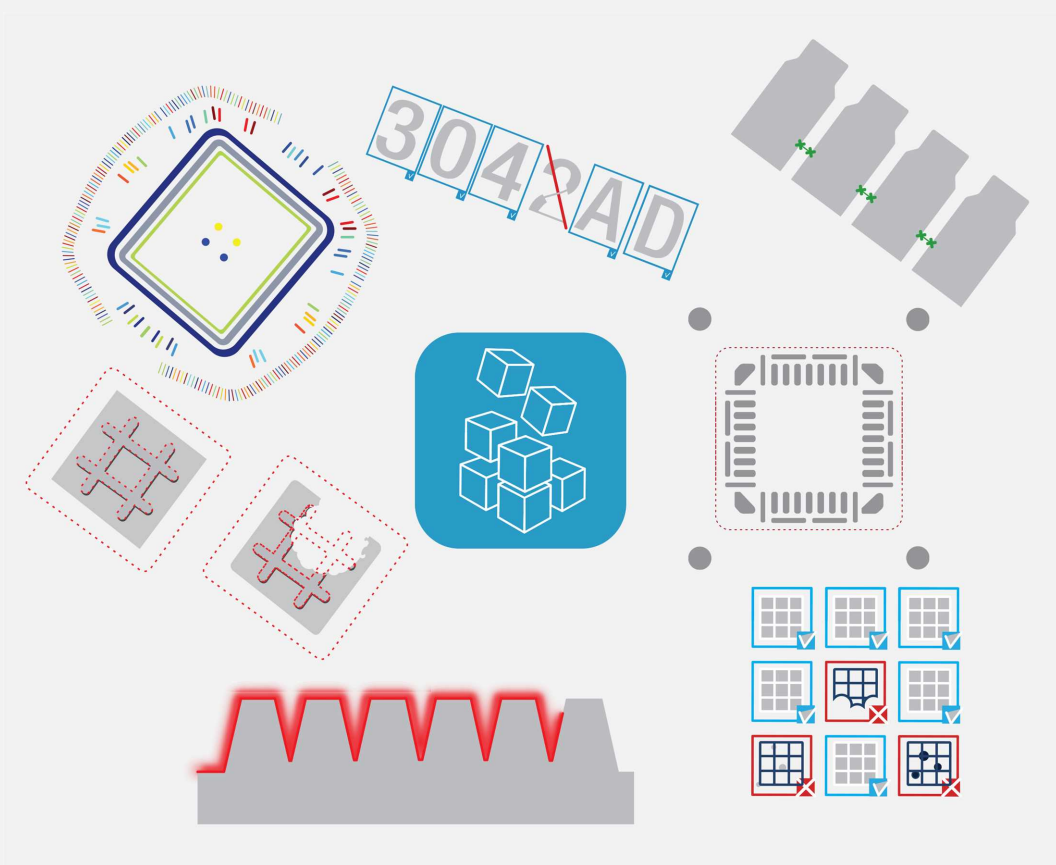


Open eVision

Release 24.02.0



This documentation is provided with **Open eVision 24.02.0** (doc build **1196**).
www.euresys.com

This documentation is subject to the General Terms and Conditions stated on the website of **EURESYS S.A.** and available on the webpage <https://www.euresys.com/en/Menu-Legal/Terms-conditions>. The article 10 (Limitations of Liability and Disclaimers) and article 12 (Intellectual Property Rights) are more specifically applicable.

Contents

1. Release Benefits	4
2. Release Specifications	6
3. End of Life and Support	8
4. Release Details	10
4.1. Features Updates	10
4.2. Breaking Changes	14
4.3. Changes	16
4.4. Solved Issues	20
5. Known Issues	22

1. Release Benefits

What's new in **Open eVision 24.02**

EasyFind

- We have significantly optimized the code of **EasyFind** and the location process is now 2 to 2.5 times faster than in the previous versions.

It is important to note that this is backward compatible.

- The algorithms of **EasyFind** have not been changed.
- When using **Open eVision 24.02**, **EasyFind** returns exactly the same results as with the previous version (but faster).

Global

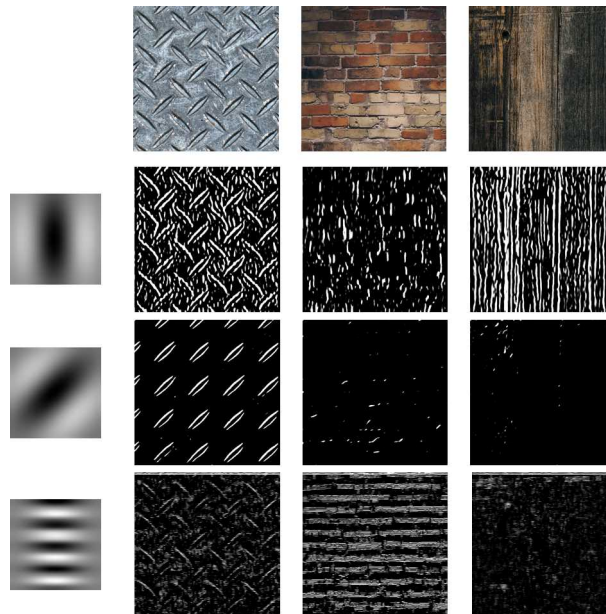
- The version numbers have been removed from the **Open eVision** headers file names, the **Open eVision** namespace and the **Open eVision** DLL file names. This will greatly simplify moving to a new version of **Open eVision**.
 - In a C++ program, add:
 - #include "Open_eVision.h" to include **Open eVision**.
 - using namespace Euresys.Open_eVision; to import the entire **Open eVision** namespace.
 - In a C# program, add:
 - The Open_eVision_NetApi.dll reference to include **Open eVision**.
 - using Euresys.Open_eVision; to import the entire **Open eVision** namespace
 - Note that different versions of **Open eVision** are still installed in separate directories. And so it is still possible to install multiple versions of **Open eVision** at the same time.
 - In C++, to choose a specific version, set the correct include search paths.
 - In .NET, include the correct reference.
 - Migrating a program from a version of **Open eVision** to another now only requires a change in the project properties and not in the source code. Recompiling is, however, still mandatory.

