

ABBYY Mobile Capture SDK

Xamarin Plugin

Table of Contents

Introduction	4
Getting Started	5
API Reference	7
MobileCapture.Forms.Common namespace	7
ICapture interface	8
StartImageCaptureAsync method	8
IImageCaptureResult interface	9
ExportPdfAsync method	9
IPageImageCaptureResult interface	10
CameraSettings class	10
Resolution enum	10
ImageCaptureSettings class	11
DocumentSize class	13
DocumentSize constructor	14
CompressionLevel enum	14
CompressionType enum	15
UISettings class	16
SupportedOrientation enum	16
MobileCapture.Core.Common namespace	17
ICropOperation interface	18
IDataCaptureCoreAPI interface	18
RecognizeAsync method	19
IDetectDocumentBoundaryOperation interface	20
IEngineFactory interface	20
GetEngine method	21
IEngine interface	21
CreateImagingCoreAPI method	22
CreateTextCaptureCoreAPI method	22
CreateDataCaptureCoreAPI method	22
IExportToJpgOperation interface	22
AddImage method	23
IExportToPdfOperation interface	23
AddImage method	24
IExportToPngOperation interface	25
AddImage method	25
IImage interface	25
IImageOperation interface	26
Apply method	26

IImagingCoreAPI interface	27
LoadImage method	27
LoadImageAsync method	28
CreateDetectDocumentBoundaryOperation method	28
CreateCropOperation method	29
CreateRotateOperation method	29
CreateQualityAssessmentForOcrOperation method	29
CreateExportToJpgOperation method	29
CreateExportToPngOperation method	30
CreateExportToPdfOperation method	30
IQualityAssessmentForOcrOperation interface	30
IRotateOperation interface	31
ITextCaptureCoreAPI interface	31
RecognizeAsync method	32
QualityAssessmentForOcrBlock struct	33
Type enum	33
CharInfo class	33
DataCaptureSettings class	34
DataField class	35
TextCaptureSettings class	36
TextBlock class	37
TextLine class	37
Compression enum	38
DetectionMode enum	38
RecognitionLanguage enum	39
RecognitionWarning enum	40
MobileCapture.Core.Droid namespace	41
Image class	41
MobileCapture.Core.iOS namespace	42
Image class	42

Copyright and Trademark Notices 43

Introduction

ABBYY Mobile Capture SDK Xamarin Plugin allows to use the Image Capture feature of ABBYY Mobile Capture SDK in apps based on the [Xamarin](#) platform.

This plugin requires the ABBYY Mobile Capture assets, which differ for Android and iOS, and a license file. You can request ABBYY Mobile Capture trial version on the [ABBYY website](#).

This manual describes .NET API. More information, including the library usage details, is available in the *ABBYY Mobile Capture SDK Developer's Guide* found in the library packages.

Getting Started

To start developing with the ABBYY Mobile Capture SDK Xamarin Plugin, you need to create a Xamarin.Forms application project, add references to provided DLL files to this project and copy the ABBYY Mobile Capture SDK assets, as described below.

1. Create a new Xamarin.Forms mobile application for Android and iOS platforms.

Note: Plugin requires Xamarin.Forms for cross-platform development.

2. In order to access the device's camera and photo library, do the following:

Android	iOS
In the project Properties check the following permissions as required: <ul style="list-style-type: none"> • CAMERA • INTERNET 	Add the following keys to your Info.plist : <ul style="list-style-type: none"> • NSCameraUsageDescription • NSPhotoLibraryUsageDescription.

3. Add to the **References** of your Xamarin.Forms application the following libraries:

Android	iOS
<ul style="list-style-type: none"> • libs/MobileCapture.Forms.Common.dll • libs/MobileCapture.Forms.Android.dll • libs/MobileCapture.Android.dll • libs/MobileCapture.UI.Android.dll 	<ul style="list-style-type: none"> • libs/MobileCapture.Forms.Common.dll • libs/MobileCapture.Forms.iOS.dll • libs/MobileCapture.iOS.dll • libs/MobileCapture.UI.iOS.dll

4. Add all files from the **assets** folder and the **MobileCapture.License** file to corresponding folder of your Xamarin.Forms application:
 - to **Assets** folder for Android
 - to **Resources** folder for iOS

5. Make sure that the following NuGet dependencies are added to your project:

Android	iOS
<ul style="list-style-type: none"> • Xamarin.Android.Support.Core.Utils • Xamarin.Forms • Xamarin.Essentials • Xamarin.Kotlin.Stdlib 	<ul style="list-style-type: none"> • Xamarin.Forms • Xamarin.Essentials

6. For iOS: Register the Mobile Capture SDK interfaces implementation.
Add to the **FinishedLaunching(UIApplication, NSDictionary)** method of the **AppDelegate** class the next string:

```
MobileCapture.Forms.iOS.DependencyManager.Register();
```

Investigate **sample** folder of your distribution to see the ready-to-use sample project.

 **Note:** *To run sample license file is required.*

API Reference

This section describes the C# API of ABBYY Mobile Capture SDK plugin module.

