XP Idea OCR SDK v1.0

Table of Contents

Symbol Reference 1

xpidea_ocr Library 1

Classes 2

OCREngine CoClass 2

Interfaces 3

ICharacter Interface 3

IImage Interface 3

IImage::AddRegion Method 4

IImage::NoiseLevel Method 4

IImage::PerformOperation Method 4

IImage::SkewAngleDetected Method 4

IOCREngine Interface 5

IOCREngine::Configure Method 5

IOCREngine::Init Method 6

IOCREngine::Recognize Method 6

IOCREngineEvents Interface 7

IOCREngineEvents::OnAfterRecognition Method 7

IOCREngineEvents::OnBeforeRecognition Method 7

IPage Interface 7

IPage::Region Method 8

IPage::RegionsCount Method 8

IReconstructor Interface 8

IReconstructor::Character Method 9

IReconstructor::EndBlock Method 9

IReconstructor::EndLine Method 9

IReconstructor::EndPage Method 10

IReconstructor::EndWord Method 10

IReconstructor::Progress Method 10

IReconstructor::StartBlock Method 10

IReconstructor::StartLine Method 10

IReconstructor::StartPage Method 10

IReconstructor::StartWord Method 11

IRegion Interface 11

IRegion::bounds Method 11

IRegion::ContentType Method 11

IRegion::Lines Method 12

IRegion::LinesCount Method 12

ITextLine Interface 12

ITextLine::bounds Method 12

ITextLine::Words Method 12

ITextLine::WordsCount Method 13

ITextWord Interface 13

ITextWord::bounds Method 13

ITextWord::Characters Method 13

ITextWord::CharactersCount Method 14

Structs, Records, Enums 14

xpidea_ocr::tagImageOperation Enumeration 14

xpidea_ocr::tagPoint Struct 14

xpidea_ocr::tagRectangle Struct 15

xpidea_ocr::tagRegionContentType Enumeration 15

Files **15**

xpidea_ocr.idl 16

Index **a**

XP Idea OCR SDK v1.0

1 Symbol Reference

1.1 xpidea_ocr Library

XP Idea OCR Library is a COM object for Windows (tm) operating systems.

XP Idea OCR COM Library is an OCR engine compatible with Visual C++, Visual Basic, Delphi, Borland C-builder, .NET and other programming languages/development platforms supporting COM technology. XP Idea OCR engine is completely implemented in Delphi 5 and does not depend on any 3rd party products and libraries.

You can evaluate the XP Idea OCR using a front end application included with this package. This is a first public release of the library, so you'll probably find a lot of issues. We strongly encourage you to submit all issues to **support@xpidea.com**. Please do not hesitate to ask Support for extension of initial 30 day trial period. This software includes automatic update feature, so you'll receive updates as soon as they available.

As a developer you can evaluate examples provided with this package and give us a feedback on the design, performance and features.

XP Idea OCR SDK will be available for purchase with licensing to OEMs on a royalty or flat annual fee basis.

Source code in Delphi 5 will be sold separately only with SDK purchase. (There will be a written contract and special license agreement with each customer purchasing the source code.)

Please address your pricing and purchase requests to **support@xpidea.com**

Classes

Class	Description
OCREngine (↗ see page 2)	CoClass of the object implementing IOCREngine (↗ see page 5).

Interfaces

Interface	Description
ICharacter (↗ see page 3)	Interface to a particular letter (character) in the word. (Implementation to be defined.)
IImage (↗ see page 3)	Interface to the document image. Provides various properties and methods allowing pre-process the image before recognition.
IOCREngine (↗ see page 5)	An interface to OCR engine. This interface allows to initialize OCR engine with specific language file and to perform recognition of the image.

