Transcode with Scalable Video Coding (SVC)

1 Introduction

This sample demonstrates how to use AMD's hardware-accelerated Media Foundation Transforms (MFTs) for playback and encoding of SVC (Scalable Video Coding) streams.

This sample generates three temporal layers with full frame rate, half frame rate, and quarter frame rate. This sample also shows how to split the temporal layers (using the SVC splitter MFT) into individual H.264 streams and to decode and display the H.264 streams.

2 Using the sample

2.1 Location

$<*installDirectory*>\samples\mediaFoundation\svcTranscode\

2.2 Contents

Package contents

Folder: $<*installDirectory*>\samples\mediaFoundation\svcTranscode\src\

File name	Description
SvcTranscodeApp.cpp	Transcode and Playback application file
SvcTranscodeDlg.cpp	Contains Transcode\Playback dialog class
TranscodeSession.cpp	Contains Transcode and playback topology
MftSvcSplitterTransform.cpp	Contains transform code for SVC splitter
MftSvcSplitterDllmain.cpp	Contains Dll interface functions
CustomStreamSink.cpp	Stream sink interface file
CustomSinkdllmain.cpp	Dll interface file
CustomSinkActivate.cpp	Sink activate interface file
CustomMediaSink.cpp	Custom media sink interface

Folder:
$<*installDirectory*>\samples\mediaFoundation\svcTranscode\inc\

File name	Description
SvcTranscodeApp.h	Header file for transcode/playback application
SvcTranscodeDlg.h	Header file which contains transcode dialog class definition
TranscodeSession.h	Header file for transcode session class
MftSvcSplitterTransformApi.h	Header file for exported function of SVC splitter
MftSvcSplitterTransform.h	Transform class definition file
MftSvcSplitterLib.h	Contains SVC layer splitter Class definition

MftSvcSplitterGuids.h	GUID for SVC splitter MFT
MftSvcSplitter.h	SVC Splitter interface file
CustomStreamSink.h	Custom Stream sink class definition file
CustomSinkLinkList.h	Contains link list class
CustomSinkActivate.h	Contains sink activate interface class
CustomMediaSinkGUIDs.h	GUID for custom media sink
CustomMediasinkActivate.h	Interface for exposing sink activate functionality
CustomMediaSink.h	Media sink class definition file

Folder:

$<*installDirectory*>\samples\mediaFoundation\svcTranscode\res\

File name	Description
SvcTranscode.ico	Icons file
SvcTranscode.rc	GUI window
SvcTranscode.rc2	GUI window
resource.h	Resource file

Folder:

$<*installDirectory*>\samples\mediaFoundation\svcTranscode\build\Windows\

File name	Description
svcTranscodeVs12.sln	Visual Studio 12 solution file
svcTranscodeVs12.vcxproj	Visual Studio 12 project file for SVC transcode
svcTranscodeVs12.vcxproj.filters	Visual Studio 12 project filter file SVC transcode
mftSvcSplitterVs12.vcxproj	Visual Studio 12 project solution file for SVC splitter
mftSvcSplitterVs12.vcxproj.filters	Visual Studio 12 project file for SVC splitter
customSinkVs12.vcxproj	Visual Studio 12 project file for custom sink
customSinkVs12.vcxproj.filters	Visual Studio 12 project filter file for custom sink
svcTranscodeVs10.sln	Visual Studio 10 solution file
svcTranscodeVs10.vcxproj	Visual Studio 10 project file for SVC transcode
svcTranscodeVs10.vcxproj.filters	Visual Studio 10 project filter file for SVC transcode
mftSvcSplitterVs10.vcxproj	Visual Studio 10 project file for SVC splitter
mftSvcSplitterVs10.vcxproj.filters	Visual Studio 10 project file for SVC splitter
customSinkVs10.vcxproj	Visual Studio 10 project file for custom sink
customSinkVs10.vcxproj.filters	Visual Studio 10 project filter file for custom sink

Folder:

$<*installDirectory*>\samples\mediaFoundation\svcTranscode\docs\

File name	Description
MediaSDK_MFT_svcTranscode.pdf	Documentation

2.3	**Compile**	1. Ensure that the following tools and SDKs are present:

- ☐ Microsoft Visual Studio 2010 or 2012
  If Windows Software Development Kit (SDK) is not installed, install it from
  http://msdn.microsoft.com/en-us/library/windows/desktop/hh852363.aspx.

2. Open one of the following two solution files:

- ☐ `<install_dir>\MediaSDK\samples\mediaFoundation\svcTranscode\build\windows\svcTranscodeVs12.sln`

- ☐ `<install_dir>\MediaSDK\samples\mediaFoundation\svcTranscode\build\windows\svcTranscodeVs10.sln`

3. Build the sample:

- ☐ Open the `svcTranscodeVs10.sln` solution file with Microsoft Visual Studio 2010 Professional Edition or the `svcTranscodeVs12.sln` solution file with Microsoft Visual Studio 2012 Professional Edition.

- ☐ To build all solutions, select `Build > Build Solution`.