Name	Description
MobileCapture.Forms.Common	The main namespace of Mobile Capture SDK API.
MobileCapture.Core.Common	The main namespace of Mobile Capture SDK API.
MobileCapture.Core.Droid	Namespace for implementing common Core API interfaces on the Android platform.
MobileCapture.Core.iOS	Namespace for implementing common Core API interfaces on the iOS platform.

MobileCapture.Forms.Common namespace

This is the namespace of Mobile Capture SDK API, intended for developing capture scenario with user interface.

Classes

[CameraSettings](#)
[DocumentSize](#)
[ImageCaptureSettings](#)
[UISettings](#)

Interfaces

[ICapture](#)
[IImageCaptureResult](#)
[IPageImageCaptureResult](#)

Enumeration

[CompressionLevel](#)
[CompressionType](#)
[Resolution](#)
[SupportedOrientation](#)

ICapture interface

Provides access to the capture method.

```
public interface ICapture
```

Methods

Name	Description
StartImageCaptureAsync	Opens Activity (if Android) or ViewController (if iOS) with controls for the image capture scenario.

StartImageCaptureAsync method

Opens Activity (if Android) or ViewController (if iOS) with controls for the image capture scenario.

Parameters passed to the method are objects that contain user interface, camera and capture process settings. All parameters can be **null**. In case **null** is passed as a parameter instead of the settings object, default settings will be used.

```
Task<IImageCaptureResult> StartImageCaptureAsync (
    UISettings uiSettings,
    CameraSettings cameraSettings,
    ImageCaptureSettings captureSettings
);
```

Parameters

uiSettings

Settings for tuning the user interface appearance, represented by a [UISettings](#) class object. Is **null** by default.

cameraSettings

Settings of the device camera, represented by a [CameraSettings](#) class object. Is **null** by default.

captureSettings

Capture scenario settings, such as required number of pages to be captured, export compression type, etc. This parameter is represented by an [ImageCaptureSettings](#) class object. Is **null** by default.

Return values

When images were captured successfully, result is returned as an [IImageCaptureResult](#) object. If capture process was interrupted by user, **TaskCanceledException** is thrown. In case error occurred during capture process, **Exception** is thrown.

IImageCaptureResult interface

Stores the whole amount of captured images. Supports export to PDF format.

```
public interface IImageCaptureResult
```

Properties

Name	Type	Description
Pages	IPageImageCaptureResult[] , read-only	Capture result in the internal format.

Methods

Name	Description
ExportPdfAsync	Saves PDF file containing captured result.

ExportPdfAsync method

The method saves PDF file, containing captured result, to the specified path.

 **Note:** This method is asynchronous.

```
Task ExportPdfAsync( string path )
```

Parameters

path

Full path for saving the result file.

IPageImageCaptureResult interface

Represents a singular captured image.

```
public interface IPageCaptureResult
```

Properties

Name	Type	Description
Image	string , read-only	Full address of the file containing result image. The result image can be in JPG or PNG format, depending on the CompressionType setting.

CameraSettings class

Manages camera resolution, defined for the capture scenario.

```
public class CameraSettings
```

Properties

Name	Type	Description
SelectedResolution	Resolution	The resolution of the images captured from the camera preview. Default: Resolution.FullHD

Resolution enum

The resolution of the images captured from the camera preview.

```
public enum Resolution
{
    HD,
    FullHD,
    UHD4K
}
```

Constants

Name	Description
HD	Captured image will have 1280x720px resolution.
FullHD	Captured image will have 1920x1080px resolution. This is the default value.
UHD4K	Captured image will have 3840x2160px resolution

ImageCaptureSettings class

Provides access to the settings of the image capture scenario.

This class provides access to such settings as required minimum document to view ratio, document size for cropping, etc.

```
public class ImageCaptureSettings
```

Properties

Name	Type	Description
LicenseName	string	Name of the license file. Default: "MobileCapture.License"
ShouldShowPreview	bool	Specifies if a preview of an image should be shown after capture. If true , a preview is shown. If false , the camera view is shown instead. Default: false.
RequiredPageCount	int	Total number of pages to be captured. Set the page-limitation mode of the image capture as following: <ul style="list-style-type: none"> 0 to allow unlimited image capture. The set of the result images can be saved or edited at any time; a positive value to set the exact number of images that should be captured. Images saving is

Name	Type	Description
		<p>enabled only when this number of images are been captured.</p> <p>Default: 0.</p>
PageCompressionType	CompressionType	<p>The type of compression for PDF files depending on the base image format.</p> <p>Default: CompressionType.Jpg.</p>
PageCompressionLevel	CompressionLevel	<p>The uniform image compression scale for lossy formats.</p> <p>Default: CompressionLevel.Low.</p>
PageSize	DocumentSize	<p>The physical size of the documents.</p> <p>Default: DocumentSize.ANY.</p>
MinDocToViewRatio	float	<p>Coefficient, signifying required document to the capture view ratio. Capture will be performed only if the ratio is greater than or equal to this property value.</p> <p>The value of this property should be in the range from 0 to 1.</p> <p>Default: 0.15.</p>
AspectRatioMin	float	<p>Lower limit of the document's aspect ratio. The value should be greater than or equal to 1 or 0 (aspect ratio is not set).</p> <p>This property is used in pair with the AspectRatioMax, defining an interval of acceptable aspect ratio values of the document to be captured. Value of AspectRatioMin should be less than value of AspectRatioMax.</p> <p>If only AspectRatioMax is set, AspectRatioMin will be set to 1.</p> <p>Setting aspect ratio will help to improve boundary detection accuracy.</p> <p>Default: 0 (aspect ratio is not set).</p>

