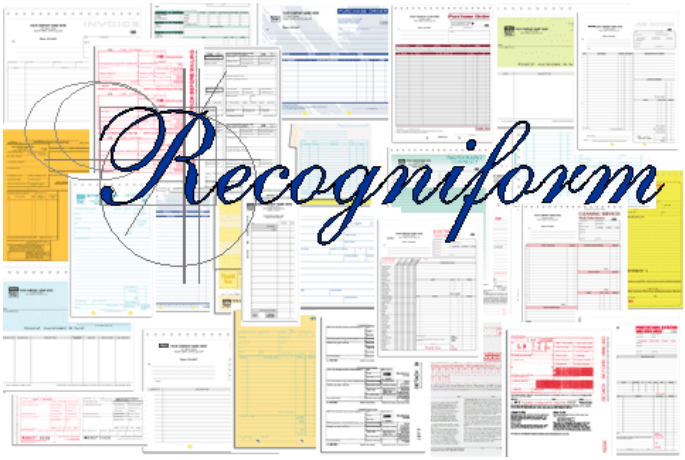


Software Development Kit

Imaging

Recogniform Technologies SpA

HOW TO CONTACT US

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

Phone : +39 0984 404174

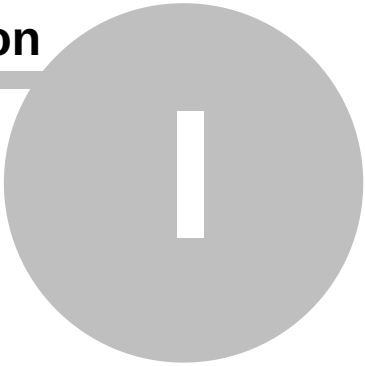
Fax : +39 0984 830299

Internet : www.recogniform.com

E-Mail : info@recogniform.com

Table of contents

Introduction



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 Recogniform Imaging Library 5.0 allows to:

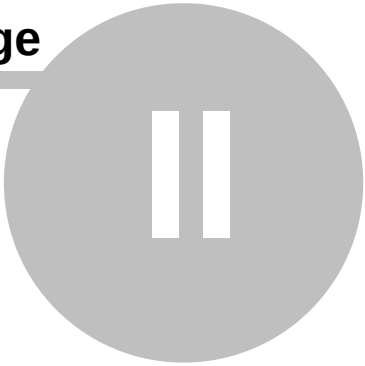
- read images from files returning standard DIB (Device Indipendent Bitmap) memory handles
- save images to files
- acquire images from any TWAIN scanner, with or without user interface

- display images also using scale-to-gray technology
- apply basic image processing (rotate 90/180/270°, flip, mirror, invert)
- extract portion of images as sub-images, returning standard DIBs
- get standard draw context to write text/graphics on images using standard GDI Windows API.
- import/export DIB from/to DDB (Device Dependent Bitmap)
- free images from memory

The file format supported both from loading and saving are:

- TIFF (Uncompressed, CCITT-G4, CCITT-G3, PackBits, Deflate, LZW, JPEG)
- BMP
- JPEG
- PDF (Images Uncompressed, CCITT-G4, CCITT-G3, JPEG)

Usage



2 Usage

2.1 Visual C++

You have to include the RECOIOAPI.C in your program. Before to execute your application make sure the *RECOIO.DLL* is available in your same .exe directory or in windows\system directory.

2.2 C#

You have to include the RECOIOAPI.CS in your program. Before to execute your application make sure the *RECOIO.DLL* is available in your same .exe directory or in windows\system directory.

2.3 Visual Basic

You have to include the RECOIOAPI.BAS in your program. Before to execute your application make sure the *RECOIO.DLL* is available in your same .exe directory or in windows\system directory.

2.4 Visual Basic .NET

You have to include the RECOIOAPI.VB in your program. Before to execute your application make sure the *RECOIO.DLL* is available in your same .exe directory or in windows\system directory.

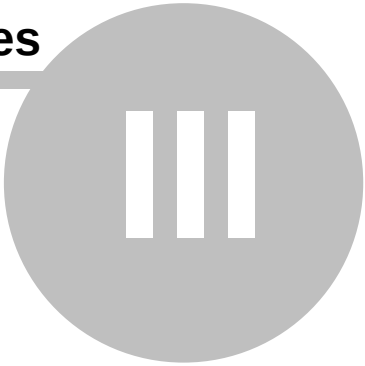
2.5 Java

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

2.6 Delphi

You have to include the RECOIOAPI.PAS in your program. Before to execute your application make sure the *RECOIO.DLL* is available in your same .exe directory or in windows\system directory.

API References



3 API References

3.1 LoadIOLibrary

C/C++ Declaration

```
long LoadIOLibrary();
```

Description

Load the IO DLL library: you have to use this function one time before to use other API functions from Visual C++ This function is not required using the API from other languages.

3.2 FreeIOLibrary

C/C++ Declaration

```
void FreeIOLibrary();
```

Description

Unload the IO DLL library: you have to use this function one time before to exit from your Visual C++ application. This function is not required using the API from other languages.

