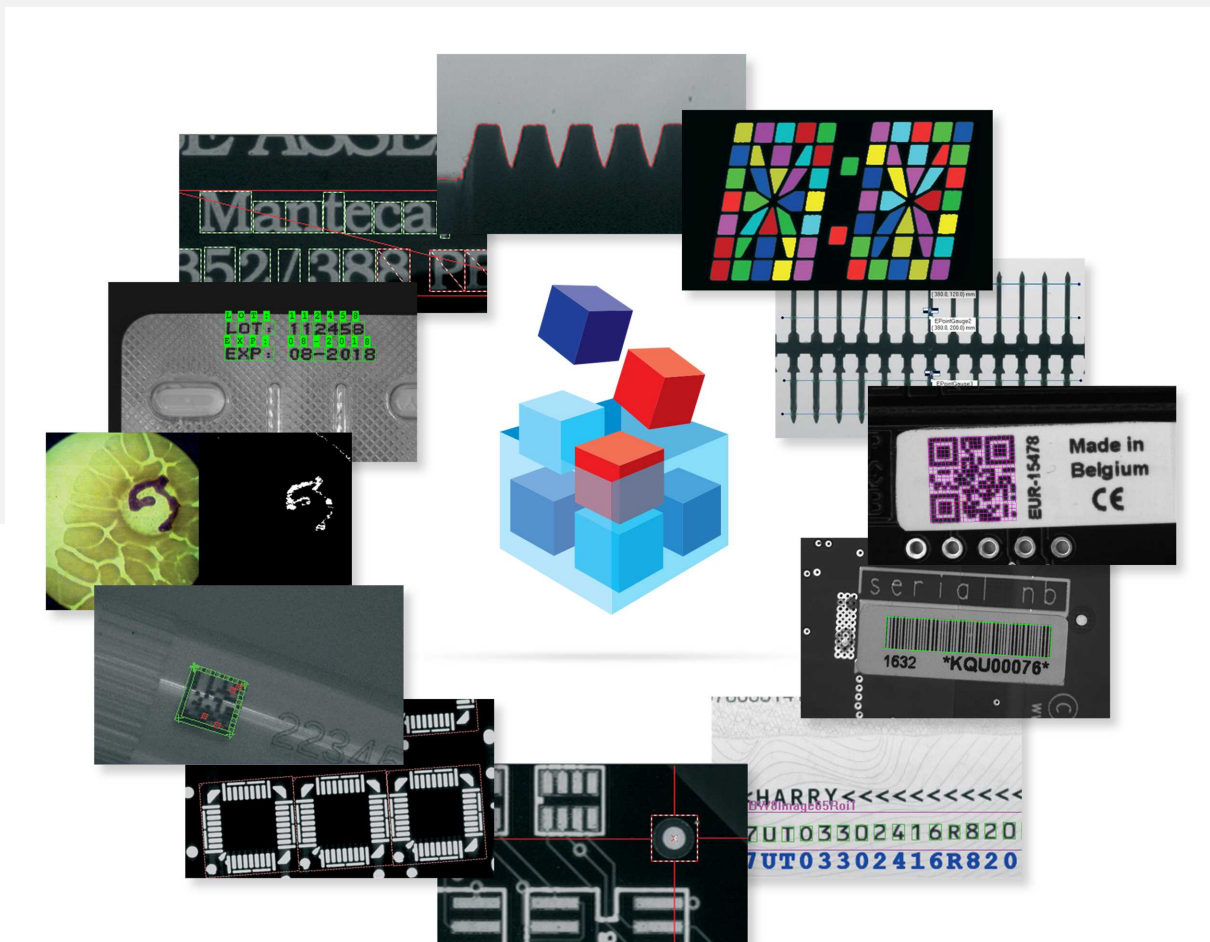


Open eVision

Release 2.7.2



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 2.7.2 (doc build 1115).
© 2019 EURESYS s.a.

Contents

- 1. Release Benefits 4
- 2. Release Specifications 6
- 3. Release Details 8
 - 3.1. Added and Improved Features 8
 - 3.2. Solved Issues 12
 - 3.3. Breaking Changes 13
 - 3.4. Changes 14
- 4. Known Issues 15

1. Release Benefits

Summary

Starting with this release 2.7, Open eVision offers:

Arbitrarily-shaped ROIs

Open eVision supports arbitrarily-shaped ROIs (Regions of Interest) to allow you to restrict the region of the image on which a particular vision tool operates.

Multicore processing

Open eVision supports multicore processing and some algorithms are optimized to take advantage of modern multicore CPUs.

EasyDeepLearning

- A classifier can generate a heat map that shows the contribution of each pixel for a given label.
- **EasyDeepLearning** can automatically determine, at the beginning of training, the batch size that maximizes the training speed.
- **EasyDeepLearning** can compute the batch size that maximizes the classification speed on a GPU.
- New documentation of memory requirements for training and classification and help on how to choose the most appropriate image size.
- Improved classification and training speed on a CPU and a GPU.