Name	Type	Description
AspectRatioMax	float	<p>Upper limit of document's aspect ratio. The value should be greater than or equal to 1 or 0 (aspect ratio is not set).</p> <p>This property is used in pair with the AspectRatioMin, defining an interval of acceptable aspect ratio values of the document to be captured. Value of AspectRatioMax should be greater than value of AspectRatioMin.</p> <p>If only AspectRatioMin is set, AspectRatioMax will be set to infinity.</p> <p>Setting aspect ratio will help to improve boundary detection accuracy.</p> <p>Default: 0 (aspect ratio is not set).</p>
ImageFromGalleryMaxSize	int	<p>Maximum available size of an image, loaded from the gallery. The size is defined as the length of the largest side of an image (in pixels).</p> <p>Default: 4096.</p>

DocumentSize class

The physical size of the documents. Can be chosen from the preset values or defined manually.

```
public class DocumentSize
```

Constructor

Name	Description
DocumentSize	Initializes a new instance of the DocumentSize class.

Properties

Name	Type	Description
width	float	Document width in millimeters.

Name	Type	Description
height	float	Document height in millimeters.

Constants

Name	Description
ANY	The captured document can have any size.
A4	Document size is 210x297 millimeters.
LETTER	Document size is 215.9x279.4 millimeters.
BUSINESS_CARD	Document size is 85.6x53.98 millimeters.

DocumentSize constructor

Initializes a new instance of the [DocumentSize](#) class.

```
public DocumentSize(
    float width,
    float height
)
```

Parameters

width

Document width in millimeters.

height

Document height in millimeters.

CompressionLevel enum

The uniform image compression scale for lossy formats.

```
public enum CompressionLevel
{
    Low,
    Normal,
```

```

    High,
    ExtraHigh
}

```

Constants

Name	Description
Low	<p>The lowest compression rate, that still might have any noticeable effect on recognition of small text.</p> <p>This is the default value.</p>
Normal	Balanced trade-off between compression and quality. Good safety margin.
High	More compression, less safety margin. Might perform poorly with small text, but generally still ok.
ExtraHigh	The maximum recommended compression rate. Will perform poorly with small text. Advisable only for relatively large text and very slow networks.

CompressionType enum

The type of compression for page files depending on the base image format.

```

public enum CompressionType
{
    Jpg,
    Png
}

```

Constants

Name	Description
Jpg	<p>The image compression type for PDF files based on the JPG format.</p> <p>This is the default value.</p>
Png	The image compression type for PDF files based on the PNG format.

UISettings class

Provides access to the settings for tuning the user interface appearance.

The elements of the user interface can be customized according to the implementation of your application needs and its design. The buttons for enabling flashlight, manual capture and choosing an image from gallery can be hidden.

```
public class UISettings
```

Properties

Name	Type	Description
isFlashlightButtonVisible	boolean	Show (true) or hide (false) the flashlight button in the image capture dialog. Default: true (flashlight visible).
isCaptureButtonVisible	boolean	Show (true) or hide (false) the button for taking photo manually. Default: true (button is visible).
IsGalleryButtonVisible	boolean	Show (true) or hide (false) the button for choosing an image from photo gallery. Default: true (button is visible).
Orientation	SupportedOrientation	Type of image orientation. Default: SupportedOrientation.Default

SupportedOrientation enum

Type of image orientation.

```
public enum SupportedOrientation
{
    Portrait,
    Landscape,
    Default
}
```


Constants

Name	Description
Portrait	Image for capture has a portrait orientation.
Landscape	Image for capture has a landscape orientation.
Default	Image for capture can have orientation of any type, depending on the device settings.

MobileCapture.Core.Common namespace

This is the namespace of Mobile Capture SDK .NET API, implementing capture mechanisms. Provides access to low-level single image core API functions. Intended for advanced users.

Classes

[CharInfo](#)
[DataField](#)
[DataCaptureSettings](#)
[TextBlock](#)
[TextLine](#)
[TextCaptureSettings](#)

Structs

[QualityAssessmentForOcrBlock](#)

Interfaces

[ICropOperation](#)
[IDataCaptureCoreAPI](#)
[IDetectDocumentBoundaryOperation](#)
[IEngine](#)
[IEngineFactory](#)
[IExportToJpgOperation](#)
[IExportToPdfOperation](#)
[IExportToPngOperation](#)
[IImage](#)
[IImagingCoreAPI](#)
[IImageOperation](#)
[IQualityAssessmentForOcrOperation](#)
[IRotateOperation](#)
[ITextCaptureCoreAPI](#)

Enumeration

[Compression](#)

[DetectionMode](#)

[RecognitionWarning](#)

[RecognitionLanguage](#)

[Type](#)

ICropOperation interface

An operation for image crop.