IOCREngineEvents (see page 7)	An OCR engine dispatch interface for the events.
IPage (see page 7)	Interface to a recognized document. This interface allows to access all regions on the page. Use RegionsCount (see page 8) method to find -out number of regions on the page. Use Region (see page 8) method to retrieve particular region by its index.
IReconstructor (see page 8)	A callback interface that used by the recognition engine to reconstruct a text from the image. Members of this interface are called during the recognition process. They provide the consumer with recognition results and provide some supporting information. Please see sample application for details.
IRegion (see page 11)	Provides access to the properties of particular region of the document as well as to text lines (in case if Content Type is a text).
ITextLine (see page 12)	Provides access to the properties of particular line of text in the region.
ITextWord (see page 13)	Provides access to the properties of the particular word in the line of text.

Structs, Records, Enums

Struct, Record, Enum	Description
tagImageOperation (see page 14)	Defines a set of operations that could be performed on the image, before it recognized.
tagPoint (see page 14)	Represents a graphical point.
tagRectangle (see page 15)	Represents graphical rectangle defined by Top Left and Bottom Right points.
tagRegionContentType (see page 15)	Defines the type of the document region - graphics or text.

1.1.1 Classes

1.1.1.1 OCREngine CoClass

CoClass of the object implementing IOCREngine (see page 5).

Class Hierarchy

File

xpidea_ocr.idl (see page 16)

Description

The CoClass describes a unique COM object which implements one or more interfaces and specifies which implemented interface is the default for the object, and optionally, which dispinterface is the default source for events.

A COM object is an instance of a CoClass, which is a class that implements one or more COM interfaces. The COM object provides the services as defined by its CoClass interfaces.

CoClasses are instantiated by a special type of object called a class factory. Whenever an object’s services are requested by a client, a class factory creates and registers an object instance for that particular client. If another client requests the object’s services, the class factory creates another object instance to service the second client.

A CoClass must have a class factory and a class identifier (CLSID) so that its COM object can be instantiated externally, that is, from another module. Using these unique identifiers for CoClasses means that they can be updated whenever new interfaces are implemented in their class. A new interface can modify or add methods without affecting older versions, which is a common problem when using DLLs.

IDL

```
[ uuid(70F6A2C2-7749-41E0-9965-25BA6822A838), version(1.0), helpstring("OCREngine Object") ]
coclass OCREngine;
```

1.1.2 Interfaces

1.1.2.1 ICharacter Interface

Interface to a particular letter (character) in the word. (Implementation to be defined.)

Class Hierarchy

File

xpidea_ocr.idl (see page 16)

Description

Used to access a character, determine its case and type.

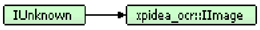
IDL

```
[ uuid(D0B34DB0-65D3-4FAF-8CF2-A58BABB215CC), version(1.0) ]
interface ICharacter : IUnknown;
```

1.1.2.2 IImage Interface

Interface to the document image. Provides various properties and methods allowing pre-process the image before recognition.

Class Hierarchy



File

xpidea_ocr.idl (see page 16)

Description

This interface is one of parameters of OnBeforeRecognition event. It allows to detect image noise level, skew angle and other properties of the document image, it also allowes to perform some image processing operations, such as noise removal and skew correction.

This interface also can be used to define recognition regions, overriding auto-detect functionality of OCR engine.

Members

Methods

Method	Description
◆ AddRegion (see page 4)	Overrides OCR engine's region auto-detection functionality, by forcing it to recognize only specified areas of the image. For instance. You can force the OCR engine to recognize only area of an invoice containing customer's name and address.
◆ NoiseLevel (see page 4)	Detects noise level (in dB) of the image.
◆ PerformOperation (see page 4)	(NOT YET IMPLEMENTED) Performs noise removal, deskew and other operations on the image, before the image is recognized.
◆ SkewAngleDetected (see page 4)	Performs skew angle detection and returns detected skew angle.

Legend

◆	Method

IDL

```
[ uuid(BCEF9328-F03E-4A31-88FB-FC6D3401FD8E), version(1.0) ]
interface IImage : IUnknown;
```

1.1.2.2.1 IImage::AddRegion Method

Overrides OCR engine's region auto-detection functionality, by forcing it to recognize only specified areas of the image.

For instance. You can force the OCR engine to recognize only area of an invoice containing customer's name and address.

Parameters

Parameters	Description
`[in] Rectangle bounds`	Rectangle (in pixels) specifying region bounds.

