

# CTR Developer's Guide - Production: Mobiclip Conversion

Nintendo of America  
Software Development Support Group  
Last updated: 2/5/2015  
Version 1.3

## Contents

1. Introduction .....	2
2. Software Environment .....	2
2.1. Required Software .....	3
2.2. Recommended Software .....	3
2.3. Optional Software .....	4
2.4. Required Environment .....	4
2.5. Installation Procedures .....	4
3. General Workflow .....	6
3.1. Traditional (2D) Movies .....	6
3.2. Stereoscopic (3D) Movies .....	7
4. 3DS Hardware and Mobiclip .....	8
5. Using VirtualDub .....	9
6. When Creating Stereoscopic Movies .....	12
7. Adjusting Quality .....	13
7.1. Adjusting Video Quality .....	14
7.2. Adjusting Audio Quality .....	15
7.3. Quantifying Bitrate .....	15
7.4. Approximating Size of Moflex File .....	16
8. Using moflextool.exe (Final Step) .....	17
9. Troubleshooting .....	19
Revision History .....	20

## 1. Introduction

This document describes how to convert your video assets into the Mobiclip format for the Nintendo 3DS (referred to as just "3DS" or its codename "CTR" throughout the remainder of this document). This includes a discussion of the software tools and their capabilities, the steps involved in the conversion, and the various considerations to take into account when performing the conversion.

This document is meant to supplement the included documentation of the Mobiclip for 3DS software package, so please consult both sources of information.

Note that the Mobiclip for 3DS middleware only provides an interface to create stereoscopic 3D movie assets based on pre-existing separate imagery for left and right perspectives. It does not provide a solution to create such 3D imagery based on 2D imagery (i.e. of a single perspective).

The final Mobiclip assets that can be viewed on 3DS hardware are *.moflex* files. After reading this document, you should get a sense of how to create *.moflex* files for your video content.

Playback of *.moflex* files on 3DS hardware can be accomplished by creating a 3DS application to play your *.moflex* files using the Mobiclip for 3DS runtime API. The Mobiclip for 3DS middleware package also comes with sample demos that can be quickly used to verify operation and illustrate use of the API. For those companies that do not have 3DS development equipment such as a CTR Debugger, and do not have the prerequisite 3DS development expertise (such as video production studios), we can provide 3DS Test hardware and software to help view and verify *.moflex* files. For details on obtaining these hardware and software materials, please email us at [support@noa.com](mailto:support@noa.com).

## 2. Software Environment

The following describes the software environment to perform conversion of video assets into Mobiclip format. All of the software mentioned below is freely available. Access to Mobiclip for 3DS requires a usage agreement to be signed and executed.

The only required software for performing the conversion is the Mobiclip for 3DS middleware. However, the included Mobiclip for 3DS codec cannot perform the

task by itself, and requires external video editing software to apply the Mobiclip compression to video content.

For the video editing software, there are freely available tools on the Internet that can do the task. What we and Mobiclip recommend are also free tools available via the Internet—these are: VirtualDub and AviSynth. Commercial video editing software may be able to handle the job as well. The thing to look for in this case would be whether or not your commercial software supports working with VFW codecs. “VFW” stands for “Video for Windows” and is a standard codec format which the Mobiclip codec adheres to.

With that said, what we recommend is using VirtualDub and AviSynth because it is a tried and tested approach. It is also an approach that we can directly provide technical support for. The procedures described in this document describe how to use the capabilities of VirtualDub and AviSynth to help perform conversions.

## 2.1. Required Software

- Mobiclip for 3DS
  - Available via WarioWorld
  - Includes:
    - 3DS Runtime Library
    - Mobiclip for 3DS VFW (Video for Windows) codec
    - moflextool.exe
  - For video conversion purposes, of the three components above, only the codec and moflextool.exe are necessary

## 2.2. Recommended Software

- VirtualDub
  - Latest version as of this writing: 1.10.4
  - Available on [www.virtualdub.org](http://www.virtualdub.org)
  - Supports .avi and .avs files
- AviSynth
  - Latest version as of this writing: 2.5.8
  - Available on [http://avisynth.nl/index.php/Main\\_Page](http://avisynth.nl/index.php/Main_Page)
  - Allows you to work with .avs files in conjunction with VirtualDub

- Only required for stereoscopic movies

Note that any commercial software that can replace the functionality of VirtualDub and AviSynth can be used as well.