EasyMatrixCode2

- **EasyMatrixCode2** is now available in Open eVision Studio.
- Improve the computational time by learning from one or multiple images and by setting the expected number of matrix codes in an image.
- Choose between faster read times or qualitatively better results.
- **EasyMatrixCode2** supports multicore processing and asynchronous computation.
- Improved reading speed and performance.
- Adding of a dedicated user guide and additional code snippets in the documentation.

EasyOCR2

- **EasyOCR2** proposes **EasyOCR2DatabaseManager**, a graphical user interface that helps you to create a large character database for training.

Easy3D

- Support by most **Easy3D** operators of a new depth map and ZMap format using 32 bit float values.
- Adding of new functions to convert between depth map and ZMap formats, with various conversion modes.
- You can save and load a point cloud using the ASCII and binary PCD formats.

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 and Improved Features

Starting with this release 2.7, Open eVision offers:

New Features

Check the documentation for detailed information and support about these new features.

System

- Open eVision supports arbitrarily-shaped ROIs.
 - Use these arbitrarily-shaped regions of interests to restrict the region of an image on which a particular vision tool operates.
 - Use ERegions, the Open eVision classes handling arbitrarily-shaped ROIs, to precisely determine which part of the image, down to a single pixel, to process.
 - The functions supporting ERegions are:
 - **EasyImage**: `Threshold`, `Histogram` and other statistics functions
 - **Easy3D**: `EDepthMapToMeshConverter::Convert` and statistics functions
 - **EasyObject**: `EImageEncoder::Encode`
 - **EasyFind**: `EPatternFinder::Find`

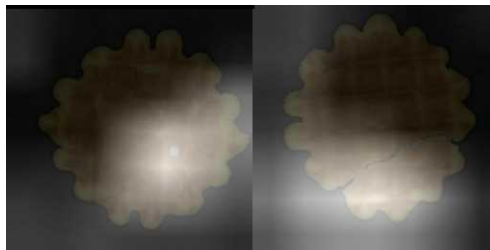
Refer to the documentation for the full list of supported methods.



- Open eVision supports multicore processing.
 - Some algorithms are now optimized to take advantage of modern multicore CPUs.
 - The libraries taking advantage of multicore processing are:
 - **EasyMatrixCode2**: [Read](#)
 - **EasyImage**: [Threshold](#) and statistics
 - **Easy3D**: [EZMapGenerator::Convert](#),
[EDepthMapToPointCloudConverter::Convert](#),
[EDepthMapToMeshConverter::Convert](#),
[ELaserLineExtractor::ExtractProfileFromFrame](#)
 - More to come in future releases.

EasyDeepLearning

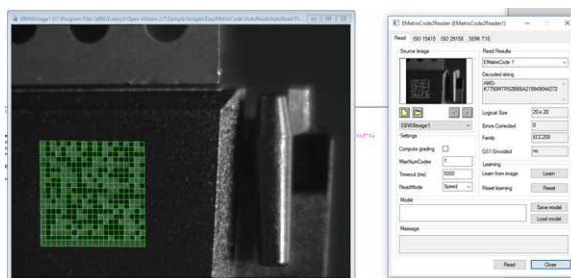
- A classifier can generate a heat map for a given image that shows the contribution of each pixel to a given label.
 - In practice, the heat map generated for the most probable label of an image highlights the region of the image that corresponds to this label.
 - In some cases, you can use this region as the general localization of an object of interest in the image.
 - This operation does not require any additional training to work.



- **EasyDeepLearning** can automatically determine, at the beginning of training, the batch size that maximizes the training speed (see [EClassifier::SetOptimizeBatchSize](#) method).
- **EasyDeepLearning** can compute the batch size that maximizes the classification speed on a GPU (see [EClassifier::GetBatchSizeForMaximumClassificationSpeed](#) method).
- The data augmentation process is improved.
- A new sample program illustrates the performing of the classification of an image. A trained classifier is available in the Additional Resources of the Open eVision download area.

EasyMatrixCode2

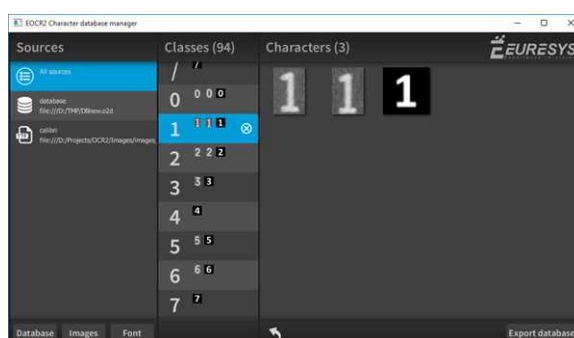
- **EasyMatrixCode2** is now available in Open eVision Studio.