Description

(NOT YET IMPLEMENTED)

To define a region, pass Rectangle structure (in pixels) to this method. To define multiple regions call this method multiple times.

IDL

```
[id(0x00000006)]
HRESULT _stdcall AddRegion([in] Rectangle bounds);
```

1.1.2.2.2 IImage::NoiseLevel Method

Detects noise level (in dB) of the image.

Returns

Image noise level in dB.

Description

(NOT YET IMPLEMENTED)

Used to determine how noisy the image is.

IDL

```
[propget, id(0x00000005)]
HRESULT _stdcall NoiseLevel([out, retval] long * Value);
```

1.1.2.2.3 IImage::PerformOperation Method

(NOT YET IMPLEMENTED)

Performs noise removal, deskew and other operations on the image, before the image is recognized.

Parameters

Parameters	Description
`[in] ImageOperation Operation`	Operation to perform.
`[in] VARIANT OperationParam1`	Operation parameters. Please see appropriate ImageOperation for details.
`[in] VARIANT OperationParam2`	Operation parameters. Please see appropriate ImageOperation for details.

IDL

```
[id(0x00000004)]
HRESULT _stdcall PerformOperation([in] ImageOperation Operation, [in] VARIANT
OperationParam1, [in] VARIANT OperationParam2);
```

1.1.2.2.4 IImage::SkewAngleDetected Method

Performs skew angle detection and returns detected skew angle.

Returns

Detected skew angle in degrees.

IDL

```
[propget, id(0x00000003), helpstring("Skew angle degrees. Positive if clockwise.")]
HRESULT _stdcall SkewAngleDetected([out, retval] double * Value);
```

1.1.2.3 IOCREngine Interface

An interface to OCR engine. This interface allows to initialize OCR engine with specific language file and to perform recognition of the image.

Class Hierarchy

IDispatch
xpidea_ocr::IOCREngine

File

xpidea_ocr.idl (see page 16)

Members

Methods

Method	Description
◆ Configure (see page 5)	Provides a way to configure the OCR engine. All sonfigration settings provided as a pair of key-value strings. List of the keys and possible values are presented in the table below.
◆ Init (see page 6)	Initializes the OCR engine with specific Language file. This is the first and only operation that needs to be performed before OCR engine can operate. Please call this method only once per recognition session.
◆ Recognize (see page 6)	Recognizes the given image. During the recognition process various members of IReconstructor (see page 8) interface are called to produce the result.

Legend

◆	Method

IDL

```
[ uuid(365DFDDE-2B08-4110-AA13-961799F6BDB6), version(1.0), helpstring("Dispatch interface
for OCREngine Object"), dual, oleautomation ]
interface IOCREngine : IDispatch;
```

1.1.2.3.1 IOCREngine::Configure Method

Provides a way to configure the OCR engine.

All sonfigration settings provided as a pair of key-value strings.

List of the keys and possible values are presented in the table below.

Key	Possible values
"NOISE LEVEL"	**Default: "2"** Group of N or fewer pixels considered as noise and ignored. Example: "3" - groups of 3 or fewer pixels will be considered as noise and ignored during recognition.
"REGION MERGE THRESHOLD"	**Default: "1.1"** Regulates text oversegmentation. (Prevents single words being recognized as separate text regions)

'DEFAULT IMAGE BINARIZATION METHOD'	**Default: "AVERAGE"** **Values:** 'AVERAGE' 'MAXIMUM ENTROPY' 'OTSU' 'SIMPLE' 'TRIANGLE' 'MINMAX' Defines default image thresholding (binarization) method to produce black and white bitmap. There are some extra settings required for by some thresholders. SIMPLE thresholder needs an integer threshold value (default 127) MINMAX thresholder needs 2 integers: Neighborhood size (3..7) default is 4, and bias value (0..255) default is 0; **Example:** **"SIMPLE,127"** **"MINMAX,4,20"**

IDL

```
[id(0x00000003)]
HRESULT _stdcall Configure([in] BSTR Setting, [in] BSTR Value);
```

1.1.2.3.2 IOCREngine::Init Method

Initializes the OCR engine with specific Language file.