3.3 IO_AcquireDC

C/C++ Declaration

```
__stdcall long IO_AcquireDC(long SessionHandle,
long DIBHandle);
```

C# Declaration

```
int IO_AcquireDC(int SessionHandle, int
DIBHandle);
```

Visual Basic Declaration

```
Function IO_AcquireDC(ByVal Session As Integer,
ByVal BMPHandle As Integer) As Integer
```

Visual Basic .NET Declaration

```
Function IO_AcquireDC(ByVal Session As Integer,
ByVal BMPHandle As Integer) As Integer
```

Delphi Declaration

```
function IO_AcquireDC(SessionHandle:Integer;
DIBHandle:Integer):Integer; stdcall;
```

Java Declaration

```
int IO_AcquireDC(int SessionHandle, int
DIBHandle);
```

Description

Acquire a draw context on the DIB. The draw context can be used with GDI Windows API to write into the image.

Parameters

SessionHandle (in) - the session handle to use
DIBHandle(in) - the DIB handle of the image where the DC has to be acquired.

Return values

The DC handle.

3.4 IO_Init

C/C++ Declaration

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

C# Declaration

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

Visual Basic Declaration

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

Visual Basic .NET Declaration

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

Delphi Declaration

```
function IO_Init(Company:PAnsiChar,
LicenseKey:PAnsiChar):Integer; stdcall;
```

Java Declaration

```
int IO_Init(String Name, String Key);
```

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 wrong values the library is initialized in evaluation mode. The evaluation mode works exactly as normal mode but some time, when you call a 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
Passwords (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

3.5 IO_CountPDFImages

C/C++ Declaration

```
__stdcall long IO_CountPDFImages(long
SessionHandle, char* FileName);
```

C# Declaration

```
int IO_CountPDFImages(int SessionHandle, string
FileName);
```

Visual Basic Declaration

```
Function IO_CountPDFImages(ByVal Session As
Integer, ByVal FileName As String) As Integer
```

Visual Basic .NET Declaration

```
Function IO_CountPDFImages(ByVal Session As
Integer, ByVal FileName As String) As Integer
```

Delphi Declaration

```
function
IO_CountPDFImages(SessionHandle:Integer;FileName:P
AnsiChar):Integer; stdcall;
```

Java Declaration

```
int IO_CountPDFImages(int SessionHandle, String
FileName);
```

Description

This is the function to call to obtain the number of pages contained in a multipage PDF file.

Parameters

SessionHandle (in) - the session handle to use
FileName(in) - the complete file name path of the file to check

Return values

The numbers of pages contained in the file.

3.6 IO_CountTIFImages

C/C++ Declaration

```
__stdcall long IO_CountTIFImages(long
SessionHandle, char* FileName);
```

C# Declaration

```
int IO_CountTIFImages(int SessionHandle, string
FileName);
```

Visual Basic Declaration

```
Function IO_CountTIFImages(ByVal Session As
Integer, ByVal FileName As String) As Integer
```

Visual Basic .NET Declaration

```
Function IO_CountTIFImages(ByVal Session As
Integer, ByVal FileName As String) As Integer
```

Delphi Declaration

```
function
IO_CountTIFImages(SessionHandle:Integer;FileName:P
AnsiChar):Integer; stdcall;
```

Java Declaration

```
int IO_CountTIFImages(int SessionHandle, String
FileName);
```

Description

This is the function to call to obtain the number of pages contained in a multipage TIFF file.

Parameters

SessionHandle (in) - the session handle to use
FileName(in) - the complete file name path of the file to check

Return values

The numbers of pages contained in the file.

3.7 IO_CountTIFImagesFromBuffer

C/C++ Declaration

```
__stdcall long IO_CountTIFImagesFromBuffer(long
SessionHandle,long* Buffer, long BufferSize);
```

C# Declaration

```
int IO_CountTIFImagesFromBuffer(int SessionHandle,
IntPtr Buffer, int BufferSize);
```

Visual Basic Declaration

```
Function IO_CountTIFImagesFromBuffer(ByVal Session
As Integer,ByVal Buffer As Long, ByVal BufferSize
As Integer) As Integer
```

Visual Basic .NET Declaration

```
Function IO_CountTIFImagesFromBuffer(ByVal Session
As Integer, ByVal Buffer As IntPtr, ByVal
BufferSize As Integer) As Integer
```

Delphi Declaration

```
function
IO_CountTIFImagesFromBuffer(SessionHandle:Integer;
Buffer:Pointer, BufferSize:Integer):Integer;
stdcall;
```

Java Declaration

```
int IO_CountTIFImagesFromBuffer(int SessionHandle,
Pointer Buffer, int BufferSize);
```

Description

This is the function to call to obtain the number of pages contained in a multipage TIFF file.

Parameters

SessionHandle (in) - the session handle to use
Buffer(in) - the buffer containing the image
BufferSize(in) - The size of image buffer

Return values

The numbers of pages contained in the file.

