JavaScript is disabled on your browser.

* [Overview](http://docs.google.com/overview-summary.html)
* [Package](http://docs.google.com/package-summary.html)
* Class
* [Tree](http://docs.google.com/package-tree.html)
* [Index](http://docs.google.com/index-all.html)
* [Help](http://docs.google.com/help-doc.html)
* Prev Class
* [Next Class](http://docs.google.com/org/opencv/videoio/Videoio.html)
* [Frames](http://docs.google.com/index.html?org/opencv/videoio/VideoCapture.html)
* [No Frames](http://docs.google.com/VideoCapture.html)
* [All Classes](http://docs.google.com/allclasses-noframe.html)
* Summary:
* Nested |
* Field |
* [Constr](#3znysh7) |
* [Method](#2et92p0)
* Detail:
* Field |
* [Constr](#3dy6vkm) |
* [Method](#26in1rg)

org.opencv.videoio

## Class VideoCapture

* java.lang.Object
  + org.opencv.videoio.VideoCapture
* public class VideoCapture  
  extends java.lang.Object

Class for video capturing from video files, image sequences or cameras. The class provides C++ API for capturing video from cameras or for reading video files and image sequences. Here is how the class can be used: INCLUDE: samples/cpp/videocapture\_basic.cpp **Note:** In REF: videoio\_c "C API" the black-box structure CvCapture is used instead of %VideoCapture. **Note:**

* + (C++) A basic sample on using the %VideoCapture interface can be found at OPENCV\_SOURCE\_CODE/samples/cpp/videocapture\_starter.cpp
  + (Python) A basic sample on using the %VideoCapture interface can be found at OPENCV\_SOURCE\_CODE/samples/python/video.py
  + (Python) A multi threaded video processing sample can be found at OPENCV\_SOURCE\_CODE/samples/python/video\_threaded.py
  + (Python) %VideoCapture sample showcasing some features of the Video4Linux2 backend OPENCV\_SOURCE\_CODE/samples/python/video\_v4l2.py

### Constructor SummaryConstructors

| Constructor and Description |
| --- |
| [**VideoCapture**](http://docs.google.com/org/opencv/videoio/VideoCapture.html#VideoCapture())() Default constructor **Note:** In REF: videoio\_c "C API", when you finished working with video, release CvCapture structure with cvReleaseCapture(), or use Ptr<CvCapture> that calls cvReleaseCapture() automatically in the destructor. |
| [**VideoCapture**](http://docs.google.com/org/opencv/videoio/VideoCapture.html#VideoCapture(int))(int index) Open a camera for video capturing |
| [**VideoCapture**](http://docs.google.com/org/opencv/videoio/VideoCapture.html#VideoCapture(int,%20int))(int index, int apiPreference) Opens a camera for video capturing |
| [**VideoCapture**](http://docs.google.com/org/opencv/videoio/VideoCapture.html#VideoCapture(java.lang.String))(java.lang.String filename) Open video file or image file sequence or a capturing device or a IP video stream for video capturing Same as VideoCapture(const String& filename, int apiPreference) but using default Capture API backends |
| [**VideoCapture**](http://docs.google.com/org/opencv/videoio/VideoCapture.html#VideoCapture(java.lang.String,%20int))(java.lang.String filename, int apiPreference) Open video file or a capturing device or a IP video stream for video capturing with API Preference |