## 2.3. Optional Software

- Huffiyuv codec
  - Latest version as of this writing: 2.1.1
  - Available on: <http://www.videohelp.com/tools/HuffYUV>
  - Huffiyuv is a lossless codec, helpful when storing intermediate files where you want to maintain source quality while still getting compression benefits in terms of reduced data size.
- Pazera Free MP4 to AVI and MOV to AVI Converters
  - Available on: <http://www.pazera-software.com/>
  - H.264/MPEG-4 (.mp4) files and Uncompressed QuickTime (.mov) files can be easily converted into an .avi file of your choice using these tools. As far as the destination compression scheme, either uncompressed or Huffiyuv v2.1.1 is recommended. QuickTime video compressed with Apple ProRes 422 is not supported.

## 2.4. Required Environment

All of the above software tools have been confirmed to work with the following Windows operating systems:

- Windows 7 64-bit
- Windows XP 32-bit with Service pack 3

## 2.5. Installation Procedures

Mobiclip for 3DS:

- Extract out the latest .zip file to a directory of your choice.
- Run the [CTR\\_MW-Mobiclip/tools/ Mobiclip\\_Tools\\_for\\_CTR.exe](#) installer.
- Obtain your license file via WarioWorld and install this via [[Start Menu > All Programs > Mobiclip Tools for NINTENDO CTR > Install a new Mobiclip license](#)].
- These steps are also described in more details in the [CTR-Mobiclip\\_SDK\\_Quick\\_Start\\_Guide-en\\_US.pdf](#) document that is included in the Mobiclip for 3DS package.

#### VirtualDub:

- Extract out the latest .zip file to a directory of your choice.
- The GUI program is [VirtualDub.exe](#).

#### AviSynth:

- Run latest installer program.

#### Huffyuv:

- Obtain the pre-built package via the "pre-built DLL" link from the URL mentioned in section 2.2.
- Extract out the .zip file to a directory of your choice.
- When installing on Windows XP:
  - Right-click on the included [huffyuv.inf](#) file, and select "Install."
- When installing on Windows 7:
  - Execute the following command from a command prompt:

```
rundll32 C:\Windows\SysWOW64\setupapi.dll,InstallHinfSection
DefaultInstall 0 C:\huffyuv-2.1.1\huffyuv.inf
```

- The above is a single command, so it should be a single line. There is no space between the comma and "InstallHinfSection", and a space between "InstallHinfSection" and "DefaultInstall."
- The above assumes that the .zip file was extracted to C:\huffyuv-2.1.1, so adjust the final argument as necessary. If the path to the .inf file contains spaces, you will need to enclose the path in quotes.

#### Pazera Free MP4 to AVI / MOV to AVI Converters:

- Both installer and portable (ZIP) versions are available.

- If using the portable version, extract the .zip file to a directory of your choice. The executable files to run are [mp4toavi.exe](#), and [movtoavi.exe](#), respectively.

### 3. General Workflow

This section goes over the general steps that are involved when converting your video assets into .moflex format. There are some additional steps involved when converting 3D movies, so the procedures are broken up between 2D and 3D movies.

#### 3.1. Traditional (2D) Movies

1. Convert video asset into .avi format (if it isn't already in .avi format).

Mobiclip compression will be applied using VirtualDub, and VirtualDub can mainly only work with .avi files as a video file format. (Some other file formats are apparently supported as well, but .avi files are the surest bet.)

Make sure the quality is not compromised when applying any conversions. Ideally, the .avi file should be in source quality. Applying Mobiclip compression on top of a video that has already been compressed with a lossy codec will be less than ideal in terms of quality.

2. Take source .avi file, and generate .avi file compressed with Mobiclip codec.

Open source .avi in VirtualDub, apply compression using the Mobiclip for CTR codec, and output a new .avi with the Mobiclip compression.

