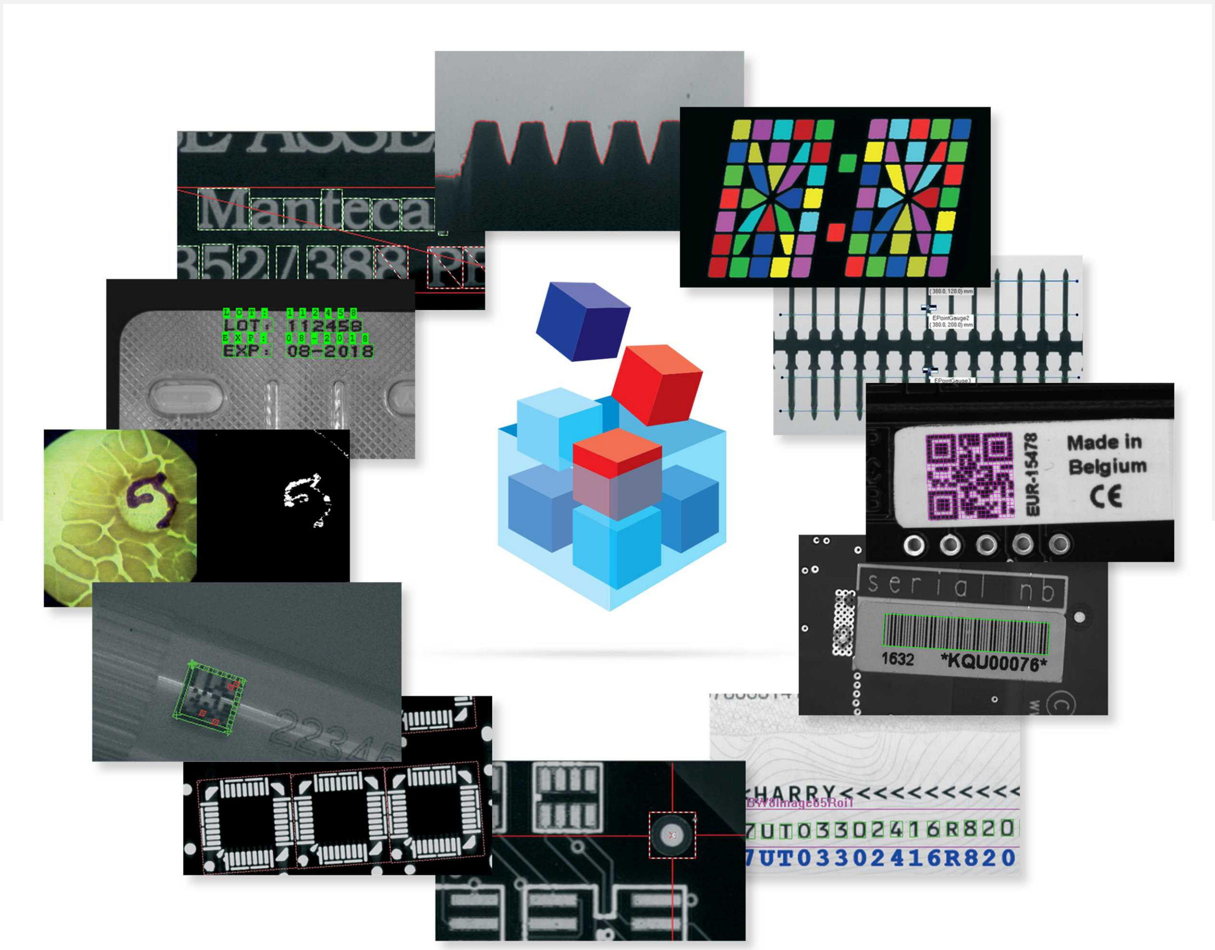
RELEASE NOTES

Open eVision

Release 2.5.1



This documentation is provided with Open eVision 2.5.1 (doc build 1107).
© 2018 EURESYS s.a.

Contents

1. Release Benefits

Summary

Starting with this release 2.5, Open eVision offers:

EasyMatrixCode2: new reader (BETA)

- A new Data Matrix code reader.

  Note: *This is a beta version of the library.*

EasyDeepLearning: new library (BETA)

- A new library for learning to classify images from a set of example images.

