

# FormatConverter

## FormatConverterOpenCv C++ library

---

v5.0.3

## Table of contents

---

- [Overview](#)
- [Versions](#)
- [Library files](#)
- [FormatConverterOpenCv class description](#)
  - [FormatConverterOpenCv class declaration](#)
  - [Default constructor](#)
  - [convert method](#)
  - [getVersion method](#)

## Overview

---

**FormatConverterOpenCv** library intended to convert pixels formats of images between each others. **FormatConverterOpenCv** supports all uncompressed pixel formats listed in **Fourcc** enum (RGB24, BGR24, YUYV, UYVY, GRAY, YUV24, NV12, NV21, YU12, YV12). FormatConverterOpenCv uses OpenCV library (version 4.5.0 and more) to convert particular pixel formats. Main file FormatConverterOpenCv.h contains declaration of **FormatConverterOpenCv** class.

## Versions

---

**Table 1** - Library versions.

Version	Release date	What's new
1.0.0	08.09.2020	First version.
2.0.0	19.11.2020	- New formats added.
3.0.0	20.12.2020	- Code optimized.
4.0.0	20.01.2023	- Programming interface changed.
5.0.0	19.03.2023	- New base Frame class added. - Programming interface changed.

Version	Release date	What's new
5.0.1	29.05.2023	- Frame class updated.
5.0.2	12.11.2023	- Frame class updated.
5.0.3	04.01.2024	- Frame class updated. - Documentation updated.

## Library files

The library is supplied only by source code. The user is given a set of files in the form of a CMake project (repository). The repository structure is shown below:

```

CMakeLists.txt ----- main CMake file
3rdparty ----- folder with 3rdparty libraries
  CMakeLists.txt ----- CMake file for 3rdparty folder
  Frame ----- folder with Frame library source code
src ----- folder with library source code
  FormatConverterOpenCv.cpp ----- library source file
  FormatConverterOpenCv.h ----- library main header file
  FormatConverterOpenCvVersion.h ----- header file with library version
  FormatConverterOpenCvVersion.h.in ----- service CMake file to generate version file
test ----- folder of test application
  CMakeLists.txt ----- CMake file of test application
  main.cpp ----- source C++ file of test application
  logo_GRAY.cpp ----- source file which contains test image data
  logo_NV12.cpp ----- source file which contains test image data
  logo_NV21.cpp ----- source file which contains test image data
  logo_RGB24.cpp ----- source file which contains test image data
  logo_UYVY.cpp ----- source file which contains test image data
  logo_YU12.cpp ----- source file which contains test image data
  logo_YUV24.cpp ----- source file which contains test image data
  logo_YUYV.cpp ----- source file which contains test image data
  logo_YV12.cpp ----- source file which contains test image data

```

## FormatConverterOpenCv class description

### FormatConverterOpenCv class declaration

**FormatConverterOpenCv.h** file contains **FormatConverterOpenCv** class declaration. Class declaration:

```

class FormatConverterOpenCv
{
public:

    /// Static method to obtain class version.

```

```

static std::string getVersion();

/// Class constructor.
FormatConverterOpenCv();

/// Class destructor.
~FormatConverterOpenCv();

/// Convert pixel format.
bool convert(Frame& src, Frame& dst);
};

```

## convert method

**convert(...)** method intended to convert **Frame** class data between pixel formats. The method determines which pixel format to convert from based on the **fourcc** field of the Frame class objects. Method declaration:

```
bool convert(Frame& src, Frame& dst);
```

Parameter	Description
src	Source Frame object. Method doesn't support H263, HEVC and JPEG pixel formats.
dst	Destination Frame object. User must at least initialize <b>fourcc</b> field of destination <b>Frame</b> object. Method doesn't support H263, HEVC and JPEG pixel formats.

**Returns:** TRUE if pixel format converted or FALSE if not.

Example:

```

// Init frame converter.
FormatConverterOpenCv converter;

// Init source image filled by 0.
Frame src(640, 480, Fourcc::BGR24);

// Init output image.
Frame dst;
dst.fourcc = Fourcc::YUV24;

// Convert.
converter.convert(src, dst);

```

# getVersion method

---

**getVersion()** method returns string of current version of **FormatConverterOpenCv** class. Method declaration:

```
static std::string getVersion();
```

Method can be used without **Frame** class instance. Example:

```
cout << "FormatConverterOpenCv class version: " << FormatConverterOpenCv::getVersion() <<  
endl;
```

Console output:

```
FormatConverterOpenCv class version: 5.0.3
```