3. Take Mobiclip-compressed .avi file and generate .moflex file.

Use moflextool.exe for this step.

4. Confirm operation of .moflex file on 3DS hardware.

## 3.2. Stereoscopic (3D) Movies

1. Convert video asset(s) of left and right perspectives into .avi format (if necessary).

Make sure the quality is not compromised when applying any conversions. Ideally, the .avi file(s) should be in source quality. Applying Mobiclip compression on top of a video that has already been compressed with a lossy codec will be less than ideal in terms of quality.

As far as the way source data can be broken up into left and right, there are two main options: either have separate .avi files for left and right perspective, or have the left and right perspectives embedded in a single .avi. When the left and right perspectives are embedded in a single .avi (and thus embedded in a single video stream), there are a few different ways this is possible:

- a. Left and right laid out in a single frame, side-by-side
- b. Left and right laid out in a single frame, top-to-bottom
- c. Left and right laid out in interleaved (i.e. alternating) frames

In the method we employ, the final .moflex file will have the left/right frames interleaved.

The moflextool.exe converter and the 3DS Mobiclip runtime supports left/right frames being side-by-side or top-to-bottom in a single frame as well, but there are compression and quality benefits to interleaving left/right frames in the final asset, so this is the approach we will employ and describe in this document.

2. Prepare .avs file to interleave frames of left and right perspectives.

If you don't have an .avi with the left/right frames already interleaved (chances are this is the case), it is simple to create an .avs file where the frames are called up from two different sources and interleaved properly, leveraging the capabilities of AviSynth.

An .avs file is a text file, so use a text editor such as Notepad to create one. For our purposes, we will describe where the left and right frames are to be obtained from, and then interleave the frames using AviSynth's descriptive

power. There is no copying of data involved.

Examples of .avs files to do the interleaving are described later in this document.

VirtualDub supports working with .avs files just like any .avi file.

3. Generate.avi file compressed with Mobiclip codec.

Open up .avs file in VirtualDub, apply compression using the Mobiclip for CTR codec, and output a new .avi with the Mobiclip compression.

4. Take Mobiclip-compressed .avi file and generate .moflex file.

Use moflextool.exe for this step.

5. Confirm operation of .moflex file on 3DS hardware.

## 4. 3DS Hardware and Mobiclip

The maximum instantaneous bitrate that is achievable for a single-screen video (whether upper screen or lower screen) using the Mobiclip runtime on 3DS is documented as 2000 kbps in Mobiclip's own documentation. But with the improvements of the 3DS runtime since then, it appears that now much higher bitrates are also possible.

Note that given a 2D movie that runs at an average bitrate of 500 kbps, a 3D movie of the same content that runs at the same quality and same perceived frame rate would run at an average bitrate of 1000 kbps as there is twice as much data.

An average bitrate of 2000 kbps for 3D movies is generally good in terms of quality. For 2D movies, 2000 kbps is excellent. To give you a comparison, the Mobiclip runtime on the DS system ran at a maximum bitrate of 400 kbps.

Here are the screen sizes for the 3DS:

- Upper screens (both left and right): 400 x 240 (5:3 aspect ratio)
- Lower screen: 320 x 240 (4:3 aspect ratio)

You generally want to have the frame size of the data in the .moflex file match the size (resolution) of the target screen to be used when playing back your content on the 3DS.

For content in 16:9, the resulting size when exactly filling up the width of the upper screen would be 400 x 225, so 15 pixels worth of letterboxing should be applied to fully fit the screen size. This can be done using VirtualDub.

Given a frame size covering the entire screen, and a standard quality setting, a frame rate of up to 60 frames/sec for single-screen 2D videos, and 30 frames/sec for single-screen 3D videos is achievable.

## 5. Using VirtualDub

The VirtualDub tool is where you will apply Mobiclip compression to your video. Other operations such as resizing, cropping, applying letterboxing, removing frames, etc. are possible as well.

Below is a description of some common operations and menus in VirtualDub:

- Opening source .avi file / .avs file in VirtualDub:

This is done via [\[File > Open video file...\]](#) or by dragging in a video file on to the GUI.