This is the first and only operation that needs to be performed before OCR engine can operate. Please call this method only once per recognition session.

Parameters

Parameters	Description
`[in] BSTR languageFileName`	String containing full path and name of the language file. Example "C:\XPIdea.OCR\English.dat"

IDL

```
[id(0x00000001)]
HRESULT _stdcall Init([in] BSTR languageFileName);
```

1.1.2.3.3 IOCREngine::Recognize Method

Recognizes the given image. During the recognition process various members of IReconstructor (⧉ see page 8) interface are called to produce the result.

Parameters

Parameters	Description
`[in] SAFEARRAY(unsigned char) imageBytes`	Array of bytes that contains the image. (Currently only 24bpp BMP image format is supported)
`[in] IReconstructor * reconstructor`	An IReconstructor (⧉ see page 8) call back interface that used to retrieve recognition results.

IDL

```
[id(0x00000002)]
HRESULT _stdcall Recognize([in] SAFEARRAY(unsigned char) imageBytes, [in] IReconstructor *
reconstructor);
```

1.1.2.4 IOCREngineEvents Interface

An OCR engine dispatch interface for the events.

Class Hierarchy

File

xpidea_ocr.idl (⧉ see page 16)

Members

Methods

Method	Description
◆ OnAfterRecognition (⧉ see page 7)	This event gets fired by recognition engine after page recognition is completed. A page parameter of this event provide access to the structure of recognized document through IPage (⧉ see page 7) inerface.
◆ OnBeforeRecognition (⧉ see page 7)	This event gets fired right before recognition happens. Please use this event to perform any image preprocessing operations, such as skew angle correction and noise removal.

Legend

◆	Method

IDL

```
[ uuid(1DC89BA1-0A91-4BC2-AD91-E70414C290CB), version(1.0), helpstring("Events interface
for OCREngine Object") ]
dispinterface IOCREngineEvents;
```

1.1.2.4.1 IOCREngineEvents::OnAfterRecognition Method

This event gets fired by recognition engine after page recognition is completed. A page parameter of this event provide access to the structure of recognized document through IPage (⧉ see page 7) inerface.

Parameters

Parameters	Description
`[in] IPage * aPage`	IPage (⧉ see page 7) interface to recognized document.

IDL

```
[id(0x00000002)]
HRESULT OnAfterRecognition([in] IPage * aPage);
```

1.1.2.4.2 IOCREngineEvents::OnBeforeRecognition Method

This event gets fired right before recognition happens. Please use this event to perform any image preprocessing operations, such as skew angle correction and noise removal.

Parameters

Parameters	Description
`[in] IImage * args`	IImage (⧉ see page 3) interface of the image to be recognize.

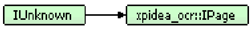
IDL

```
HRESULT OnBeforeRecognition([in] IImage * args);
```

1.1.2.5 IPage Interface

Interface to a recognized document. This interface allows to access all regions on the page. Use RegionsCount (⧉ see page 8) method to find -out number of regions on the page. Use Region (⧉ see page 8) method to retrieve particular region by its index.

Class Hierarchy



File

xpidea_ocr.idl (see page 16)

Members

Methods

Method	Description
◆ Region (see page 8)	Retrieves region IRegion (see page 11) by given index.
◆ RegionsCount (see page 8)	Number of regions detected or defined on the page.

Legend

◆	Method

IDL

```
[ uuid(162B2F74-1639-4E81-9DC9-30A7EE378775), version(1.0) ]
interface IPage : IUnknown;
```

1.1.2.5.1 IPage::Region Method

Retrieves region IRegion (see page 11) by given index.

Parameters

Parameters	Description
`[in] long index`	index of the region to retrieve.
`[out, retval] IRegion ** Value`	IRegion (see page 11) interface of the region with given index.

Returns

IRegion (see page 11) interface with given index.