3.8 IO_DebugBuffer

C/C++ Declaration

```
__stdcall long IO_DebugBuffer(long
SessionHandle,long* Buffer, long BufferSize);
```

C# Declaration

```
int IO_DebugBuffer(int SessionHandle, IntPtr
Buffer, int BufferSize);
```

Visual Basic Declaration

```
Function IO_DebugBuffer(ByVal Session As
Integer,ByVal Buffer As Long, ByVal BufferSize As
Integer) As Integer
```

Visual Basic .NET Declaration

```
Function IO_DebugBuffer(ByVal Session As Integer,
ByVal Buffer As IntPtr, ByVal BufferSize As
Integer) As Integer
```

Delphi Declaration

```
function IO_DebugBuffer(SessionHandle:Integer;
Buffer:Pointer, BufferSize:Integer):Integer;
stdcall;
```

Java Declaration

```
int IO_DebugBuffer(int SessionHandle, Pointer
Buffer, int BufferSize);
```

Description

Show a message with the buffer length and its first character

Parameters

SessionHandle (in) - the session handle to use
Buffer(in) - the buffer containing the image
BufferSize(in) - The size of image buffer

Return values

None

3.9 IO_Done

C/C++ Declaration

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

C# Declaration

```
void IO_Done(int SessionHandle);
```

Visual Basic Declaration

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

Visual Basic .NET Declaration

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

Delphi Declaration

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

Java Declaration

```
void IO_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

3.10 IO_Duplicate

C/C++ Declaration

```
__stdcall long IO_Duplicate(long SessionHandle,
long DIBHandle);
```

C# Declaration

```
int IO_Duplicate(int SessionHandle, long
DIBHandle);
```

Visual Basic Declaration

```
Function IO_Duplicate(ByVal Session As Integer,
ByVal BMPHandle As Integer) As Integer
```

Visual Basic .NET Declaration

```
Function IO_Duplicate(ByVal Session As Integer,
ByVal BMPHandle As Integer) As Integer
```

Delphi Declaration

```
function IO_Duplicate(SessionHandle:Integer;
DIBHandle:Integer):Integer; stdcall;
```

Java Declaration

```
int IO_Duplicate(int SessionHandle, int
DIBHandle);
```

Description

Create an image duplicate.

Parameters

SessionHandle (in) - the session handle to use
DIBHandle(in) - the DIB handle of the image to use.

Return values

The DIB handle of the new duplicated image.

3.11 IO_ExportDDB

C/C++ Declaration

```
__stdcall void IO_ExportDDB(long SessionHandle,
long DIBHandle, long* Bitmap, long* Palette);
```

C# Declaration

```
void IO_ExportDDB(int SessionHandle, int
DIBHandle, ref int Bitmap, ref int Palette);
```

Visual Basic Declaration

```
Sub IO_ExportDDB(ByVal Session As Integer, ByVal
DIBHandle As Integer, ByRef DDBHandle As Integer,
ByRef Palette As Integer)
```

Visual Basic .NET Declaration

```
Sub IO_ExportDDB(ByVal Session As Integer, ByVal
DIBHandle As Integer, ByRef DDBHandle As Integer,
ByRef Palette As Integer)
```

Delphi Declaration

```
procedure IO_ExportDDB(SessionHandle:Integer;
DIBHandle:Integer; Var Bitmap:Integer; Var
Palette:Integer); stdcall;
```

Java Declaration

```
void IO_ExportDDB(int SessionHandle, int
DIBHandle, int[] Bitmap, int[] Palette);
```

Description

Export a DIB as a DDB (Device Dependent Bitmap)

Parameters

SessionHandle (in) - the session handle to use
DIBHandle(in) - the DIB handle of the image to export.
Bitamap (out) - the DDB handle
Palette (out) - the DDB palette handle

Return values

n/a

3.12 IO_FreeImage

C/C++ Declaration

```
__stdcall void IO_FreeImage(long SessionHandle,
long DIBHandle);
```

C# Declaration

```
void IO_FreeImage(int SessionHandle, int
DIBHandle);
```

Visual Basic Declaration

```
Sub IO_FreeImage(ByVal Session As Integer, ByVal
BMPHandle As Integer)
```

Visual Basic .NET Declaration

```
Sub IO_FreeImage(ByVal Session As Integer, ByVal
BMPHandle As Integer)
```

Delphi Declaration

```
procedure IO_FreeImage(SessionHandle:Integer;
DIBHandle:Integer); stdcall;
```

Java Declaration

```
void IO_FreeImage(int SessionHandle, int
DIBHandle);
```

Description

This is the function to call to deallocate memory used by an image.

Parameters

SessionHandle (in) - the session handle to use
DIBHandle(in) - the DIB handle to free.

Return values

n/a

3.13 IO_Flip

C/C++ Declaration

```
__stdcall void IO_Flip(long SessionHandle, long
DIBHandle);
```

C# Declaration

```
void IO_Flip(int SessionHandle, int DIBHandle);
```

Visual Basic Declaration

```
Sub IO_Flip(ByVal Session As Integer, ByVal
BMPHandle As Integer)
```

Visual Basic .NET Declaration

```
Sub IO_Flip(ByVal Session As Integer, ByVal
BMPHandle As Integer)
```

Delphi Declaration

```
function IO_Flip(SessionHandle:Integer;
DIBHandle:Integer):Integer; stdcall;
```

Java Declaration

```
void IO_Flip(int SessionHandle, int DIBHandle);
```

Description

Convert the image in the fliped version.

Parameters

SessionHandle (in) - the session handle to use
DIBHandle(in) - the DIB handle of the image to use.

Return values

n/a

3.14 IO_GetImageInfo

C/C++ Declaration

