

## Introduction

The MainConcept FFmpeg Plugins enable you to use MainConcept's industry-leading codecs directly within FFmpeg. If you are already familiar with FFmpeg, working with the MainConcept plugins will be easy. However, there are still some things that should be considered, and if you are new to FFmpeg you need to know some basics before getting started.

This tutorial explains the command line structure required for working with the MainConcept FFmpeg Plugins. We try not to focus on either AVC or HEVC but instead cover both these powerful video codecs that are widely used today. As you will discover, the process to use each of these MainConcept video encoders in FFmpeg is almost identical.

## Command Line Structure

Rather than covering the full installation procedure here, we have created video tutorials that can walk you through the installation using [Windows](#), [Ubuntu](#) and [CentOS Linux](#). For now, let's jump right into how to use the plugins. To begin, launch FFmpeg from the location where you have it installed together with the corresponding MainConcept Plugins (if you're unsure how to do this, details are available in the above video tutorials).

Different libraries are required to get the whole MainConcept FFmpeg Plugin product framework working. A special OMX interface makes sure that the communication between FFmpeg and the plugins is working correctly. The workflow is based on four items:

- The modified FFmpeg executable exposing the **OMX interface** (*ffmpeg*).
- The **MainConcept OMX Core** that handles the communication and loading of the plugins (*omxil\_core*).
- The actual **MainConcept OMX Codec Plugins** for each product (*omxil\_encavc*, *omxil\_encevc*).
- The actual **MainConcept Video Encoder** Low Level libraries (*demo/mc\_enc\_avc*, *demo/mc\_enc\_hevc*).

The modified FFmpeg using our patch enables the MainConcept Plugin to be queried and dynamically loaded at FFmpeg runtime.

To a certain extent, this structure is also reflected within the FFmpeg command line itself. As a first item you need to specify the *ffmpeg* executable in the command line prompt.

# Command Line Structure & Usage

## MainConcept FFmpeg Plugin Tutorial

```
Demo Plug-Ins for FFmpeg
C:\Program Files\MainConcept\FFmpeg-PlugIns\bin>cmd.exe
Microsoft Windows [Version 10.0.17134.1550]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Program Files\MainConcept\FFmpeg-PlugIns\bin>ffmpeg _
```

Next, specify the input stream (-i) followed by FFmpeg general parameters, such as the resolution, bitrate, framerate, etc. Of course, many of the parameters can be specified using the built-in MainConcept presets or within the MainConcept Config files.

Next you need to specify the video codec (-c:v) you want to use, i.e. the MainConcept AVC/H.264 or HEVC/H.265 video encoder. The two possible options are *omx\_enc\_avc* or *omx\_enc\_hevc*.

```
Demo Plug-Ins for FFmpeg
C:\Program Files\MainConcept\FFmpeg-PlugIns\bin>cmd.exe
Microsoft Windows [Version 10.0.17134.1550]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Program Files\MainConcept\FFmpeg-PlugIns\bin>ffmpeg -i C:\Users\fschoenberger\Desktop\Test\Wildlife.wmv -vf scale=416:234 -c:v omx_enc_avc
```

Now specify the core of the MainConcept FFmpeg Plugins (-omx\_core). This is responsible for loading and unloading the plugin components. The core basically manages the communication between FFmpeg and the actual encoder plugins. It is the same for all MainConcept FFmpeg Plugins [*omxil\_core.dll* (Windows) or *omxil\_core.so* (Linux)] regardless of the codec.

```
Demo Plug-Ins for FFmpeg
C:\Program Files\MainConcept\FFmpeg-PlugIns\bin>cmd.exe
Microsoft Windows [Version 10.0.17134.1550]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Program Files\MainConcept\FFmpeg-PlugIns\bin>ffmpeg -i C:\Users\fschoenberger\Desktop\Test\Wildlife.wmv -vf scale=416:234 -c:v omx_enc_avc -omx_core omxil_core.dll
```