IDL

```
[propget, id(0x00000003)]
HRESULT _stdcall Region([in] long index, [out, retval] IRegion ** Value);
```

1.1.2.5.2 IPage::RegionsCount Method

Number of regions detected or defined on the page.

IDL

```
[propget, id(0x00000001)]
HRESULT _stdcall RegionsCount([out, retval] long * Value);
```

1.1.2.6 IReconstructor Interface

A callback interface that used by the recognition engine to reconstruct a text from the image. Members of this interface are called during the recognition process. They provide the consumer with recognition results and provide some supporting information. Please see sample application for details.

Class Hierarchy

IUnknown
xpidea_ocr::IReconstructor

File

xpidea_ocr.idl (see page 16)

Members

Methods

Method	Description
◆ Character (⧉ see page 9)	Called by OCR engine to give recognized character(s) to consumer.
◆ EndBlock (⧉ see page 9)	Called after recognition of each region is completed. If you treat each region as a text paragraph, you can use this method to create the end of paragraph mark in recognized text.
◆ EndLine (⧉ see page 9)	Called after recognition of each line of text. You can put caret return/line feed symbols (CRLF) into a recognized text when this method is called.
◆ EndPage (⧉ see page 10)	Called after recognition of image is completed.You can put a page break symbol into the recognized text when this method is called.
◆ EndWord (⧉ see page 10)	Called after recognition of word is completed. You can put a space into the recognized text when this method is called.
◆ Progress (⧉ see page 10)	This method is called by recognition engine, to notify client on the progress of recognition.
◆ StartBlock (⧉ see page 10)	Called before recognition of the image region begins. Provides consumer with the information about the region to be recognized. If ContentType of the region is graphic - you can use bounds parameter as a clip rectangle to insert the image into recognized text.
◆ StartLine (⧉ see page 10)	Called before recognition of each text line. Provides consumer with the information about the line location.
◆ StartPage (⧉ see page 10)	Called by OCR engine right after page segmentation and before character recognition. Parameters are bounds of the page and IPage (⧉ see page 7) interface to a page being recognized.
◆ StartWord (⧉ see page 11)	Called before recognition of each word. Provides consumer with the information about the word location.

Legend

◆	Method

IDL

```
[ uuid(BD40B2FC-BA70-406F-B1E2-199DE78C6AE5), version(1.0) ]
interface IReconstructor : IUnknown;
```

1.1.2.6.1 IReconstructor::Character Method

Called by OCR engine to give recognized character(s) to consumer.

Parameters

Parameters	Description
`[in] BSTR ch`	Character that was recognize.

IDL

```
[id(0x00000009)]
HRESULT _stdcall Character([in] BSTR ch);
```

1.1.2.6.2 IReconstructor::EndBlock Method

Called after recognition of each region is completed. If you treat each region as a text paragraph, you can use this method to create the end of paragraph mark in recognized text.

IDL

```
[id(0x00000004)]
HRESULT _stdcall EndBlock(void);
```

1.1.2.6.3 IReconstructor::EndLine Method

Called after recognition of each line of text. You can put caret return/line feed symbols (CRLF) into a recognized text when this method is called.

IDL

```
[id(0x00000006)]
HRESULT _stdcall EndLine(void);
```

1.1.2.6.4 IReconstructor::EndPage Method

Called after recognition of image is completed.You can put a page break symbol into the recognized text when this method is called.

IDL

```
[id(0x00000002)]
HRESULT _stdcall EndPage(void);
```

1.1.2.6.5 IReconstructor::EndWord Method

Called after recognition of word is completed. You can put a space into the recognized text when this method is called.

IDL

```
[id(0x00000008)]
HRESULT _stdcall EndWord(void);
```

1.1.2.6.6 IReconstructor::Progress Method

This method is called by recognition engine, to notify client on the progress of recognition.

IDL

```
[id(0x0000000A)]
HRESULT _stdcall Progress([in] long Percent, [in] BSTR Status);
```

1.1.2.6.7 IReconstructor::StartBlock Method

Called before recognition of the image region begins. Provides consumer with the information about the region to be recognized.