```
__stdcall void IO_GetImageInfo(long SessionHandle,
long DIBHandle, long* Width, long* Height, long*
XRes, long* YRes, long* Bits);
```

C# Declaration

```
int IO_GetImageInfo(int SessionHandle, int
DIBHandle, ref int Width, ref int Height, ref int
XRes, ref int YRes, ref int Bits);
```

Visual Basic Declaration

```
Sub IO_GetImageInfo(ByVal Session As Integer,
ByVal BMPHandle As Integer, ByRef Width As
Integer, ByRef Height As Integer, ByRef XRes As
Integer, ByRef YRes As Integer, ByRef Bits As
Integer)
```

Visual Basic .NET Declaration

```
Sub IO_GetImageInfo(ByVal Session As Integer,
ByVal BMPHandle As Integer, ByRef Width As
Integer, ByRef Height As Integer, ByRef XRes As
Integer, ByRef YRes As Integer, ByRef Bits As
Integer)
```

Delphi Declaration

```
procedure IO_GetImageInfo(SessionHandle:Integer;
DIBHandle:Integer; Var Width, Height, XRes, YRes,
Bits:Integer); stdcall;
```

Java Declaration

```
int IO_GetImageInfo(int SessionHandle, int
DIBHandle, int[] Width, int[] Height, int[] XRes,
int[] YRes, int[] Bits);
```

Description

Retrieves info about the image.

Parameters

SessionHandle (in) - the session handle to use
DIBHandle(in) - the DIB handle of the image to use.
Width(out) - the width in pixel of the image.
Height(out) - the height in pixel of the image.
XRes(out) - the horizontal resolution in DPI of the image.
YRes(out) - the vertical resolution in DPI of the image.
Bits (out) - the number of bits per pixel of the image: 1= monochrome, 8=gray level, 24=true-color.

Return values

n/a

3.15 IO_GetNextAsyncScannedImage

C/C++ Declaration

```
__stdcall long IO_GetNextAsyncScannedImage(long
SessionHandle);
```

C# Declaration

```
int IO_GetNextAsyncScannedImage(int
SessionHandle);
```

Visual Basic Declaration

```
Function IO_GetNextAsyncScannedImage(ByVal Session
As Integer) As Integer
```

Visual Basic .NET Declaration

```
Function IO_GetNextAsyncScannedImage(ByVal Session
As Integer) As Integer
```

Delphi Declaration

```
function IO_GetNextAsyncScannedImage
(SessionHandle:Integer):integer; stdcall;
```

Java Declaration

```
int IO_GetNextAsyncScannedImage(int
SessionHandle);
```

Description

Get the next scanned image.

Parameters

SessionHandle (in) - the session handle to use

Return values

The DIB handle of the next scanned image.

3.16 IO_GetPixel

C/C++ Declaration

```
__stdcall long IO_GetPixel(long SessionHandle,
long DIBHandle, long X, long Y);
```

C# Declaration

```
int IO_GetPixel(int SessionHandle, int DIBHandle,
int X, int Y);
```

Visual Basic Declaration

```
Function IO_GetPixel(ByVal Session As Integer,
Byval DIBHandle As Integer, Byval X As Integer,
Byval Y As Integer) As Integer
```

Visual Basic .NET Declaration

```
Function IO_GetPixel(ByVal Session As Integer,
Byval DIBHandle As Integer, Byval X As Integer,
Byval Y As Integer) As Integer
```

Delphi Declaration

```
function IO_GetPixel(SessionHandle:Integer;
DIBHandle,X,Y:Integer):Integer; stdcall;
```

Java Declaration

```
int IO_GetPixel(int SessionHandle, int DIBHandle,
int X, int Y);
```

Description

Gets the color of a specific pixel element using x and y coordinates

Parameters

SessionHandle (in) - the session handle to use
intImageHandle: integer value corresponding to the image handle.
intX: integer value corresponding to the column.
intY: integer value corresponding to the line.

Return values

Numerical value showing the indicated pixel color

3.17 IO_GetScaleToGrayImage

C/C++ Declaration

```
__stdcall long IO_GetScaleToGrayImage(long
SessionHandle, long DIBHandle, long ZoomFactor);
```

C# Declaration

```
int IO_GetScaleToGrayImage(int SessionHandle, int
DIBHandle, int ZoomFactor);
```

Visual Basic Declaration

```
Function IO_GetScaleToGrayImage(ByVal Session As
Integer, ByVal BMPHandle As Integer, ByVal Zoom As
Integer) As Integer
```

Visual Basic .NET Declaration

```
Function IO_GetScaleToGrayImage(ByVal Session As
Integer, ByVal BMPHandle As Integer, ByVal Zoom As
Integer) As Integer
```

Delphi Declaration

```
function
IO_GetScaleToGrayImage(SessionHandle:Integer;
DIBHandle:Integer; ZoomFactor:Integer):Integer;
```

```
stdcall;
```

Java Declaration

```
int IO_GetScaleToGrayImage(int SessionHandle, int
DIBHandle, int ZoomFactor);
```

Description

Create a new gray image scaling a monochrome image: very useful for display large images.

Parameters

SessionHandle (in) - the session handle to use
DIBHandle(in) - the DIB handle of the image to use.
ZoomFactor(in) - the zoom factor to use (1-100)

Return values

The DIB handle of the new scaled image in gray scale.

3.18 IO_GetSubImage

C/C++ Declaration

