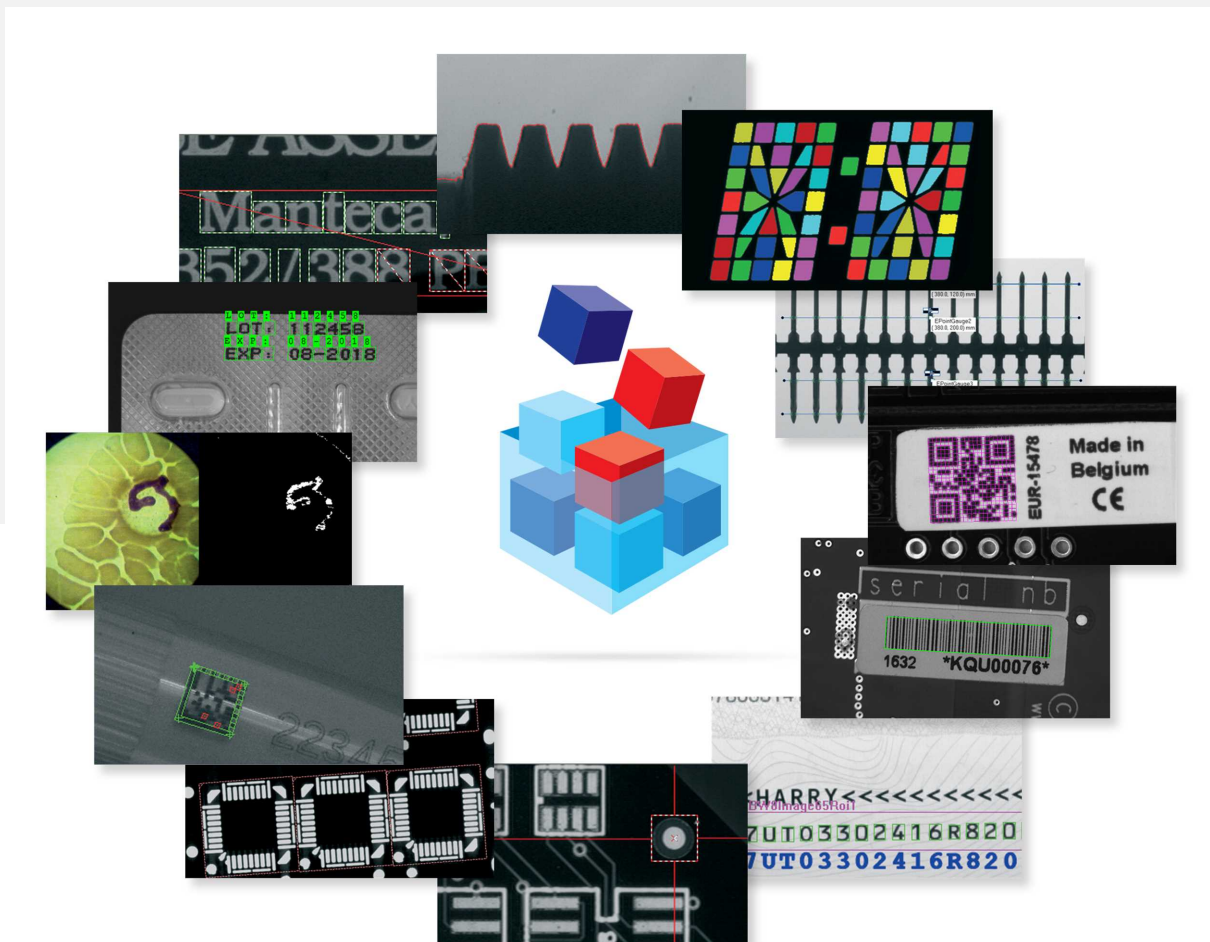


Open eVision

Release 2.6.1



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 2.6.1 (doc build 1110).
© 2018 EURESYS s.a.