Easy

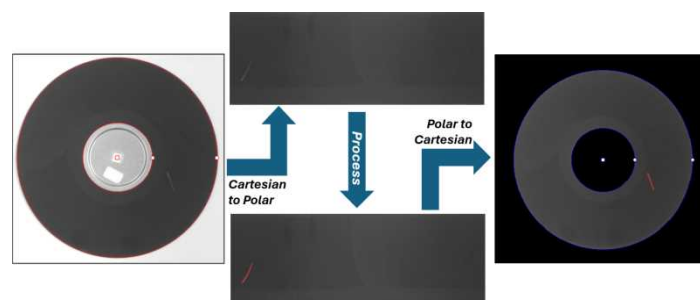
- Use the new class `EGenericDrawAdapter` to draw in C24A (color with transparency) images.
 - It is a draw adapter designed to work on all platforms with a similar rendering on all of them.
 - It uses CPU processing.
 - It is mainly designed to make off-screen rendering.

EasyImage

- A new function `EasyImage.ConvGabor` is now available to apply Gabor filters to images.
 - Gabor filters are helpful in texture analysis and edge detection applications by enhancing some specific features in the image.



- Use the new inverse circle warp function `SetInvCircleWarp` to setup a polar to cartesian image transformation.
 - This function generates look-up-tables that can be efficiently used by the **EasyImage** "Warp" function.



Deprecated items

Soft-based / host licenses

- As of 24.02, **Open eVision** does not support "host-based" (**FlexNet**) licenses anymore.
 - The change is related to **Open eVision** licenses with a Product Code ranging from 4250 to 4289.
 - The host-based (**FlexNet**) licensing system of **Open eVision** has been deprecated in 2022 and is replaced by the **Neo Licensing System**.

2. Release Specifications

OS and processor architectures

Windows OS

- **Open eVision** is a 64-bit library that requires a processor compatible with the SSE4 instruction set.
- **Open eVision** runs on the following Windows operating systems:

OS version	Additional information	
Windows 11®	64-bit	—
Windows 10®	64-bit	—
Windows 8®	64-bit	—
Windows 7®	64-bit	The recommended version is 6.1.7601 (Windows 7 Service Pack 1)

Linux OS

- **Open eVision** is compatible with x86_64 and aarch64 (ARMv8-A) CPUs.
- **Open eVision** and the **Neo License Manager** are designed to be distribution-independent on x86_64 platforms.
- The minimum requirements are:
 - gcc version 7.5
 - glibc version 2.23
 - libglvnd
- This release has been validated with the following distributions and their default gcc compilers and cmake programs:
 - **Ubuntu LTS** 16.04 to 22.04

NOTE: Ubuntu 16.04:

- On arm64: not supported.
- On x86-64 (Intel/AMD processors): you must add additional repositories to install GCC 7.5 and libglvnd (see readme.txt in the installation archives).
- **Fedora** 33 to 38
- **Debian** 10 and 11
- **CentOS** 8

- This release has been validated on the following embedded systems:
 - **Raspberry Pi** 4 and Raspberry Pi Zero 2 W
 - **NVidia** Jetson Nano, Xavier and Orin
 - Smart camera **Mindvision** MV-ITE130GM

NOTE: - For **CMake** samples: the client programs must be linked against the pthread library.
 - For **Qt** samples: it is not necessary to make this dependency explicit because a program using Qt automatically depends on the pthread library.

[Qt compatibility](#)

Some **Open eVision** samples use the **Qt** framework to create a user interface and display images with a graphical overlay.

- We recommend the **Qt** versions 5.12 to 5.15.

[Remote connections and virtual machines](#)

- 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
 - Virtual machines are supported. **Microsoft Hyper-V**, **Oracle VirtualBox** and **libvirt** hypervisors have been successfully tested.
 - Only the **Neo Licensing System** is compatible with virtualization.

[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 2017®	C++	.NET Assembly
Microsoft Visual Studio 2019®	C++	.NET Assembly
Microsoft Visual Studio 2022®	C++	.NET Assembly
QtCreator 4.15 with Qt 5.12 (*)	C++	

NOTE: (*) A C++ compiler like **MSVC** must be installed

[Required system resources](#)

- Processor:
 - Intel compatible processor supporting the SSE4 instruction set
 - ARMv8-A compatible processor
- Memory:
 - Execution: minimum 2 GB RAM to run an **Open eVision** application
 - Compilation: minimum 8 GB RAM to compile an **Open eVision** application
- Hard disk space:
 - **Open eVision** libraries: 100 MB - 2 GB (depending on selected options)

3. End of Life and Support

End of life announcements

Open eVision soft-based / host licensing system

- The **Open eVision** soft-based / host licensing system will be progressively phased out.
 - It was introduced in 2007 with the first version of **Open eVision** and it is based on an old, and now obsolete technology.
 - It is superseded by the **Neo Licensing System**.

Milestones of the phase out period

- Since 1 January 2023:
 - There are no more sales of **Open eVision** soft-based / host licenses.
 - The related product codes are 4250 to 4289.
 - These products are removed from the price lists.
- Starting 1 January 2024:
 - The support for the soft-based / host licenses will be removed from the **Open eVision** libraries.
 - The **Open eVision** releases 24.02 and later will not be able to detect and use any soft-based / host licenses activated on the platform.
- Starting 1 January 2029:
 - The soft-based license operation server will be shut down.
 - Activating or recovering a soft-based / host license will not be possible after 2028.