```
__stdcall long IO_GetSubImage(long SessionHandle,
long DIBHandle, long left, long top, long right,
long bottom);
```

C# Declaration

```
int IO_GetSubImage(int SessionHandle, int
DIBHandle, int left, int top, int right, int
bottom);
```

Visual Basic Declaration

```
Function IO_GetSubImage(ByVal Session As Integer,
ByVal BMPHandle As Integer, ByVal Left As Integer,
ByVal Top As Integer, ByVal Right As Integer,
ByVal Bottom As Integer) As Integer
```

Visual Basic .NET Declaration

```
Function IO_GetSubImage(ByVal Session As Integer,
```

```
ByVal BMPHandle As Integer, ByVal Left As Integer,
ByVal Top As Integer, ByVal Right As Integer,
ByVal Bottom As Integer) As Integer
```

Delphi Declaration

```
function IO_GetSubImage(SessionHandle:Integer;
DIBHandle:Integer;
Left,Top,Right,Bottom:Integer):Integer; stdcall;
```

Java Declaration

```
int IO_GetSubImage(int SessionHandle, int
DIBHandle, int left, int top, int right, int
bottom);
```

Description

Create a new subimage from the image.

Parameters

SessionHandle (in) - the session handle to use
DIBHandle(in) - the DIB handle of the image to use.
Left(in) - the left coordinate of the rectangular area to copy.
Top(in) - the top coordinate of the rectangular area to copy.
Right(in) - the right coordinate of the rectangular area to copy.
Bottom(in) - the bottom coordinate of the rectangular area to copy.

Return values

The DIB handle of the new sub-image.

3.19 IO_ImportDDB

C/C++ Declaration

```
__stdcall long IO_ImportDDB(long SessionHandle,
long Bitmap, long Palette);
```

C# Declaration

```
int IO_ImportDDB(int SessionHandle, int Bitmap,
int Palette);
```

Visual Basic Declaration

```
Function IO_ImportDDB(ByVal Session As Integer,
ByVal DDBHandle As Integer, ByVal Palette As
Integer) As Integer
```

Visual Basic .NET Declaration

```
Function IO_ImportDDB(ByVal Session As Integer,
ByVal DDBHandle As Integer, ByVal Palette As
Integer) As Integer
```

Delphi Declaration

```
function IO_ImportDDB(SessionHandle:Integer;
DIBHandle:Integer; Bitmap:Integer;
Palette:Integer):integer; stdcall;
```

Java Declaration

```
int IO_ImportDDB(int SessionHandle, int Bitmap,
int Palette);
```

Description

Import a DDB (Device Dependent Bitmap) as a DIB

Parameters

SessionHandle (in) - the session handle to use
Bitamap (in) - the DDB handle
Palette (in) - the DDB palette handle

Return values

The DIB handle of the imported image.

3.20 IO_Invert

C/C++ Declaration

```
__stdcall void IO_Invert(long SessionHandle, long
DIBHandle);
```

C# Declaration

```
void IO_Invert(int SessionHandle, int DIBHandle);
```

Visual Basic Declaration

```
Sub IO_Invert(ByVal Session As Integer, ByVal
BMPHandle As Integer)
```

Visual Basic .NET Declaration

```
Sub IO_Invert(ByVal Session As Integer, ByVal
BMPHandle As Integer)
```

Delphi Declaration

```
function IO_Invert(SessionHandle:Integer;
DIBHandle:Integer):Integer; stdcall;
```

Java Declaration

```
void IO_Invert(int SessionHandle, int DIBHandle);
```

Description

Invert the colors, creating a negative version of the image.

Parameters

SessionHandle (in) - the session handle to use
DIBHandle(in) - the DIB handle of the image to use.

Return values

n/a

3.21 IO_LoadBMPImage

C/C++ Declaration

```
__stdcall long LoadBMPImage(long SessionHandle,
char* FileName);
```

C# Declaration

```
int LoadBMPImage(int SessionHandle, string
FileName);
```

Visual Basic Declaration

```
Function IO_LoadBMPImage(ByVal Session As Integer,
ByVal FileName As String) As Integer
```

Visual Basic .NET Declaration

```
Function IO_LoadBMPImage(ByVal Session As Integer,
ByVal FileName As String) As Integer
```

Delphi Declaration

```
function IO_LoadBMPImage(SessionHandle:Integer;
FileName:PAnsiChar):Integer; stdcall;
```

Java Declaration

```
int LoadBMPImage(int SessionHandle, String
FileName);
```

Description

This is the function performing the loading of an image from a file: you have to call this function after library initialization to perform the loading.

Parameters

SessionHandle (in) - the session handle to use
FileName(in) - the complete file name path of the file to load

Return values

The DIB handle of the loaded image.

Notes

The returned value is an handle to a standard DIB images. Refer to Microsoft documentation for additional info about Device Indipendent Bitmaps.

3.22 IO_LoadBMPImageFromBuffer

C/C++ Declaration