- To check and confirm properties of the video, use [\[File > File Information...\]](#).

Information on various properties of the video such as resolution, frame rate, compression format, bitrate, length, as well as audio properties is displayed, making this an invaluable tool for sanity-checking and troubleshooting.

Note that when viewing information about an .avi for 3D display with the left and right frames interleaved, the displayed frame rate will be twice that of the perceived frame rate.

- Creating an output .avi file:

The changes you have made to the video and audio properties of the source video via the [\[Video\]](#) and [\[Audio\]](#) menus will be applied in the output .avi

that is saved out.

The output .avi is generated via the following menu options:

- [\[File > Save as AVI...\]](#) to immediately execute a single conversion.
- [\[File > Queue batch operation > Save as AVI...\]](#) to queue up a conversion in a batch that can be run together at once at a later time.
  - After queuing, make changes and queue up next job, repeat process.
  - Run all queued jobs at once via [\[File > Job control...\]](#).
- [\[Video\]](#) menu:

Operations related to adjusting the properties of the video are done via this menu.

Use [\[Filters...\]](#) to resize, crop, add letterboxing, etc . if necessary:

- There are a variety of filters that can be added, and multiple filters can be applied in succession as well.
- Can use [\[null transform\]](#) for cropping.
- Can use [\[resize\]](#) for adjusting size, and adding letterboxing. See [\[Framing Options\]](#) for letterboxing.

Use the options in [\[Frame Rate... > Frame rate conversion\]](#) to adjust frame rate if necessary.

Use [\[Compression...\]](#) to apply compression format:

- Select and configure settings in [\[Mobiclip for NINTENDO CTR\]](#) when creating Mobiclip-compressed .avi file. More details on this later.
- Huffyuv v2.1.1 is a good codec to use when creating intermediate files.

If any of the above sub-menus are disabled, select [\[Full processing mode\]](#) from the [\[Video\]](#) menu to enable them. [\[Direct stream copy\]](#) is useful when you want to exactly maintain the properties of the video stream (for example, when only modifying the audio stream, or when saving out only a specific time segment of the .avi, etc.).

- [\[Audio\]](#) menu:

Operations related to adjusting the properties of the audio are done via this menu.

Enable [\[Direct stream copy\]](#) when simply maintaining the source audio into the output .avi.

On the other hand, with [\[Full processing mode\]](#) enabled, the following properties can be adjusted:

- The sampling rate, and number of channels (mono versus stereo) via [\[Conversion...\]](#)
- The volume level via [\[Volume...\]](#)
- The compression scheme via [\[Compression...\]](#)

Note that for a Mobiclip-compressed .avi, due to input restrictions in the moflextool.exe, the audio must have the following properties:

- Uncompressed PCM audio
- 16-bit sampling precision, either mono or stereo

- Use [\[Mark in\]](#) and [\[Mark out\]](#) arrows at bottom of timeline to limit conversion to a specific time segment of the video.
- Use [\[File > Append AVI segment...\]](#) to sequentially stitch together multiple .avi files into a single one.
  - After appending an .avi file to another one, you must use [\[File > Save as AVI...\]](#) to create the .avi with both segments stitched together.
  - Note that various video and audio properties of the .avi files being appended to one another must be identical. On the video side, this includes the video codec, frame size, and frame rate. On the audio side, this includes the audio codec, sampling rate, channels, and sampling precision. If the video codecs do not match up, changing the codec scheme of each .avi file to Huffiyuv v2.1.1 is a good solution.

VirtualDub can also be run in a command-line environment, allowing the entire conversion process to be automated. For details, consult "Advanced operation >

Command-line operation" in the VirtualDub.chm help file. The general step involved is to save processing configuration information into a .vcf file using [File > Save processing settings...] and then using these settings by specifying the .vcf file when invoking VirtualDub via the command-line. Note that you will need to add the VirtualDub directory to your PATH environment variable, which is done via [My Computer > Properties > Advanced > Environment Variables].

