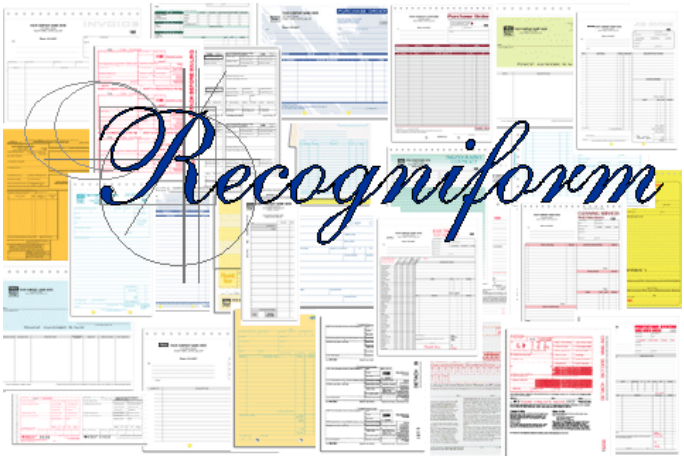


Software Development Kit

Deskew

Recogniform Technologies SpA

HOW TO CONTACT US

Recogniform Technologies SpA
Contrada Concistocchi
87036 Rende (CS), Italy

Phone : +39 0984 404174

Fax : +39 0984 830299

Internet : www.recogniform.com

E-Mail : info@recogniform.com

Table of contents

Introduction 5

- Copyright 5
- License 5
- Overview 5

Usage 7

- Visual C++ 7
- C# 7
- Visul Basic .NET 7
- Visul Basic 7
- Java 7
- Delphi 7

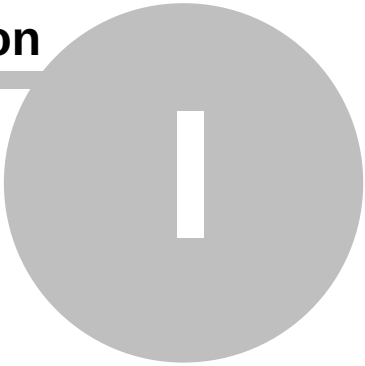
API References 9

- DESKEW_Init 9
- DESKEW_Done 10
- DESKEW_DetectAndCorrect 11
- DESKEW_DetectUsingBlackBorders 12
- DESKEW_Detect 14
- DESKEW_Correct 15
- DESKEW_CorrectEx 17
- LoadDeskewLibrary 18
- FreeDeskewLibrary 19

Sample 21

- Code Sample 21

Introduction



1 Introduction

1.1 Copyright

The software and the documentation are property of:

Recogniform Technologies SpA
Contrada Concistocchi
87036 Rende (CS)
Italy
www.recogniform.com
info@recogniform.com

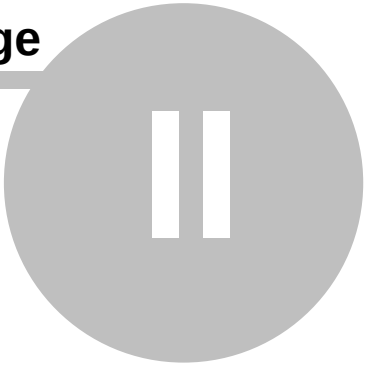
1.2 License

1.3 Overview

The library allows to detect and recognize the skew on images acquired using a scanner or a fax.

The deskewing process is fast and accurate.

Usage



2 Usage

2.1 Visual C++

You have to include the RECODESKEWAPI.C in your program (see sample application). Before to execute your application make sure the *RECODESKEW.DLL* is available in your same .exe directory or in windows\system directory.

2.2 C#

You have to include the RECODESKEWAPI.CS in your program (see sample application). Before to execute your application make sure the *RECODESKEW.DLL* is available in your same .exe directory or in windows\system directory.

2.3 Visul Basic .NET

You have to include the RECODESKEWAPI.VB in your program (see sample application). Before to execute your application make sure the *RECODESKEW.DLL* is available in your same .exe directory or in windows\system directory.