This operation not only crops the image but also corrects perspective distortion if needed.

Use the [Apply](#) method of the [IImageOperation](#) interface to apply the operation to the image.

```
interface ICropOperation : IImageOperation
```

Properties

Name	Type	Description
DocumentWidth	float	The document width in millimeters.
DocumentHeight	float	The document height in millimeters.
DocumentBoundary	Point[]	Document boundary defined by the four vertex points of the bounding quadrangle. The vertices are indexed clockwise starting from the bottom left.
Resolution	Rectangle , read-only	The image resolution as calculated from image size and physical page size.

IDataCaptureCoreAPI interface

Provides access to functionality of data capture scenario for a single image.

```
interface IDataCaptureCoreAPI
```

Properties

Name	Type	Description
Settings	DataCaptureSettings , read-only	Settings specific for data capture scenario.

Methods

Name	Description
RecognizeAsync	Extracts data from a still image.

RecognizeAsync method

Extracts data from a still image returns a collection of captured data fields.

Note: This method is asynchronous.

```
Task<List<DataField>> RecognizeAsync(
    ImageSource image,
    Func<int, RecognitionWarning, bool> onProgress,
    Action<int> onOrientationDetected
);
```

Parameters

image

Captured image, represented by an [ImageSource](#) object. Data will be extracted from this image.

onProgress

Callback, signaling the progress of recognition process. This function receives 2 parameters and returns a boolean value:

- [in] progress in percents as integer value;
- [in] warning that occurred during processing, represented by a [RecognitionWarning](#) constant;
- [out] **true** to proceed recognition, **false** to stop recognition.

onOrientationDetected

The callback informing you when the image orientation is detected. The *angle* parameter can take values of 0, 90, 180, and 270, and means the angle on which the image should be rotated to get normal orientation.

Return values

When data are extracted, result is returned as a list of [DataField](#) objects.

IDetectDocumentBoundaryOperation interface

An operation for image boundaries detection. Use the [Apply](#) method of the [IImageOperation](#) interface to apply the operation to the image.

```
interface IDetectDocumentBoundaryOperation : IImageOperation
```

Properties

Name	Type	Description
AreaOfInterest	Rectangle	The rectangle specifying the area of interest in the image coordinates.
Mode	DetectionMode	Document boundary detection mode. The mode influences the crop speed and accuracy.
DocumentWidth	float	The document width in millimeters.
DocumentHeight	float	The document height in millimeters.
DocumentBoundary	Point[] , read-only	The detected document boundary defined by the four vertex points of the bounding quadrangle. The vertices are indexed clockwise starting from the bottom left.

IEngineFactory interface

Provides method for loading engine.

```
interface IEngineFactory
```

Methods

Name	Description
GetEngine	Loads the ABBYY Mobile Capture SDK engine.

GetEngine method

Loads the ABBYY Mobile Capture SDK engine.

Creating the [Engine](#) and initializing the library may take up a lot of time, because all the resources need to be loaded. Therefore you should call this method only once, during initialization, and reuse the [Engine](#) object every time you need to start a new recognition operation.

```
IEngine GetEngine(string licenseName);
```

Parameters

licenseName

The license file name. Please note, that the license file should be placed the **Assets** directory for Android and **Resources** directory for iOS .

Return values

The method returns an instance of the [Engine](#) object.

IEngine interface

ABBYY Mobile Capture SDK engine via which all other objects may be created.

Creating the **Engine** and initializing the library may take up a lot of time, since all the resources have to be loaded. Therefore this object should only be created once (using the [GetEngine](#) method of the [IEngineFactory](#) interface), during initialization, and you should reuse it every time you need to start a new recognition operation.

```
interface IEngine
```

Methods

Name	Description
CreateImagingCoreAPI	Returns an IImagingCoreAPI object, providing access to the methods for image capture scenario.
CreateTextCaptureCoreAPI	Returns an ITextCaptureCoreAPI object, providing access to the methods and settings for text capture scenario.

Name	Description
CreateDataCaptureCoreAPI	Returns an IDataCaptureCoreAPI object, providing access to the methods and settings for data capture scenario.

CreateImagingCoreAPI method

Creates an object, providing access to the methods for image capture scenario.

```
IIImagingCoreAPI CreateImagingCoreAPI();
```

Return values

The method returns an instance of the [IIImagingCoreAPI](#) object.

CreateTextCaptureCoreAPI method

Creates an object, providing access to the methods and settings for text capture scenario.

```
ITextCaptureCoreAPI CreateTextCaptureCoreAPI();
```

The method returns an instance of the [ITextCaptureCoreAPI](#) object.

CreateDataCaptureCoreAPI method

Creates an object, providing access to the methods and settings for data capture scenario.

Note: *The functionality is currently supported for business cards recognition only.*

```
IDataCaptureCoreAPI CreateDataCaptureCoreAPI(string profile);
```

The method returns an instance of the [IDataCaptureCoreAPI](#) object.

