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 2.9.0 (doc build 1120).
© 2019 EURESYS s.a.
Open eVision  Release Notes

Contents

1. Release Benefits ................................................................. 4
2. Release Specifications ............................................................ 5
3. Release Details ................................................................. 7
   3.1. New Products .......................................................... 7
   3.2. New and Improved Features ........................................... 8
   3.3. Breaking Changes ...................................................... 9
   3.4. Solved Issues ......................................................... 10
4. Known Issues ................................................................. 12
1. Release Benefits

Summary

*Easy3DObject*

- The new *Easy3DObject* library extracts 3D objects and their features from a ZMap.
- Use *Easy3DObject* to inspect PCBs, connectors or any 3D model with local and consistent objects.
- Use various metric criteria to define the extraction domain (length, width, height, slope...).
- Display the resulting objects in 2D and 3D.
- Use the full-featured demo application (*3D Object Extractor*) and the 2 sample programs to start working.

*New 3D libraries structure*

In order to better fit our customers' requirements, we have changed the structure of Open eVision 3D libraries. The laser line extraction and calibration functions of *Easy3D* have been moved into a new library called *Easy3DLaserLine* and *Easy3D* now only includes the 3D core functions and classes of Open eVision.

You can purchase *Easy3D*, *Easy3DLaserLine* and *Easy3DObject* separately. However, please note that *Easy3D* is required for all 3D developments. For that reason, the *Easy3DLaserLine* and the *Easy3DObject* licenses include *Easy3D*.

*EGrabberBridge*

- Use the new *EGrabberImage* and *EGrabberDepthMap* classes to use images coming from an *EGrabber* instance (from a *Coaxlink* frame grabber) directly into the Open eVision tools.

*EasyOCR2*

- *EasyOCR2* now offers a new segmentation method that uses a global threshold instead of a local one.
- It is faster than the default method for images with a uniform background.
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:

<table>
<thead>
<tr>
<th>OS version</th>
<th>Additional information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Windows 10®</td>
<td>—</td>
</tr>
<tr>
<td>Windows 10®</td>
<td>—</td>
</tr>
<tr>
<td>Windows 8®</td>
<td>—</td>
</tr>
<tr>
<td>Windows 8®</td>
<td>—</td>
</tr>
<tr>
<td>Windows 7®</td>
<td>The recommended version is 6.1.7601</td>
</tr>
<tr>
<td>Windows 7®</td>
<td>(Windows 7 Service Pack 1)</td>
</tr>
</tbody>
</table>

- 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:

<table>
<thead>
<tr>
<th>IDE</th>
<th>Programming language</th>
<th>IDE</th>
<th>Programming language</th>
</tr>
</thead>
<tbody>
<tr>
<td>Microsoft Visual Studio 2008® SP1</td>
<td>C++, .NET Assembly</td>
<td>Microsoft Visual Studio 2010®</td>
<td>C++, .NET Assembly</td>
</tr>
</tbody>
</table>

**NOTE**

(*) Visual C++ MFC MBSC 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 Products

Starting with this release 2.9, Open eVision offers the following new product:

**Easy3DObject**

*Easy3DObject* is a new library that extracts 3D objects and their features from a ZMap.

- It enables the inspection of PCBs, connectors or any 3D model featuring local and consistent objects.
- You can use various metric criteria to define the extraction domain (length, width, height, slope...).
- You can display the resulting E3DObject instances in 2D and 3D.
- It requires the new Easy3DObject license.
- A full featured demo application and 2 sample programs are available.
3.2. New and Improved Features

**New Features**

*EGrabberBridge*

- Open eVision now includes the new `EGrabberImage` and `EGrabberDepthMap` classes that allow to seamlessly use images coming from an EGrabber instance (and a Coaxlink card) with the Open eVision tools. They are members of the new `EGrabberBridge` namespace.
- Open eVision includes the "EGrabberBridge" sample programs to show how to use these classes.

**Improvements**

*Easy3D*

- The `E3DViewer` is now able to display the features of an `E3DObject`.
- You can configure the style rendering of these objects.

*EasyOCR2*

- *EasyOCR2* now supports a new segmentation method that is computationally faster than the previous (default) segmentation. It uses a global threshold value rather than a local one and it is thus less suitable for images with a non-uniform background.
- Use `EOCR2.SetSegmentationMethod()` to activate it.

*EasyMatrixCode2*

- The reading performance is slightly improved.
- The computational time is slightly reduced.

*Easy*

- You can now load and save `EZMap32f` instances (32-bit floating point pixels) in TIFF format.

*Open eVision Studio applications*

- The icons of the Open eVision Studio applications have been updated.
3.3. Breaking Changes

**Easy3D**

- The following classes related to Laser Line Extraction and Calibration are now part of a new library called *Easy3DLaserLine*. They require the new *Easy3DLaserLine* license.

  These classes are:
  - `EExplicitGeometricCalibrationModel`
  - `EObjectBasedCalibrationGenerator`
  - `EObjectBasedCalibrationModel`
  - `ELaserLineExtractor`

- The new structure of Open eVision 3D libraries is as follows:
  - *Easy3D* now only includes the 3D core functions and classes of Open eVision.
  - *Easy3DLaserLine* includes the laser line extraction and calibration functions.
  - *Easy3DObject* includes the functions to extract 3D objects and their features from a ZMap.

**NOTE**

All libraries are sold separately and have their own license, but the *Easy3DLaserLine* and the *Easy3DObject* licenses automatically include an *Easy3D* license.

- All the classes included in the *Easy3D* namespace that previously didn’t need a license now require the *Easy3D* license.
3.4. Solved Issues

The following issues have been fixed in Open eVision 2.9:

**EasyDeepLearning**

- The multi-GPU mode does not lower the training performance anymore (lower accuracy and higher error rate after a given number of iterations).
- Training could crash when using more than one processing thread. This has been fixed.
- **EasyDeepLearning** now runs properly with image files that have a relative path.
- There are no performance difference anymore between a freshly trained classifier and its serialization.

**Easy**

- The pivot point used in the `ScaleRotate()` dialog box of Open eVision Studio is now consistent with the API.
- The different implementations of an `ERegion` are now in a valid state after a drag operation.

**EasyOCV**

- Open eVision Studio displays correct values for the `SampleForegroundArea` in the characters quality dialog box.
- When `MarginWidth` is set, the computation of the quality indicators is correct.
● MarginWidth, when different from 0, expands the calculation area by the given number of pixels for the selected character.

After changing MarginWidth (for one or several characters):

□ Perform a new training.
□ The values of the Template quality indicators are updated.
□ During the inspection, the sample quality indicators are calculated with the same MarginWidth and are compared to the Template value.
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:

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

becomes:

```csharp
-codedImage.SetThreshold((uint)EThresholdMode.MinResidue));
```

Reserved keywords

The following keywords are reserved by Open eVision:

- `EUnit_um`, `EUnit_mm`, `EUnit_cm`, `EUnit_dm`
- `EUnit_m`, `EUnit_dm`, `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#**

```csharp
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#**

```csharp
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++**

```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#

```csharp
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, deselected 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.