### Method SummaryMethods

| Modifier and Type | Method and Description |
| --- | --- |
| static [VideoCapture](http://docs.google.com/org/opencv/videoio/VideoCapture.html) | [**\_\_fromPtr\_\_**](http://docs.google.com/org/opencv/videoio/VideoCapture.html#__fromPtr__(long))(long addr) |
| double | [**get**](http://docs.google.com/org/opencv/videoio/VideoCapture.html#get(int))(int propId) Returns the specified VideoCapture property |
| java.lang.String | [**getBackendName**](http://docs.google.com/org/opencv/videoio/VideoCapture.html#getBackendName())() Returns used backend API name **Note:** Stream should be opened. |
| long | [**getNativeObjAddr**](http://docs.google.com/org/opencv/videoio/VideoCapture.html#getNativeObjAddr())() |
| boolean | [**grab**](http://docs.google.com/org/opencv/videoio/VideoCapture.html#grab())() Grabs the next frame from video file or capturing device. |
| boolean | [**isOpened**](http://docs.google.com/org/opencv/videoio/VideoCapture.html#isOpened())() Returns true if video capturing has been initialized already. |
| boolean | [**open**](http://docs.google.com/org/opencv/videoio/VideoCapture.html#open(int))(int index) Open a camera for video capturing Parameters are same as the constructor VideoCapture(int index) |
| boolean | [**open**](http://docs.google.com/org/opencv/videoio/VideoCapture.html#open(int,%20int))(int cameraNum, int apiPreference) Open a camera for video capturing Parameters are similar as the constructor VideoCapture(int index),except it takes an additional argument apiPreference. |
| boolean | [**open**](http://docs.google.com/org/opencv/videoio/VideoCapture.html#open(java.lang.String))(java.lang.String filename) Open video file or a capturing device or a IP video stream for video capturing Parameters are same as the constructor VideoCapture(const String& filename) |
| boolean | [**open**](http://docs.google.com/org/opencv/videoio/VideoCapture.html#open(java.lang.String,%20int))(java.lang.String filename, int apiPreference) Open video file or a capturing device or a IP video stream for video capturing with API Preference Parameters are same as the constructor VideoCapture(const String& filename, int apiPreference) |
| boolean | [**read**](http://docs.google.com/org/opencv/videoio/VideoCapture.html#read(org.opencv.core.Mat))([Mat](http://docs.google.com/org/opencv/core/Mat.html) image) Grabs, decodes and returns the next video frame. |
| void | [**release**](http://docs.google.com/org/opencv/videoio/VideoCapture.html#release())() Closes video file or capturing device. |
| boolean | [**retrieve**](http://docs.google.com/org/opencv/videoio/VideoCapture.html#retrieve(org.opencv.core.Mat))([Mat](http://docs.google.com/org/opencv/core/Mat.html) image) Decodes and returns the grabbed video frame. |
| boolean | [**retrieve**](http://docs.google.com/org/opencv/videoio/VideoCapture.html#retrieve(org.opencv.core.Mat,%20int))([Mat](http://docs.google.com/org/opencv/core/Mat.html) image, int flag) Decodes and returns the grabbed video frame. |
| boolean | [**set**](http://docs.google.com/org/opencv/videoio/VideoCapture.html#set(int,%20double))(int propId, double value) Sets a property in the VideoCapture. |

### Methods inherited from class java.lang.Objectequals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

### Constructor Detail

#### VideoCapture public VideoCapture() Default constructor **Note:** In REF: videoio\_c "C API", when you finished working with video, release CvCapture structure with cvReleaseCapture(), or use Ptr<CvCapture> that calls cvReleaseCapture() automatically in the destructor.

#### VideoCapture public VideoCapture(int index) Open a camera for video capturingParameters:index - camera\_id + domain\_offset (CAP\_\*) id of the video capturing device to open. To open default camera using default backend just pass 0. Use a domain\_offset to enforce a specific reader implementation if multiple are available like cv::CAP\_FFMPEG or cv::CAP\_IMAGES or cv::CAP\_DSHOW. e.g. to open Camera 1 using the MS Media Foundation API use index = 1 + cv::CAP\_MSMF SEE: cv::VideoCaptureAPIs

#### VideoCapture public VideoCapture(int index, int apiPreference) Opens a camera for video capturingParameters:index - id of the video capturing device to open. To open default camera using default backend just pass 0. (to backward compatibility usage of camera\_id + domain\_offset (CAP\_\*) is valid when apiPreference is CAP\_ANY)apiPreference - preferred Capture API backends to use. Can be used to enforce a specific reader implementation if multiple are available: e.g. cv::CAP\_DSHOW or cv::CAP\_MSMF or cv::CAP\_V4L2. SEE: cv::VideoCaptureAPIs

