

# V4L2\_META\_FMT\_D4XX ('D4XX')

Intel D4xx UVC Cameras Metadata

## Description

Intel D4xx (D435 and other) cameras include per-frame metadata in their UVC payload headers, following the Microsoft(R) UVC extension proposal [1]. That means, that the private D4XX metadata, following the standard UVC header, is organised in blocks. D4XX cameras implement several standard block types, proposed by Microsoft, and several proprietary ones. Supported standard metadata types are MetadataId\_CaptureStats (ID 3), MetadataId\_CameraExtrinsics (ID 4), and MetadataId\_CameraIntrinsics (ID 5). For their description see [1]. This document describes proprietary metadata types, used by D4xx cameras.

V4L2\_META\_FMT\_D4XX buffers follow the metadata buffer layout of V4L2\_META\_FMT\_UVC with the only difference, that it also includes proprietary payload header data. D4xx cameras use bulk transfers and only send one payload per frame, therefore their headers cannot be larger than 255 bytes.

Below are proprietary Microsoft style metadata types, used by D4xx cameras, where all fields are in little endian order:

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\linux-master\Documentation\userspace-api\media\v4l\linux-master)(Documentation)(userspace-api)(media)(v4l)pixfmt-meta-d4xx.rst, line 32)**

Unknown directive type "tabularcolumns".

```
.. tabularcolumns:: |p{5.0cm}|p{12.5cm}|
```

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\linux-master\Documentation\userspace-api\media\v4l\linux-master)(Documentation)(userspace-api)(media)(v4l)pixfmt-meta-d4xx.rst, line 35)**

Unknown directive type "flat-table".

```
.. flat-table:: D4xx metadata
   :widths: 1 2
   :header-rows: 1
   :stub-columns: 0

   * - **Field**
     - **Description**
   * - :cspan:`1` *Depth Control*
   * - __u32 ID
     - 0x80000000
   * - __u32 Size
     - Size in bytes (currently 56)
   * - __u32 Version
     - Version of this structure. The documentation herein corresponds to
       version xxx. The version number will be incremented when new fields are
       added.
   * - __u32 Flags
     - A bitmask of flags: see [2_] below
   * - __u32 Gain
     - Gain value in internal units, same as the V4L2_CID_GAIN control, used to
       capture the frame
   * - __u32 Exposure
     - Exposure time (in microseconds) used to capture the frame
   * - __u32 Laser power
     - Power of the laser LED 0-360, used for depth measurement
   * - __u32 AE mode
     - 0: manual; 1: automatic exposure
   * - __u32 Exposure priority
     - Exposure priority value: 0 - constant frame rate
   * - __u32 AE ROI left
     - Left border of the AE Region of Interest (all ROI values are in pixels
       and lie between 0 and maximum width or height respectively)
   * - __u32 AE ROI right
     - Right border of the AE Region of Interest
   * - __u32 AE ROI top
     - Top border of the AE Region of Interest
   * - __u32 AE ROI bottom
     - Bottom border of the AE Region of Interest
   * - __u32 Preset
     - Preset selector value, default: 0, unless changed by the user
```

```

* - __u32 Laser mode
  - 0: off, 1: on
* - :cspan: `1` *Capture Timing*
* - __u32 ID
  - 0x80000001
* - __u32 Size
  - Size in bytes (currently 40)
* - __u32 Version
  - Version of this structure. The documentation herein corresponds to
    version xxx. The version number will be incremented when new fields are
    added.
* - __u32 Flags
  - A bitmask of flags: see [3_] below
* - __u32 Frame counter
  - Monotonically increasing counter
* - __u32 Optical time
  - Time in microseconds from the beginning of a frame till its middle
* - __u32 Readout time
  - Time, used to read out a frame in microseconds
* - __u32 Exposure time
  - Frame exposure time in microseconds
* - __u32 Frame interval
  - In microseconds = 1000000 / framerate
* - __u32 Pipe latency
  - Time in microseconds from start of frame to data in USB buffer
* - :cspan: `1` *Configuration*
* - __u32 ID
  - 0x80000002
* - __u32 Size
  - Size in bytes (currently 40)
* - __u32 Version
  - Version of this structure. The documentation herein corresponds to
    version xxx. The version number will be incremented when new fields are
    added.
* - __u32 Flags
  - A bitmask of flags: see [4_] below
* - __u8 Hardware type
  - Camera hardware version [5_]
* - __u8 SKU ID
  - Camera hardware configuration [6_]
* - __u32 Cookie
  - Internal synchronisation
* - __u16 Format
  - Image format code [7_]
* - __u16 Width
  - Width in pixels
* - __u16 Height
  - Height in pixels
* - __u16 Framerate
  - Requested frame rate per second
* - __u16 Trigger
  - Byte 0: bit 0: depth and RGB are synchronised, bit 1: external trigger

```

[1] <https://docs.microsoft.com/en-us/windows-hardware/drivers/stream/uvc-extensions-1-5>

[2] Depth Control flags specify which fields are valid:

```

0x00000001 Gain
0x00000002 Exposure
0x00000004 Laser power
0x00000008 AE mode
0x00000010 Exposure priority
0x00000020 AE ROI
0x00000040 Preset

```

[3] Capture Timing flags specify which fields are valid:

```

0x00000001 Frame counter
0x00000002 Optical time
0x00000004 Readout time
0x00000008 Exposure time
0x00000010 Frame interval
0x00000020 Pipe latency

```

[4] Configuration flags specify which fields are valid:

```

0x00000001 Hardware type
0x00000002 SKU ID
0x00000004 Cookie
0x00000008 Format

```

0x00000010 Width  
0x00000020 Height  
0x00000040 Framerate  
0x00000080 Trigger  
0x00000100 Cal count

**[5] Camera model:**

0 DS5  
1 IVCAM2

**[6] 8-bit camera hardware configuration bitfield:**

[1:0] depthCamera  
00: no depth  
01: standard depth  
10: wide depth  
11: reserved  
[2] depthIsActive - has a laser projector  
[3] RGB presence  
[4] Inertial Measurement Unit (IMU) presence  
[5] projectorType  
0: HPTG  
1: Princeton  
[6] 0: a projector, 1: an LED  
[7] reserved

**[7] Image format codes per video streaming interface:**

**Depth:**

1 Z16  
2 Z

**Left sensor:**

1 Y8  
2 UYVY  
3 R8L8  
4 Calibration  
5 W10

**Fish Eye sensor:**

1 RAW8