## 6. When Creating Stereoscopic Movies

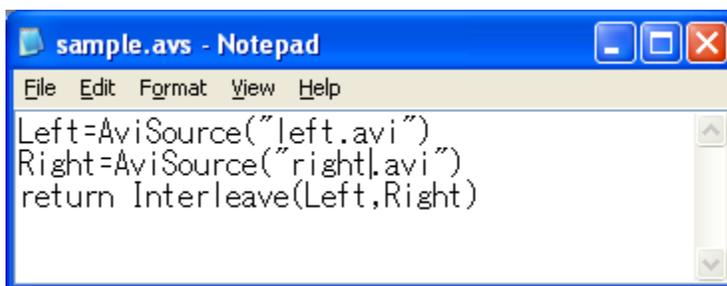
The moflextool.exe supports three different video layouts for 3D movies:

- Interleaved (400 x 240, with double the frame rate)
- Side-by-side (800 x 240)
- Top-to-bottom (400 x 480)

However, as mentioned earlier, the interleaved format is highly recommended. To help with the interleaving of frames, we can utilize .avs files. The following information describes two main scenarios, and illustrates how to write an .avs file to properly interleave frames for each scenario.

**Scenario 1.** Given two separate files, one for left, one for right:

- Create an .avs file in the same directory containing the left and right files.
- Assuming the .avi files are named "left.avi" and "right.avi", create an .avs file with the following text using a text editor:

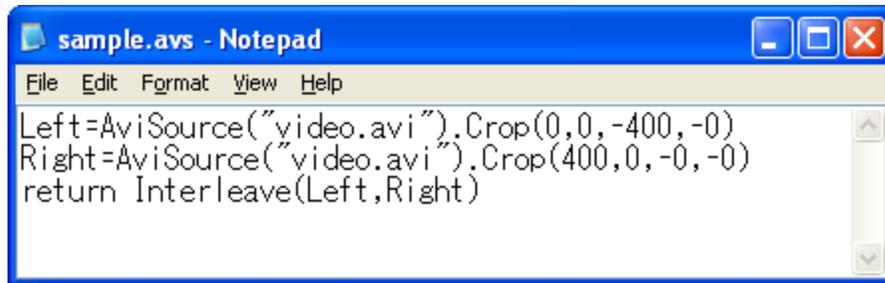


```
sample.avs - Notepad
File Edit Format View Help
Left=AviSource("left.avi")
Right=AviSource("right.avi")
return Interleave(Left,Right)
```

- Open the .avs file in VirtualDub just like an .avi, and process as necessary.
- Note that the .avi files can be referenced in the .avs file using relative or absolute paths as well.

**Scenario 2.** Given one file with left and right frames laid out side-by-side (800 x 240):

- Create an .avs file in the same directory containing the left and right files.
- Assuming the .avi file is named "video.avi", create an .avs file with the following text using a text editor:



```
sample.avs - Notepad
File Edit Format View Help
Left=AviSource("video.avi").Crop(0,0,-400,-0)
Right=AviSource("video.avi").Crop(400,0,-0,-0)
return Interleave(Left,Right)
```

- The syntax of the crop filter in AviSynth is:

```
Crop(left_amount, top_amount, -right_amount, -bottom_amount)
```

- Open the .avs file in VirtualDub just like an .avi, and process as necessary.
- Alternatively, you can crop left and right into separate files first using the "null transform" filter in VirtualDub, and then apply the same steps as Scenario 1.
- Note that the .avi file can be referenced in the .avs file using relative or absolute paths as well.

Note that it is probably best to interleave the files with the left perspective first, just as the above examples show, because the moflextool.exe by default assumes that the left perspective comes first. It is also possible to change this behavior of moflextool.exe via its "firsteye" option, but it is simpler to not to have to specify that option, and standardize on a format.

## 7. Adjusting Quality

The tradeoff between quality and data size is something to always keep in mind. Minimizing the data size while maintaining a good level of quality is a goal to strive for when converting video.

Also, depending on the distribution method of the video (such as streaming over the network, versus downloading, versus via physical media) there may be bitrate and size guidelines to follow. The following provides information on how to adjust the quality of the .moflex files in terms of both the video and the audio, as well as

some supplementary information to help quantify the relationships of all the related parameters.

