and each chain (pipeline) will work on a different data stream.

  - For example, an MF application can be used to play a video file with closed captioning data and audio
  - To play this file, an application would need 3 chains of MF components:
    - Decode and display the video
    - Decode and render the audio
    - Display the subtitle data stream

MEDIA FOUNDATION PIPELINE EXAMPLE

- Steps necessary to play back a video file
  - 1. Load the file from disk
  - 2. Unpack the data streams from the file
  - 3. Separate the audio and video streams for processing by their respective codecs
  - 4. Decode
    - A. Decompress audio data
    - B. Decompress video data
  - 5. Present the uncompressed and decoded information to the user
    - A. Send the audio data to the audio hardware on the PC, and eventually the speakers
    - B. Send the video data to the video card, and display the video on the monitor



- 1-3 are done by file source
- 4 is done by the audio and video decoders
- 5 is done by the renderers

COMPONENTS

- MF sources –load the multiplexed (intertwined) data streams from a file or the network, unpack the elementary audio or video streams from the container, and send them to the other objects in the topology

- Media Foundation Transforms (MFTs) – transform the data in various ways. They accept compressed data as input, transform the data by decoding it, and produce uncompressed information. All MFTs have at least one input link and at least one output link

- MF sinks – responsible for rendering content on the screen or to the audio card, saving data to the hard drive, or sending it over the network. Sinks are essentially the components that extract data from the topology and pass it to the external entities

- The internal implementation of these objects is hidden from the application and the programmer. The only way to communicate to the components is through COM interfaces

- Objects are MF components if and only if they implement specific Media Foundation interfaces

MEDIA TYPES

- A MF media type object describes the type of media in a data stream that is produced or consumed by an MF component

- 2 most important values in a media type are :
  - **Major type** – defines the generic type of data handled by a component (i.e. audio, video, closed captioning, or custom iTV data)
  - **Subtype** – indicates the specific format of the data (i.e. MP3, MPEG2 or H.264)

- A media type can also contain any number of custom data structures and parameters with specific information required to decode the data.

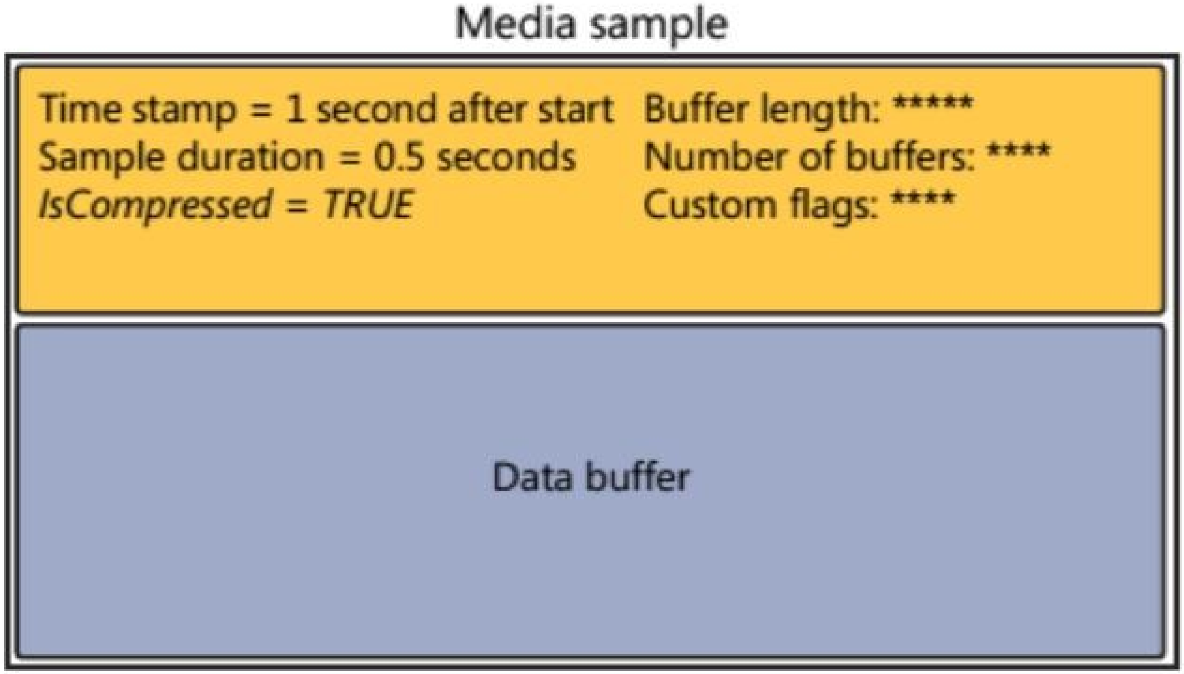
  - For instance, a video media type would usually contain:
    - Frame size
    - Sample size
    - Pixel aspect ratio
    - Frame rate of the video stream
    - And other parameters