Contents

- 1. Release Benefits 4
- 2. Release Specifications 5
- 3. Release Details 7
 - 3.1. Added Products 7
 - 3.2. Updated Development Tools Support 8
 - 3.3. Added and Improved Features 8
 - 3.4. Solved Issues 11
 - 3.5. Breaking Changes 12
- 4. Known Issues 14

1. Release Benefits

Summary

Starting with this release 2.6, Open eVision offers:

EasyMatrixCode: improvements

- Adding of the new `EasyMatrixCode2` object that improves the library capacity to read difficult data matrix codes.

Note: *This is still a beta version of the object.*

EasyDeepLearning: improvements

- Changes to the library to improve the data augmentation process.
- Improvements regarding the use of very large datasets.

Easy3D: namespace

- Migration of all classes and methods into the Easy3D namespace.
- Renaming of most classes for coherence.
- Minor enhancements to various functions.

Easy3D Studio: a new studio

- This new studio application provides an easy way to configure a complete 3D laser line inspection setup.

Licensing on Dongles

- Updating of the support for Dongles to avoid malfunction issues introduced after Windows 10 update 1803 (“April Update”).

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	—
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:

IDE	Programming language	
	C++	C#, VB.NET, C++/CLI
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
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

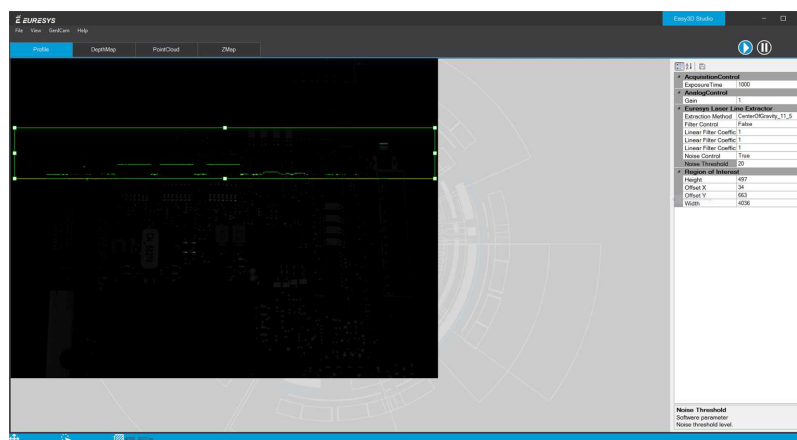
3.1. Added Products

Starting with this release 2.6, Open eVision offers the following new products:

Easy3D Studio

Easy3D Studio is a graphical user interface providing a user-friendly experience and feedback to the following processes:

- Configuration of the parameters of the laser line extraction algorithm.
- Configuration of the size and parameters for the depth map acquisition.
- Configuration of the calibration parameters for the point cloud generation.
- Configuration of the projection parameters for the ZMap generation.



3.2. Updated Development Tools Support

Starting with this release 2.6:

- Borland / Embarcadero support is discontinued.
- ActiveX and OCX interfaces support is discontinued.

Note: *Open eVision 2.5 is the last version supporting these IDE and interfaces.*

3.3. Added and Improved Features

Starting with this release 2.6, Open eVision offers:

EasyMatrixCode2 (BETA)