Notes

- All applications using a soft-based / host license will continue to work “forever”, as long as the licenses have been activated and don’t require recovery.
- The **Open eVision Neo Licensing System** replaces the soft-based / host licensing system.
- The **Neo Licensing System** supports software and **Neo Software Containers** for licenses.
- The usage and features of the **Neo Software Containers** are the same as the old soft-based / host licenses.
- **eVision** licenses are not affected by these changes.

Support of older environments

32-bit libraries

- The **32-bit Open eVision** libraries (only available for **Windows**) are removed from **Open eVision** distributions in July 2023.
 - You should migrate to 64-bit if you need newer versions of **Open eVision**.
 - ▶ The last version with 32-bit support is **Open eVision 23.04**.

Soft-based / host licenses

- The support for the soft-based / host licenses is removed from the **Open eVision** libraries.
 - ▶ The last version to support soft-based / host licenses is **Open eVision 23.12**.

OS and processor architectures

OS		Last version with support
Windows 11®	32-bit	Open eVision 23.04
Windows 10®	32-bit	
Windows 8®	32-bit	
Windows 7®	32-bit	
Windows Vista	—	Open eVision 2.3.3.10777
Windows XP	—	Open eVision 2.2.2.10255
Windows 2000	—	Open eVision 1.0.1.5222

Supported IDE

IDE	Last version with support
Microsoft Visual Studio 2008	Open eVision 22.12
Microsoft Visual Studio 2005	Open eVision 2.7
Microsoft Visual Studio 2003	
Microsoft Visual Studio 6.0	Open eVision 2.5.1.1107

Programming languages

Programming language	Last version with support
Borland C++ Builder 6.0	Open eVision 2.5.1.1107
CodeGear Delphi 2009	
CodeGear C++ Builder 2009	
ActiveX API	
Embarcadero RAD Studio XE4 and XE5	Open eVision 2.4.1.11114
Borland Delphi 6.0 and 2006	Open eVision 1.0.1.5222
Borland C++ Builder 2006	

Legacy Headers

- The **Legacy Headers** are removed from **Open eVision** distributions since the end of 2022.
 - The **Legacy Headers** help the customers to migrate an existing application using **eVision** (last major release in 2006) to **Open eVision**.
 - ▶ The last version with support is **Open eVision 22.12**.

4. Release Details

4.1. Features Updates

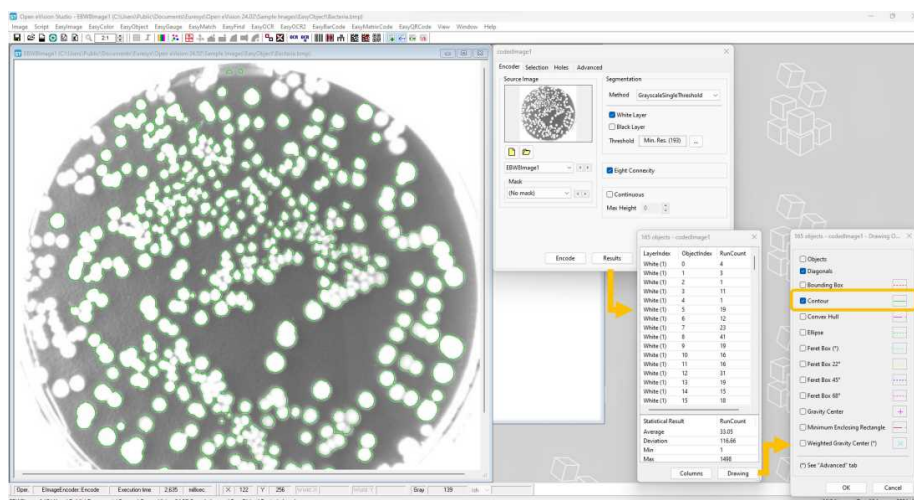
New features

Easy

- Use the new method `LogInMemento` to send messages directly to **eGrabber Memento**.
- Use the new class `EGenericDrawAdapter` to draw in C24A (color with transparency) images.
 - It is a draw adapter designed to work on all platforms with a similar rendering on all of them.
 - It uses the CPU processing.
 - It is mainly designed to make off-screen rendering.
- The .NET version of the **EGrabberBridge** classes have been adapted to be compatible with the new **eGrabber** assembly provided since **eGrabber 24.01**.
 - **EGrabberBridge**, however, remains compatible with the assemblies of the previous **eGrabber** versions.

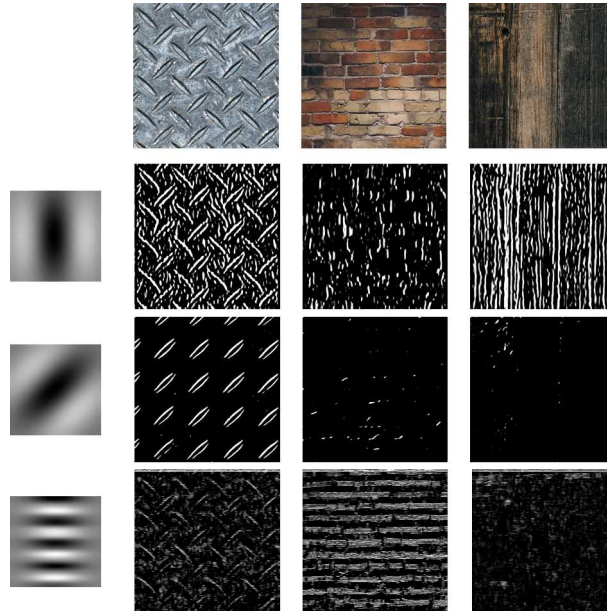
EasyObject

- Use the new feature `EDrawableFeature_Contour` and the new method `ECodedElement.GetContourPath` to draw or get the contour of the objects extracted with **EasyObject**.
 - The contour path is computed with `EasyImage.Contour` using the mode `EContourMode_AnticlockwiseAlwaysClosed` mode.
 - The drawn is computed using the method `DrawClosedContour` of `EPathVector`.