## 7.1. Adjusting Video Quality

The parameters of the video quality of the Mobiclip content are configured via the Mobiclip for CTR codec configuration dialog. This dialog can be brought up in one of two ways:

- Via [[Start Menu > Mobiclip Tools for NINTENDO CTR > Mobiclip VFW codec](#)].
- Via VirtualDub:

Usually it is most convenient to bring it up here, since you will be using VirtualDub to apply the compression.

Given the same footage, quality is proportional to the bitrate. Perceived quality can depend on the type of footage as well (.e.g. static footage versus dynamic footage), so the same bitrate for different types of footage may not result in the same degree of quality we perceive.

In the Mobiclip codec dialog, there are a number of different rate control types:

- [[Single pass – CBR](#)]: This applies (on average) a constant bitrate over the life of the video. This is the most straightforward approach to reach a target bitrate/size that you are looking for.
- [[Single pass – Constant Quality](#)]: This option is simple and intuitive, but it is difficult to predict the resulting bitrate of the output video. A quality setting of 50 may yield two different bitrates for two different videos. It may be best for testing purposes.
- [[Multipass – VBR](#)]: This applies a variable bitrate. This helps when you want to compress down static parts of the footage and maintain high quality in highly dynamic parts of the footage, all while working within the constraints of an average bitrate. You probably need to pay attention to spikes in instantaneous bitrate however, and ensure that the .moflex file does not exhibit frame drops and hiccups when played back on target hardware, as this can happen if the bitrate spikes exceed the capabilities of the system (see section 4, 3DS Hardware and Mobiclip). Applying multiple passes may improve quality.

For more details on using the Mobiclip VFW codec, refer to the [CTR-Mobiclip\\_SDK\\_Encoding\\_Guide-en\\_US.pdf](#) document, included in the Mobiclip for 3DS middleware package.

## 7.2. Adjusting Audio Quality

There are two different locations where you can adjust the properties of the audio.

Things you can do in VirtualDub (which is also detailed in section 5, Using VirtualDub):

- Change sampling rate
- Change number of channels (mono ↔ stereo)

Things you can do in moflextool.exe:

- Adjust compression format:

Sampling precision of each available audio compression format:

- PCM – 16 bits/sample per channel
- ADPCM – 4 bits/sample per channel (default)
- Fast Audio – 1.25 bits/sample per channel

## 7.3. Quantifying Bitrate

The average bitrate of the video stream can be obtained from within VirtualDub via the [File > File Information...] window. Note that the actual bitrate here may not be exactly identical to the bitrate level set in the Mobiclip codec dialog (if using CBR or VBR methods), but will be in the ballpark of that value.

The audio bitrate can also be obtained from the same dialog. However, when converting, there is no audio bitrate setting to key this off of. Instead, use the following equation to base how to configure the parameters to reach the target audio bitrate you are looking for, for the .moflex file that you are working on.

$$\text{Audio bitrate} = \text{Sampling precision} \times \text{Sampling frequency} \times \text{Number of channels}$$

Here are some examples:

- Fast Audio, 48000 Hz, stereo (2 channels):

$$\text{Audio bitrate} = 1.25 \text{ bits/sample} \times 48000 \text{ samples/sec} \times 2 \text{ channels} = 120000 \text{ bits/sec} = 120 \text{ kbps}$$

- ADPCM, 32000 HZ, mono (1 channel):

$$\text{Audio bitrate} = 4 \text{ bits/sample} \times 32000 \text{ samples/sec} \times 1 \text{ channel} = 128000 \text{ bits/sec} = 128 \text{ kbps}$$

## 7.4. Approximating Size of Moflex File

If you are working with a target size for the entire .moflex file in mind, you can approximate the size before doing the conversion using the following relationships:

$$\text{Total video size} = \text{Average bitrate} \times \text{Length of video}$$

$$\text{Total audio size} = \text{Audio bitrate} \times \text{Length of audio}$$

$$\text{Total moflex size} = \text{Total video size} + \text{Total audio size}$$

Here is an example, with unit conversions:

- Average bitrate of video: 1500 kbps
- Length: 2 min 30 secs
- Audio: ADPCM @ 48 kHz @ Stereo

- video size  $\approx \frac{1500 \text{ kbits}}{1 \text{ sec}} \times \frac{1000 \text{ bits}}{1 \text{ kbit}} \times 150 \text{ secs} \times \frac{1 \text{ byte}}{8 \text{ bits}} \times \frac{1 \text{ KB}}{1024 \text{ bytes}} \times \frac{1 \text{ MB}}{1024 \text{ KB}} = 26.82 \text{ MB}$

- audio size  $\approx \frac{4 \text{ bits}}{\text{sample}} \times \frac{48000 \text{ samples}}{1 \text{ sec}} \times 2 \text{ channels} \times 150 \text{ secs} \times \frac{1 \text{ MB}}{8 \times 1024 \times 1024 \text{ bits}} = 6.87 \text{ MB}$

- total size  $\approx 33.69 \text{ MB}$

## 8. Using moflextool.exe (Final Step)

The moflextool.exe is a command-line tool that is used to convert an .avi file into a file format (i.e. .moflex) that is supported by the Mobiclip for 3DS runtime.

The moflextool.exe requires that the input .avi file have the following characteristics:

- Video codec used must be Mobiclip for CTR codec.
- Audio must be raw PCM audio with 16-bit sampling precision, either mono or stereo.

There are many options involved in invoking the moflextool.exe properly, so the best method is to create a Windows batch (.bat) file using a text editor such as Notepad, and include the command(s) to call moflextool.exe there. The .bat file can then be run either by double-clicking it or from a command prompt.

To see a list of all the moflextool.exe options, enter "moflextool" at a command prompt. If you do not see a list of all the options displayed, the tool was not installed properly at the expected location (e.g. C:\Program Files\Mobiclip Inc\Mobiclip Tools for NINTENDO CTR\converter).

Below is a list of the main options, and simple descriptions of what they do. For all the details of each option, refer to the help text output by executing "moflextool" at a command prompt, and the [CTR-Mobiclip\\_SDK\\_Encoding\\_Guide-en\\_US.pdf file](#), included in the Mobiclip for 3DS middleware package.

Main options:

- -avi: specify path to input .avi file
- -wav: specify path to input .wav file (only necessary when you wish to add an audio track from an external source; it is also possible to ignore the audio track in an .avi file)
- -out: specify path to output .moflex file
- -video: <suboptions, such as stereo layout>
  - The suboptions can be left unspecified for default behavior.
- -audio: <suboptions, such as audio compression format>
  - The suboptions can be left unspecified for default behavior.
- -timeline: <suboptions, such as file index>
  - The suboptions can be left unspecified for default behavior.
  - If you are submitting .moflex files to Nintendo for distribution, please use the -timeline option.
- -info: use this to print helpful diagnostic information about files

- -moflex: use this to specify a moflex file which you want to do a sanity check on.

Below are some example moflextool.exe commands that you may find useful. Note that each of the following is a single command, so each should be on a single line in your .bat file.

In you are targeting the 3DS screen size of 400x240 and your .avi file reflects that, for most standard cases the moflextool will be able to figure out the proper settings to convert your video based on the specs of the input .avi, such as its framerate and frame size. To let moflextool handle this, execute the following command at a command prompt.

### **Example 1:** Generic command that handles a lot of cases

```
moflextool -avi input.avi -video -audio -timeline -out output.moflex -info
```

This single command can handle both stereoscopic and non-stereoscopic videos, as moflextool configures its stereo option based on a set of criteria listed in its help text if the stereo option is left unspecified (as above). To specify stereo configuration explicitly, see the moflextool help text.

For the above command, the output audio will be in ADPCM format as that is the default value when the audio format option is left unspecified. By specifying the audio format explicitly, you can choose a different format such as Fast Audio, or PCM.

Adding the -info option at the end of the command (as above) is a good idea in general. Note that specifying -info at any other place other than the last option appears to cause the moflextool to fail.