As the next step, you need to specify the actual OMX encoder plugin (AVC or HEVC) using the correct identifier (-omx\_name). Depending on which encoder plugin you want to use (*OMX.MainConcept.encavc.video* or *OMX.MainConcept.enchevc.video*), it loads the corresponding MainConcept video encoder libraries. Based on this, it enables specific MainConcept parameters that can be defined on the command line.

```
Demo Plug-Ins for FFmpeg
C:\Program Files\MainConcept\FFmpeg-PlugIns\bin>cmd.exe
Microsoft Windows [Version 10.0.17134.1550]
(c) 2018 Microsoft Corporation. All rights reserved.

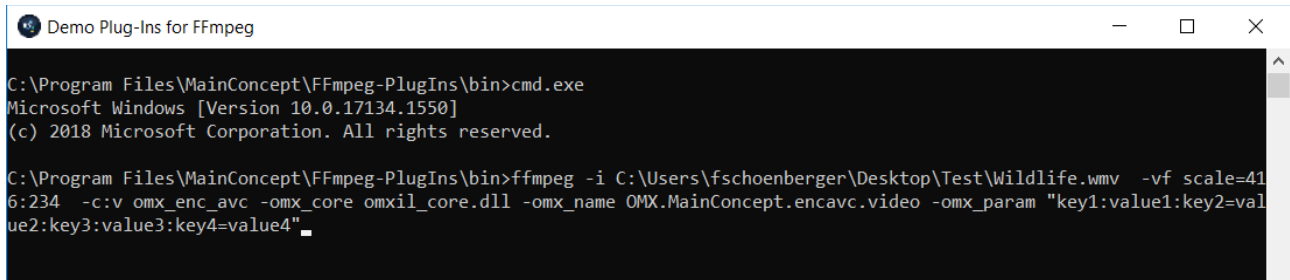
C:\Program Files\MainConcept\FFmpeg-PlugIns\bin>ffmpeg -i C:\Users\fschoenberger\Desktop\Test\Wildlife.wmv -vf scale=416:234 -c:v omx_enc_avc -omx_core omxil_core.dll -omx_name OMX.MainConcept.encavc.video _
```



# Command Line Structure & Usage

## MainConcept FFmpeg Plugin Tutorial

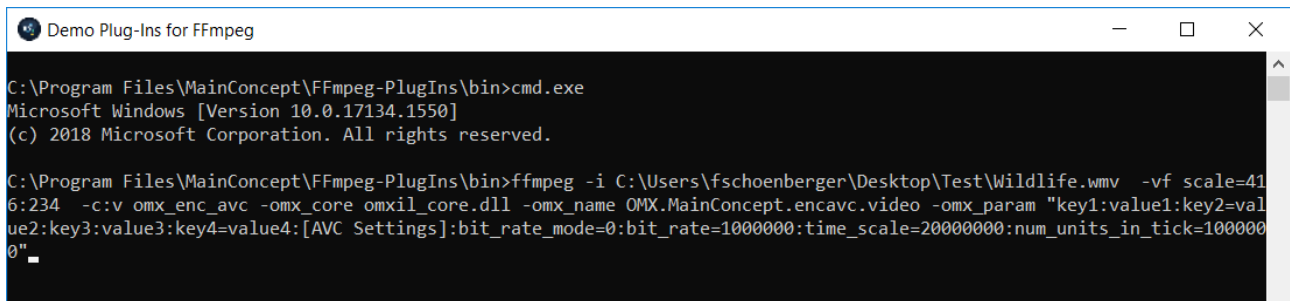
After you have specified these options, it is time to start defining the general MainConcept parameters, e.g. a built-in preset, performance level, software / hardware encoding, loading of Config file, etc. Please have a look at the user guides that come with MainConcept FFmpeg Plugins for more information about the available settings. These optional parameters are introduced with `-omx_param`. Keep in mind that the syntax is always a string of "key=value" pairs separated by colons (:).



```
C:\Program Files\MainConcept\FFmpeg-PlugIns\bin>cmd.exe
Microsoft Windows [Version 10.0.17134.1550]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Program Files\MainConcept\FFmpeg-PlugIns\bin>ffmpeg -i C:\Users\fschoenberger\Desktop\Test\Wildlife.wmv -vf scale=416:234 -c:v omx_enc_avc -omx_core omxil_core.dll -omx_name OMX.MainConcept.encavc.video -omx_param "key1:value1:key2=value2:key3:value3:key4=value4"
```