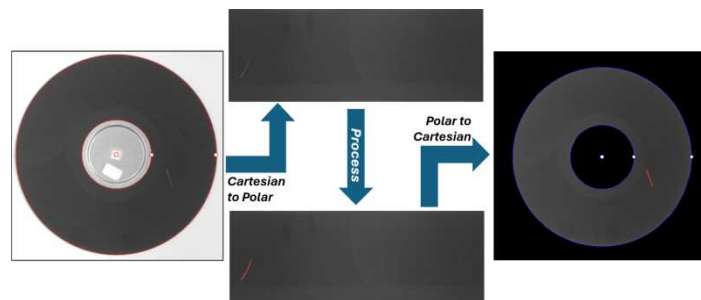


EasyImage

- Use the new function `EasyImage.ConvGabor` to apply the Gabor filters to images.
 - These filters enhance the features in the image for an improved texture analysis and edge detection.

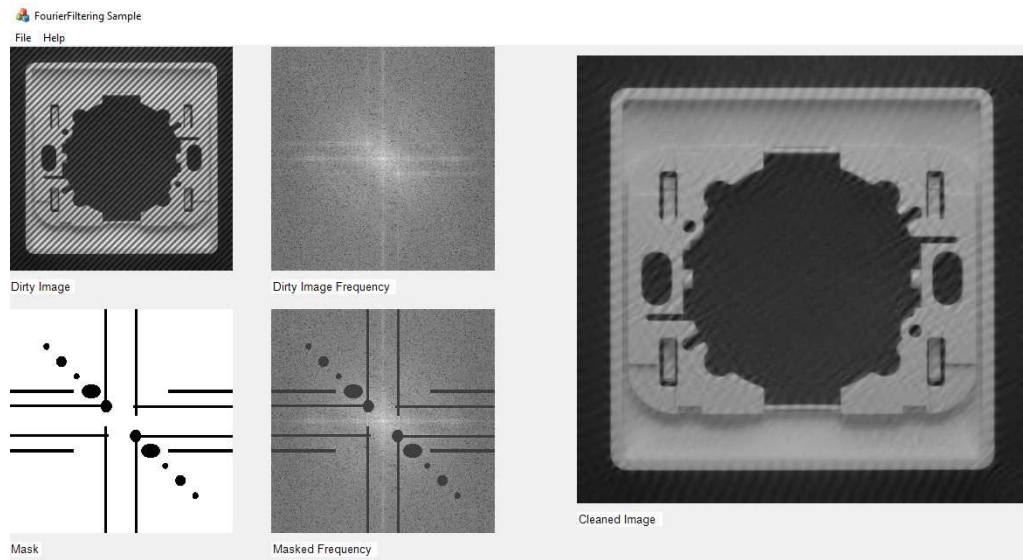


- Use the new inverse circle warp function `SetInvCircleWarp` to setup a polar to cartesian image transformation.
 - This function generates look-up-tables that can be efficiently used by the **EasyImage** "Warp" function.



Code Samples

- Use `FourierFiltering`, a new C++ MsVc sample, to remove periodic patterns in a specified image by masking specific parts of the frequency domain.



Improvements

EasyBarcode2

- The decoding is now more robust in some specific cases.

Deep Learning Studio

- The shortcuts used to change a label (F1 to F10 keys) have been improved.
 - For **EasyClassify**, they apply to a selection of images.
 - For **EasyLocate**:
 - In "object creation" mode, use them to change the label of the next object.
 - In "edit object" mode, use them to change the label of the object under the mouse.
 - For **EasySegment Supervised**, they change the label for the next drawing operation.
- For training and inference tasks, **Deep Learning Studio** displays "Initializing..." while the engine and/or the dataset is initialized.
- For inference tasks, the engine initialization is performed when the previous task used a different tool or engine, or when the batch size is set to Optimize in the **Tools** tab.
 - In the previous releases, the engine initialization was always performed.
- For **EasyLocate** and **EasySegment Supervised**, in the **Validation** tab, the new button **Edit prediction as ground truth** behaves as the **Use prediction as ground truth** but also opens the ground truth editor to refine and adapt the imported prediction.
- For **EasyClassify**, in the **Validation and results** tab and in the **Inference tests** tab, the probability for the predicted label is displayed in the image list (both in the thumbnail and in the table view).

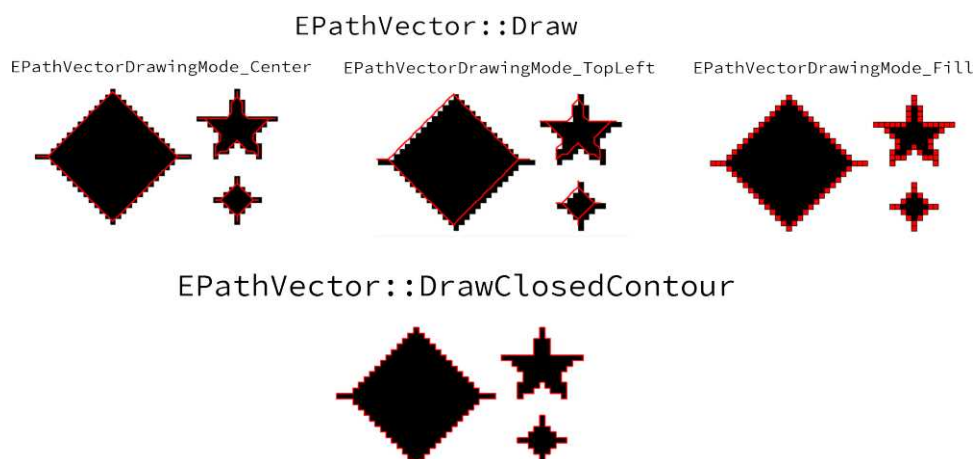
- The confusion matrix is improved and now displays the statistics about its rows and columns.

