**MeOCRLib**.**Puma DLL API Reference**

This is the interface for the meocr100.dll. Included are four example projects. Two are written in C# the other two in VB.Net. Please refer to the examples along with this document to implement the library in your application. As you can see in the sample code below MeOCRLib has been designed to be as simple as possible to use. Both examples below are fully functional programs. Most of the code is boiler plate. The actual code that calls MeOCRLib is contained in just a few calls:

**C#:**

Puma ocrPuma = new Puma(); //Declare a new instance of engine

ocrPuma.Init(); //Initialize the engine

rText = ocrPuma.Recognize(imgPuma); //Recognize the image and return the text output

**VB.NET:**

Dim ocrPuma As New MeOCRLib.Puma //Declare a new instance of engine

ocrPuma.Init() //Initialize the engine

rText = ocrPuma.Recognize(imgPuma) //Recognize the image and return the text output

Below are two of the examples provided:

**C# Example:**

namespace CSharpEx\_01

{

public partial class frmCSharpEx\_01 : Form

{

public frmCSharpEx\_01()

{

InitializeComponent();

}

private void frmCSharpEx\_01\_Load(object sender, EventArgs e)

{

//Declare a new instance og engine

Puma ocrPuma = new Puma();

//Declare a few needed varaibles

Image imgPuma;

string rText;

string FileName;

FileName = System.IO.Path.GetDirectoryName(Environment.GetCommandLineArgs()[0]) + @"\Sample\_02.tif";

//Initialize the engine

ocrPuma.Init();

//Load the image

imgPuma = Image.FromFile(FileName);

//Set the language

ocrPuma.Language = 0; // 0 for English

//Set output format

ocrPuma.OutputFormat = 2; // Formatted text

//Recognize the image

rText = ocrPuma.Recognize(imgPuma);

//Close the engine

ocrPuma.Close();

}

}

}

**VB.NET Example:**

Public Class VBasicEx\_01

Private Sub VBasicEx\_01\_Load(sender As Object, e As EventArgs) Handles MyBase.Load

'Declare a new instance og engine

Dim ocrPuma As New MeOCRLib.Puma

'Declare a few needed varaibles

Dim imgPuma As Image

Dim rText As String

Dim FileName As String

FileName = System.IO.Path.GetDirectoryName(Environment.GetCommandLineArgs()(0)) & "\sample\_02.tif"

'Initialize the engine

ocrPuma.Init()

'Load the image

imgPuma = Image.FromFile(FileName)

'Set the language

ocrPuma.Language = 0 ' 0 for English

'Set output format

ocrPuma.OutputFormat = 2 ' Formatted text

'recognize the image

rText = ocrPuma.Recognize(imgPuma)

'Close the engine

ocrPuma.Close()

End Sub

End Class

**Class Constructor:**

public **Puma**()

**Description:**

Used to create a new instance of Puma OCR Engine.

**Use:**

**Puma** *ocrPuma* = new **Puma**();

**Class Members:**

public **bool** **Init**(**MeOCRLib.Puma.ProgressStepPtr** *pStep = null*)

**Description:**

Used to initialize a new instance of Puma OCR Engine. Must be called right after creation new

instance is created.

**Parameters:**

***pStep*:** is an optional parameter and is the name of your callback delegate function for displaying the OCR progress. See documentation and examples of **ProgressStepPtr** below for details.

**Use:**

*ocrPuma*.Init(*pStep*);

public **void** **Close**()

**Description:**

Used to close an instance of Puma OCR Engine. Should be called at program exit to clean up.

**Use:**

*ocrPuma*.Close();

public **string** **Recognize**(**System.Drawing.Image** *SrcImg***,** [**System.Drawing.Rectangle** *sRect* = null])

**Description:**

Used to recognize an image.

**Parameters:**

***SrcBmp*:** is a **System.Drawing.Image** object of the image.

***sRect****:* is a **System.Drawing.Rectangle** object containing the coordinates of a zone to recognize rather than the whole image. This parameter is optional. If it is not specified then the entire image is recognized.