2.4 Visul Basic

You have to include the RECODESKEWAPI.BAS in your program (see sample application). Before to execute your application make sure the *RECODESKEW.DLL* is available in your same .exe directory or in windows\system directory.

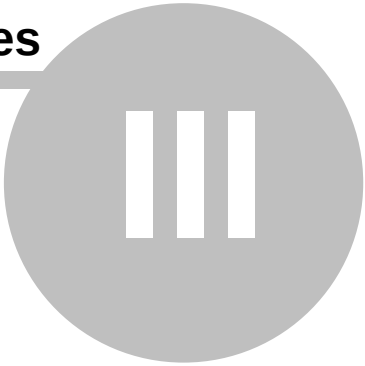
2.5 Java

You have to use 32 bit JVM and you have to include the RECODESKEWAPI.JAVA in your program. Before to execute your application make sure the *RECODESKEW.DLL* is available in your same .jar directory.

2.6 Delphi

You have to include the RECODESKEWAPI.PAS in your program (see sample application). Before to execute your application make sure the *RECODESKEW.DLL* is available in your same .exe directory or in windows\system directory.

API References



3 API References

3.1 DESKEW_Init

C/C++ Declaration

```
__stdcall long DESKEW_Init(char* Name, char* Key);
```

C# Declaration

```
int DESKEW_Init(string Name, string Key);
```

Visual Basic Declaration

```
Function DESKEW_Init(ByVal Name As String, ByVal
Key As String) As Integer
```

Visual Basic .NET Declaration

```
Function DESKEW_Init(ByVal Name As String, ByVal
Key As String) As Integer
```

Delphi Declaration

```
function DESKEW_Init(Company:PAnsiChar;
LicenseKey:PAnsiChar):Integer; stdcall;
```

Java Declaration

```
int DESKEW_Init(String Company,String LicenseKey);
```

Description

This is the first function to call: initialize the library and returns a session handle to use in next calls. When you buy the library you receive an "user" and a "password" string necessary to initialize the library in normal mode: without this value or with wrang values the library is initialized in evaluation mode. The evaluation mode works exactly as normal mode but some time, when you call the deskew function, is displayed a warning dialog box remembering the evaluation state: you can close it and continue to work with no problems.

Parameters

User (in) - then user name string

Passwors (in) - then password string

Return values

The session handle if the library is initialized, 0 otherwise.

Note

The “status” string, shown in the windows dialog, is built by 1 and 0 meaning OK or KO for this checking in the order from left to right:

- SDK initialized with valid user/key
- License file integrity
- License file matching sdk version
- Computer verification done or not required
- Dongle verification done or not required
- Date expiration done or not required
- SDK option used unlocked

3.2 DESKEW_Done

C/C++ Declaration

```
__stdcall void DESKEW_Done(long SessionHandle);
```

C# Declaration

```
void DESKEW_Done(int SessionHandle);
```

Visual Basic Declaration

```
Sub DESKEW_Done(ByVal Session As Integer)
```

Visual Basic .NET Declaration

```
Sub DESKEW_Done(ByVal Session As Integer)
```

Delphi Declaration

```
procedure DESKEW_Done(SessionHandle:Integer);
stdcall;
```

Java Declaration

```
void DESKEW_Done(int SessionHandle);
```

Description

This is the last function to call when you don't need more services from the library: deinitialize the library and free all used resources.

Parameters

SessionHandle (in) - the session handle to free

Return values

n/a

3.3 DESKEW_DetectAndCorrect

C/C++ Declaration

```
__stdcall double DESKEW_DetectAndCorrect(long
SessionHandle, long DIBIn);
```

C# Declaration

```
double DESKEW_DetectAndCorrect(int SessionHandle,
int DIBIn);
```

Visual Basic Declaration

```
Function DESKEW_DetectAndCorrect(ByVal Session As
Integer, ByVal DIBHandle As Integer) As Double
```

Visual Basic .NET Declaration

```
Function DESKEW_DetectAndCorrect(ByVal Session As
Integer, ByVal DIBHandle As Integer) As Double
```

Delphi Declaration

```
function
DESKEW_DetectAndCorrect(SessionHandle:Integer;DIBH
andle:Integer):double; stdcall;
```

Java Declaration

```
double DESKEW_DetectAndCorrect(int SessionHandle,
int DIBHandle);
```

Description

This function allows to detect and correct the skew in full automatic mode.