IExportToJpgOperation interface

An operation for image export into JPG format. Property of the interface predefine compression for the exported image.

```
interface IExportToJpgOperation : IDisposable
```

Properties

Name	Type	Description
Compression	Compression	A uniform image compression scale for all lossy formats.

Methods

Name	Description
AddImage	Defines the image to be exported.

AddImage method

Defines the image to be exported.

```
void AddImage (IImage image);
```

Parameters

image

Captured image to be exported, represented by an [IImage](#) object.

Note: Please note, that only one captured image can be exported via this method. On attempt to call this method repeatedly an exception will be thrown.

IExportToPdfOperation interface

An operation for image export into PDF format. Use properties of this interface to tune result PDF-file metadata and settings.

```
interface IExportToPdfOperation : IDisposable
```

Properties

Name	Type	Description
Compression	Compression	A uniform image compression scale for all lossy

Name	Type	Description
		formats.
PageWidth	int	Result PDF document width in millimeters.
PageHeight	int	Result PDF document height in millimeters.
PdfInfoTitle	string	Title of the result PDF document.
PdfInfoSubject	string	Subject of the result PDF document.
PdfInfoKeywords	string	Keywords defined for the result PDF document.
PdfInfoAuthor	string	Information about the result PDF document author.
PdfInfoCompany	string	Information about the result PDF document company.
PdfInfoCreator	string	Creator of the result PDF document.
PdfInfoProducer	string	Producer of the result PDF document.

Methods

Name	Description
AddImage	Defines the image to be exported.

AddImage method

Defines the image to be exported.

```
void AddImage (IImage image);
```

Parameters

image

Captured image to be exported, represented by an [IImage](#) object.

IExportToPngOperation interface

An operation for image export into PNG format.

```
interface IExportToPngOperation : IDisposable
```

Methods

Name	Description
AddImage	Defines the image to be exported.

AddImage method

Defines the image to be exported.

```
void AddImage(IImage image);
```

Parameters

image

Captured image to be exported, represented by an [IImage](#) object.

! **Note:** Please note, that only one captured image can be exported via this method. On attempt to call this method repeatedly an exception will be thrown.

IImage interface

Represents a single captured image, giving access to such properties as geometrical parameters. It allows you to get a bitmap handle corresponding to this image.

```
interface IImage : IDisposable
```

Properties

Name	Type	Description
ImageSize	Size , read-only	Width and height of captured image.

Name	Type	Description
ImageSource	ImageSource , read-only	Captured image.

IImageOperation interface

Image operation interface.

After the image is captured a sequence of specified operations can be applied to it:

- image crop
- document boundary detection
- export to different formats
- rotation
- quality for OCR assessment

Create an operation using one of the [IImagingCoreAPI](#) interface methods, set up the operation and then call **Apply** method of this interface to captured image.

```
interface IImageOperation
```

Methods

Name	Description
Apply	Applies chosen operation to the image.

Apply method

Applies chosen operation to the image.

```
void Apply(IImage image);
```

Parameters

image

The image to which the operation is to be applied.

IImagingCoreAPI interface

Provides access to functionality for single image capture scenario.

```
interface IImagingCoreAPI
```

Methods

Name	Description
LoadImage	Loads an image from a byte buffer.
LoadImageAsync	Loads an image asynchronously from an ImageSource object.
CreateDetectDocumentBoundaryOperation	Creates an operation for document boundary detection.
CreateCropOperation	Creates an operation for image crop.
CreateRotateOperation	Creates an operation for rotating the image.
CreateQualityAssessmentForOcrOperation	Creates an operation for estimating if an image quality is suitable for OCR.
CreateExportToJpgOperation	Creates an operation for exporting image to JPG.
CreateExportToPngOperation	Creates an operation for exporting image to PNG.
CreateExportToPdfOperation	Creates an operation for exporting image to PDF.

LoadImage method

Loads an image from a byte buffer.

```
IImage LoadImage(byte[] imageBuffer, int imageWidth, int imageHeight, int
orientation);
```

Parameters

imageBuffer

The byte buffer to be filled with image data. Only NV21 format is currently supported.

imageWidth

The width of the image in millimeters.

imageHeight

The height of the image in millimeters.

orientation

The orientation of the image. The orientation is used to rotate the final image when getting the result.

Return values

The method returns the image object implementing the [IImage](#) interface.

LoadImageAsync method

Loads an image asynchronously from [ImageSource](#).

! *Note:* This method is asynchronous.

```
Task<IImage> LoadImageAsync(ImageSource imageSource);
```

Parameters

imageSource

Image as an [ImageSource](#) object to be loaded.

Return values

The method returns the image object implementing the [IImage](#) interface.

CreateDetectDocumentBoundaryOperation method

Creates an operation for document boundary detection.

```
IDetectDocumentBoundaryOperation CreateDetectDocumentBoundaryOperation();
```

Return values

The method returns an object implementing the [IDetectDocumentBoundaryOperation](#) interface.

CreateCropOperation method

Creates an operation for image crop. This operation not only crops the image but also corrects perspective distortion if needed.

```
ICropOperation CreateCropOperation();
```

Return values

The method returns an object implementing the [ICropOperation](#) interface.