```
__stdcall long LoadBMPImageFromBuffer(long
SessionHandle, long* Buffer , long BufferSize);
```

C# Declaration

```
int LoadBMPImageFromBuffer(int SessionHandle,
IntPtr Buffer , int BufferSize);
```

Visual Basic Declaration

```
Function LoadBMPImageFromBuffer(ByVal Session As
Integer,ByVal Buffer As Long, ByVal BufferSize As
Integer) As Integer
```

Visual Basic .NET Declaration

```
Function LoadBMPImageFromBuffer(ByVal Session As
Integer, ByVal Buffer As Intptr, ByVal BufferSize
As Integer) As Integer
```

Delphi Declaration

```
function
LoadBMPImageFromBuffer(SessionHandle:Integer;
Buffer:Pointer; BufferSize:Integer):Integer;
stdcall;
```

Java Declaration

```
int LoadBMPImageFromBuffer(int SessionHandle,
Pointer Buffer , int BufferSize);
```

Description

This is the function performing the loading of an BMP image from a buffer: you have to call this function after library initialization to perform the loading.

Parameters

SessionHandle (in) - the session handle to use
Buffer(in) - the buffer containing the image

BufferSize(in) - The size of image buffer

Return values

The DIB handle of the loaded image.

Notes

The returned value is an handle to a standard DIB images. Refer to Microsoft documentation for additional info about Device Indipendent Bitmaps.

3.23 IO_LoadJPGImage

C/C++ Declaration

```
__stdcall long IO_LoadJPGImage(long SessionHandle,
char* FileName);
```

C# Declaration

```
int IO_LoadJPGImage(int SessionHandle, string
FileName);
```

Visual Basic Declaration

```
Function IO_LoadJPGImage(ByVal Session As Integer,
ByVal FileName As String) As Integer
```

Visual Basic .NET Declaration

```
Function IO_LoadJPGImage(ByVal Session As Integer,
ByVal FileName As String) As Integer
```

Delphi Declaration

```
function IO_LoadJPGImage(SessionHandle:Integer;
FileName:PAnsiChar):Integer; stdcall;
```

Java Declaration

```
int IO_LoadJPGImage(int SessionHandle, String
FileName);
```

Description

This is the function performing the loading of an image from a file: you have to call this function after library initialization to perform the loading.

Parameters

SessionHandle (in) - the session handle to use
FileName(in) - the complete file name path of the file to load

Return values

The DIB handle of the loaded image.

Notes

The returned value is an handle to a standard DIB images. Refer to Microsoft documentation for additional info about Device Indipendent Bitmaps.

3.24 IO_LoadJPGImageFromBuffer

C/C++ Declaration

```
__stdcall long LoadJPGImageFromBuffer(long
SessionHandle, long* Buffer , long BufferSize);
```

C# Declaration

```
int LoadJPGImageFromBuffer(int SessionHandle,
IntPtr Buffer , int BufferSize);
```

Visual Basic Declaration

```
Function LoadJPGImageFromBuffer(ByVal Session As
Integer,ByVal Buffer As Long, ByVal BufferSize As
Integer) As Integer
```

Visual Basic .NET Declaration

```
Function LoadJPGImageFromBuffer(ByVal Session As
Integer, ByVal Buffer As Intptr, ByVal BufferSize
As Integer) As Integer
```

Delphi Declaration

```
function
LoadJPGImageFromBuffer(SessionHandle:Integer;
```

```
Buffer:Pointer; BufferSize:Integer):Integer;
stdcall;
```

Java Declaration

```
int LoadJPGImageFromBuffer(int SessionHandle,
Pointer Buffer , int BufferSize);
```

Description

This is the function performing the loading of an JPG image from a file: you have to call this function after library initialization to perform the loading.

Parameters

SessionHandle (in) - the session handle to use
Buffer(in) - the buffer containing the image
BufferSize(in) - The size of image buffer

Return values

The DIB handle of the loaded image.

Notes

The returned value is an handle to a standard DIB images. Refer to Microsoft documentation for additional info about Device Indipendent Bitmaps.

3.25 IO_LoadPDFImage

C/C++ Declaration

```
__stdcall long IO_LoadPDFImage(long SessionHandle,
char* FileName, long PageNumber, long
DPIRendering);
```

C# Declaration

```
int IO_LoadPDFImage(int SessionHandle, string
FileName, int PageNumber, int DPIRendering);
```

Visual Basic Declaration

```
Function IO_LoadPDFImage(ByVal Session As Integer,
```

```
ByVal FileName As String, ByVal Page As Integer,
ByVal DPIRendering As Integer ) As Integer
```

Visual Basic .NET Declaration

```
Function IO_LoadPDFImage(ByVal Session As Integer,
ByVal FileName As String, ByVal Page As Integer,
ByVal DPIRendering As Integer) As Integer
```

Delphi Declaration

```
function IO_LoadPDFImage(SessionHandle:Integer;
FileName:PAnsiChar; ImageIndex:Integer;
DPIRendering:Integer):Integer; stdcall;
```

Java Declaration