If ContentType of the region is graphic - you can use bounds parameter as a clip rectangle to insert the image into recognized text.

Parameters

Parameters	Description
`[in] Rectangle bounds`	Region bounds in pixels.
`[in] RegionContentType ContentType`	Specifies type of the region (graphic or text).
`[in] Font * font`	Detected font that used in the region. (In this version of the software just estimates the font size. Does not detect the actual type face.)

IDL

```
[id(0x00000003)]
HRESULT _stdcall StartBlock([in] Rectangle bounds, [in] RegionContentType ContentType, [in]
Font * font);
```

1.1.2.6.8 IReconstructor::StartLine Method

Called before recognition of each text line. Provides consumer with the information about the line location.

IDL

```
[id(0x00000005)]
HRESULT _stdcall StartLine([in] Rectangle bounds);
```

1.1.2.6.9 IReconstructor::StartPage Method

Called by OCR engine right after page segmentation and before character recognition.

Parameters are bounds of the page and IPage (see page 7) interface to a page being recognized.

IDL

```
[id(0x00000001)]
HRESULT _stdcall StartPage([in] Rectangle bounds, [in] Rectangle margins, [in] IPage *
aPage);
```

1.1.2.6.10 IReconstructor::StartWord Method

Called before recognition of each word. Provides consumer with the information about the word location.

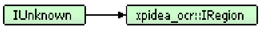
IDL

```
[id(0x00000007)]
HRESULT _stdcall StartWord([in] Rectangle bounds);
```

1.1.2.7 IRegion Interface

Provides access to the properties of particular region of the document as well as to text lines (in case if Content Type is a text).

Class Hierarchy



File

xpidea_ocr.idl (see page 16)

Members

Methods

Method	Description
◆ bounds (see page 11)	Bounds of the region expressed in pixels.
◆ ContentType (see page 11)	Type of the region - Text or Graphics. If content type is Text - you can access text lines through Lines (see page 12) and LinesCount (see page 12) methods. For graphic regions LinesCount (see page 12) always returns 0.
◆ Lines (see page 12)	Provides indexed access to ITextLine (see page 12) member of the region.
◆ LinesCount (see page 12)	Returns number of text lines in the region. Used for indexed access to the text lines.

Legend

◆	Method

IDL

```
[ uuid(45F4CE79-AE7C-4666-A14B-F977DA4F0307), version(1.0) ]
interface IRegion : IUnknown;
```

1.1.2.7.1 IRegion::bounds Method

Bounds of the region expressed in pixels.

IDL

```
[propget, id(0x00000004)]
HRESULT _stdcall bounds([out, retval] Rectangle * Value);
```

1.1.2.7.2 IRegion::ContentType Method

Type of the region - Text or Graphics. If content type is Text - you can access text lines through Lines (see page 12) and LinesCount (see page 12) methods. For graphic regions LinesCount (see page 12) always returns 0.

IDL

```
[propget, id(0x00000001)]
HRESULT _stdcall ContentType([out, retval] RegionContentType * Value);
```

1.1.2.7.3 IRegion::Lines Method

Provides indexed access to ITextLine (see page 12) member of the region.

Parameters

Parameters	Description
`[in] long index`	Line index. Please use LinesCount (see page 12) to determine number of lines in the region.
`[out, retval] ITextLine ** Value`	ITextLine (see page 12) interface to a text line.

IDL

```
[propget, id(0x00000003)]
HRESULT _stdcall Lines([in] long index, [out, retval] ITextLine ** Value);
```

1.1.2.7.4 IRegion::LinesCount Method

Returns number of text lines in the region. Used for indexed access to the text lines.

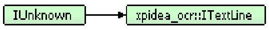
IDL

```
[propget, id(0x00000002)]
HRESULT _stdcall LinesCount([out, retval] long * Value);
```

1.1.2.8 ITextLine Interface

Provides access to the properties of particular line of text in the region.

