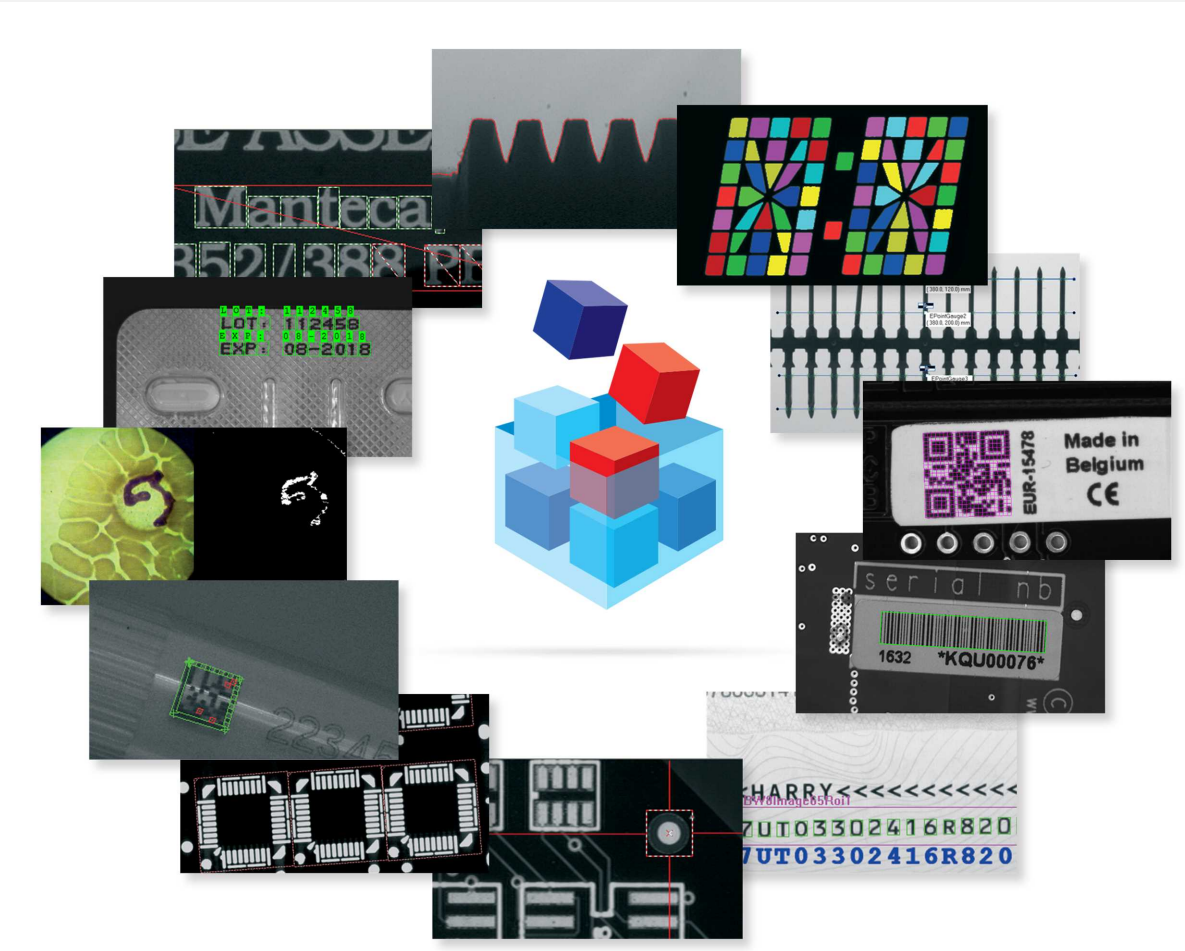
EURESYS™
Excellence in vision

RELEASE NOTES

Open eVision

Release 2.10.0



© EURESYS s.a. 2019 - Document D101EN-Release Notes-Open eVision-2.10.0.1121 built on 2019-10-03

Terms of Use

EURESYS s.a. shall retain all property rights, title and interest of the documentation of the hardware and the software, and of the trademarks of EURESYS s.a.

All the names of companies and products mentioned in the documentation may be the trademarks of their respective owners.

The licensing, use, leasing, loaning, translation, reproduction, copying or modification of the hardware or the software, brands or documentation of EURESYS s.a. contained in this book, is not allowed without prior notice.

EURESYS s.a. may modify the product specification or change the information given in this documentation at any time, at its discretion, and without prior notice.

EURESYS s.a. shall not be liable for any loss of or damage to revenues, profits, goodwill, data, information systems or other special, incidental, indirect, consequential or punitive damages of any kind arising in connection with the use of the hardware or the software of EURESYS s.a. or resulting of omissions or errors in this documentation.