Confusion matrix

		True labels		
		Good	Bad	%
Predicted labels	Good	16	0	100.00 %
	Bad	0	8	100.00 %
%		100.00 %	100.00 %	100.00 %

Easy

- The drawing operations on the path vectors are improved (EPathVector, EBW8PathVector, EBW16PathVector and EC24PathVector).
- The method Draw has the additional optional argument EPathVectorDrawOption whose value can be:
 - EPathVectorDrawOption_TopLeft: links the top left corner of the pixels in the path.
 - EPathVectorDrawOption_Center: links the center of the pixels in the path.
 - EPathVectorDrawOption_Fill: fills all the pixels in the path.
 - (*Breaking change*) The default value was EPathVectorDrawOption_TopLeft but is now EPathVectorDrawOption_Center.
- Use the new method DrawClosedContour to draw closed contours.
 - This method requires the contour mode (EContourMode) used to compute the path vector.
 - The drawing is a vector rendering of the outer edges of all the pixels in the path.



EasyFind

- **EasyFind** is now about 2 to 2.5 times faster than before.

4.2. Breaking Changes

Starting with this release **24.02**, **Open eVision** implements the following changes:

Easy

- The method Draw has the additional optional argument EPathVectorDrawOption whose value can be:
 - EPathVectorDrawOption_TopLeft: links the top left corner of the pixels in the path.
 - EPathVectorDrawOption_Center: links the center of the pixels in the path.
 - EPathVectorDrawOption_Fill: fills all the pixels in the path.
 - (*Breaking change*) The default value was EPathVectorDrawOption_TopLeft but is now EPathVectorDrawOption_Center.

Global

- (*24.02.0 - Breaking change*) Starting from 24.02, the version numbers have been removed from the **Open eVision** headers file names, the **Open eVision** namespace and the **Open eVision** DLL file names. This will allow you to more easily update your version of **Open eVision** in the future.
 - To include **Open eVision** in a C++ program, use #include "Open_eVision.h" and using namespace Euresys.Open_eVision; to import the entire **Open eVision** namespace.
 - To include **Open eVision** in a C# program, add the Open_eVision.NetApi.dll reference and use using Euresys.Open_eVision; to import the entire **Open eVision** namespace.
 - Different versions of **Open eVision** are still installed in different directories. The choice of a specific version is now a matter of setting the correct include search paths during compilation in C++. And in .NET to include the correct reference.
 - Migrating a program from one version of **Open eVision** to another one now only requires a change in the project properties and not in the source code anymore. Recompiling is, however, still mandatory.
- (*24.02.0 - Breaking change*) Starting from 24.02, **Open eVision** does not support host-based licenses (**FlexNet**) anymore.
 - The related product codes range from 4250 to 4289.

>> For software licenses (host locked), use the **Neo Licensing System**.
- (*24.02.0 - Breaking change*) For historical reasons, some **Open eVision** represented booleans using either C++ bool or Windows BOOL. C++ bool are now used nearly everywhere.
- (*24.02.0 - Breaking change*) In C++, the attributes `[[deprecated]]` are added in all APIs considered deprecated.
 - This may cause potential compilation issues in projects using these deprecated APIs and using non-default compiler flags to indicate the usage of deprecated APIs as compilation errors instead of warnings.

>> In this case, we recommend to replace the usage of the deprecated APIs according to the indications provided in **Open eVision** documentation or to remove those non-default flags.

Legacy License Manager

- (24.02.0 - *Breaking change*) The Legacy License Manager does not handle host-based licenses anymore.
>> For software licenses (host locked), use the **Neo Licensing System**.

Easy3D

- (24.02.0 - *Breaking change*) Some functions (mostly ObjectBasedCalibration, PlaneFinder and 3DMatch) had slight differences between Windows and Linux due to implementation variations in random number generation.
 - This has been uniformized.
 - As a consequence, the results of these functions have slightly changed on Windows.
- (24.02.0 - *Breaking change*) EZMap.Create and ECalibrationModel.Create are modified to returned a unique_ptr.

EasyMatch

- (24.02.0 - *Breaking change*) The method EMatcher.CopyTo that returned a pointer has been removed.
>> Instead, use the other overload.

Easy

- (24.02.0 - *Breaking change*) In the C++ API, the method parameters pointer types are changed to void* for: ESerializer.CreateMemoryWriter, ESerializer.CreateMemoryReader, ESha1Hasher.Update, ESha1Hasher.GetHash and ESha1Hasher.Get32BitHash.
 - The fonctionnalities remain strictly identical but this change may cause minor compilation differences.

EasyObject

- (24.02.0 - *Breaking change*) The **EasyObject** API is reworked to improve its "constness". This means that all methods that does not change the equality between two objects are now const in the classes EImageEncoder, ECodedImage2, ECodedElement, EObject, EHole and EObjectSelection. This can cause minor breaking changes in your code.
 - A lot of these methods are drawing methods or feature getters.
 - There were previously not const because the features are lazily evaluated (their computation is delayed until we actually need their value). They are still lazily evaluated but the cache is changed to make storing and retrieving lazily evaluated values in the cache a const operation.