Class Hierarchy



File

xpidea_ocr.idl (see page 16)

Members

Methods

Method	Description
◆ bounds (see page 12)	Bounds of the text line expressed in pixels.
◆ Words (see page 12)	Provides indexed access to ITextWord (see page 13) member of the text line.
◆ WordsCount (see page 13)	Returns detected number of words in the line of text. Used for indexed access to the words.

Legend

◆	Method

IDL

```
[ uuid(8AD37025-7DFF-4C8F-A460-D325547CED6C), version(1.0) ]
interface ITextLine : IUnknown;
```

1.1.2.8.1 ITextLine::bounds Method

Bounds of the text line expressed in pixels.

IDL

```
[propget, id(0x00000003)]
HRESULT _stdcall bounds([out, retval] Rectangle * Value);
```

1.1.2.8.2 ITextLine::Words Method

Provides indexed access to ITextWord (see page 13) member of the text line.

Parameters

Parameters	Description
`[in] long index`	index of the word.
`[out, retval] ITextWord ** Value`	ITextWord (↗ see page 13) interface.

IDL

```
[propget, id(0x00000002)]
HRESULT _stdcall Words([in] long index, [out, retval] ITextWord ** Value);
```

1.1.2.8.3 ITextLine::WordsCount Method

Returns detected number of words in the line of text. Used for indexed access to the words.

IDL

```
[propget, id(0x00000001)]
HRESULT _stdcall WordsCount([out, retval] long * Value);
```

1.1.2.9 ITextWord Interface

Provides access to the properties of the particular word in the line of text.

Class Hierarchy

IUnknown
xpidea_ocr::ITextWord

File

xpidea_ocr.idl (↗ see page 16)

Members

Methods

Method	Description
◆ bounds (↗ see page 13)	Word location and size on the image expressed in pixels.
◆ Characters (↗ see page 13)	Provides indexed access to a character(letter) ICharacter (↗ see page 3) in the word.
◆ CharactersCount (↗ see page 14)	Returns detected number of letters in the word. Used for indexed access to the letters.

Legend

◆	Method

IDL

```
[ uuid(B929589E-EC9A-417E-8597-1FFE79C3CB31), version(1.0) ]
interface ITextWord : IUnknown;
```

1.1.2.9.1 ITextWord::bounds Method

Word location and size on the image expressed in pixels.

IDL

```
[propget, id(0x00000003)]
HRESULT _stdcall bounds([out, retval] Rectangle * Value);
```

1.1.2.9.2 ITextWord::Characters Method

Provides indexed access to a character(letter) ICharacter (↗ see page 3) in the word.

Parameters

Parameters	Description
`[in] long index`	Index of the letter in the word.
`[out, retval] ICharacter ** Value`	ICharacter (↗ see page 3) interface.

IDL

```
[propget, id(0x00000002)]
HRESULT _stdcall Characters([in] long index, [out, retval] ICharacter ** Value);
```

1.1.2.9.3 ITextWord::CharactersCount Method

Returns detected number of letters in the word. Used for indexed access to the letters.

IDL

```
[propget, id(0x00000001)]
HRESULT _stdcall CharactersCount([out, retval] long * Value);
```

1.1.3 Structs, Records, Enums

1.1.3.1 xpidea_ocr::tagImageOperation Enumeration

Defines a set of operations that could be performed on the image, before it recognized.

File

xpidea_ocr.idl (↗ see page 16)

Members

Members	Description
`Rotate = 0`	Rotates the image on specified number of degrees. Used to deskew the image by angle opposite to detected by the SkewAngleDetected method. First parameter of PerformOperation method is angle in degrees (of double type).
`RemoveNoise = 1`	Tries to reduce the noise in the image. Should be applied to a noisy images. No parameters needed for PerformOperation method.
`AutoContrast = 2`	Automatically adjusts contrast of the image. Should be applied to a low contrast or very dark images. No parameters needed for PerformOperation method.
`Threshold = 3`	Thresholds (creates black and white) image with the given threshold method and optional threshold value. Currently use Otsu automatic global thresholding method and no parameters needed for PerformOperation method.