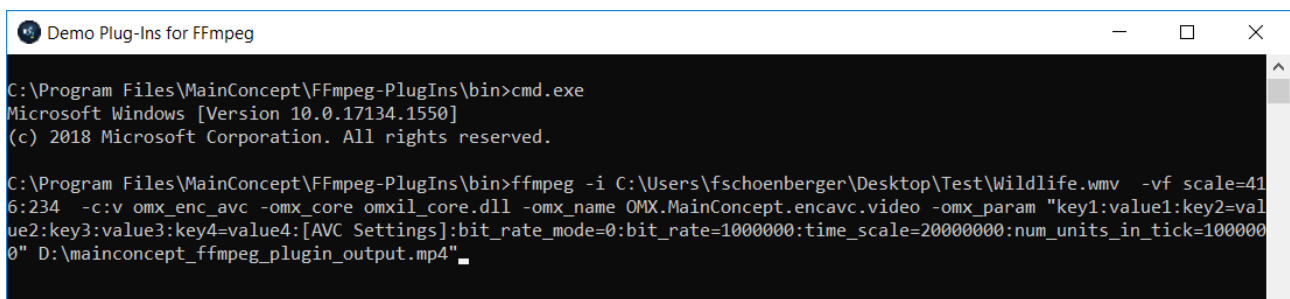
As mentioned above, you can either load a separate Config file with all settings pre-defined or specify the specific parameters directly on the command line. In this case, the arguments are matching the video encoder configurations from the corresponding MainConcept SDKs. The user guides that are included in the product packages provide a detailed overview of the encoder settings. The structure and the order must be the same as it is written in the MainConcept Config (\*.ini) files, e.g. *[AVC Settings]*, *[HEVC Settings]* and/or *[HEVC Layer 0000]*. The syntax of the command line is similar to what has been introduced in the previous paragraph.



```
C:\Program Files\MainConcept\FFmpeg-PlugIns\bin>cmd.exe
Microsoft Windows [Version 10.0.17134.1550]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Program Files\MainConcept\FFmpeg-PlugIns\bin>ffmpeg -i C:\Users\fschoenberger\Desktop\Test\Wildlife.wmv -vf scale=416:234 -c:v omx_enc_avc -omx_core omxil_core.dll -omx_name OMX.MainConcept.encavc.video -omx_param "key1:value1:key2=value2:key3:value3:key4=value4:[AVC Settings]:bit_rate_mode=0:bit_rate=1000000:time_scale=20000000:num_units_in_tick=1000000"
```

Finally, don't forget to specify the output filename and location. Once this is done, you are ready to start the encoding.



```
C:\Program Files\MainConcept\FFmpeg-PlugIns\bin>cmd.exe
Microsoft Windows [Version 10.0.17134.1550]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Program Files\MainConcept\FFmpeg-PlugIns\bin>ffmpeg -i C:\Users\fschoenberger\Desktop\Test\Wildlife.wmv -vf scale=416:234 -c:v omx_enc_avc -omx_core omxil_core.dll -omx_name OMX.MainConcept.encavc.video -omx_param "key1:value1:key2=value2:key3:value3:key4=value4:[AVC Settings]:bit_rate_mode=0:bit_rate=1000000:time_scale=20000000:num_units_in_tick=1000000" D:\mainconcept_ffmpeg_plugin_output.mp4"
```

Now that you understand the fundamentals, in the following section we will take a closer look at a complete command line to execute an encoding using the MainConcept FFmpeg Plugins.



### MainConcept FFmpeg Plugin Usage

In the previous section you learned about the general structure and syntax of the MainConcept FFmpeg Plugins. Now let's take a closer look at an example for AVC/H.264 encoding.