4.3. Changes

Starting with this release **24.02**, **Open eVision** implements the following changes:

Global

- (24.02.0 - *Breaking change*) Starting from 24.02, the version numbers have been removed from the **Open eVision** headers file names, the **Open eVision** namespace and the **Open eVision** DLL file names. This will allow you to more easily update your version of **Open eVision** in the future.
 - To include **Open eVision** in a C++ program, use `#include "Open_eVision.h"` and using namespace `Euresys.Open_eVision`; to import the entire **Open eVision** namespace.
 - To include **Open eVision** in a C# program, add the `Open_eVision_NetApi.dll` reference and use `using Euresys.Open_eVision`; to import the entire **Open eVision** namespace.
 - Different versions of **Open eVision** are still installed in different directories. The choice of a specific version is now a matter of setting the correct include search paths during compilation in C++. And in .NET to include the correct reference.
 - Migrating a program from one version of **Open eVision** to another one now only requires a change in the project properties and not in the source code anymore. Recompiling is, however, still mandatory.
- (24.02.0 - *Breaking change*) Starting from 24.02, **Open eVision** does not support host-based licenses (**FlexNet**) anymore.
 - The related product codes range from 4250 to 4289.

>> For software licenses (host locked), use the **Neo Licensing System**.
- (24.02.0) On Windows, you may observe slight speedups (2-3%) due to an update of the compiler used internally by **Open eVision**.
- (24.02.0 - *Breaking change*) For historical reasons, some **Open eVision** represented booleans using either C++ `bool` or Windows `BOOL`. C++ `bool` are now used nearly everywhere.
- (24.02.0 - *Breaking change*) In C++, the attributes `[[deprecated]]` are added in all APIs considered deprecated.
 - This may cause potential compilation issues in projects using these deprecated APIs and using non-default compiler flags to indicate the usage of deprecated APIs as compilation errors instead of warnings.

>> In this case, we recommend to replace the usage of the deprecated APIs according to the indications provided in **Open eVision** documentation or to remove those non-default flags.
- (24.02.0) In C#, the attribute `[Obsolete]` is added to all APIs considered deprecated.
 - This is not considered as a breaking change as the only effect it may have is to alert with warnings regarding the usage of deprecated APIs.

Legacy License Manager

- (24.02.0 - *Breaking change*) The Legacy License Manager does not handle host-based licenses anymore.
 - >> For software licenses (host locked), use the **Neo Licensing System**.

ERegion

- (24.02.0) ERegion.Prepare is now automatically called internally when performing an operation that requires a prepared region.
 - It is no longer necessary to use Prepare in your code.

Easy3D

- (24.02.0 - *Breaking change*) Some functions (mostly ObjectBasedCalibration, PlaneFinder and 3DMatch) had slight differences between Windows and Linux due to implementation variations in random number generation.
 - This has been uniformized.
 - As a consequence, the results of these functions have slightly changed on Windows.
- (24.02.0 - *Breaking change*) EZMap.Create and ECalibrationModel.Create are modified to returned a unique_ptr.
- (24.02.0) **OpenGL** is now an optional dependency for **Open eVision**.
 - If **OpenGL** library can not be dynamically loaded, the creation of an E3DViewer fails (but the other **Open eVision** features are available).
 - **OpenGL ES** is not supported by E3DViewer.
 - The minimum required version is **OpenGL 2.1**.
 - The Eye Dome Lighting shading option is only supported with **OpenGL 4.4** or above.

Deep Learning Studio

- (24.02.0) The **EasyClassify** setting `Automatic image reformat` is removed. It is now always enabled.
- (24.02.0) Some **EasyClassify** settings are now hidden by default in the `Advanced parameters` category.
- (24.02.0) The behavior of the image list has changed:
 - Before, the image list was always up to date with respect to the filters set in place.
 - Now, the image list contains the images that matched the filter at the moment it was put in place.
 - A message invites you to refresh the list. This allows you to better keep track of your recent annotation work by keeping the images in the list even if they no longer match the filter.
- (24.02.0) The behavior of the image viewer has changed:
 - Before, the image viewer always displayed the image that was highlighted in the image list.
 - Now, the image viewer displays the last image you individually selected.
 - A message is displayed when the image viewer displays an image that is not in the list.
- (24.02.0) `With/No ground truth object labeling` texts are shortened to `Annotated/Not annotated`. These texts are used in the image viewer and confusion matrix to indicate whether an image is annotated or not.
- (24.02.0) The Windows **Deep Learning Studio** installer no longer bundles the **Deep Learning** additional resources.
>> You need to download and install the **Deep Learning** additional resources separately.

EasyMatch

- (24.02.0 - *Breaking change*) The method `EMatcher.CopyTo` that returned a pointer has been removed.
>> Instead, use the other overload.

Easy

- (24.02.0) The methods `Draw` are now `const` for: `EVector`, `EPeakVector`, `EBW8Vector`, `EBW16Vector`, `EC24Vector`, `EBWHistogramVector`, `EPathVector`, `EBW8PathVector`, `EBW16PathVector` and `EC24PathVector`.
- (24.02.0) The classes `EPen`, `EBrush` and `EFont` now have serialization capabilities.
- (24.02.0 - *Breaking change*) In the C++ API, the method parameters pointer types are changed to `void*` for: `ESerializer.CreateMemoryWriter`, `ESerializer.CreateMemoryReader`, `ESha1Hasher.Update`, `ESha1Hasher.GetHash` and `ESha1Hasher.Get32BitHash`.
 - The functionalities remain strictly identical but this change may cause minor compilation differences.

