nanocosmos H.264 Video Encoder Filter

Author: Ulrich Pflüger

Date: 2011-10-10

Am Borsigturm 40

D-13507 Berlin

info@nanocosmos.de

Module / Version

nanocosmos H.264 DirectShow Video Encoder Filter
nh264enc.ax Version 2.5.5.2

DirectShow Connectivity

The input is accepting connections to video source, capture and decoder filters matching the following media types:

Major types:
MEDIATYPE_Video
Subtypes:
MEDIASUBTYPE_YV12,
MEDIASUBTYPE_I420,
MEDIASUBTYPE_YUY2,
MEDIASUBTYPE_RGB24,
MEDIASUBTYPE_RGB32,
MEDIASUBTYPE_ARGB32,
MEDIASUBTYPE_RGB565,
Formats:
FORMAT_VideoInfo
FORMAT_VideoInfo2

The output supports these media types:

Major types:
MEDIATYPE_Video
Subtypes:
FourCCs: H264, h264
Formats:
FORMAT_MPEG2_VIDEO,
FORMAT_NONE

Configuration

The encoding configuration may be set by using either the property page or the COM Interface INanoCodecOpts as declared in header file INanoCodecOpts.h .

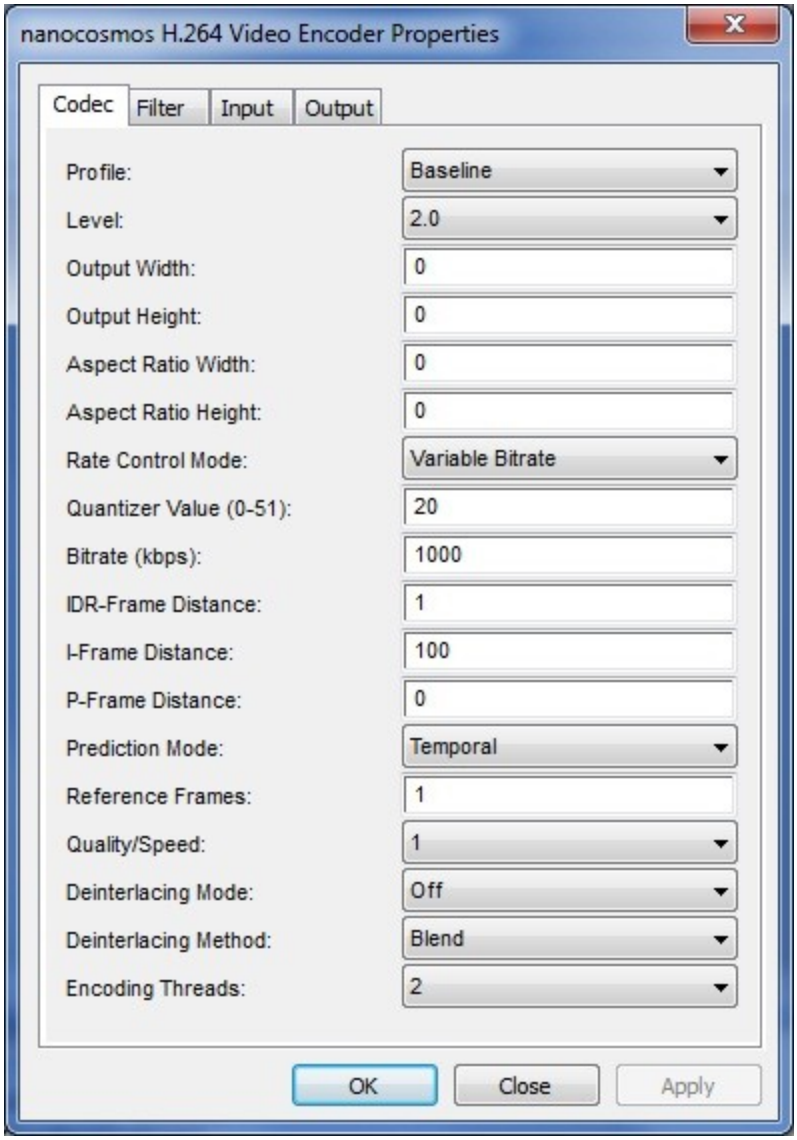
```
// Filter GUID
// {A88889A8-3C2A-4a32-8EAA-755D491D02A0}
DEFINE_GUID(CLSID_Nanocosmos_H264_Encoder,
0xa88889a8, 0x3c2a, 0x4a32, 0x8e, 0xaa, 0x75, 0x5d, 0x49, 0x1d, 0x2,
0xa0);

// Property Page GUID
// {7A8F0A5A-AEDC-4fa5-AB94-2E3FB1268DB3}
DEFINE_GUID(CLSID_Nanocosmos_H264_Encoder_PropPage,
0x7a8f0a5a, 0xaedc, 0x4fa5, 0xab, 0x94, 0x2e, 0x3f, 0xb1, 0x26, 0x8d,
0xb3);

// Configuration Interface INanoCodecOpts
// {698E0F57-B828-4c40-8867-095FF49F77D6}
DEFINE_GUID(IID_INanoCodecOpts,
0x698e0f57, 0xb828, 0x4c40, 0x88, 0x67, 0x9, 0x5f, 0xf4, 0x9f, 0x77,
0xd6);

// Configuration Interface ICodecProp
// {0F817204-82C8-4c12-884A-F45FB2F33A6E}
DEFINE_GUID(IID_ICodecProp,
0xf817204, 0x82c8, 0x4c12, 0x88, 0x4a, 0xf4, 0x5f, 0xb2, 0xf3, 0x3a, 0x6e);


// ICodecProp: IID_nanoPeg_LicenseString
// type: BSTR / Unicode string
// Set license string to unlock filter
// {1788F0B0-5985-4a19-B7FE-8AAC1BFC14B3}
DEFINE_GUID(IID_nanoPeg_LicenseString,
0x1788f0b0, 0x5985, 0x4a19, 0xb7, 0xfe, 0x8a, 0xac, 0x1b, 0xfc, 0x14,
0xb3);
```