MEDIA SAMPLES

- Data is passed between individual components in chunks or packets, called **media samples**

- Each media sample is an object with a data buffer and a set of information describing the data

- Media sample object:



Media sample

Time stamp = 1 second after start
Sample duration = 0.5 seconds
IsCompressed = TRUE

Buffer length: *****
Number of buffers: ****
Custom flags: ****

Data buffer

DATA FLOW EXAMPLE

- 1. The file source loads data from a file, generates a new media sample, and fills it with some of the MP3-encoded audio bits

- 2. The MP3 audio decoder consumes the incoming MP3 audio samples, extracts the compressed audio data from the samples, and releases them.
  - It then decodes the audio data, generates new samples, stores the decoded audio data in them, and then sends those uncompressed samples to the audio renderer

- 3. The audio renderer receives the samples with uncompressed audio and holds onto them.
  - The renderer compares the time stamps in the samples to the current time, and sends the sample data to the audio hardware (through the driver), which in turn generates the sounds.
  - After the renderer is done with the samples, it releases them and request the next sample from the upstream MF components

MEDIA FOUNDATION TOPOLOGIES

- To build an MF pipeline a MF **topology** is needed
- Applications usually use the MF topology builder components provided with Windows
- Topology builders receive various hints about the topology from the application and then automatically discover which components need to be loaded to create a working pipeline
- To give a topology builder the information needed, an application provides it with a partial topology
  - Partial topology usually contains only the source nodes and their corresponding sink nodes
- The topology builder then searches the registry for all MF transforms, instantiates them, and attempts to insert them between the source and the sink
- Custom MFT are used if editing of audio or video is needed

COMPARISON

MEDIA FOUNDATION OR DIRECTSHOW?

COMPARISON

DIRECTSHOW

- Scalable for HD content and DRM
- Excellent resilience to CPU, I/O and memory stress for low-latency glitch-free playback of audio and video
- Well supported by third parties applications and codecs
- Thoroughly tested
- Widely used

MEDIA FOUNDATION

- Scalable for HD content and DRM
- Improved resilience to CPU, I/O and memory stress for low-latency glitch-free playback of audio and video
- Improved video processing support that enables high color spaces and enhanced full-screen playback
- Enables different content protection to operate together
- Uses the Multimedia Class Scheduler Service (MMCSS) to ensure time-sensitive processing receives prioritized access to CPU resources

FUTURE

WHAT IS THE PURPOSE?

FUTURE OF MEDIA FOUNDATION

- Media Foundation has been designed to meet the challenges posed by the demanding HD content, and does provide improvements which are likely to be needed more and more in the future as audio and video quality keep improving

- MF is a fairly new technology and is likely to take time for developers to switch over

- Media Foundation may also seem to be an attempt by Microsoft to pressure users to upgrade at cost to future Windows releases beyond Windows 8

- Whatever the real reason for Microsoft re-inventing their playback programming interface, the fact is it remains easier to develop using DirectShow

APPLICATION SUPPORT

Applications that support Media Foundation include:

- Windows Media Player in Windows Vista and later
- Windows Media Center in Windows Vista and later
- Firefox v24 and later on Windows 7 and later (only for H.264 playback)
- GoldWave 5.60 and later relies on Media Foundation for importing and exporting audio
- Any application that uses Protected Media Path in Windows also uses Media Foundation