- You can improve the computational time by learning from one or multiple images.
- You can improve the computational time by setting the expected number of matrix codes in an image.
- You can set a timeout.
- You can choose between faster read times or qualitatively better results.
- You can turn on or off the grading of matrix codes.
- You can save and load the Reader parameter settings.
- **EasyMatrixCode2** supports multicore processing.
- **EasyMatrixCode2** supports asynchronous computation, with the ability to stop a Read operation from another thread.

EasyOCR2

- **EasyOCR2** proposes **EasyOCR2DatabaseManager**, a graphical user interface that helps you to create large character databases for training.



Improvements

EasyDeepLearning

- Documentation
 - Documentation of memory requirements for training and classification.
 - Help on how to choose the most appropriate image size for you application.
- Improved classification speed (up to 30% faster on a CPU and a GPU depending on parametrization).
 - Classification of a single image is 30% faster on a CPU (single thread) and 5% faster on a GPU.
 - Classification of a batch of 128 images is 19% faster on a CPU and 30% faster on a GPU.
- Improved training speed (up to 20% faster on a CPU and 25% on a GPU depending on parametrization).
- Using of a bicubic interpolation in data augmentation

EasyMatrixCode2

- Improved reading speed.
- Improved reading performance, you can read images in more challenging images.
- Adding of a dedicated user guide and additional code snippets in the documentation.

EasyOCR2

- Adding of a code snippet showing how to obtain the bitmap that is extracted for each detected character.

Easy3D

- The CAD files for the 3D calibration models are now available as additional resources in the Open eVision download area.
- Support by most **Easy3D** operators of a new depth map and ZMap format using 32 bits float values.
- Adding of new functions to convert between depth map and ZMap formats, with various conversion modes.
- You can save and load a point cloud using the ASCII and binary PCD formats.
- In **EZMapGenerator**, use the `SetZMapToWorldTransform()` method to set the transformation.
- Use the new `GetZMapPositionFromPixelPosition(u, v)` query on a ZMap to get the 3D position in the ZMap space at the given pixel position.

3.2. Solved Issues

Starting with the release 2.7, Open eVision does not exhibit the following known issues anymore:

Licensing

- The error code returned by an exception of the "license missing" type is now `EError_LicenseMissing` as expected.

EasyGauge

- The function `ELine::GetIntersectionOfLines` returns correct results when called with the `limited` parameter set to `True` (segment intersection check).
- **EasyGauge** resets correctly the "Average Distance" calculation of rectangle, line and wedge gauges when it does not find any fit. Open eVision Studio displays this value in the Fitting tab of the EasyGauge dialog.

EasyDeepLearning

- It is possible to train a classifier with the minimum image size of 128 x 128.
- All combinations of data augmentation produce correct results
- The classifier works properly when switching between a GPU and a CPU processing after loading a trained classifier.
- When using serialization, the second call to `EClassifier::Save()` in a program works correctly.
- When an image is not square, the hue and the saturation data augmentations are applied correctly and do not lead to memory corruption.
- The batch classification produces correct results in all cases.
- **EasyDeepLearning Studio** includes the required `vcomp140.dll` file.
- **EasyDeepLearning** finds the CUDA 10.0 dll files in all cases.
- The display and interaction in **EasyDeepLearning Studio** are improved.

EasyDeepLearning Studio

- The display and interaction are correct in the interface.

EasyOCR

- When a character does not match the font (for example because the character dimensions are smaller than the characters in the reference font), the output of `ReadText()` and `Recognize()` is always a " " (space) for the unmatched character, with the error 1.

EasyOCR2

- The `detect` and `read` methods work correctly with an ROI (partially) outside its parent image.
- Setting the ROI height or width to zero in the `detect` or `read` method works correctly .
- The detection is complete when a detected word contains zero characters.

EasyMatrixCode2

- If a matrix code encodes a byte with the value 0x00, **EasyMatrixCode2** returns a complete `DecodedString`.

Easy3D

- In `Easy3D::EPrincipalAxisExtractor`, the axis of the returned transform matrix are correctly ordered.
- The `EZMapGenerator` performs a correct Z clipping when generating a ZMap.
- When the `EZMapGenerator` is used without the reference plane, the generated reference plane optimizes the dynamic range and a correct plane offset is calculated to project all the point cloud in the ZMap value range.

3.3. Breaking Changes

Starting with this release 2.7, Open eVision implements the following breaking changes:

Easy3D

- In the `EObjectBasedCalibrationGenerator`, the methods `SetRange_X/Y/Z` and `GetRange_X/Y/Z` are renamed `SetRangeX/Y/Z` and `GetRangeX/Y/Z`.
- Easy3D Studio projects saved with Open eVision 2.6 are not compatible with the new Easy3D Studio 2.7.

EasyOCR2

- The bitmaps extracted for each detected character are no longer padded with empty pixels. So they do no longer have a uniform width.

3.4. Changes

Starting with this release 2.7, Open eVision implements the following changes:

Installer

- The Open eVision installer now places the samples and the associated images into the public document folder instead of the program files folder.

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.