EasyMatrixCode

- (24.02.0) The property `OverallSymbolGrade` of the structures `EMatrixCodeIso15415GradingParameters` and `EMatrixCodeIso19258GradingParameters` are renamed `ScanGrade` to be more in line with the relevant standards.

EasyQRCode

- (24.02.0) The property `OverallSymbolGrade` of the structures `EQRCodeIso15415GradingParameters` and `EQRCodeIso19258GradingParameters` are renamed `ScanGrade` to be more in line with the relevant standards.

EasyObject

- (24.02.0 - *Breaking change*) The **EasyObject** API is reworked to improve its "constness". This means that all methods that does not change the equality between two objects are now `const` in the classes `EImageEncoder`, `ECodedImage2`, `ECodedElement`, `EObject`, `EHole` and `EObjectSelection`. This can cause minor breaking changes in your code.
 - A lot of these methods are drawing methods or feature getters.
 - There were previously not `const` because the features are lazily evaluated (their computation is delayed until we actually need their value). They are still lazily evaluated but the cache is changed to make storing and retrieving lazily evaluated values in the cache a `const` operation.
- (24.02.0) The `Copy` constructor and the assignment operator now exist for `EImageEncoder`, `ECodedImage2`, `EObjectSelection` and `ECodedElement`.
- (24.02.0) Use the new `ECodedImage2.ToRegion` and `EObjectSelection.ToRegion` to convert a coded image or an object selection to an `ERegion`.
- (24.02.0) The serialization is now available for `EImageEncoder` and its segmenters.

- (24.02.0) Internal changes are made to `ECodedImage2` and `EObjectSelection` to improve the memory management.
 - These changes may have a small impact on the performance of some operations (creating object selection and copying objects).
 - The objects `EObjectSelection` or `ECodedElement` are still valid when the corresponding `ECodedImage2` is destroyed.
 - Previously, the address of a `ECodedElement` (pointer) could be used to compare the equality of two coded elements. This is no longer the case and there is now an equality operator.
- (24.02.0) The reference image segmenter (class `Segmenters.EReferenceImageSegmenter`, type `ESegmentationMethod_ReferenceImage`) and the image range segmenter (class `Segmenters.EImageRangeSegmenter`, type `ESegmentationMethod_ImageRange`) now copy the given reference, low or high images.
 - Before, these two segmenters only stored the pointers towards the images, meaning that the images had to continue existing as long as the segmenters were used.
 - The new behavior is slower but impacts only the configuration of an `EImageEncoder` which is usually done once for multiple uses.

Open eVision Studio

- (24.02.0) In the dialogs for **EasyMatrixCode**, **EasyMatrixCode2** and **EasyQRCode**, the `Overall Symbol Grade` of the **ISO/IEC 15415** and **ISO/IEC 29158** standards are renamed `Scan Grade` to be more in line with these standards.

EasyGauge

- (24.02.0) The following methods are renamed:
 - `ERectangleGauge.GetNumSamples(x|X|y|Y)`
=> `ERectangleGauge.GetNumSamples(Left|Right|Lower|Upper)Edge`
 - `ERectangleGauge.GetSample(x|X|y|Y)`
=> `ERectangleGauge.GetSample(Left|Right|Lower|Upper)Edge`
 - `EWedgeGauge.GetNumSamples(a|A|r|R)`
=> `EWedgeGauge.GetNumSamples(Left|Right|Outer|Inner)Edge`
 - `EWedgeGauge.GetSample(a|A|r|R)`
=> `EWedgeGauge.GetSample(Left|Right|Outer|Inner)Edge`
 - The replaced methods are still available but tagged as `[[deprecated]]`.

Linux ARM

- (24.02.0) The **Deep Learning** additional resources package no longer exists for **Linux arm64** platforms.
 - The GPU processing is only supported on **NVIDIA Jetson** boards and uses the **NVIDIA CUDA** and **TensorRT** libraries that come with the system.

Windows installers

- (24.02.0) The Windows **Deep Learning Studio** installer no longer bundles the **Deep Learning** additional resources.
 - >> You need to download and install the **Deep Learning** additional resources separately.

4.4. Solved Issues

The following issues have been fixed in **Open eVision 24.02**:

[EasyMatrixCode](#)

- (24.02.0) The grading of the axis-aligned data matrices with `EMatrixCodeReader` could cause a crash. This has been fixed.

[EasyQRCode](#)

- (24.02.0) The copy constructor operator caused a crash. This has been fixed.

[EasyOCR2](#)

- (24.02.0) When no topology was specified, a call to `EasyOCR2.Detect` (and thus to `EasyOCR2.Read`) could take an infinite amount of time. This has been fixed.

[EasyGauge](#)

- (24.02.0) The wrong display of the handles of `EWedgeShape` and `EWedgeGauge` has been fixed.
- (24.02.0) The `EPolygonGauge` resulting measures could be wrong when the gauge is attached to an `EFrameShape`. This has been fixed.

[Easy3D](#)

- (24.02.0) The overloads `EPhotometricStereoImager.Compute` and `EPhotometricStereoImager.CalibrateFromSphere` that take 4 `EROIBW8` as input threw an exception when the 4 ROIs were actually `EImageBW8`. This has been fixed.

[Deep Learning tools](#)

- (24.02.0) Since 23.04, the multicore processing settings were not respected when training or performing an inference on a GPU. This has been fixed.
 - When using a GPU, the multicore processing is used to load images and apply data augmentation.
- (24.02.0) On Linux, importing a dataset that was exported on Windows didn't work. This has been fixed.