IDL

```
[ uuid(04F53DBB-A2FE-4C6F-8D3C-4755547C7A7A), version(1.0) ]
enum tagImageOperation {
  Rotate = 0,
  RemoveNoise = 1,
  AutoContrast = 2,
  Threshold = 3
};
```

1.1.3.2 xpidea_ocr::tagPoint Struct

Represents a graphical point.

File

xpidea_ocr.idl (↗ see page 16)

Members

Members	Description
`long X;`	X-coordinate.
`long Y;`	Y-coordinate.

IDL

```
[ uuid(E56DC05D-5BD4-456F-B0A9-09760C246846), version(1.0) ]
struct tagPoint {
  long X;
  long Y;
};
```

1.1.3.3 xpidea_ocr::tagRectangle Struct

Represents graphical rectangle defined by Top Left and Bottom Right points.

File

xpidea_ocr.idl (↗ see page 16)

Members

Members	Description
`Point TopLeft;`	Top left (location) point of the rectangle.
`Point BottomRight;`	Bottom right corner location of the rectangle.

IDL

```
[ uuid(87C6CD2D-C8C2-4ACE-926B-1B9B40F1A3F0), version(1.0) ]
struct tagRectangle {
  Point TopLeft;
  Point BottomRight;
};
```

1.1.3.4 xpidea_ocr::tagRegionContentType Enumeration

Defines the type of the document region - graphics or text.

File

xpidea_ocr.idl (↗ see page 16)

Members

Members	Description
`Text = 0`	Region with this type contains text.
`Graphic = 1`	Region with this type contains a graphic object.

IDL

```
[ uuid(FF5D8D46-458B-4D43-97A8-3CCAAAAC732F), version(1.0) ]
enum tagRegionContentType {
  Text = 0,
  Graphic = 1
};
```

1.2 Files

1.2.1 xpidea_ocr.idl

Interface Definition (IDL) File

Libraries

Library	Description
xpidea_ocr (↗ see page 1)	XP Idea OCR Library is a COM object for Windows (tm) operating systems. XP Idea OCR COM Library is an OCR engine compatible with Visual C++, Visual Basic, Delphi, Borland C-builder, .NET and other programming languages/development platforms supporting COM technology. XP Idea OCR engine is completely implemented in Delphi 5 and does not depend on any 3rd party products and libraries. You can evaluate the XP Idea OCR using a front end application included with this package. This is a first public release of the library, so you'll probably find a lot of issues. We strongly encourage you to submit all issues to **support@xpidea.com**. Please do not hesitate to ask Support for extension of initial 30 day trial period. This software includes automatic update feature, so you'll receive updates as soon as they available. As a developer you can evaluate examples provided with this package and give us a feedback on the design, performance and features. XP Idea OCR SDK will be available for purchase with licensing to OEMs on a royalty or flat annual fee basis. Source code in Delphi 5 will be sold separately only with SDK purchase. (There will be a written contract and special license agreement with each customer purchasing the source code.) Please address your pricing and purchase requests to **support@xpidea.com**

Index

I

ICharacter interface 3

IImage interface 3

AddRegion 4

NoiseLevel 4

PerformOperation 4

SkewAngleDetected 4

IOCREngine interface 5

Configure 5

Init 6

Recognize 6

IOCREngineEvents interface 7

OnAfterRecognition 7

OnBeforeRecognition 7

IPage interface 7

Region 8

RegionsCount 8

IReconstructor interface 8

Character 9

EndBlock 9

EndLine 9

EndPage 10

EndWord 10

Progress 10

StartBlock 10

StartLine 10

StartPage 10

StartWord 11

IRegion interface 11

bounds 11

ContentType 11

Lines 12

LinesCount 12

ITextLine interface 12

bounds 12

Words 12

WordsCount 13

ITextWord interface 13

bounds 13

Characters 13

CharactersCount 14

O

OCREngine coclass 2

X

xpidea_ocr library 1

xpidea_ocr.idl 16

xpidea_ocr::tagImageOperation enumeration 14

xpidea_ocr::tagPoint struct 14

xpidea_ocr::tagRectangle struct 15

xpidea_ocr::tagRegionContentType enumeration 15