Setting the license to unlock filter

The filter can be unlocked either through a license key entry in the windows registry or by setting the license key through COM interface ICodecProp::SetProperty with the property `IID_nanoPeg_LicenseString` as first parameter. The second license parameter has to be a wide/unicode string!

Configuration through DirectShow filter property page

The filter's property page offers a subset of encoding parameters, containing the most important options.



Configuration through INanoCodecOpts interface

1. Version check by calling *GetCodecOptsVersion* (optional)
2. Instantiating a parameter structure of type *MPDX4_H264EncoderParams*
3. Setting desired values for profile and level *(profile_idc, level_idc)*
4. Initializing the parameter struct by calling *InitCodecOptions* will set all parameters to default values for the selected profile and level
5. Setting custom values for resolution and bitrate
6. Applying settings by calling *SetCodecOptions*

Common Encoder Settings

Parameter	Default values	Description
profile_idc, level_idc	Baseline, 1.2	H.264 Profile and Level
base_video.bitrate_kb	Profile and Level dependend	Video Bitrate in kBits/second
base_video.resolution.width	0 – use input width	Picture Coding Width
base_video.resolution.height	0 – use input height	Picture Coding Height
entropy_coding_mode	Baseline+Extended: 0-CAVLC other profiles: 1-CABAC	[0,1] Entropy Coding Mode
coding_type	0 – Frame/progressive	[0,1] Frame/Field coding type
num_ref_frames	2	[1..16] number of ref frames
key_frame_controls.distance_i	20	Intra Frame Distance / vop len.
distance_p	Baseline 1, other profiles 2	P Frame Distance

Code sample:

```
HRESULT hr;

MPDX4_H264EncoderParams optionsH264;

// set lowest desired level here, lower level means lower latency
// level will be automatically increased by encoder if bitrate,
// resolution and framerate settings make a higher level necessary
optionsH264.profile_idc = MPDX4_H264_BASE_PROFILE;
optionsH264.level_idc = MPDX4_H264LEVEL_30;

// perform profile/level based initialization of parameter structure
hr = m_EncoderOpt->InitCodecOptions((MPDX4_BaseCodecOpts*) &optionsH264,
INANOCODECOPTS_VERSION);

// choose constant bitrate for network streaming
optionsH264.rate_controls.method = MPDX4_H264RATEM_CBR;

// set desired bitrate in kByte
optionsH264.base_video.bitrate_kb = 800; // only example value

// set GOP structure
// distance between two i-frames (gop length)
optionsH264.key_frame_controls.distance_i = 50; // only example value

// distance between two p-frames, meaning:
// 1: no b-frames, 2: 1 b-frames, 3: 2 b-frames
// in Baseline profile this settings will be overridden (1) by encoder
optionsH264.distance_p = 1; // (no B-frames allowed in Baseline profile)

hr = m_EncoderOpt->SetCodecOptions((MPDX4_BaseCodecOpts*) &optionsH264,
INANOCODECOPTS_VERSION);
```

Debug-Log Configuration Registry Settings

Key: HKEY_CURRENT_USER\Software\DebugNano\nh264enc.ax

File name

Sets the output file name. The folder must already exist.
Value name: LogToFile
Value type: REG_SZ / String
Valid values: a valid output file name to enable file logging or an empty string

Logging level

A higher value increases the amount of logging messages sent, and messages get more detailed.
Value name: TRACE
Value type: REG_DWORD
Valid values:
0 – minimal logging
…
9 – maximal logging

Embedding / Customizing nanocosmos technology

Nanocosmos also offers special OEM and customization service. Dependent on your needs, we may offer different models of licensing or application development.

About nanocosmos

nanocosmos informationstechnologien gmbh is a software company providing professional video coding and streaming products. We are partners for digital TV and video/audio streaming for intra- and internet applications. Nanocosmos provides encoding and decoding solutions for any video standard on any platform, with specific expertise in plugin architectures and DirectShow filters. Based on core technology, additional support and development service is available for custom applications.

Contact us at http://www.nanocosmos.net or info@nanocosmos.de for further information.