Windows

Contents

dshow

Uses the `dshow` (DirectShow) input device which is the preferred option for Windows users. See the wiki article about DirectShow and the dshow input device documentation for more information.

vfwcap

> **Warning:** `vfwcap` is outdated. Use `dshow` instead if possible. See DirectShow for more information.

Uses the outdated `vfwcap` input device. See the vfwcap input device documentation for more information.

List devices

To list the supported, connected capture devices:

```
ffmpeg -y -f vfwcap -i list
```

Example output:

```
…
libavutil     50.36. 0 / 50.36. 0
libavcore      0.16. 1 /  0.16. 1
libavcodec    52.108. 0 / 52.108. 0
libavformat   52.93. 0 / 52.93. 0
libavdevice   52. 2. 3 / 52. 2. 3
libavfilter    1.74. 0 /  1.74. 0
libswscale     0.12. 0 /  0.12. 0
[vfwcap @ 01c6d150] Driver 0
[vfwcap @ 01c6d150]  Microsoft WDM Image Capture (Win32)
[vfwcap @ 01c6d150]  Version:  5.1.2600.5512
list: Input/output error
```

Encoding example

Example to encode video from the camera:

```
ffmpeg -y -f vfwcap -r 25 -i 0 out.mp4
```

`-i 0` is the index (zero based) in the list of present capture devices (`Driver 0` in this instance).

Linux

Uses the `video4linux2` (or simply `v4l2`) input device to capture live input such as from a webcam. See the v4l2 input device documentation for more information.

List devices

To list the supported, connected capture devices you can use the `v4l-ctl` tool. This example shows two connected webcams: `/dev/video0` and `/dev/video1`.

```
$ v4l2-ctl --list-devices
USB2.0 PC CAMERA (usb-0000:00:1d.7-1):
        /dev/video1

UVC Camera (046d:0819) (usb-0000:00:1d.7-2):
        /dev/video0
```

List device capabilities

To list available formats (supported pixel formats, video formats, and frame sizes) for a particular input device:

```
$ ffmpeg -f v4l2 -list_formats all -i /dev/video0
…
[video4linux2,v4l2 @ 0xf07d80] Raw       :   yuyv422 :     YUV 4:2:2 (YUYV
[video4linux2,v4l2 @ 0xf07d80] Compressed:     mjpeg :               MJPE
```

Alternatively you could use `v4l2-ctl --list-formats-ext` to list available formats.

Encoding example

Example to encode video from `/dev/video0`:

```
ffmpeg -f v4l2 -framerate 25 -video_size 640x480 -i /dev/video0 output.mkv
```

Adjusting camera functions

Brightness, zoom, focus, etc, can be adjusted with `v4l2-ctl`. Display all controls and their menus:

```
v4l2-ctl -L
```

Then adjust the value:

```
v4l2-ctl -c <option>=<value>
```

OS X

OS X users can use the avfoundation and qtkit input devices for grabbing integrated iSight cameras as well as cameras connected via USB or FireWire:

- AVFoundation is available on Mac OS X 10.7 (Lion) and later. Since then, Apple recommends AVFoundation for stream grabbing on OS X and iOS devices.
- QTKit is available on Mac OS X 10.4 (Tiger) and later. QTKit has been marked deprecated since OS X 10.7 (Lion) and may not be available on future releases.

AVFoundation

To list the supported, connected capture devices:

```
ffmpeg -f avfoundation -list_devices true -i ""
```

To use the default device which is usually the first device in the listing the user can either use an empty name string or `default`:

```
ffmpeg -f avfoundation -i "" out.mpg
```

or

```
ffmpeg -f avfoundation -i "default" out.mpg
```

To use one of these devices for capturing the user has to specify either the name of the device or the index shown in the device listing. Abbreviations using just the beginning of the device name are possible. Thus, to capture from a device named `Integrated iSight-camera`:

```
ffmpeg -f avfoundation -i "Integrated" out.mpg
```

To use the device's index provide the index either as the input or use the `-video_device_index` option that will override any given input name:

```
ffmpeg -f avfoundation -i "2" out.mpg
```

and

```
ffmpeg -f avfoundation -video_device_index 2 -i "default" out.mpg
```

will use the device with the index `2` ignoring the default device in the second case.

QTKit

To list the supported, connected capture devices:

```
ffmpeg -f qtkit -list_devices true -i ""
```

To use the default device which is usually the first device in the listing the user can either use an empty name string or `default`:

```
ffmpeg -f qtkit -i "" out.mpg
```

or

```
ffmpeg -f qtkit -i "default" out.mpg
```

To use one of these devices for capturing the user has to specify either the name of the device or the index shown in the device listing. Abbreviations using just the beginning of the device name are possible. Thus, to capture from a device named `Integrated iSight-camera`:

```
ffmpeg -f qtkit -i "Integrated" out.mpg
```

To use the device's index provide the index either as the input or use the `-video_device_index` option that will override any given input name:

```
ffmpeg -f qtkit -i "2" out.mpg
```

and

```
ffmpeg -f qtkit -video_device_index 2 -i "default" out.mpg
```

will use the device with the index `2` ignoring the default device in the second case.

Last modified on 06/03/14 23:45:13