  Note: *This is a beta version of the library.*

2. Release Specifications

OS and processor architecture

- 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	—
Windows 7®	64-bit	—

- Remote connection:
  - You can install and use Open eVision licenses on a remote connection using remote desktop, TeamViewer or any other similar software.
- Virtual machine:
  - 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				Notes
IDE	C++	Basic	C#, BV.NET, C++/CLI	Object Pascal	
Microsoft Visual Studio 6.0® SP6	C++	ActiveX Library	—	—	
Microsoft Visual Studio .NET 2003® SP1	C++	—	—	—	
Microsoft Visual Studio 2005® SP1	C++	—	.NET Assembly	—	
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	—	

	Programming language				Notes
IDE	**C++**	**Basic**	**C#, BV.NET, C++/CLI**	**Object Pascal**	
Microsoft Visual Studio 2017®	C++	—	.NET Assembly	—	
CodeGear Delphi 2009®	—	—	—	ActiveX Library	
Borland C++ Builder 6.0® update 4	C++	—	—	—	
CodeGear C++ Builder 2009®	C++	—	—	—	See "Known Issues" on page 11
Embarcadero RAD Studio XE4	C++	—	—	ActiveX Library	
Embarcadero RAD Studio XE5	C++	—	—	ActiveX Library	

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: 25-300 MB (depending on selected options)
  - Open eVision development tools: 100-600 MB (depending on selected options)

3. Release Details

3.1. Added Products

Starting with this release 2.5, Open eVision offers the following new products:

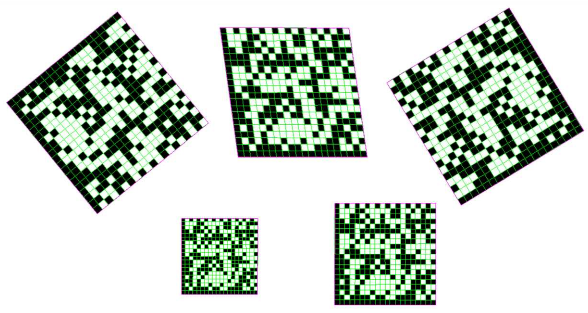
EasyMatrixCode2

Important: *This library is currently in beta release. Its performance and features may be substantially modified before it is commercially released. It must not be used except for the purpose of evaluating the new functionality. Euresys makes no warranties, express or implied, with respect to the performance and features of this beta release.*

Starting with release 2.5, Open eVision introduces a new data matrix code reader, named "EasyMatrixCode2". In the future, EasyMatrixCode2 will be able to replace the current EasyMatrixCode reader (called "EasyMatrixCode1" from now on) with the following benefits:

- Ability to read multiple data matrix codes in an image.
- Improved consistency of reading and grading results.
- Improved consistency of processing time.
- Improved handling of distorted data matrix codes (this feature is not available yet and will be in a later release).

Note: *EasyMatrixCode1 and EasyMatrixCode2 are both available in the EasyMatrixCode library.*



EasyDeepLearning

> **Important:** *This library is currently in beta release. Its performance and features may be substantially modified before it is commercially released. It must not be used except for the purpose of evaluating the new functionality. Euresys makes no warranties, express or implied, with respect to the performance and features of this beta release.*

EasyDeepLearning is a library that classifies images using deep convolutional neural networks (CNNs). You can use it, for example, to identify a product in an image or to detect if the product is good or defective.

Contrary to traditional machine vision techniques, EasyDeepLearning does not require an explicit model of what to recognize inside an image. Instead, it learns this model from a set of example images. Thus it is able to solve machine vision problems where an explicit model is too complex to build.

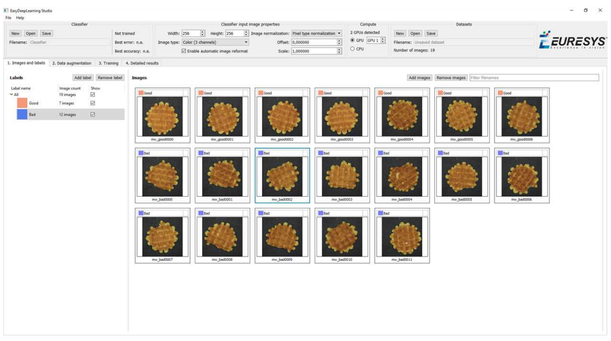
To accelerate computations, we strongly recommend running the EasyDeepLearning library on a recent NVIDIA GPU.

> **Note:** *The EasyDeepLearning library runs only on x64 platforms.*

EasyDeepLearning Studio

The EasyDeepLearning Studio is a graphical user interface for:

- Creating datasets and labeling images,
- Configuring and visualizing the data augmentation transformations,
- Training a classifier,
- Analyzing the performance of the classifier.