- ☐ In the Solutions Explorer, select the project file.
  The executable `svcTranscode.exe` is created in the following folders for 32-bit builds and 64-bit builds:
  `$<installDirectory>\samples\mediaFoundation\svcTranscode\bin\x86\`
  `$<installDirectory>\samples\mediaFoundation\svcTranscode\bin\x86_64\`
  This sample uses the DLL from the DirectX 11 video rendering sample from Microsoft sample. This DLL is copied from:

- ☐ `<installDirectory>\thirdParty\components\DirectX video rendering sample\C++\Win32\Debug` or `Release\DX11VideoRenderer.dll` for 32-bit builds, and from

- ☐ `<installDirectory>\thirdParty\components\DirectX video rendering sample\C++\x64\Debug` or `Release\DX11VideoRenderer.dll` for 64-bit builds.

3 Run

The sample can be executed on an AMD platform that includes the VCE and UVD hardware blocks.
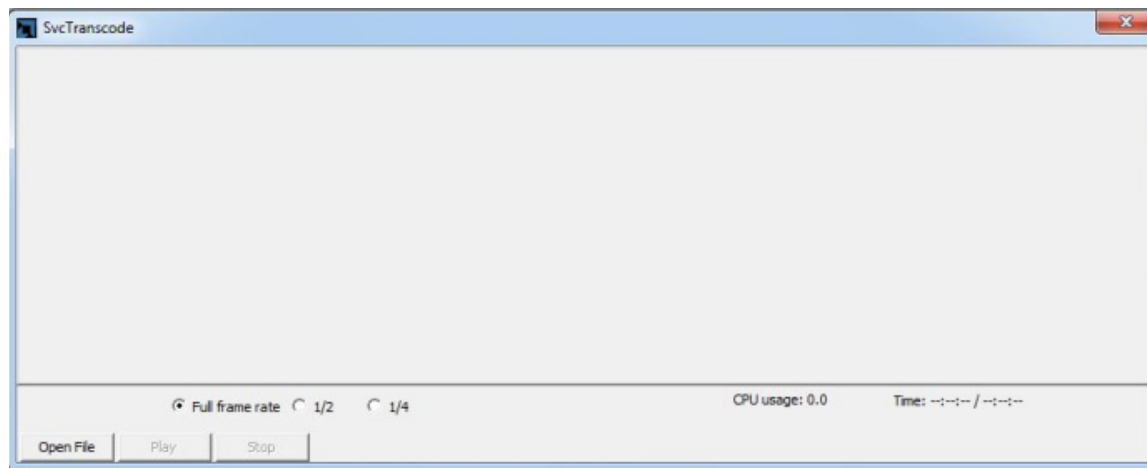
To run the sample, press Ctrl+F5.
The following screen is displayed.
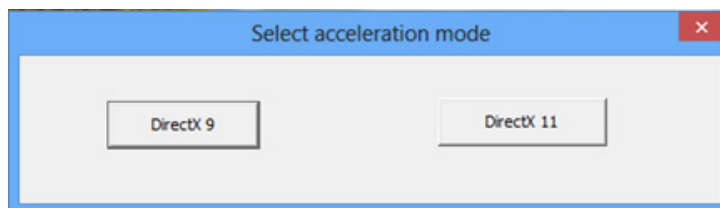
This screen is not displayed on Windows 7 as only DirectX 9 is supported.

On Windows 8/8.1, select the appropriate DirectX version. The following screen is displayed.

- `Open File`: Use this option to browse an input media file (for e.g. AVI, MP4, WMV, ASF, MPG, MPEG)

- `Play`: Use this option to play the input media file.

- `Stop`: Use this option to stop playing the input media file.

4 Implementation Details

The sample implements the following playback pipeline:
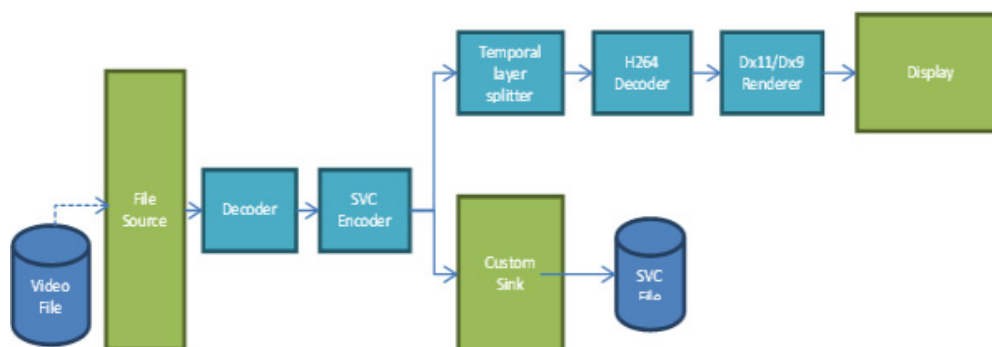
5 Supported formats

The following file formats are supported:

- Input file/container formats: .avi, .mp4

- Video decoders supported: H264

Transcode with Scalable Video Coding (SVC)

6 Known limitations

The sample is currently supported on the following platforms:

- Windows 7 (DirectX 9)

- Windows 8/8.1 (DirectX 9 and DirectX 11)