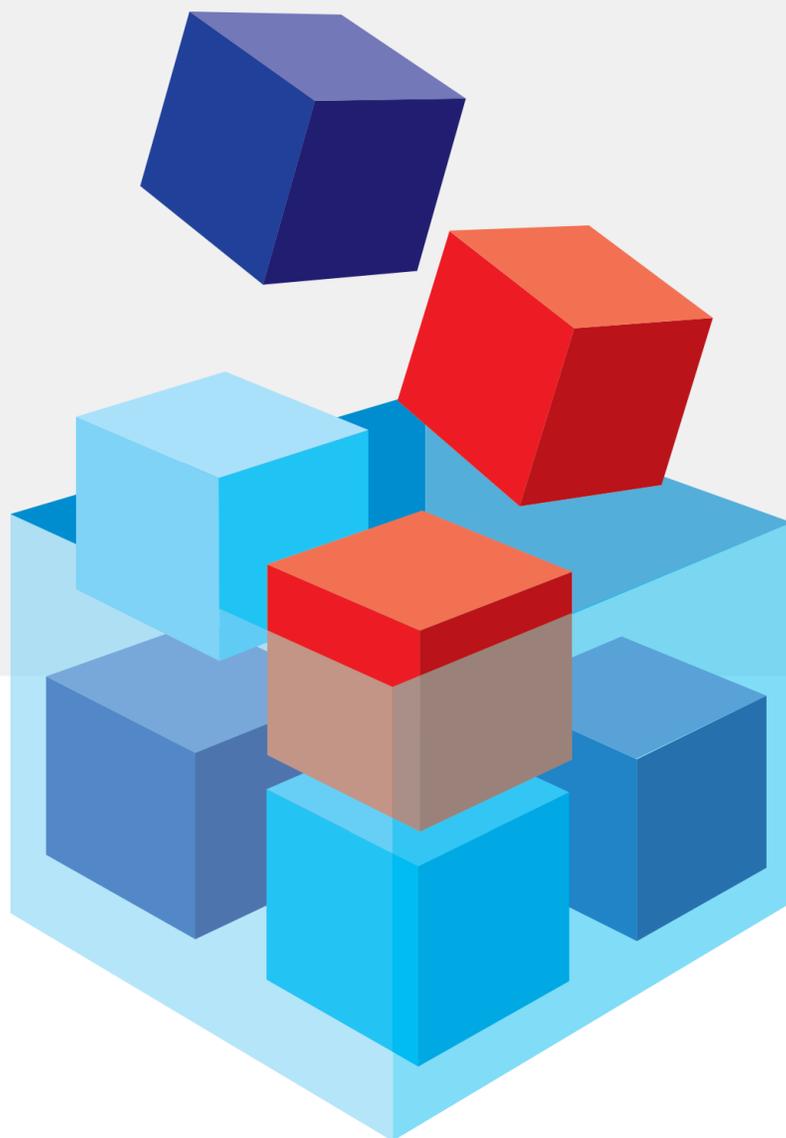


# Open eVision 2.1



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.

# Contents

<b>Release Benefits</b>	<b>4</b>
A New Library: Easy3D.....	4
<b>Release Specifications</b>	<b>5</b>
Operating Systems and Processor Architecture .....	5
Supported Integrated Development Environments and Programming Languages .....	6
Required system resources.....	7
<b>Release Details</b>	<b>8</b>
A New Library: Easy3D.....	8
TIFF Image Loading Corrections.....	8
Improved Text Rotation Detection in EasyOCR2 .....	8
<b>Known Issues</b>	<b>9</b>
Reserved keywords .....	9
ActiveX: Object cleanup .....	9
.NET: Object cleanup.....	10
Basic Types: retrieving and setting pixel values .....	11
Basic Types: ROI zooming and panning issue .....	12
Basic Types: miscellaneous issues .....	12
EasyBarCode .....	12
EasyQRCode .....	13
EasyObject.....	13
EasyMatch .....	13
EasyGauge .....	13
EasyMatrixCode.....	14
Open eVision Studio.....	14
Open eVision Installer .....	14
Open eVision License Manager .....	14
Open eVision Documentation.....	15

# Release Benefits

## A New Library: Easy3D

A new library, Easy3D, has been added to Open eVision. This library features classes and basic processing features that will serve as the foundations of the future Open eVision 3D tools.