Parameters

SessionHandle (in) - the session handle to use
DIBIn (in) - the handle of the monochrome DIB to deskew

Return values

The detected and corrected skew angle, in degree.

Notes

The library works with standard DIBs images. Refer to Microsoft documentation for additional info about Device Indipendent Bitmaps.

3.4 DESKEW_DetectUsingBlackBorders

C/C++ Declaration

```
__stdcall double
DESKEW_DetectUsingBlackBorders(long SessionHandle,
long DIBIn, double MaxAngle, bool HasNoise, long
BordersToUse);
```

C# Declaration

```
double DESKEW_DetectUsingBlackBorders(int
SessionHandle, int DIBIn, double MaxAngle, bool
HasNoise, int BordersToUse);
```

Visual Basic Declaration

```
Function DESKEW_DetectUsingBlackBorders(ByVal
Session As Integer, ByVal DIBHandle As Integer,
ByVal MaxAngle As Double, ByVal HasNoise as
Boolean, ByVal BordersToUse as Integer) As Double
```

Visual Basic .NET Declaration

```
Function DESKEW_DetectUsingBlackBorders(ByVal
Session As Integer, ByVal DIBHandle As Integer,
ByVal MaxAngle As Double, ByVal HasNoise As
```

```
Boolean, ByVal BordersToUse As Integer) As Double
```

Delphi Declaration

```
function
DESKEW_DetectUsingBlackBorders(SessionHandle:Integ
er; DIBHandle:Integer; MaxAngle,HasNoise:Boolean ;
BordersToUse:NativeInt):double; stdcall;
```

Java Declaration

```
double DESKEW_DetectUsingBlackBorders(int
SessionHandle, int DIBHandle, double
MaxAngle,boolean HasNoise, int BordersToUse);
```

Description

This function allows to detect the skew on the image using the presence of black borders

Parameters

SessionHandle (in) - the session handle to use
DIBIn (in) - the handle of the monochrome DIB to deskew
MaxAngle (in) - the max angle, in degree, to check for skew: the interval -MaxAngle to +MaxAngle is checked
HasNoise (in) - boolean value that take in account the noise presence
BordersToUse (in) - the position where consider the border. It is an add/or of 1=left, 2=top,4=right and 8=bottom

Return values

The detected skew angle, in degree.

Notes

The library works with standard DIBs images. Refer to Microsoft documentation for additional info about Device Indipendent Bitmaps.

3.5 DESKEW_Detect

C/C++ Declaration

```
__stdcall double DESKEW_Detect(long SessionHandle,
long DIBIn, double MaxAngle, double AnlgeAccuracy,
long Step,long EdgePixelsToSkip );
```

C# Declaration

```
double DESKEW_Detect(int SessionHandle, int DIBIn,
double MaxAngle, double AnlgeAccuracy, int Step,
int EdgePixelsToSkip);
```

Visual Basic Declaration

```
Function DESKEW_Detect(ByVal Session As Integer,
ByVal DIBHandle As Integer, ByVal MaxAngle As
Double, ByVal AngleAccuracy as Double, ByVal Step
as Integer, ByVal EdgePixelsToSkip as Integer) As
Double
```

Visual Basic .NET Declaration

```
Function DESKEW_Detect(ByVal Session As Integer,
ByVal DIBHandle As Integer, ByVal MaxAngle As
Double, ByVal AngleAccuracy As Double, ByVal aStep
As Integer, ByVal EdgePixelsToSkip as Integer) As
Double
```

Delphi Declaration

```
function DESKEW_Detect(SessionHandle:Integer;
DIBHandle:Integer; MaxAngle,Accuracy:double;
Step:Integer, EdgePixelsToSkip: Integer):double;
stdcall;
```

Java Declaration

```
double DESKEW_Detect(int SessionHandle, int
DIBHandle, double MaxAngle,double Accuracy, int
Step);
```

Description

This function allows to detect only the skew on the image.

