cv::VideoWriter Class Reference

Video I/O

Video writer class. More...

```
#include "videoio.hpp"
```

Public Member Functions

	VideoWriter () Default constructors. More...
	VideoWriter (const **String** &filename, int **fourcc**, double fps, **Size** frameSize, bool isColor=true)
	VideoWriter (const **String** &filename, int apiPreference, int **fourcc**, double fps, **Size** frameSize, bool isColor=true)
virtual	**~VideoWriter** () Default destructor. More...
virtual double	**get** (int propId) const Returns the specified **VideoWriter** property. More...
virtual bool	**isOpened** () const Returns true if video writer has been successfully initialized. More...
virtual bool	**open** (const **String** &filename, int **fourcc**, double fps, **Size** frameSize, bool isColor=true) Initializes or reinitializes video writer. More...
bool	**open** (const **String** &filename, int apiPreference, int **fourcc**, double fps, **Size** frameSize, bool isColor=true)
virtual **VideoWriter** &	**operator<<** (const **Mat** &image) Stream operator to write the next video frame. More...
virtual void	**release** () Closes the video writer. More...
virtual bool	**set** (int propId, double value) Sets a property in the **VideoWriter**. More...
virtual void	**write** (const **Mat** &image) Writes the next video frame. More...

Static Public Member Functions

static int **fourcc** (char c1, char c2, char c3, char c4)
Concatenates 4 chars to a fourcc code. More...

Static Protected Member Functions

static **Ptr**< IVideoWriter > **create** (const **String** &filename, int **fourcc**, double fps, **Size** frameSize, bool isColor=true)

Protected Attributes

Ptr< IVideoWriter > **iwriter**

Ptr< **CvVideoWriter** > **writer**

Detailed Description

Video writer class.

The class provides C++ API for writing video files or image sequences.

Examples:

samples/cpp/tutorial_code/videoio/video-write/video-write.cpp, **samples/cpp/videowriter_basic.cpp**, and **samples/tapi/hog.cpp**.

Constructor & Destructor Documentation

§ VideoWriter() [1/3]

cv::VideoWriter::VideoWriter ( )

Python:

```
<VideoWriter object> = cv.VideoWriter(                                          )
<VideoWriter object> = cv.VideoWriter( filename, fourcc, fps, frameSize[, isColor]                 )
<VideoWriter object> = cv.VideoWriter( filename, apiPreference, fourcc, fps, frameSize[, isColor] )
```

Default constructors.

The constructors/functions initialize video writers.

- On Linux FFMPEG is used to write videos;
- On Windows FFMPEG or VFW is used;
- On MacOSX QTKit is used.

§ VideoWriter() [2/3]

```
cv::VideoWriter::VideoWriter ( const String &  filename,
                               int             fourcc,
                               double          fps,
                               Size            frameSize,
                               bool            isColor = true
                             )
```

Python:

```
<VideoWriter object> = cv.VideoWriter(                                          )
<VideoWriter object> = cv.VideoWriter( filename, fourcc, fps, frameSize[, isColor]                )
<VideoWriter object> = cv.VideoWriter( filename, apiPreference, fourcc, fps, frameSize[, isColor] )
```

This is an overloaded member function, provided for convenience. It differs from the above function only in what argument(s) it accepts.

Parameters

filename	Name of the output video file.
fourcc	4-character code of codec used to compress the frames. For example, **VideoWriter::fourcc**('P','I','M','1') is a MPEG-1 codec, **VideoWriter::fourcc**('M','J','P','G') is a motion-jpeg codec etc. List of codes can be obtained at Video Codecs by FOURCC page. FFMPEG backend with MP4 container natively uses other values as fourcc code: see ObjectType, so you may receive a warning message from OpenCV about fourcc code conversion.
fps	Framerate of the created video stream.
frameSize	Size of the video frames.
isColor	If it is not zero, the encoder will expect and encode color frames, otherwise it will work with grayscale frames (the flag is currently supported on Windows only).

Tips:

- With some backends `fourcc=-1` pops up the codec selection dialog from the system.
- To save image sequence use a proper filename (eg. `img_%02d.jpg`) and `fourcc=0` OR `fps=0`. Use uncompressed image format (eg. `img_%02d.BMP`) to save raw frames.
- Most codecs are lossy. If you want lossless video file you need to use a lossless codecs (eg. FFMPEG FFV1, Huffman HFYU, Lagarith LAGS, etc...)
- If FFMPEG is enabled, using `codec=0; fps=0;` you can create an uncompressed (raw) video file.

§ VideoWriter() [3/3]

```
cv::VideoWriter::VideoWriter ( const String &  filename,
                               int             apiPreference,
                               int             fourcc,
                               double          fps,
                               Size            frameSize,
                               bool            isColor = true
                             )
```

Python:

```
<VideoWriter object> = cv.VideoWriter(                                                )
<VideoWriter object> = cv.VideoWriter( filename, fourcc, fps, frameSize[, isColor]    )
<VideoWriter object> = cv.VideoWriter( filename, apiPreference, fourcc, fps, frameSize[, isColor] )
```

This is an overloaded member function, provided for convenience. It differs from the above function only in what argument(s) it accepts. The `apiPreference` parameter allows to specify API backends to use. Can be used to enforce a specific reader implementation if multiple are available: e.g. **cv::CAP_FFMPEG** or **cv::CAP_GSTREAMER**.