# Release Specifications

## Operating Systems and Processor Architecture

Open eVision 2.1 is a 32-bit and 64-bit library that requires a processor compatible with the SSE2 instruction set. Open eVision 2.1 can be used 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	—
Windows Vista®	32-bit	Service Pack 1
Windows XP®	32-bit	Service Pack 3
Windows Server 2008®	32-bit	—
Windows Server 2008® R2	64-bit	—
Windows Server 2003®	32-bit	Service Pack 1
Windows Embedded Standard 2009®	32-bit	—

**Note:** The Open eVision 2.1 installer does not allow the installation of the product on virtual machines.

## Supported Integrated Development Environments and Programming Languages

IDE	Language	Open eVision API module
<b>Microsoft Visual Studio 6.0® SP6</b>	C++	C++
	Basic	ActiveX Library
<b>Microsoft Visual Studio .NET 2003® SP1</b>	C++	C++
<b>Microsoft Visual Studio 2005® SP1</b>	C++	C++
	C#, VB .NET, C++/CLI	.NET Assembly
<b>Microsoft Visual Studio 2008® SP1</b>	C++	C++
	C#, VB .NET, C++/CLI	.NET Assembly
<b>Microsoft Visual Studio 2010®</b>	C++	C++
	C#, VB .NET, C++/CLI	.NET Assembly
<b>Microsoft Visual Studio 2012®</b>	C++	C++
	C#, VB .NET, C++/CLI	.NET Assembly
<b>Microsoft Visual Studio 2013®</b>	C++	C++
	C#, VB .NET, C++/CLI	.NET Assembly
<b>Borland C++ Builder 6.0® update 4</b>	C++	C++
<b>CodeGear C++ Builder 2009®</b>	C++	C++
<b>CodeGear Delphi 2009®</b>	Object Pascal	ActiveX Library
<b>Embarcadero RAD Studio XE4</b>	C++	C++
	Object Pascal	ActiveX Library
<b>Embarcadero RAD Studio XE5</b>	C++	C++
	Object Pascal	ActiveX Library

## Required system resources

- Minimal display size: 800 x 600. 1280 x 1024 recommended.
- Minimal color depth: 16 bits. 32 bits recommended.
- Between 25 MB and 300 MB free hard disk space needed for Open eVision Libraries, depending upon selected options.
- Between 100 MB and 600 MB free hard disk space needed for Open eVision Development Tools, depending upon selected options.

# Release Details

## A New Library: Easy3D

A new library, Easy3D, has been added to Open eVision. This library features classes and basic processing features that will serve as the foundations of the future Open eVision 3D tools.

Please refer to the Open eVision Documentation for more information about all the features of Easy3D

## TIFF Image Loading Corrections

In some rare cases, a TIFF file could be detected as a RAW file and loading of that file into Open eVision would fail. This has been corrected.

## Improved Text Rotation Detection in EasyOCR2

In some cases, EasyOCR2 could fail to detect correctly the value of text rotation and return a value with an error of 180°. This has been fixed.

# Known Issues

## 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.

Variables, functions, methods, macros and such should not be named using those to avoid conflict.

## ActiveX: Object cleanup

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 VB

```
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
```

Moreover, if you used a nested object, like the segmenter properties in EasyObject's encoder objects, it is important to remember to call **Dispose** on that object before calling **Dispose** on the parent object.