Here is a command-line example:

```
ffmpeg -r 25.000000 -pix_fmt yuv420p -s 1920x1080 -i "1920x1080p_25p_YV12.yuv" -vf
scale=1280:720 -b:v 3500k -c:v omx_enc_avc -omx_core omxil_core.dll -omx_name
OMX.MainConcept.encavc.video -omx_param "preset=main:perf_level=10:acc_type=sw:[AVC
Settings]:bit_rate_mode=0:bit_rate=100000:time_scale=20000000:num_units_in_tick=1000000"
"1920x1080p_25p_YV12_ffmpeg.mp4"
```

At first glance it looks pretty simple, but it has some special considerations you should be aware of.

The first part of the command line deals with the input stream. It specifies a YUV 4:2:0 RAW stream that has 1920x1080 resolution and a framerate of 25 fps:

```
ffmpeg -r 25.000000 -pix_fmt yuv420p -s 1920x1080 -i "1920x1080p_25p_YV12.yuv"
```

The following options are general parameters from FFmpeg itself. In this case, we create a filter graph to downscale the input stream to 720p (`-vf scale`) and set the output video bitrate to 3.5 mbit (`-b:v`):

```
ffmpeg -r 25.000000 -pix_fmt yuv420p -s 1920x1080 -i "1920x1080p_25p_YV12.yuv" -vf
scale=1280:720 -b:v 3500k
```

Now the MainConcept FFmpeg Plugins come into play. As outlined in the previous section, you need to specify the codec identifier (`-c:v`), the MainConcept OMX core (`-omx_core`) as well as plugin identifier (`-omx_name`) on the command line.

```
ffmpeg -r 25.000000 -pix_fmt yuv420p -s 1920x1080 -i "1920x1080p_25p_YV12.yuv" -vf
scale=1280:720 -b:v 3500k -c:v omx_enc_avc -omx_core omxil_core.dll -omx_name
OMX.MainConcept.encavc.video
```

The general MainConcept parameters are up next, introduced by the `-omx_param` option. We specify the generic AVC/H.264 Main preset (`-preset`) with performance level 10 (`-perf_level`) to get a very good compromise between quality and speed. In addition, we do software encoding (`-aac_type`) using the renowned MainConcept AVC/H.264 Video Encoder.

```
ffmpeg -r 25.000000 -pix_fmt yuv420p -s 1920x1080 -i "1920x1080p_25p_YV12.yuv" -vf
scale=1280:720 -b:v 3500k -c:v omx_enc_avc -omx_core omxil_core.dll -omx_name
OMX.MainConcept.encavc.video -omx_param "preset=main:perf_level=10:acc_type=sw:[AVC
Settings]:bit_rate_mode=0:bit_rate=100000:time_scale=20000000:num_units_in_tick=1000000"
```



# Command Line Structure & Usage

## MainConcept FFmpeg Plugin Tutorial

As you can see in example above, the *[AVC Settings]* parameter starts the configuration of the codec specific settings on the command line. The Bitrate Control mode is set to Constant Bitrate (*bit\_rate\_mode=0*). The actual target bitrate in this example is adjusted to 10 mbit (*bit\_rate=10000*).

Do you remember the "special considerations" raised at the beginning of this section? Now it is time to come back to them. As you can see in the command line, we specified the bitrate twice. First, we set the bitrate in FFmpeg generic global options (*-b:v 3500k*) and then also in the OMX parameters, i.e. in the MainConcept Video Encoder directly (*bit\_rate=100000*). And the latter value will be used! Why? This is simple! In this case, the MainConcept codec-specific settings are prioritized over the FFmpeg generic global ones because we have explicitly specified them on the command line in the *-omx\_param* options. In general, the MainConcept encoder-specific settings always have higher priority than the FFmpeg ones. In the end, the encoded file should have a bitrate of 10 mbit.

Furthermore, we set two additional parameters: the number of time units that pass in one second (*time\_scale=20000000*) and the number of time units of a clock operating at the frequency *time\_scale* that corresponds to one clock tick (*num\_units\_in\_tick=1000000*). This sets the target video framerate. Please refer to the [user guide](#) that comes with the MainConcept AVC Encoder Plugin for FFmpeg to learn more about these settings.