CreateRotateOperation method

Creates an operation for rotating the image.

```
IRotateOperation CreateRotateOperation();
```

Return values

The method returns an object implementing the [IRotateOperation](#) interface.

CreateQualityAssessmentForOcrOperation method

Creates an operation for estimating if an image quality is suitable for OCR.

```
IQualityAssessmentForOcrOperation CreateQualityAssessmentForOcrOperation();
```

Return values

The method returns an object implementing the [IQualityAssessmentForOcrOperation](#) interface.

CreateExportToJpgOperation method

Creates an operation for exporting image to JPG.

```
IExportToJpgOperation CreateExportToJpgOperation(Stream stream);
```

Parameters

stream

The output stream to which the exported data will be written.

Return values

The method returns an object implementing the [IExportToJpgOperation](#) interface.

CreateExportToPngOperation method

Creates an operation for exporting image to PNG.

```
IExportToPngOperation CreateExportToPngOperation(Stream stream);
```

Parameters

stream

The output stream to which the exported data will be written.

Return values

The method returns an object implementing the [IExportToPngOperation](#) interface.

CreateExportToPdfOperation method

Creates an operation for exporting image to PDF.

```
IExportToPdfOperation CreateExportToPdfOperation(Stream stream);
```

Parameters

stream

The output stream to which the exported data will be written.

Return values

The method returns an object implementing the [IExportToPdfOperation](#) interface.

IQualityAssessmentForOcrOperation interface

An operation for estimating if an image quality is suitable for OCR. The whole image is represented as a set of rectangles. A type of rectangle can be either text or unknown. The quality assessment of the image for OCR is calculated based on the rectangles collection.

Use the [Apply](#) method of the [IImageOperation](#) interface to apply the operation to the image.

```
interface IQualityAssessmentForOcrOperation : IImageOperation
```

Properties

Name	Type	Description
DocumentBoundary	Point[]	Document boundary defined by the four vertex points of the bounding quadrangle. The vertices are indexed clockwise starting from the bottom left.
QualityAssessmentForOcrBlocks	QualityAssessmentForOcrBlock[] , read-only	Collection of rectangles with corresponding quality assessment for each of them.

IRotateOperation interface

An operation for rotating the image to a specified angle. Use the [Apply](#) method of the [IImageOperation](#) interface to apply the operation to the image.

```
interface IRotateOperation : IImageOperation
```

Properties

Name	Type	Description
Angle	int	The angle in degrees. Available values of the angle: 0, 90, 180, 270.

ITextCaptureCoreAPI interface

Provides access to functionality for text capture scenario for a single image.

```
interface ITextCaptureCoreAPI
```

Properties

Name	Type	Description
Settings	TextCaptureSettings	Settings specific for text capture scenario.

Name	Type	Description
	read-only	

Methods

Name	Description
RecognizeAsync	Performs recognition of an image.

RecognizeAsync method

Performs recognition of an image and returns a collection of recognized text lines.

Note: This method is asynchronous.

```
Task<List<TextBlock>> RecognizeAsync(
    ImageSource image,
    Func<int, RecognitionWarning, bool> onProgress,
    Action<int> onOrientationDetected);
```

Parameters

image

Captured image, represented by an [ImageSource](#) object. Data will be extracted from this image.

onProgress

Callback, signaling the progress of recognition process. This function receives 2 parameters and returns a boolean value:

- [in] progress in percents as integer value;
- [in] warning that occurred during processing, represented by a [RecognitionWarning](#) constant;
- [out] **true** to proceed recognition, **false** to stop recognition.

onOrientationDetected

The callback informing you when the image orientation is detected. The *angle* parameter can take values of 0, 90, 180, and 270, and means the angle on which the image should be rotated to get normal orientation.

Return values

When document is recognized, result is returned as a list of [TextBlock](#) objects.

QualityAssessmentForOcrBlock struct

Collection of rectangles with corresponding quality assessment for each of them.

```
struct QualityAssessmentForOcrBlock
```

Properties

Name	Type	Description
BlockType	Type , read-only	Stores type of the rectangle.
Quality	int , read-only	Value from 0 to 100 that indicates suitability of the text for OCR.
Rect	Rectangle , read-only	Current rectangle.

Type enum

Rectangle type for quality for OCR estimating.

```
public enum Type{
    Text,
    Unknown
}
```

Constants

Name	Description
Text	Current rectangle contains text.
Unknown	Content of current rectangle is undefined.

CharInfo class

Extended information about character formatting and properties.

```
class CharInfo
```

Properties

Name	Type	Description
Quadrangle	List<Point>	The four vertex points of the bounding quadrangle. The vertices are indexed clockwise starting from the bottom left.
BoundingRect	Rectangle	Rectangle, containing the character. Rectangle is defined by width, height and upper-left corner.
IsItalic	bool	Indicates if character font is italic.
IsBold	bool	Indicates if character font is bold.
IsUnderlined	bool	Indicates if character font is underlined.
IsStrikethrough	bool	Indicates if character font is strikethrough.
IsSmallcaps	bool	Indicates if character style is small caps.
IsSuperscript	bool	Indicates if character is a superscript.
IsUncertain	bool	Indicates if character is uncertainly recognized.