### Example in VB

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

## .NET: Object cleanup

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 ();
    }
}
```

Moreover, if you used a nested object, like the segmenter properties in EasyObject's encoder objects, it is important to 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 many calls are made (regardless of the language used).

In order to greatly speed up ROI/image buffer access, you can embed the buffer access in your own code.

You can find some examples below, using the new Open eVision API. For the sake of readability, 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 barcodes may increase after a failed read. This happens only in debug mode and under Visual C++ 2012.
- EasyBarCode requires that a quiet zone of at least one full module is present around the whole barcode to be read.

- EasyBarcode is currently unable to read barcode with curved or bended bars. For reliable reading, the bars must be as straight as possible.
- EasyBarcode is currently not multithread-safe.

## EasyQRCode

- MicroQR code support is not available in EasyQRCode.

## EasyObject

- **ECodedImage2** and **EHarrisDetector** results draw slowly when there are many results.

## EasyMatch

- The maximum size for a pattern in EasyMatch is 1791x1791. This is by design.
- 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.
- EasyMatch interpolation does not work by default on 15x15 and smaller patterns. As a workaround, for pattern sizes smaller than 16x16, the **MinReduced** area needs to be adjusted to fit **MinreducedArea < W\*H/4** (if interpolation is needed).

## EasyGauge

- Under .NET, the **EPointGauge.GetMeasuredPoint()** overload with no argument is unavailable. To get the default measured point, use -1 as index.
- By design, an **ELineGauge**, **ERectangleGauge**, **ECircleGauge** or **EWedgeGauge** is reported invalid if at least one of its sample points is invalid. Moreover, these invalid sample points cannot be drawn since they haven't been measured successfully.
- The **EWedgeGauge::SetActiveEdges()** method incorrectly gets the **EDragHandle\_Edge\_r** and **EDragHandle\_Edge\_RR** bits mixed up when processing its argument. As a workaround, in order to activate the inner circle, the **EDragHandle\_Edge\_RR** flag needs to be set and, conversely, the **EDragHandle\_Edge\_r** value will toggle the outer circle.
- Using a gauge on an ROI leads to drawing problems. As a workaround, use the gauge on the parent image instead.
- In the custom **EDraggingMode\_ToEdges** dragging mode, it is not possible to resize the nominal wedge gauge position using the on-screen handles, be it in a custom application or in Open eVision Studio or Open eVision Eval. As a workaround, enter numerical values for the wedge gauge position.

## EasyMatrixCode

- When grading is enabled, optimizations are made in order to get accurate grading rather than have the best read possible. As such, the number of decoding errors reported when grading can be higher than when not grading.
- When inspecting images with a lot of details, even if those are low contrast, the time effectively spent into EasyMatrixCode can be significantly superior to 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 treeview does not activate the ROI overlay in the image window. This can prevent you to graphically interact with it. To avoid that issue, click first on the ROI in the treeview and, immediately after, inside its overlay in the image window. This two-step sequence will allow you to interact properly with the ROI overlay.
- In the EasyQRCode dialog, to avoid crashes, deselecting all detection methods will revert to the default detection method. In some cases, however, the dialog might not refresh automatically.
- In the detection method selection control of the EasyQRCode dialog, clicking beside a text might select or deselect it anyway.
- 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 will fail and rollback. This port is typically used by National Instrument software such as LabView.  
To work around this issue, please install Open eVision first, and only then the other software.
- Prior to installing any Euresys product, the OS must be up-to-date (using Microsoft Update). Otherwise, problems can occur.

## Open eVision License Manager

- Under Windows XP, the License Manager may refuse to start if the .NET Framework 2.0 is not installed.
- Using the license manager to activate a license requires an internet connection and a secure SSL transaction to Euresys' servers. As such, on older systems, like Windows XP SP3, it is important to ensure that the root certificates are up to date. If that is not the case, the secure connection will be refused and the license will not be 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 before trying again.

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

### [Open eVision Documentation](#)

- The HTML documentation uses recent features of HTML for its search functionality. As such, on older OSes like Windows XP it might be necessary to upgrade the web browser to ensure full compatibility.