```
int IO_LoadPDFImage(int SessionHandle, String
FileName, int PageNumber, int DPIRendering);
```

Description

This is the function performing the loading of an image from a PDF file.

Parameters

SessionHandle (in) - the session handle to use
FileName(in) - the complet file name path of the file to load
PageNumber(in) - the page number to load if the file is multipage. First page is 0.
DPIRendering(in)- DPI Rendering resolution

Return values

The DIB handle of the loaded image.

Notes

The returned value is an handle to a standard DIB images. Refer to Microsoft documentation for additional info about Device Indipendent Bitmaps.

3.26 IO_LoadPDFImageFromBuffer

C/C++ Declaration

```
__stdcall long IO_LoadPDFImageFromBuffer(long
SessionHandle, long* Buffer, long BufferSize, long
ImageIndex, long DPIRendering );
```

C# Declaration

```
int IO_LoadPDFImageFromBuffer(int SessionHandle,
IntPtr Buffer, int BufferSize, int ImageIndex, int
DPIRendering);
```

Visual Basic Declaration

```
Function IO_LoadPDFImageFromBuffer(ByVal Session
As Integer, ByVal Buffer As Long, ByVal BufferSize
As Integer,  ByVal ImageIndex As Integer,  ByVal
DPIRendering As Integer) As Integer
```

Visual Basic .NET Declaration

```
Function IO_LoadPDFImageFromBuffer(ByVal Session
As Integer, ByVal Buffer As Intptr, ByVal
BufferSize As Integer, ByVal ImageIndex As
Integer, ByVal DPIRendering As Integer) As Integer
```

Delphi Declaration

```
function
IO_LoadPDFImageFromBuffer(SessionHandle:Integer;
Buffer:Pointer; BufferSize:Integer;
ImageIndex:Integer; DPIRendering:Integer):Integer;
stdcall;
```

Java Declaration

```
int IO_LoadPDFImageFromBuffer(int SessionHandle,
Pointer Buffer, int BufferSize, int ImageIndex,
int DPIRendering);
```

Description

This is the function performing the loading of an PDF image from a buffer.

Parameters

SessionHandle (in) - the session handle to use
Buffer(in) - the buffer containing the image
BufferSize(in) - The size of image buffer
ImageIndex(in)- the page number to load if the file is multipage. First page is 0.
DPIRendering(in)- DPI Rendering resolution

Return values

The DIB handle of the loaded image.

Notes

The returned value is an handle to a standard DIB images. Refer to Microsoft documentation for additional info about Device Indipendent Bitmaps.

3.27 IO_LoadTIFImage

C/C++ Declaration

```
__stdcall long IO_LoadTIFImage(long SessionHandle,
char* FileName, long PageNumber);
```

C# Declaration

```
int IO_LoadTIFImage(int SessionHandle, string
FileName, int PageNumber);
```

Visual Basic Declaration

```
Function IO_LoadTIFImage(ByVal Session As Integer,
ByVal FileName As String, ByVal Page As Integer)
As Integer
```

Visual Basic .NET Declaration

```
Function IO_LoadTIFImage(ByVal Session As Integer,
ByVal FileName As String, ByVal Page As Integer)
As Integer
```

Delphi Declaration

```
function IO_LoadTIFImage(SessionHandle:Integer;
```

```
FileName:PAnsiChar; ImageIndex:Integer):Integer;
stdcall;
```

Java Declaration

```
int IO_LoadTIFImage(int SessionHandle, String
FileName, int PageNumber);
```

Description

This is the function performing the loading of an image from a TIF file.

Parameters

SessionHandle (in) - the session handle to use
FileName(in) - the complet file name path of the file to load
PageNumber(in) - the page number to load if the file is multipage. First page is 0.

Return values

The DIB handle of the loaded image.

Notes

The returned value is an handle to a standard DIB images. Refer to Microsoft documentation for additional info about Device Indipendent Bitmaps.

3.28 IO_LoadTIFImageFromBuffer

C/C++ Declaration

```
__stdcall long IO_LoadTIFImageFromBuffer(long
SessionHandle, long* Buffer, long BufferSize, long
ImageIndex );
```

C# Declaration

```
int IO_LoadTIFImageFromBuffer(int SessionHandle,
IntPtr Buffer, int BufferSize, int ImageIndex);
```

Visual Basic Declaration

```
Function IO_LoadTIFImageFromBuffer(ByVal Session
```

```
As Integer, ByVal Buffer As Long, ByVal BufferSize
As Integer,  ByVal ImageIndex As Integer) As
Integer
```

Visual Basic .NET Declaration

```
Function IO_LoadTIFImageFromBuffer(ByVal Session
As Integer, ByVal Buffer As Intptr, ByVal
BufferSize As Integer, ByVal ImageIndex As
Integer) As Integer
```

Delphi Declaration

```
function
IO_LoadTIFImageFromBuffer(SessionHandle:Integer;
Buffer:Pointer; BufferSize:Integer;
ImageIndex:Integer):Integer; stdcall;
```

Java Declaration

```
int IO_LoadTIFImageFromBuffer(int SessionHandle,
Pointer Buffer, int BufferSize, int ImageIndex);
```

Description

This is the function performing the loading of an TIF image from a buffer.

Parameters

SessionHandle (in) - the session handle to use
Buffer(in) - the buffer containing the image
BufferSize(in) - The size of image buffer
ImageIndex(in)- the page number to load if the file is multipage. First page is 0.

Return values

The DIB handle of the loaded image.

Notes

The returned value is an handle to a standard DIB images. Refer to Microsoft documentation for additional info about Device Indipendent Bitmaps.

3.29 IO_Mirror

C/C++ Declaration