DataCaptureSettings class

Settings specific for data capture scenario.

```
class DataCaptureSettings
```

Properties

Name	Type	Description
IsTextOrientationDetectionEnabled	bool	Enables or disables detection of the image orientation while preprocessing.

Name	Type	Description
		<p>If the property is set to true, the image top is detected and correct orientation can be used for image rotation.</p> <p>You can set this property to false for speeding the process up.</p> <p> Note: Disable the image detection only if you can be sure that the captured image has correct orientation. Otherwise the text on image will not be detected and recognized.</p> <p>The default value of this property is true (enabled).</p>
Languages	ISet<RecognitionLanguage>	List of languages, supported for data capture profile.

DataField class

A recognized data field. Provides field contents, location and included data fields, if applicable.

Note that a field may have several components — for example, it can contain two or more words. Component details are available from the **Components** array. Each element of this array is a **DataField** object with its own **Text** property (for example, a word) and **Quadrangle** property (the bounding quadrangle of this component). The field's **Text** property contains its entire text, and the field's **Quadrangle** property represents the whole area of a field: this quadrangle encloses the quadrangles of all components.

The **Components** array always contains at least one element. When a field contains only one component, the **Text** and **Quadrangle** properties of the field and this component are identical.

```
class DataField
```

Properties

Name	Type	Description
Id	string	The internal field identifier. May be null .
Name	string	The human-readable name of the field.

Name	Type	Description
Text	string	The text of the field.
Quadrangle	List<Point>	The four vertex points of the bounding quadrangle. The vertices are indexed clockwise starting from the bottom left.
CharsInfo	List<CharInfo>	Extended characters' information as an array of CharInfo objects.
Components	List<DataField>	An array of data fields, representing one complex field found in the image, with all additional information.

TextCaptureSettings class

Settings specific for text recognition scenario.

```
class TextRecognizerSettings
```

Properties

Name	Type	Description
Languages	ISet<RecognitionLanguage>	Languages to be used for recognition.
AreaOfInterest	Rectangle?	Width and height of the recognition area. The area of interest is centered in the preview frame, its width and height are relative to the preview frame size and should be in the [0.0, 1.0] range.
IsTextOrientationDetectionEnabled	bool	<p>Enables or disables detection of the image orientation while preprocessing.</p> <p>If the property is set to true, the image top is detected and correct orientation can be used for image rotation.</p>

Name	Type	Description
		<p>You can set this property to false for speeding the process up.</p> <p>! <i>Note: Disable the image detection only if you can be sure that the captured image has correct orientation. Otherwise the text on image will not be detected and recognized.</i></p> <p>The default value of this property is true (enabled).</p>

TextBlock class

A collection of recognized text lines found in a text area (block) on the image.

```
class TextBlock
```

Properties

Name	Type	Description
Lines	List< TextLine >	The lines of recognized text.

TextLine class

A line of recognized text; the location and additional information are also available.

```
class TextLine
```

Properties

Name	Type	Description
Text	string	The text of the field.
Quadrangle	List< Point >	The four vertex points of the bounding quadrangle. The vertices are indexed clockwise starting from the

Name	Type	Description
		bottom left.
BoundingRect	Rectangle	Rectangle, containing recognized text line. Rectangle is defined by width, height and upper-left corner.
CharsInfo	List < CharInfo >	Extended characters' information as an array of CharInfo objects.

Compression enum

A uniform image compression scale for all lossy formats.

```
enum Compression{
    Low,
    Normal,
    High,
    ExtraHigh
}
```

Constants

Name	Description
Low	The lowest compression rate that still might have any noticeable effect on recognition of small text
Normal	[Default] Balanced trade-off between compression and quality.
High	More compression, less safety margin.
ExtraHigh	The maximum compression rate.

DetectionMode enum

The type of document boundary detection and crop.

```
enum DetectionMode {
    Default,
```

```

    Fast
}

```

Constants

Name	Description
Default	[Default] Balanced mode, that combines optimal processing speed and high quality.
Fast	Fast mode, that signifies processing speed.

RecognitionLanguage enum

The language of the text.

```

enum RecognitionLanguage{
    Afrikaans,
    Albanian,
    Basque,
    Belarusian,
    Breton,
    Bulgarian,
    Catalan,
    Chechen,
    ChineseSimplified,
    ChineseTraditional,
    CrimeanTatar,
    Croatian,
    Czech,
    Danish,
    Dutch,
    DutchBelgian,
    English,
    Estonian,
    Fijian,
    Finnish,
    French,
    German,
    GermanNewSpelling,
    Greek,
    Hawaiian,
    Hungarian,
    Icelandic,
    Indonesian,
    Irish,
    Italian,
    Japanese,
    Kabardian,