#### VideoCapture public VideoCapture(java.lang.String filename) Open video file or image file sequence or a capturing device or a IP video stream for video capturing Same as VideoCapture(const String& filename, int apiPreference) but using default Capture API backendsParameters:filename - automatically generated

#### VideoCapture public VideoCapture(java.lang.String filename, int apiPreference) Open video file or a capturing device or a IP video stream for video capturing with API PreferenceParameters:filename - it can be:

* + - * name of video file (eg. video.avi)
      * or image sequence (eg. img\_%02d.jpg, which will read samples like img\_00.jpg, img\_01.jpg, img\_02.jpg, ...)
      * or URL of video stream (eg. protocol://host:port/script\_name?script\_params|auth)
      * or GStreamer pipeline string in gst-launch tool format in case if GStreamer is used as backend Note that each video stream or IP camera feed has its own URL scheme. Please refer to the documentation of source stream to know the right URL.

apiPreference - preferred Capture 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\_IMAGES or cv::CAP\_DSHOW.SEE: cv::VideoCaptureAPIs

### Method Detail

#### \_\_fromPtr\_\_ public static [VideoCapture](http://docs.google.com/org/opencv/videoio/VideoCapture.html) \_\_fromPtr\_\_(long addr)

#### get public double get(int propId) Returns the specified VideoCapture propertyParameters:propId - Property identifier from cv::VideoCaptureProperties (eg. cv::CAP\_PROP\_POS\_MSEC, cv::CAP\_PROP\_POS\_FRAMES, ...) or one from REF: videoio\_flags\_others Returns:Value for the specified property. Value 0 is returned when querying a property that is not supported by the backend used by the VideoCapture instance. **Note:** Reading / writing properties involves many layers. Some unexpected result might happens along this chain. VideoCapture -> API Backend -> Operating System -> Device Driver -> Device Hardware The returned value might be different from what really used by the device or it could be encoded using device dependent rules (eg. steps or percentage). Effective behaviour depends from device driver and API Backend

#### getBackendName public java.lang.String getBackendName() Returns used backend API name **Note:** Stream should be opened.Returns:automatically generated

#### getNativeObjAddr public long getNativeObjAddr()

#### grab public boolean grab() Grabs the next frame from video file or capturing device.Returns:true (non-zero) in the case of success. The method/function grabs the next frame from video file or camera and returns true (non-zero) in the case of success. The primary use of the function is in multi-camera environments, especially when the cameras do not have hardware synchronization. That is, you call VideoCapture::grab() for each camera and after that call the slower method VideoCapture::retrieve() to decode and get frame from each camera. This way the overhead on demosaicing or motion jpeg decompression etc. is eliminated and the retrieved frames from different cameras will be closer in time. Also, when a connected camera is multi-head (for example, a stereo camera or a Kinect device), the correct way of retrieving data from it is to call VideoCapture::grab() first and then call VideoCapture::retrieve() one or more times with different values of the channel parameter. REF: tutorial\_kinect\_openni

#### isOpened public boolean isOpened() Returns true if video capturing has been initialized already. If the previous call to VideoCapture constructor or VideoCapture::open() succeeded, the method returns true.Returns:automatically generated

#### open public boolean open(int index) Open a camera for video capturing Parameters are same as the constructor VideoCapture(int index)Parameters:index - automatically generated Returns:true if the camera has been successfully opened. The method first calls VideoCapture::release to close the already opened file or camera.

#### open public boolean open(int cameraNum, int apiPreference) Open a camera for video capturing Parameters are similar as the constructor VideoCapture(int index),except it takes an additional argument apiPreference. Definitely, is same as open(int index) where index=cameraNum + apiPreferenceParameters:cameraNum - automatically generatedapiPreference - automatically generated Returns:true if the camera has been successfully opened.

#### open public boolean open(java.lang.String filename) Open video file or a capturing device or a IP video stream for video capturing Parameters are same as the constructor VideoCapture(const String& filename)Parameters:filename - automatically generated Returns:true if the file has been successfully opened The method first calls VideoCapture::release to close the already opened file or camera.