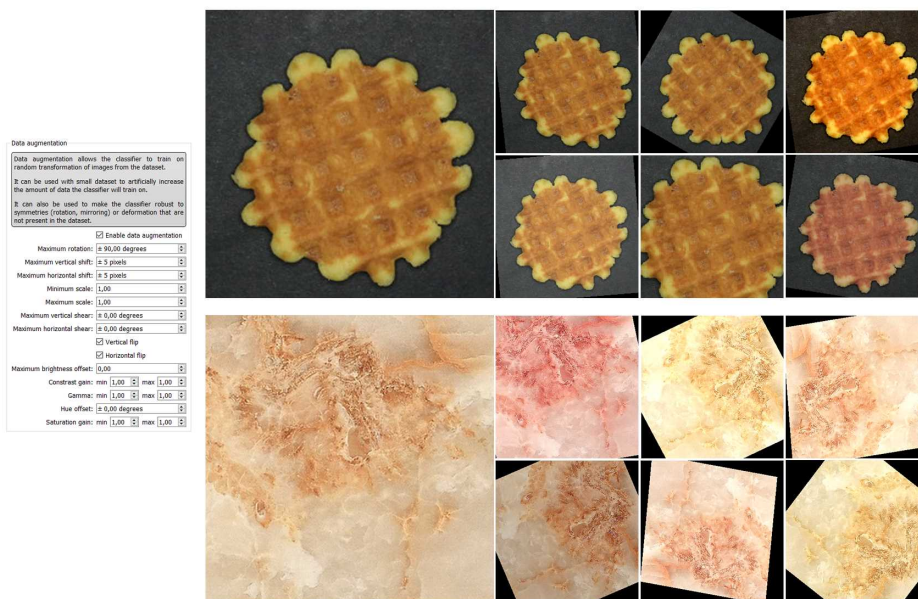
- Improvements to the library capacity of reading difficult data matrix codes.

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

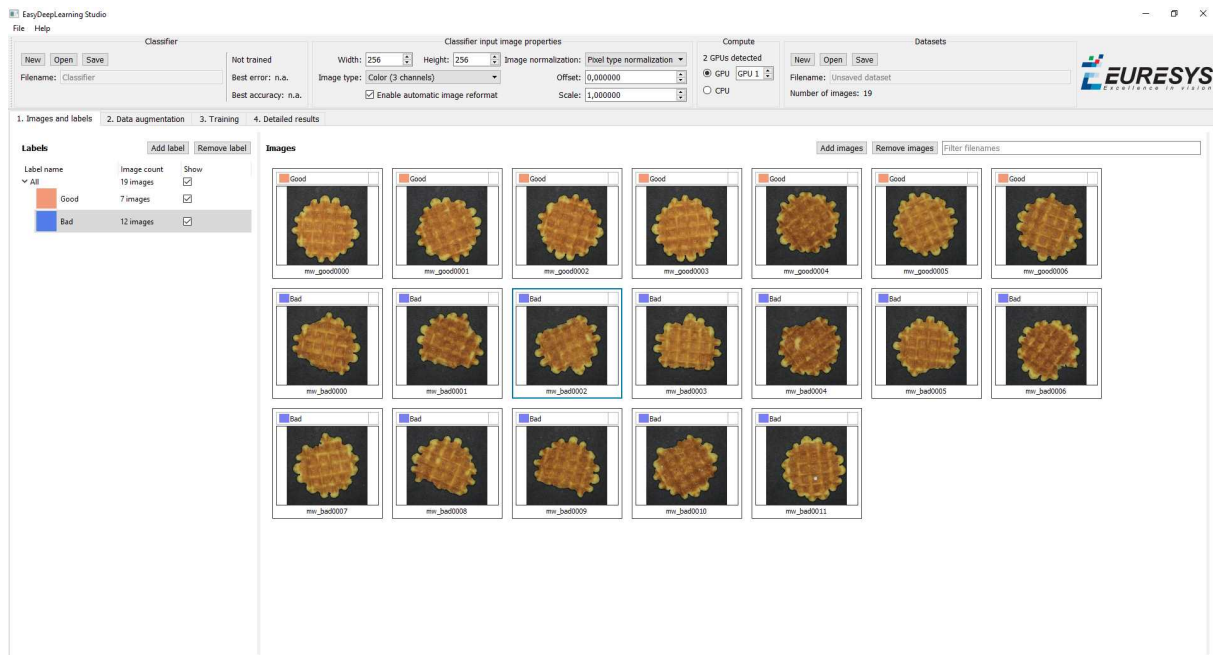
EasyDeepLearning

EasyDeepLearning is now fully available and is not in BETA version anymore.