[C++ wrappers](#)

- (24.02.0) An invalid free causing a crash could occur when calling a method taking a non-const `std::string` reference as argument and an exception occurred in that method (for example: `E3DViewer.GetTextLabel`). This has been fixed.

Deep Learning Studio

- (24.02.0) For **EasyClassify**, in the tab **Validation and results**, the fields **GT** and/or **Pred** could be wrongly positioned. This has been fixed.
- (24.02.0) When removing images, the results and the metrics could become corrupted and invalid. This has been fixed.
- (24.02.0) Renaming a project while training a tool could lead to crashes. This has been fixed by preventing the renaming of the project while a training or an inference is under way.
- (24.02.0) When **EasyLocate** predicted a 0 size, an infinite size or a very large bounding box, **Deep Learning Studio** could crash. This has been fixed by never predicting objects with such size. See the related **EasyLocate** solved issue.

EasyImage

- (24.02.0) In rare instances, when using ER0IBW16 or ER0IC24 with morphological operators or convolutions, there was a crash. This has been fixed.
- (24.02.0) Some results with morphological operators or convolutions could be slightly translated when using a region. In rare instances, this could lead to a crash. This has been fixed.
- (24.02.0) A memory leak that appeared when an error occurred while using in-place morphological operators or convolutions with regions has been fixed.

Easy

- (24.02.0) In the C# version of **Open eVision**, the methods `ESerializer.CreateMemoryWriter`, `ESerializer.CreateMemoryReader`, `ESha1Hasher.Update`, `ESha1Hasher.GetHash` and `ESha1Hasher.Get32BitHash` that took invalid arguments, rendering them unusable for their intended purpose, have been fixed.

EasyLocate

- (24.02.0) Objects predicted by **EasyLocate** could have a 0 size, an infinite size, be completely outside of the image or have a size much larger than the size of the image. This could happen after a very short training where the neural network did not yet learn to correctly predict the object location and size. These kind of unwanted predictions are now filtered out.

EasySegment Supervised

- (24.02.0) In some cases, when manipulating the ground truth of images, the metrics would become corrupted and the image level confusion matrix would display wrong information. This has been fixed
- (24.02.0) The precision, recall, average precision and F-Score values could be wrongly computed in some cases and appear out of sync with the value of the confusion matrix. This has been fixed.

Open eVision Studio

- **Open eVision Studio** crashed when the name given to an image was longer than 255 characters. This has been fixed.

5. Known Issues

Open eVision License Manager

- The **Open eVision License Manager** might not start if the **.NET Framework 4.8** 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 7**, 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.

Deep Learning tools

- (2.15.0) The Deep Learning tool objects (EClassifier, EUnsupervisedSegmenter, ESupervisedSegmenter and ELocator) can leak CPU and GPU memory at destruction.
So, it is not recommended to create and delete a lot of Deep Learning tools in the same program.

Samples

- (22.04.0) In the Qt sample Easy3DMatchAlign, on **Ubuntu 16.04** and with **Qt version 5.5.1**, it is not possible to rotate or translate the 3D view.

Licensing

On some installations, the licensing systems can take a long time to start (from 10 seconds up to a few minutes). If you have this issue, you can try the following procedures:

- Clean your software license cache.
 - The software license cache can become bloated by usage.
 - It can also happen if you use only dongles, as the system checks the presence of software licenses in all cases.
 - To clean the cache, use the `LicenseManager.exe /DeleteLicenseFiles` command.



WARNING

This command deletes all the licenses that are not managed by the **Neo License Manager** on the system. Reactivate these licenses after the cleaning.

- Update your system root certificates.
 - If your root certificates are expired, the validation of the licensing system signatures might fail and timeout.
 - This only happens if the computer is on a network, even if the network is not connected to the Internet.
- Enable only the licensing system(s) you use.
 - By default, all the supported licensing systems are enabled.
 - Use the new (available from 2.13) `Preconfiguration::SelectLicensingModels` method to select exactly the licenses you want to enable and avoid issues arising from the usage of the other ones.

.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.

Memory leaks

- If you use the CRT library to detect memory leaks in your program, it can falsely detect some memory leaks when you use the **Open eVision** library.
 - This is a known limitation of the CRT library memory leak detection scheme.
See: <https://docs.microsoft.com/en-us/visualstudio/debugger/finding-memory-leaks-using-the-crt-library>
 - It happens when the memory leak detection scheme is ended before the **Open eVision** DLL is unloaded or the code in the **Open eVision** headers is uninitialized.

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".
 - Use UTF-8 encoded strings to handle filenames with non-latin characters.
- `Easy::GetBestMatchingImageType()` only works for BW8 and C24 images.

Multithreading

- In multithread applications, if `Easy::Initialize` is not called before launching new threads that call **Open eVision** functions, then the number of **Open eVision** processing threads in these new threads may be wrongly initialized to use all the cores that are available on the machine.

EasyBarCode

- `EasyBarCode` requires that a quiet zone of at least one full module is present around the whole bar code to be read.

EasyOCR2

- (2.13.0) The detection of a topology with ranged characters was always failing with the proportionnal detection method. This functionality is now disabled and an error is thrown.

EasyObject

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

EasyMatch

- 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.

Easy3DObject

- The `E3DObjectExtractor` objects saved with **Open eVision 2.11** cannot be loaded with **Open eVision 2.12** and higher.

Easy3D

- (2.16.0) When using the class `EMeshToZMapConverter` without extension, some triangles of the mesh close to the `ZMap` border may be removed.
- (22.04.1) The Linux virtual machines do not support EDL in the `E3DViewer`.
- (23.04.0) The `E3DViewer` does not work on **Nvidia Jetson TX2**, **Nano** and **Xavier** series. It works on **Nvidia Jetson Orin** series.

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.