Of course, you can specify as many options on the command line as needed to tweak your settings. If you do not want to enter all options on the command line, you can also load a MainConcept encoder config file (\*.ini). Here is an example:

```
ffmpeg -r 25.000000 -pix_fmt yuv420p -s 1920x1080 -i "1920x1080p_25p_YV12.yuv" -c:v omx_enc_avc  
-omx_core omxil_core.dll -omx_name OMX.MainConcept.encavc.video -omx_param  
"cfg_file_path=D:\Configs\avc_config.ini" "1920x1080p_25p_YV12.mp4"
```

You might be wondering why we have neither specified an audio encoder nor a multiplexer. In this case, the FFmpeg default settings will be applied, i.e. FFmpeg automatically uses the built-in AAC encoder, and – defined by the file extension – uses the MP4 Multiplexer to create a valid file.

As a final step, we need to enter the output filename and location on the FFmpeg command line.

```
ffmpeg -r 25.000000 -pix_fmt yuv420p -s 1920x1080 -i "1920x1080p_25p_YV12.yuv" -vf  
scale=1280:720 -b:v 3500k -c:v omx_enc_avc -omx_core omxil_core.dll -omx_name  
OMX.MainConcept.encavc.video -omx_param "preset=main:perf_level=10:acc_type=sw:[AVC  
Settings]:bit_rate_mode=0:bit_rate=100000:time_scale=20000000:num_units_in_tick=1000000"  
"1920x1080p_25p_YV12_ffmpeg.mp4"
```

Start the encoding in FFmpeg. As you can see, FFmpeg and the MainConcept AVC/H.264 Video Encoder share some useful statistics while processing the file.



# Command Line Structure & Usage

## MainConcept FFmpeg Plugin Tutorial

```
Demo Plug-Ins for FFmpeg
Bits/sec:  avg =  144025

-----

I frames:      QP avg = 35.750  QP min = 33  QP max = 38
P frames:      QP avg = 37.863  QP min = 34  QP max = 41
B frames:      QP avg = 40.703  QP min = 38  QP max = 43
All frames:    QP avg = 39.170  QP min = 33  QP max = 43

-----

Total size: 541835 bytes (529.14 KB)
Total time: 00:00:00:04.655, time per frame: 0.005 (0.003) seconds

-----

Frames (incoming/encoding): 902/902
Speed (incoming/encoding): 0.00/193.77(372.42) fps
Load (average):              0.00
Delay (average):              922.937 ms

-----

Average CPU load:  22.2 %, process 19.5 %
Peak memory usage: 107.910 Mb
Peak memory commit: 139.234 Mb

-----

[aac @ 0000018817CA9AC0] Qavg: 582.589
C:\Program Files\MainConcept\FFmpeg-PlugIns\bin>
```

Once the encoding is done, FFmpeg will inform you.

## Closing Words

If you are already familiar with FFmpeg, working with the MainConcept Plugins is very easy. However, even if you are new to FFmpeg, with the MainConcept Plugins you quickly obtain great quality encoding output in tremendous speed. You can modify the video encoder settings in the same way as with MainConcept SDKs. Once you know the syntax and structure of these FFmpeg Plugins, your video production workflows will be powered by MainConcept's industry leading codec technology.

## More of a visual learner?

Check out our video tutorial series. We cover everything from the basics of encoding in FFmpeg, to more advanced topics like hardware acceleration. Check out the full video series, [here](#).



## Credits

*Copyright © 2020 MainConcept GmbH or its affiliates. All rights reserved.*

*MainConcept® and its logos are registered trademarks of MainConcept GmbH or its affiliates. This software is protected by copyright law and international treaties. Unauthorized reproduction or distribution of any portion is prohibited by law.*

*This manual, as well as the software described in it, is furnished under license and may only be used or copied in accordance with the terms of such license. The information in this manual is furnished for informational use only, is subject to change without notice, and should not be construed as a commitment or representation by MainConcept GmbH or its affiliates. MainConcept GmbH and its affiliates assumes no responsibility or liability for any errors or inaccuracies that may appear in this book and use is at your sole risk.*

*Except as permitted by such license, no part of the publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, recording, or otherwise, without the prior written permission of MainConcept GmbH.*

*Edition: August 2020*