§ ~VideoWriter()

```
virtual cv::VideoWriter::~VideoWriter ( )
```

virtual

Default destructor.

The method first calls **VideoWriter::release** to close the already opened file.

Member Function Documentation

§ create()

```
static Ptr<IVideoWriter> cv::VideoWriter::create ( const String &  filename,
                                                   int              fourcc,
                                                   double           fps,
                                                   Size             frameSize,
                                                   bool             isColor = true
                                                  )
```

static protected

§ fourcc()

```
static int cv::VideoWriter::fourcc ( char  c1,
                                     char  c2,
                                     char  c3,
                                     char  c4
                                    )
```

static

Python:

```
retval = cv.VideoWriter_fourcc( c1, c2, c3, c4 )
```

Concatenates 4 chars to a fourcc code.

Returns

a fourcc code

This static method constructs the fourcc code of the codec to be used in the constructor **VideoWriter::VideoWriter** or **VideoWriter::open**.

Examples:

samples/cpp/videowriter_basic.cpp.

§ get()

virtual double cv::VideoWriter::get (int propId) const

virtual

Python:

retval = cv.VideoWriter.get(propId)

Returns the specified **VideoWriter** property.

Parameters

propId Property identifier from **cv::VideoWriterProperties** (eg. **cv::VIDEOWRITER_PROP_QUALITY**) or one of **Additional flags for video I/O API backends**

Returns

Value for the specified property. Value 0 is returned when querying a property that is not supported by the backend used by the **VideoWriter** instance.

§ isOpened()

virtual bool cv::VideoWriter::isOpened ( ) const

virtual

Python:

retval = cv.VideoWriter.isOpened( )

Returns true if video writer has been successfully initialized.

Examples:

samples/cpp/videowriter_basic.cpp, and **samples/tapi/hog.cpp**.

§ open() [1/2]

```
virtual bool cv::VideoWriter::open ( const String &  filename,
                                     int             fourcc,
                                     double          fps,
                                     Size            frameSize,
                                     bool            isColor = true
                                   )                                    virtual
```

Python:

```
retval = cv.VideoWriter.open( filename, fourcc, fps, frameSize[, isColor]                 )
retval = cv.VideoWriter.open( filename, apiPreference, fourcc, fps, frameSize[, isColor] )
```

Initializes or reinitializes video writer.

The method opens video writer. Parameters are the same as in the constructor **VideoWriter::VideoWriter**.

Returns

`true` if video writer has been successfully initialized

The method first calls **VideoWriter::release** to close the already opened file.

Examples:

samples/cpp/tutorial_code/videoio/video-write/video-write.cpp, **samples/cpp/videowriter_basic.cpp**, and **samples/tapi/hog.cpp**.

§ open() [2/2]

```
bool cv::VideoWriter::open ( const String &  filename,
                             int             apiPreference,
                             int             fourcc,
                             double          fps,
                             Size            frameSize,
                             bool            isColor = true
                           )
```

Python:

```
retval = cv.VideoWriter.open( filename, fourcc, fps, frameSize[, isColor]                 )
retval = cv.VideoWriter.open( filename, apiPreference, fourcc, fps, frameSize[, isColor] )
```

This is an overloaded member function, provided for convenience. It differs from the above function only in what argument(s) it accepts.

§ operator<<()

virtual **VideoWriter**& cv::VideoWriter::operator<< (const **Mat** & image) virtual

Stream operator to write the next video frame.

See also

write

§ release()

virtual void cv::VideoWriter::release ( ) virtual

Python:

None = cv.VideoWriter.release( )

Closes the video writer.

The method is automatically called by subsequent **VideoWriter::open** and by the **VideoWriter** destructor.

§ set()

virtual bool cv::VideoWriter::set (int propId,
double value
) virtual

Python:

retval = cv.VideoWriter.set(propId, value)

Sets a property in the **VideoWriter**.

Parameters

propId Property identifier from **cv::VideoWriterProperties** (eg. **cv::VIDEOWRITER_PROP_QUALITY**) or one of **Additional flags for video I/O API backends**

value Value of the property.

Returns

`true` if the property is supported by the backend used by the **VideoWriter** instance.

§ write()

virtual void cv::VideoWriter::write (const **Mat** & image) virtual

Python:

None = cv.VideoWriter.write(image)

Writes the next video frame.

Parameters

image The written frame. In general, color images are expected in BGR format.

The function/method writes the specified image to video file. It must have the same size as has been specified when opening the video writer.

Examples:

samples/cpp/videowriter_basic.cpp.

Member Data Documentation

§ iwriter

Ptr<IVideoWriter> cv::VideoWriter::iwriter protected

§ writer

Ptr<CvVideoWriter> cv::VideoWriter::writer protected

The documentation for this class was generated from the following file:

- videoio/include/opencv2/**videoio.hpp**

Generated on Wed Aug 29 2018 10:30:12 for OpenCV by doxygen 1.8.12