This documentation is provided with Open eVision 2.10.0 (doc build 1121).
© 2019 EURESYS s.a.

Contents

1. Release Benefits

Summary

EasyOCR2

- Open eVision now supports `ERegion` when calling `Read` and `Detect` methods.

EasyFind

- The speed for `Learn` and `Find` operations using `ERegion` is improved.

EasyImage

- The **EasyImage** processing functions are optimized (some functions are up to 200 × faster than in the previous releases).

2. Release Specifications

OS and processor architectures

- Open eVision is a 32-bit and 64-bit library that requires a processor compatible with the SSE2 instruction set.
- Open eVision runs on the following operating systems:

OS version	Additional information	
Windows 10®	32-bit	—
Windows 10®	64-bit	—
Windows 8®	32-bit	—
Windows 8®	64-bit	—
Windows 7®	32-bit	The recommended version is 6.1.7601 (Windows 7 Service Pack 1)
Windows 7®	64-bit	

- Remote connections:
  - You can install and use Open eVision licenses on a remote connection using remote desktop, TeamViewer or any other similar software.
- Virtual machines:
  - You cannot install Open eVision on virtual machines.

Supported IDE and programming languages

Select the recommended API Module according to your IDE and programming language:

	Programming language	
IDE	C++	C#, VB.NET, C++/CLI
Microsoft Visual Studio 2008® SP1	C++	.NET Assembly
Microsoft Visual Studio 2010®	C++	.NET Assembly
Microsoft Visual Studio 2012®	C++	.NET Assembly
Microsoft Visual Studio 2013® (*)	C++	.NET Assembly
Microsoft Visual Studio 2015®	C++	.NET Assembly
Microsoft Visual Studio 2017®	C++	.NET Assembly

NOTE
(*) Visual C++ MFC MBCS Library for Visual Studio 2013 must be installed.

Required system resources

- Display size:
  - Minimum: 800 x 600
  - Recommended: 1280 x 1024
- Display color depth:
  - Minimum: 16 bits
  - Recommended: 32 bits
- Hard disk space:
  - Open eVision libraries: 100 MB - 2 GB (depending on selected options)

3. Release Details

3.1. New and Improved Features

New features

EasyOCR2

- Open eVision now supports `ERegion` when calling `Read` and `Detect` methods.

Improvements

EasyFind

- The speed for `Learn` and `Find` operations with `ERegion` is improved.

EasyImage

- The following processing functions are optimized.
  - The listed speed up factors are relative to the processing time measured with the release 2.9.
  - The processed images are 512 × 512 pixels, 8-bit gray, 16-bit gray and 24-bit color format.
  - The processing is executed single thread on 32- and 64-bit architectures.

Processor architecture:		32-bit			64-bit		
	Pixel format:	8-bit	16-bit	24-bit	8-bit	16-bit	24-bit
Convolution functions (Sobel, Prewit, High / LowPass, Laplacian...)		1.0	1.5	1.0	1.0	1.5	0.7
Gaussian and uniform convolutions	3 × 3 kernel	1.0			1.0		
	11 × 11 kernel	1.5			1.5		
	21 × 21 kernel	2.0			2.0		
Convolution with user defined kernel	3 × 3 kernel	1.4			1.5		
	7 × 7 kernel	2.1			2.0		
	21 × 21 kernel	2.1			2.2		
Morphological operations (tested with 3 × 3 to 21 × 21 kernel size)	ErodeBox / DilateBox	1.0 -2.7	20 -200	1.1-2.0	1.0-2.8	20-190	1.0-2.3
	OpenBox / CloseBox	1.0-5.0	6.0-120	1.0-1.4	1.0-3.4	21-200	0.7
	BlackTopHat/WhiteTopHat	1.0-2.7	6.0-112	1.0-1.4	1.0-2.8	17-190	0.7
LinearTransform, ScaleRotate	Nearest	1.6			2.4		
	Linear interpolation	1.8			2.5		
	Cubic interpolation	3.0			3.0		

3.2. Changes

Starting with this release 2.10, Open eVision implements the following changes:

EasyImage

- The internal rounding conventions are changed.

  This can lead to:

  - Small differences in the pixel values for the image borders when using `EasyImage::LinearTranform` and `EasyImage::ScaleRotate`.
  - Slightly different results for `EHarrisCornerDetector` when using the gaussian filter (set with the `EHarrisCornerDetector::SetDerivationScale` function).
  - Slightly different results for `ECannyEdgeDetector` when using the gaussian filter (set with the `ECannyEdgeDetector::SetSmoothingScale` function).

EasyOCR2

- The `Read` and `Detect` methods on a ROI with a dimension smaller than 3 now throws an exception.