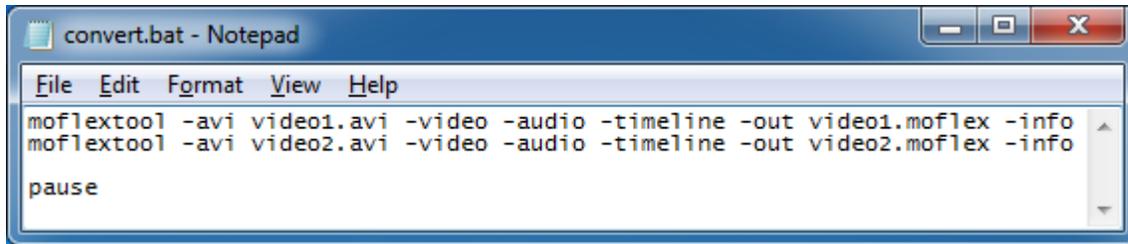
If you wish to check the settings of a .moflex file after you have created one, you can do so with the following command.

### **Example 2:** Checking a .moflex file

```
moflextool -moflex output.moflex -info
```

Information on the video, audio, and timeline streams of the .moflex file will be displayed.

Note that you can include one or more commands such as the above in a .bat file and execute all the conversions at once, as shown below.



```
convert.bat - Notepad
File Edit Format View Help
moflextool -avi video1.avi -video -audio -timeline -out video1.moflex -info
moflextool -avi video2.avi -video -audio -timeline -out video2.moflex -info
pause
```

Adding the “pause” command at the end as shown above will allow you to double-click the .bat file to execute it and not have the command prompt window disappear as soon as the command finishes executing.

## 9. Troubleshooting

Here are some issues that can come up during conversion:

- When trying to save out an .avi with Mobiclip compression, a “license expired” message shows up.
  - Download a new license file from WarioWorld and install it.
- Although the .avi files can be opened in VirtualDub and worked with, an .avs file referencing the .avi files causes an error in VirtualDub and cannot be opened.
  - This can occur if the the codec used in the .avi files is not supported by AviSynth although supported by VirtualDub. The best thing to do in this situation would be to use VirtualDub to convert your .avi files using Huffiyuv v2.1.1, and then call up the converted files from your .avs file.
- When running moflextool.exe, no .moflex file shows up.
  - Check the information displayed by moflextool.exe to diagnose what is wrong with the output. If running a .bat file from Windows by double-clicking it causes the output to fly by, add a “pause” command in the final line of your .bat file to force the command prompt not to disappear.

When confirming operation on 3DS hardware, if:

- Stereoscopic effect looks strange, and is extremely hard to watch:
  - Check to make sure left and right perspectives are not switched.
  - Make sure each corresponding frame for left and right match up properly with just a perspective shift.
  
- Frames are being dropped:
  - Reduce bitrate and re-convert video.
  
- Other strange behavior:
  - Make sure the video and audio of the moflex file are within the constraints of the specs.

If you are having issues other than the above, consult the included documentation within the Mobiclip for CTR package, and/or contact [support@noa.com](mailto:support@noa.com).

## Revision History

Version	Date	Changes
1.3	2/5/2015	<ul style="list-style-type: none"> <li>• Section 2: Updated software links, versions. Added mention of MP4 to AVI converter.</li> <li>• Section 4: Revised text on achievable bitrates.</li> <li>• Section 8: Revised to reflect changes to moflex tool.</li> </ul>
1.2	8/9/2011	<ul style="list-style-type: none"> <li>• Section 2: Revised description on prerequisite usage agreement for obtaining access to Mobiclip for 3DS.</li> </ul>
1.1	7/6/2011	<ul style="list-style-type: none"> <li>• Section 1: Revised description of how to view and verify .moflex files.</li> <li>• Section 3.1 and 3.2: Revised description of how to view and verify .moflex files.</li> <li>• Section 5: Added supplementary information about appending .avi files together.</li> <li>• Section 6: Added note about using paths in .avs files.</li> <li>• Section 8: Revised note about use of -timeline option.</li> </ul>
1.0	6/27/2011	Initial revision.