Parameters

SessionHandle (in) - the session handle to use
DIBIn (in) - the handle of the monochrome DIB to deskew
MaxAngle (in) - the max angle, in degree, to check for skew: the interval -MaxAnglo to +MaxAngle is checked
AngleAccuracy (in) - the angle accuracy, in degree, to check for skew: eg. 0.1 (1/10 of degree) or 0.25 (1/4 of degree)
Step (in) - the step to skip rows in image analisys. eg. 10 (use a line every 10) or -1 (auto calculate the best value)
EdgePixelsToSkip (in) - if the value is >0 represents the number of pixels from the edge of the image to be skipped in the analysis, while if it is <0 it represents the percentage calculated on the side of the image

Return values

The detected skew angle, in degree.

Notes

The library works with standard DIBs images. Refer to Microsoft documentation for additional info about Device Indipendent Bitmaps.

3.6 DESKEW_Correct

C/C++ Declaration

```
__stdcall void DESKEW_Correct(long SessionHandle,
long DIBIn, double SkewAngle, long FillWhite, long
Interpolation, long Resize);
```

C# Declaration

```
void DESKEW_Correct(int SessionHandle, int DIBIn,
double SkewAngle, int FillWhite, int
Interpolation, int Resize);
```

Visual Basic Declaration

```
Sub DESKEW_Correct(ByVal Session As Integer, ByVal
DIBHandle As Integer, ByVal Angle As Double, ByVal
FillWhite As Integer, ByVal Interpolation As
Integer,ByVal Resize As Integer )
```

Visual Basic .NET Declaration

```
Sub DESKEW_Correct(ByVal Session As Integer, ByVal
DIBHandle As Integer, ByVal Angle As Double, ByVal
FillWhite As Integer, ByVal Interpolation As
Integer, ByVal Resize As Integer)
```

Delphi Declaration

```
procedure DESKEW_Correct(SessionHandle:Integer;
DIBHandle:Integer; SkewAngle:double;
FillWhite,Interpolation,Resize:Integer); stdcall;
```

Java Declaration

```
void DESKEW_Correct(int SessionHandle,int
DIBHandle, double SkewAngle, int FillWhite,int
Interpolation, int Resize);
```

Description

This function allows to correct the skew on the image.

Parameters

SessionHandle (in) - the session handle to use
DIBIn (in) - the handle of the monochrome DIB to deskew
SkewAngle(in) - the skew angle to correct, in degree
FilleWhite(in) - flag allowing to set the color of new background: 1 (white) or 0 (black)
Intepolation(in) - flag allowing to set the interpolation: 1 (active) or 0 (inactive). Using interpolation you will have better results on gray-scale or color images, but the speed will decrease.
Resize(in) - flag allowing to set the resize the image: not 0 (active) or 0 (inactive). The value not null increase the processing time.

Return values

n/a

Notes

The library works with standard DIBs images. Refer to Microsoft documentation for additional info about Device Indipendent Bitmaps. If after deskewing you will use black border removal feature, you should use 0 (black) as *FillWhite* flag value.

3.7 DESKEW_CorrectEx

C/C++ Declaration

```
__stdcall void DESKEW_CorrectEx(long
SessionHandle, long DIBIn, double SkewAngle, long
FillWhite, long Interpolation);
```

C# Declaration

```
void DESKEW_CorrectEx(int SessionHandle, int
DIBIn, double SkewAngle, int FillWhite, int
Interpolation);
```

Visual Basic Declaration

```
Sub DESKEW_CorrectEx (ByVal Session As Integer,
ByVal DIBHandle As Integer, ByVal HSkewAngle As
Double, ByVal VSkewHangle As Double, ByVal
FillColor As Integer, ByVal Interpolation As
Integer, ByVal Resize As Integer)
```

Visual Basic .NET Declaration

```
Sub DESKEW_CorrectEx (ByVal Session As Integer,
ByVal DIBHandle As Integer, ByVal HSkewAngle As
Double, ByVal VSkewHangle As Double, ByVal
FillColor As Integer, ByVal Interpolation As
Integer, ByVal Resize As Integer)
```