**Use:**

**string** *txt* = *ocrPuma*,Recognize(*imgPuma*);

public **string** **Recognize**(**System.Drawing.Bitmap** *SrcBmp***,** [**System.Drawing.Rectangle** *sRect* = null])

**Description:**

Used to recognize an image.

**Parameters:**

***SrcBmp*:** is a **System.Drawing.Bitmap** object of the image.

***sRect****:* is a **System.Drawing.Rectangle** object containing the coordinates of a zone to recognize rather than the whole image. This parameter is optional. If it is not specified then the entire image is recognized.

**Use:**

**string** *txt* = *ocrPuma*,Recognize(*bmpPuma*);

public **string** **LanguageString**(**short** *index*)

**Description:**

Used to get the name of the language who is assigned to the numeric value in *index*.

**Parameters:**

***index*:** is the numeric index of the language.

**Use:**

**string** *languaget* = *ocrPuma*,LanguageString(*index*);

**Class Properties:**

public **short** **Language** { set; get; }

**Description:**

Used to get or set the recognition language. Below is a list of values for specific languages supported.

**Languages codes:**

ENGLISH 0

GERMAN 1

FRENCH 2

RUSSIAN 3

SWEDISH 4

SPANISH 5

ITALIAN 6

RUS\_ENG 7

UKRAINIAN 8

SERBIAN 9

CROATIAN 10

POLISH 11

DANISH 12

PORTUGUESE 13

DUTCH 14

RESERVED 15

RESERVED 16

RESERVED 17

RESERVED 18

CZECH 19

ROMANIAN 20

HUNGARIAN 21

BULGARIAN 22

SLOVENIAN 23

LATVIAN 24

LITHUANIAN 25

ESTONIAN 26

TURKISH 27

**Use:**

*ocrPuma*.Language = 0;

public **short** **OutputFormat** { set; get; }

**Description:**

Used to get or set the recognized text output format.

**Output format codes:**

RAW 0

PLAIN 1

FORMATTED 2

RTF 3

**Use:**

*ocrPuma*.OutputFormat = 0;

public **short** **UnrecogChar** { set; get; }

**Description:**

Used to get or set the character ASCII code to be used when a character is unrecognized.

**Use:**

*ocrPuma*.Language = 128;

**Class Delegate:**

public delegate **void** **ProgressStepPtr**(**int** *step***, string** *stepName***, int** *lPercent*)

**Description:**

The callback delegate used as an event to update the OCR progress.

**Use:**

*ocrPuma*.Init(*pStep*);

void *pStep*(int step, string stepName, int lPercent)

{

// Update the progress

}

**Format Descriptions:**

**Plain:** This format outputs plain text with no structure or formatting.

**Formatted:** This format outputs text with formatting preserved.

**RTF:** This format outputs text with formatting preserved and rtf codes.

**Raw:** This format species the coordinates and attributes for each character. Refer to **Figure 1** below as you read this section.

Each character's information is delimited by a carriage return **(\r or vbCr)**.

Each property member of a character is delimited by ASCII code (01).

The first item in the format data is the **number of records** in the data. in the example below there are 10 character records in the data. Each record consists of 8 values as follows:

1- Character value

2 - Left position in pixels

3 - Top position in pixels

4 - Right position in pixels

5 - Bottom position in pixels

6 - Internal Value

7 - Internal Value

8 - Internal Value

If the "Character value" is a **{32)**  it is the end of a word.

If the "Character value" is a **{02)**  it is the end of a line fragment. A line fragment is a group of words on a line separated by single spaces. In the line below there are 3 line fragments. Notice there is more than one space between the line fragments.