```

```

        Korean,
        KoreanHangul,
        Latin,
        Latvian,
        Lithuanian,
        NorwegianBokmal,
        NorwegianNynorsk,
        Macedonian,
        Malay,
        Maori,
        Moldavian,
        Mongol,
        Ossetic,
        Polish,
        Portuguese,
        PortugueseBrazilian,
        Provencal,
        RhaetoRomanic,
        Romanian,
        Russian,
        Samoan,
        Serbian,
        Slovak,
        Slovenian,
        Spanish,
        Swahili,
        Swedish,
        Tagalog,
        Tatar,
        Turkish,
        Ukrainian,
        Welsh
    }

```

RecognitionWarning enum

A warning that occurred during processing.

```

enum RecognitionWarning{
    NoWarning,
    RecognitionIsSlow,
    ProbablyLowQualityImage,
    ProbablyWrongLanguage,
    WrongLanguage,
    TextTooSmall
}

```


Constants

Name	Description
NoWarning	Processing succeeded without warnings.
RecognitionIsSlow	Recognition takes too much time. Check if there is some problem.
ProbablyLowQualityImage	The image quality (contrast, resolution) may not be good enough for accurate results.
ProbablyWrongLanguage	The recognition language may be specified incorrectly.
WrongLanguage	The recognition language is specified incorrectly.
TextTooSmall	The text is too small. Advise the end user to move the camera closer or zoom in.

MobileCapture.Core.Droid namespace

Namespace for implementing common Core API interfaces on the Android platform.

Classes

[Image](#)

Image class

Implementation of the [IImage](#) interface, providing access to the image.

```
class Image : IImage
```

Properties

Name	Type	Description
Bitmap	Bitmap , read-only	Android native image object, referring to the captured image.

Name	Type	Description
ImageSource	ImageSource , read-only	Captured image source.

MobileCapture.Core.iOS namespace

Namespace for implementing common Core API interfaces on the iOS platform.

Classes

[Image](#)

Image class

Implementation of the [IImage](#) interface, providing access to the image.

```
class Image : IImage
```

Properties

Name	Type	Description
UIImage	UIImage , read-only	iOS native image object, referring to the captured image.
ImageSource	ImageSource , read-only	Captured image source.

Copyright and Trademark Notices

ABBYY® Mobile Capture © 2019 ABBYY Production LLC.

ABBYY is a registered trademark or a trademark of ABBYY Software Ltd.

Kotlin stdlib

Copyright 2010-2018 JetBrains s.r.o. Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at <http://www.apache.org/licenses/LICENSE-2.0> Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

Libjpeg

This software is based in part on the work of the Independent JPEG Group.

BRISK

BRISK - Binary Robust Invariant Scalable Keypoints Reference implementation of [1] Stefan Leutenegger, Margarita Chli and Roland Siegwart, BRISK: Binary Robust Invariant Scalable Keypoints, in Proceedings of the IEEE International Conference on Computer Vision (ICCV2011). Copyright (C) 2011 The Autonomous Systems Lab (ASL), ETH Zurich, Stefan Leutenegger, Simon Lynen and Margarita Chli. Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met: * Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer. * Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution. * Neither the name of the ASL nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission. THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDERS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE

AGAST++

Copyright (c) 2010, Elmar Mair All rights reserved. Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met: * Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer. * Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution. * Neither the name of the owner nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission. THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDERS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO,

PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

AKAZE

Copyright (c) 2013, Pablo Fernandez Alcantarilla, Jesus NuevoAll Rights ReservedRedistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met: * Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer. * Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution. * Neither the name of the copyright holders nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

KAZE

Copyright (c) 2012, Pablo Fernández AlcantarillaAll Rights ReservedRedistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met: * Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer. * Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution. * Neither the name of the copyright holders nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

MD5 message digest algorithm reference implementation

This software is derived in part from the RSA DataSecurity, Inc. MD5 Message-Digest Algorithm

Xamarin SDK

The MIT License (MIT)

Copyright (c) .NET Foundation Contributors

All rights reserved.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Xamarin.Essentials

The MIT License (MIT)

Copyright (c) Microsoft Corporation

All rights reserved.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Xamarin.Android SDK

The MIT License (MIT)

Copyright (c) .NET Foundation Contributors

All rights reserved.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights

to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Xamarin.iOS

Xamarin SDK

The MIT License (MIT)

Copyright (c) .NET Foundation Contributors

All rights reserved.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

OpenTK

The Open Toolkit library license

Copyright (c) 2006 - 2009 The Open Toolkit library.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Third parties

OpenTK.Platform.Windows and OpenTK.Platform.X11 include portions of the Mono class library. These portions are covered by the following license:

Copyright (c) 2004 Novell, Inc.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

OpenTK.Compatibility offers the Tao namespace, which includes portions of the Tao Framework library. These portions are covered by the following license:

Copyright ©2003-2007 Tao Framework Team

<http://www.taoframework.com>

All rights reserved.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

OpenTK.Half offers Half-to-Single and Single-to-Half conversions based on OpenEXR source code, which is covered by the following license:

Copyright (c) 2002, Industrial Light & Magic, a division of Lucas Digital Ltd. LLC. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- * Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- * Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- * Neither the name of Industrial Light & Magic nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

All other trademarks and copyrights are the property of their respective owners.