#### open public boolean open(java.lang.String filename, int apiPreference) Open video file or a capturing device or a IP video stream for video capturing with API Preference Parameters are same as the constructor VideoCapture(const String& filename, int apiPreference)Parameters:filename - automatically generatedapiPreference - automatically generated Returns:true if the file has been successfully opened The method first calls VideoCapture::release to close the already opened file or camera.

#### read public boolean read([Mat](http://docs.google.com/org/opencv/core/Mat.html) image) Grabs, decodes and returns the next video frame.Parameters:image - automatically generated Returns:false if no frames has been grabbed The method/function combines VideoCapture::grab() and VideoCapture::retrieve() in one call. This is the most convenient method for reading video files or capturing data from decode and returns the just grabbed frame. If no frames has been grabbed (camera has been disconnected, or there are no more frames in video file), the method returns false and the function returns empty image (with %cv::Mat, test it with Mat::empty()). **Note:** In REF: videoio\_c "C API", functions cvRetrieveFrame() and cv.RetrieveFrame() return image stored inside the video capturing structure. It is not allowed to modify or release the image! You can copy the frame using cvCloneImage and then do whatever you want with the copy.

#### release public void release() Closes video file or capturing device. The method is automatically called by subsequent VideoCapture::open and by VideoCapture destructor. The C function also deallocates memory and clears \\*capture pointer.

#### retrieve public boolean retrieve([Mat](http://docs.google.com/org/opencv/core/Mat.html) image) Decodes and returns the grabbed video frame.Parameters:image - automatically generated Returns:false if no frames has been grabbed The method decodes and returns the just grabbed frame. If no frames has been grabbed (camera has been disconnected, or there are no more frames in video file), the method returns false and the function returns an empty image (with %cv::Mat, test it with Mat::empty()). SEE: read() **Note:** In REF: videoio\_c "C API", functions cvRetrieveFrame() and cv.RetrieveFrame() return image stored inside the video capturing structure. It is not allowed to modify or release the image! You can copy the frame using cvCloneImage and then do whatever you want with the copy.

#### retrieve public boolean retrieve([Mat](http://docs.google.com/org/opencv/core/Mat.html) image, int flag) Decodes and returns the grabbed video frame.Parameters:flag - it could be a frame index or a driver specific flagimage - automatically generated Returns:false if no frames has been grabbed The method decodes and returns the just grabbed frame. If no frames has been grabbed (camera has been disconnected, or there are no more frames in video file), the method returns false and the function returns an empty image (with %cv::Mat, test it with Mat::empty()). SEE: read() **Note:** In REF: videoio\_c "C API", functions cvRetrieveFrame() and cv.RetrieveFrame() return image stored inside the video capturing structure. It is not allowed to modify or release the image! You can copy the frame using cvCloneImage and then do whatever you want with the copy.

#### set public boolean set(int propId, double value) Sets a property in the VideoCapture.Parameters:propId - Property identifier from cv::VideoCaptureProperties (eg. cv::CAP\_PROP\_POS\_MSEC, cv::CAP\_PROP\_POS\_FRAMES, ...) or one from REF: videoio\_flags\_othersvalue - Value of the property. Returns:true if the property is supported by backend used by the VideoCapture instance. **Note:** Even if it returns true this doesn't ensure that the property value has been accepted by the capture device. See note in VideoCapture::get()

* [Overview](http://docs.google.com/overview-summary.html)
* [Package](http://docs.google.com/package-summary.html)
* Class
* [Tree](http://docs.google.com/package-tree.html)
* [Index](http://docs.google.com/index-all.html)
* [Help](http://docs.google.com/help-doc.html)
* Prev Class
* [Next Class](http://docs.google.com/org/opencv/videoio/Videoio.html)
* [Frames](http://docs.google.com/index.html?org/opencv/videoio/VideoCapture.html)
* [No Frames](http://docs.google.com/VideoCapture.html)
* [All Classes](http://docs.google.com/allclasses-noframe.html)
* Summary:
* Nested |
* Field |
* [Constr](#3znysh7) |
* [Method](#2et92p0)
* Detail:
* Field |
* [Constr](#3dy6vkm) |
* [Method](#26in1rg)

Generated on 2021-04-02 03:15:03 / OpenCV 3.4.14