3.3. Solved Issues

The following issues have been fixed in Open eVision 2.10:

EasyImage

- The convolution with user-defined even-dimension kernel does not produce artifacts anymore.
- The morphological operation on 16-bit images are now correct at image borders.
- The `ConvolKernel` function returns a correct result when the kernel size is larger than the image size.

EasyOCR

- The `RecognizeWide` function does not crash and does not throw an exception anymore.

EasyGauge

- No detected points are considered as valid anymore when the shape (rectangle, circle...) fitting process fails.

EasysMatrixCode2

- The Cell Modulation Grade calculation is now correct.

Open eVision Studio

- The code generated for the **EasyMatrixCode2** timeout is now correct.
- **EasyMatrixCode2** does not crash anymore when you execute a learn operation with no image selected.

4. Known Issues

.NET API and unsigned integer parameters

Since this release 2.5 of Open eVision, unsigned integer parameters in the C++ API are not exposed in the .NET API as signed integer anymore, but as unsigned integers. This brings the .NET API closer to the C++ one.

This change does not cause any issue except when you want to pass an enumerate value as one of these parameters. In these specific cases, update your casting operation as in the following example:

```
codedImage.SetThreshold((int)EThresholdMode.MinResidue);
```

becomes:

```
codedImage.SetThreshold(unchecked((uint)EThresholdMode.MinResidue)); .
```

Reserved keywords

The following keywords are reserved by Open eVision:

- EUnit_um, EUnit_mm, EUnit_cm, EUnit_dm
- EUnit_m, EUnit_dam, EUnit_hm, EUnit_km
- EUnit_mil, EUnit_inch, EUnit_foot, EUnit_yard, EUnit_mile
- EasyWorld

TIP
To avoid conflict, do not use these keywords to name variables, functions, methods, macros...

Image formats

- If you use some types of 96-bit RGB Tiff image, Open eVision may crash.

Object cleanup: .NET

As a rule, it is highly recommended to call `Dispose()` on Open eVision .NET objects when they are not useful anymore.

TIP
Not doing so might result in unnecessarily high memory usage and crashes.

Example in C#

```
using(EImageBW8 src = new EImageBW8())
using(EPatternFinder finder = new EPatternFinder())
{
  src.Load(ImageFilePath);
  EFoundPattern[] foundPatterns = finder.Find(src);
  ...
  foreach(EFoundPattern foundPattern in foundPatterns)
  {
    foundPattern.Dispose();
  }
}
```

In addition, if you use a nested object (such as the segmenter properties in EasyObject encoder objects), remember to call `Dispose()` on that object before calling `Dispose()` on the parent object.

Example in C#

```
imageEncoder.GrayscaleSingleThresholdSegmenter.BlackLayerEncoded = true;
...
imageEncoder.GrayscaleSingleThresholdSegmenter.Dispose();
imageEncoder.Dispose();
```

Basic types: retrieving and setting pixel values

Using the `GetPixel()` and `SetPixel()` methods of the various ROI classes can sometimes be slow if you make many calls (regardless of the language used).

- In order to greatly speed up the ROI/image buffer access, embed the buffer access in your own code.
- See the examples below that use the new Open eVision API.

NOTE
For a better readability of these examples, the variable declarations and initializations have been omitted when possible.

Example in C++

```
void* pixAddr;
UINT8 pix;
...
for (int y = 0; y < height; ++y)
{
  pixAddr = bw8Image.GetImagePtr(0,y);
  for (int x = 0; x < width; ++x)
  {
    pix = *(reinterpret_cast<UINT8*>(pixAddr)+x);
  }
}
```

Example in C#

```
using System.RunTime.InteropServices;
...
IntPtr pixAddr;
byte pix;
...
for (int y = 0; y < height; ++y)
{
  pixAddr = bw8Image.GetImagePtr(0,y)
  for (int x = 0; x < width; ++x)
  {
    pix = Marshal.ReadByte(pixAddr,x)
  }
}
```

Basic types: ROI zooming and panning issue

- When drawing an ROI with a zoom factor, applying panning (retrieved from a scroll bar) causes the ROI display to be shifted. Consequently, the `HitTest()` and `Drag()` functions fail because the handles do not appear at their actual positions.

  Workaround: The panning values should be divided by the zoom factor before calling the `DrawFrame()`, `HitTest()` and `Drag()` functions.

Basic Types: miscellaneous issues

- TIFF files containing RGB values + alpha values are not supported.
- Filenames with multibyte characters are not supported. The error is "Unrecognized file format".
- `Easy::GetBestMatchingImageType()` only works for BW8 and C24 images.