- Adding of data augmentation mechanisms:
 - Skew
 - Brightness
 - Contrast
 - Colors



- You can make an equalization of the images using a histogram.
- You can use a dataset larger than the memory capacity.



Easy3D

- All the elements of the Easy3D library are in their own namespace: `Euresys::Open_eVision2_6::Easy3D` (see also "[Breaking Changes](#)" on page 12).
- Depth maps provide settings to control the reference axis system used to interpret the internal buffer data.
- ZMaps provide settings to control the X/Y/Z resolution used to interpret the internal buffer data.
- The explicit geometric calibration uses the camera height instead of the camera-laser plane distance.
- The object based calibration method supports anisotropic scale of the reference object.

EasyOCR2

- Adding of a new VB.NET sample program: `Ocr2ReadProportional`.

3.4. Solved Issues

Starting with the release 2.6, Open eVision does not exhibit the following known issues anymore:

Easy3D

- `Easy3D::E3DViewer` does not throw an exception when closed in stand-alone mode.

EasyQRCode

- The timeout displayed in the Open eVision Studio script is correct and consistent with the API unit.

EasyMatrixCode2

- The `Mxc2Read` sample program provides correct solution files for Visual Studio 2008 and Visual Studio 2010.

EasyDeepLearning

- The training or making of a prediction on a EROI works correctly.
- In EasyDeepLearning Studio, the gamma correction used to display data augmentation results is correct.

EasyOCR

- Processing the whole `EBW8Image` (not a ROI) works correctly.

EasyOCV

- The code generated in Open eVision Studio is correct.
- Performing an inspection after the statistical adjustments does not generate an “Unable to access image memory” exception.
- In the dialog boxes in Open eVision Studio:
 - In the character quality indicators panel, the input of `Margin Width` is correct.
 - The `Minimum` and `Maximum` labels in the “Text position parameters” panel are renamed `Average` and `Deviation`.

EasyGauge

- The method `SetMinNumFitSamples(a, b, c, d)` correctly takes into account all 4 parameters.

Note: This correction affects all gauge types: `ERectangleGauge`, `EWedgeGauge`, `ECircleGauge` and `ELineGauge`.

Licensing

- The support for Dongles is updated to avoid malfunction issues introduced after Windows 10 update 1803 (“April Update”).

3.5. Breaking Changes

Starting with this release 2.6, Open eVision implements the following breaking changes:

Easy3D

- All classes, structures and enums are now included in the `Euresys::Open_eVision_2_6::Easy3D` namespace. Most of them have been renamed from “E3D...” to “E...”.
- The `Easy3D` class, which contained static functions operating on 3D objects, is splitted into two new classes:
 - `Euresys::Open_eVision_2_6::Easy3D::EFilters`
 - `Euresys::Open_eVision_2_6::Easy3D::EStatistics`
- The `E3DObject` class is renamed `EMesh`.

EasyDeepLearning

- The `Epoch` concept has been renamed into `Iteration`.
This change has an impact on the `EClassifier` class:
 - `WaitForEpochCompletion => WaitForIterationCompletion`
 - `GetTrainingFinishedEpochs => GetCurrentTrainingFinishedIterations`
 - `GetTrainingNumEpochs => GetCurrentTrainingNumIterations`
 - `GetNumTrainedEpochs => GetNumTrainedIterations`
 - `GetBestEpoch => GetBestIteration`
- `EClassifier::GetTrainingProgression()` becomes `EClassifier::GetCurrentTrainingProgression()`
- `ENormalizationMethod` is removed and replaced by the histogram equalization method.

Open eVision supported IDE and interfaces

- ActiveX and OCX interfaces are discontinued.
- Borland / Embarcadero support is discontinued

Documentation

The Open eVision installer does not include the documentation anymore.

The documentation is now available as independent packages in the following languages:

- English only
- English and Chinese
- English and Japanese
- English and Korean

The Euresys website (<https://documentation.euresys.com/>) provides the online documentation in all languages.

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

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

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)
    }
}
```

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.