Filename: Classifier
Not trained
Best error: n.a.
Best accuracy: n.a.
Width: 256 Height: 256 Image normalization: Pixel type normalization
Image type: Color (3 channels) Offset: 0,000000
☑ Enable automatic image reformat Scale: 1,000000
2 GPUs detected
● GPU GPU 1
○ CPU
Filename: Unsaved dataset
Number of images: 19

Label name	Image count	Show
All	19 images	☑
Good	7 images	☑
Bad	12 images	☑

3.2. Added and Improved Features

Starting with this release 2.5, Open eVision offers:

EasyMatch improvement

- The learning algorithm is updated:
  - Enable the option *AdvancedLearning* (which is the default settings) to fine tune the resolution of the pattern using the learning image context.
  - This improvement is significant when processing tiled and periodic images.
  - To disable the advanced learning, call *SetAdvancedLearning(false)*.

Easy3D improvements

- Object based calibration: the multipass-calibration is added. More precise calibration models are computed when multi-pass is enable.
- Depth maps now support a "YFlipped" flag, indicating that the depth map is stored upside-down.
- You can keep the current view in E3DViewer after submitting a new rendering source.

Open eVision Studio and the documentation

- Links to the documentation are again available in Open eVision Studio.
  - If the documentation is not installed locally on the computer, the online version is automatically displayed.

3.3. Solved Issues

Starting with the release 2.5, Open eVision does not exhibit the following known issues anymore:

Easy

- The *EROI*::Drag* does not resize the ROI with negative values anymore.

EasyImage

- The *ConvolGaussian* computes the convolution correctly even on large images.

Easy3D

- DepthMap and ZMap serialization in Open eVision format is fixed.

Open eVision Studio and EasyColor

- The gain slider is now updated correctly on the Adjust Gain/Offset dialog when the range is not 0-1.

3.4. Breaking Changes

Starting with this release 2.5, Open eVision implements the following breaking changes:

Easy3D

- The *E3DViewer.SetPointCloud* deprecated method is removed. Use the *E3DViewer.ConfigureRenderSource* method instead.
- *E3DTransformMatrix:isRigid* becomes *E3DTransformMatrix:IsRigid*.
- *void E3DCalibrationModel::Apply(const E3DPoint&, E3DPoint&)* becomes *E3DPoint E3DCalibrationModel::Apply(E3DPoint)*.
- Due to depth map and ZMap serialization rework, previously saved files may not be compatible with 2.5 (and later) version.
- Depth maps and ZMaps can be stored and restored to and from a file using the *Save* and *Load* methods (image pixels and all other object attributes). To save and load ONLY the image associated with the depth map and the ZMap to and from regular image files, use the methods *SaveImage* and *LoadImage*.
- *E3DObjectBasedCalibrationModel* objects saved with version 2.4.1 (or earlier) are not compatible with version 2.5. You should recreate these calibration models.

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)); .
```

Borland development tools

Since the release 2.4.1, Open eVision does not work with the following Borland C++ compilers:

- CodeGear C++ Builder 2009®

Note: *If you are using this development environment, please wait for a coming update solving this issue .*

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

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

Object cleanup: ActiveX

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

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

Example in Visual Basic

```
src = New EImageBW8
finder = New EPatternFinder
src.Load ImageFilePath
foundPatterns = finder.Find(src)
...
For Each foundPattern In foundPatterns
  foundPattern.Dispose
Next
finder.Dispose
src.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 Visual Basic

```
imageEncoder.GrayscaleSingleThresholdSegmenter.BlackLayerEncoded = True
...
imageEncoder.GrayscaleSingleThresholdSegmenter.Dispose
imageEncoder.Dispose
```

Object cleanup: .NET

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

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)
  }
}
```

Example in Visual Basic 6.0

```
Private Declare Sub GetMem1 Lib "msvbvm60" (ByVal Addr As Long, RetVal As Byte)
...
Dim curAddr As Long
Dim pix As Byte
...
For y = 0 To h - 1
  curAddr = bw8Image.GetImagePtrXY(0, y)
  For x = 0 To w - 1
    GetMem1 curAddr, pix
    curAddr = curAddr + 1
  Next x
Next y
```

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 bended bars. For reliable reading, the bars must be as straight as possible.
- EasyBarCode is currently not multithread-safe.

EasyQRCode

- EasyQRCode does not support MicroQR code.

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.

Open eVision Documentation

- The HTML documentation uses recent features of HTML for its search functionality. On older OS such as Windows XP it might be necessary to upgrade the web browser to ensure full compatibility.
- For best viewing experience, Mozilla Firefox is recommended.