EasyBarCode

- Due to a bug in the debugger of Visual C++ 2012, the reading time of bar codes may increase after a failed reading. This happens only in debug mode with Visual C++ 2012.
- EasyBarCode requires that a quiet zone of at least one full module is present around the whole bar code to be read.
- EasyBarCode is currently unable to read bar codes with curved or distorted bars. For reliable reading, the bars must be as straight as possible.
- EasyBarCode is currently not multithread-safe.

EasyQRCode

- EasyQRCode does not support MicroQR codes.

EasyObject

- The `ECodedImage2` and `EHarrisDetector` results are drawn slowly when there are many results.

EasyMatch

- By design, the maximum size for a pattern in EasyMatch is 1791 x 1791.
- Matching a vertically symmetric pattern with an angle tolerance around 180° and in the original image can lead to an error of 1 pixel on the detected position.
- By default, EasyMatch interpolation does not work on 15 x 15 and smaller patterns.

  Workaround: For pattern sizes smaller than 16 x 16, adjust the `MinReduced` area to fit the `MinReducedArea < W*H/4` (if interpolation is needed).

EasyGauge

- In .NET, the `EPointGauge.GetMeasuredPoint()` overload with no argument is not available. To get the default measured point, use -1 as index.
- By design, an `ELineGauge`, `ERectangleGauge`, `ECircleGauge` or `EWedgeGauge` is reported as invalid if at least one of its sample points is invalid. In addition, these invalid sample points cannot be drawn as they have not been measured successfully.
- The `EWedgeGauge::SetActiveEdges()` method incorrectly gets the `EDragHandle_Edge_r` and `EDragHandle_Edge_RR` bits mixed up when processing its argument.

  Workaround: In order to activate the inner circle, set the `EDragHandle_Edge_RR` flag and use the `EDragHandle_Edge_r` flag to activate the outer circle.
- Using a gauge on an ROI leads to drawing problems.

  Workaround: Use the gauge on the parent image.
- In the custom `EDraggingMode_ToEdges` dragging mode, you cannot resize the nominal wedge gauge position using the on-screen handles, neither in a custom application nor in Open eVision Studio or in Open eVision Eval.

  Workaround: Enter numerical values for the wedge gauge position.

EasyMatrixCode

- When grading is enabled, the optimizations are made in order to get accurate grading rather than have the best possible reading. As a result, the number of decoding errors reported with grading can be higher than without grading.
- Inspecting images with a lot of details, even if they are low contrast, can require much more time spent in EasyMatrixCode than the `TimeOut` set previously.
- In .NET, retrieving the coordinates of a MatrixCode using `EMatrixCode.GetCorner()` or `EMatrixCode.Center()` can lead to an unhandled exception when the garbage collection starts up. To avoid this problem, call `Dispose()` on the `EPoint` objects returned by these functions when they are no longer needed.

Open eVision Studio

- In the ROI management dialog, clicking on a ROI in the tree view does not activate the ROI overlay in the image window. This can prevent you to graphically interact with it.

  To avoid this issue and to properly interact with the ROI overlay:

  a. Click on the ROI in the tree view.

  b. Immediately after, click inside its overlay in the image window.

- To avoid crashes, deselecting all detection methods in the EasyQRCode dialog box reverts to the default detection method. In some cases, the dialog might not refresh automatically.
- In the detection method selection control of the EasyQRCode dialog box, clicking beside a text might select or deselect it.
- When managing the EasyOCR2 topology, the potential characters option is not available.

Open eVision installer

- There is a conflict between the Open eVision installer and any program using the UDP:6001 port. When a software is already using this port, the installation fails and rolls back.

  Workaround: Install Open eVision first, and then the other software.

> **NOTE**
> This port is typically used by National Instrument software such as LabView.

- Before installing any Euresys product, make sure that your OS is up-to-date (using Microsoft Update), otherwise, problems might occur.

Open eVision License Manager

- Under Windows XP, the Open eVision License Manager might not start if the .NET Framework 2.0 is not installed.
- Using the Open eVision License Manager to activate a license requires an Internet connection and a secure SSL transaction to EURESYS s.a. servers.

> **NOTE**
> On older systems, such as Windows XP SP3, ensure that the root certificates are up-to-date otherwise the secure connection is refused and the license is not activated.

- When activating an emergency license, the following error may occur: "`Error Message: Loading of the ASR failed!`"

  This error occurs when all 3 emergency licenses have already been used and the computer has been formatted.

- Using Open eVision License Manager in English language mode on a Chinese or Japanese Windows version can lead to truncated text being displayed. This is an issue linked to the automatic font selection and there is currently no workaround. Please note however that, by default, the Open eVision License Manager runs in the OS language, including Chinese and Japanese.