**This is line fragment 1 This is line fragment 2 This is line fragment 3**

**Figure 1.**

**10(\r or vbCr) <--------------------------------- Specifies the number of records to follow.**

**E**(01)**665**(01)**201**(01)**697**(01)**236**(01)**22**(01)**4**(01)**1(\r or vbCr)**

**n**(01)**700**(01)**209**(01)**733**(01)**235**(01)**22**(01)**4**(01)**1(\r or vbCr)**

**d**(01)**736**(01)**198**(01)**771**(01)**236**(01)**22**(01)**4**(01)**1(\r or vbCr)**

**{32)** (01)**0**(01)**0**(01)**0**(01)**0**(01)**0**(01)**0**(01)**0(\r or vbCr)**

**h**(01)**771**(01)**209**(01)**802**(01)**236**(01)**22**(01)**4**(01)**1(\r or vbCr)**

**o** (01)**807**(01)**209**(01)**838**(01)**235**(01)**22**(01)**4**(01)**1(\r or vbCr)**

**r**(01)**843**(01)**209**(01)**871**(01)**236**(01)**22**(01)**4**(01)**1(\r or vbCr)**

**s**(01)**876**(01)**208**(01)**907**(01)**235**(01)**22**(01)**4**(01)**1(\r or vbCr)**

**e**(01)**908**(01)**208**(01)**946**(01)**234**(01)**22**(01)**4**(01)**1(\r or vbCr)**

**{32)** (01)**0**(01)**0**(01)**0**(01)**0**(01)**0**(01)**0**(01)**0(\r or vbCr)**

**{02)** (01)**0**(01)**0**(01)**0**(01)**0**(01)**0**(01)**0**(01)**0(\r or vbCr)**

Here is some sample code for parsing the Raw Format. Thus code can also be found in the example programs provided:

**C# Example:**

//-----------------------------------------------------------

//Demonstrates parsing of Raw Format

//-----------------------------------------------------------

void ParseResults(string chars)

{

int numItems = -1;

if (chars != "")

{

string[] charToks = chars.Split('\r');

foreach (string Tok in charToks)

{

if (numItems == -1)

{

numItems = Convert.ToInt32(Tok);

}

else

{

string[] TokData = Tok.Split('\x01');

if (TokData[0] != "")

{

switch (TokData[0][0])

{

case ' ': //Specifies the end of a word.

break;

case '\x02': //Specifies the end of a line fragment.

break;

default: //OCR character

char OCRChar = TokData[0][0];

int Left = Convert.ToInt32(TokData[1]);

int Top = Convert.ToInt32(TokData[2]);

int Right = Convert.ToInt32(TokData[3]);

int Bottom = Convert.ToInt32(TokData[4]);

break;

}//switch (TokData[0][0])

}//if (TokData[0] != "")

}//if (numItems == -1)

}//foreach (string Tok in charToks)

}//if (chars != "")

}

**VB.NET Example:**

'-----------------------------------------------------------

'Demonstrates parsing of Raw Format

'-----------------------------------------------------------

Private Sub ParseResults(chars As String)

Dim numItems As Integer = -1

If chars <> "" Then

Dim charToks As String() = chars.Split(ControlChars.Cr)

For Each Tok As String In charToks

If numItems = -1 Then

numItems = Convert.ToInt32(Tok)

Else

Dim TokData As String() = Tok.Split(ChrW(1))

If TokData(0) <> "" Then

Select Case TokData(0)(0)

Case " "

'Specifies the end of a word.

Exit Select

Case ChrW(2)

'Specifies the end of a line fragment.

Exit Select

Case Else

'OCR character

Dim OCRChar As String = TokData(0)(0)

Dim Left As Integer = Convert.ToInt32(TokData(1))

Dim Top As Integer = Convert.ToInt32(TokData(2))

Dim Right As Integer = Convert.ToInt32(TokData(3))

Dim Bottom As Integer = Convert.ToInt32(TokData(4))

Exit Select

'switch (TokData[0][0])

End Select

'if (TokData[0] != "")

End If

'if (numItems == -1)

End If

'foreach (string Tok in charToks)

Next

End If

'if (chars != "")

End Sub

For further information contact us at support@meocr.com or visit our website www.meocr.com