Delphi Declaration

```
procedure DESKEW_CorrectEx(SessionHandle:Integer;
DIBHandle:Integer; HSkewAngle,VSkewHangle:double;
FillColor,Interpolation,Resize:NativeInt);
stdcall;
```

Java Declaration

```
void DESKEW_CorrectEx(int SessionHandle,int
DIBHandle, double HSkewAngle,doubleVSkewHangle,
int FillColor,int Interpolation,int Resize);
```

Description

This function allows to correct the horizontal and vertical skew on the image

Parameters

SessionHandle (in) - the session handle to use
DIBIn (in) - the handle of the monochrome DIB to deskew
HSkewAngle(in) - the horizontal skew angle to correct, in degree
VSkewAngle(in) - the vertical skew angle to correct, in degree
FillColor(in) - flag allowing to set the RGB color of new background
Intepolation(in) - flag allowing to set the interpolation: not 0 (active) or 0 (inactive). Using interpolation you will have better results on gray-scale or color images, but the speed will decrease.
Resize(in) - flag allowing to set the resize the image: not 0 (active) or 0 (inactive). The value not null increase the processing time.

Return values

n/a

Notes

The library works with standard DIBs images. Refer to Microsoft documentation for additional info about Device Indipendent Bitmaps.

3.8 LoadDeskewLibrary

C/C++ Declaration

```
long LoadDeskewLibrary();
```

Description

Load the DESKEW DLL library: you have to use this function one time before to use other API functions

3.9 FreeDeskewLibrary

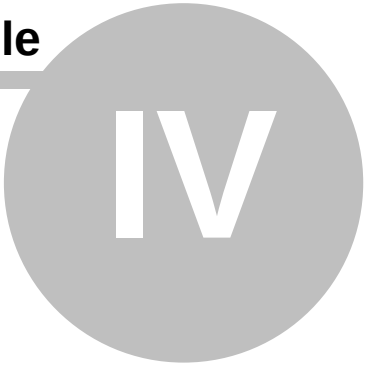
C/C++ Declaration

```
void FreeDeskewLibrary();
```

Description

Unload the DESKEW.DLL library: you have to use this function one time before to exit from your application.

Sample



4 Sample

4.1 Code Sample

```
#include "stdafx.h"
#include <stdio.h>
#include "recodeskew.c"

int APIENTRY WinMain(HINSTANCE hInstance,
                     HINSTANCE hPrevInstance,
                     LPSTR     lpCmdLine,
                     int       nCmdShow)
{

        // Load dynamically the library
        LoadDeskewLibrary();

        // Init the DESKEW session
        int Session= DESKEW_Init("demo", "demo");

        long hBitmap;

        // Check if a DIB is available in clipboard
        bool bAvail= IsClipboardFormatAvailable(CF_DIB);

        hBitmap = 0;



        if (bAvail)
        {

                // Open the Clipboard
                ::OpenClipboard(NULL);

                // Retrieve the DIB from clipboard
                hBitmap = (long) GetClipboardData(CF_DIB);


                // Detect and correct skew on the DIB
                double SkewAngle=
DESKEW_DetectAndCorrect(Session, (long) hBitmap);

                ::EmptyClipboard();

                // Set the deskewed DIB to clipboard
        SetClipboardData(CF_DIB, (void*)hBitmap);

                // Close the Clipboard
                ::CloseClipboard();

                char  buffer[200];

                // Format the output message
        if (SkewAngle!=0)

                        sprintf( buffer, "Image deskewed by %f
degree ! You can paste in your application...", SkewAngle);
```

```
                else

            sprintf( buffer, "Image don't need to be deskewd
!");


        // Show the result
                MessageBox(NULL, buffer, "RESULT", MB_OK);


        }
        // Show an error message
        else MessageBox(NULL, "Unable to paste DIB", "ERROR",
MB_OK);


        // Close the session
        DESKEW_Done(Session);

        // Unload the library
        FreeDeskewLibrary();

        return 0;
}
```

Annotation