```
__stdcall void IO_Mirror(long SessionHandle, long
DIBHandle);
```

C# Declaration

```
void IO_Mirror(int SessionHandle, int DIBHandle);
```

Visual Basic Declaration

```
Sub IO_Mirror(ByVal Session As Integer, ByVal
BMPHandle As Integer)
```

Visual Basic .NET Declaration

```
Sub IO_Mirror(ByVal Session As Integer, ByVal
BMPHandle As Integer)
```

Delphi Declaration

```
function IO_Mirror(SessionHandle:Integer;
DIBHandle:Integer):Integer; stdcall;
```

Java Declaration

```
void IO_Mirror(int SessionHandle, int DIBHandle);
```

Description

Convert the image in the mirrored version.

Parameters

SessionHandle (in) - the session handle to use
DIBHandle(in) - the DIB handle of the image to use.

Return values

n/a

3.30 IO_ReleaseDC

C/C++ Declaration

```
__stdcall void IO_ReleaseDC(long SessionHandle,
long DIBHandle);
```

C# Declaration

```
int IO_ReleaseDC(int SessionHandle, int
DIBHandle);
```

Visual Basic Declaration

```
Sub IO_ReleaseDC(ByVal Session As Integer, ByVal
BMPHandle As Integer)
```

Visual Basic .NET Declaration

```
Sub IO_ReleaseDC(ByVal Session As Integer, ByVal
BMPHandle As Integer)
```

Delphi Declaration

```
procedure IO_ReleaseDC(SessionHandle:Integer;
DIBHandle:Integer); stdcall;
```

Java Declaration

```
int IO_ReleaseDC(int SessionHandle, int
DIBHandle);
```

Description

Release the draw context on the DIB, previously acquired using IO_AcquireDC

Parameters

SessionHandle (in) - the session handle to use
DIBHandle(in) - the DIB handle of the image where the DC has to be released.

Return values

n/a

3.31 IO_Rotate

C/C++ Declaration

```
__stdcall void IO_Rotate(long SessionHandle, long
DIBHandle, long Angle);
```

C# Declaration

```
void IO_Rotate(int SessionHandle, int DIBHandle,
int Angle);
```

Visual Basic Declaration

```
Sub IO_Rotate(ByVal Session As Integer, ByVal
BMPHandle As Integer, ByVal Angle As Integer)
```

Visual Basic .NET Declaration

```
Sub IO_Rotate(ByVal Session As Integer, ByVal
BMPHandle As Integer, ByVal Angle As Integer)
```

Delphi Declaration

```
function IO_Rotate(SessionHandle:Integer;
DIBHandle:Integer; Angle:Integer):Integer;
stdcall;
```

Java Declaration

```
void IO_Rotate(int SessionHandle, int DIBHandle,
int Angle);
```

Description

Rotate the image by 90, 180 or 270 degrees clockwise.

Parameters

SessionHandle (in) - the session handle to use
DIBHandle(in) - the DIB handle of the image to use.
Angle (in) - the rotation angle

Return values

n/a

3.32 IO_SaveBMPImage

C/C++ Declaration

```
__stdcall void IO_SaveBMPImage(long SessionHandle,
long DIBHandle, char* FileName);
```

C# Declaration

```
void IO_SaveBMPImage(int SessionHandle, int
DIBHandle, string FileName);
```

Visual Basic Declaration

```
Sub IO_SaveBMPImage(ByVal Session As Integer,
ByVal BMPHandle As Integer, ByVal FileName As
String)
```

Visual Basic .NET Declaration

```
Sub IO_SaveBMPImage(ByVal Session As Integer,
ByVal BMPHandle As Integer, ByVal FileName As
String)
```

Delphi Declaration

```
procedure IO_SaveBMPImage(SessionHandle:Integer;
DIBHandle:Integer; FileName:PAnsiChar); stdcall;
```

Java Declaration

```
void IO_SaveBMPImage(int SessionHandle, int
DIBHandle, String FileName);
```

Description

This is the function performing the saving of an image to a file in BMP format.

Parameters

SessionHandle (in) - the session handle to use
DIBHandle (in) - the handle of the DIB to save
FileName(in) - the complete file name path of the file to save

Return values

n/a

3.33 IO_SaveJPGImage

C/C++ Declaration

```
__stdcall void IO_SaveJPEGImage (long
SessionHandle, long DIBHandle, char* FileName,
long Quality);
```

C# Declaration

```
void IO_SaveJPEGImage(int SessionHandle, int
DIBHandle, string FileName, int Quality);
```

Visual Basic Declaration

```
Sub IO_SaveJPGImage(ByVal Session As Integer,
ByVal BMPHandle As Integer, ByVal FileName As
String, ByVal Quality As Integer)
```

Visual Basic .NET Declaration

```
Sub IO_SaveJPGImage(ByVal Session As Integer,
ByVal BMPHandle As Integer, ByVal FileName As
String, ByVal Quality As Integer)
```

Delphi Declaration

```
procedure IO_SaveJPGImage(SessionHandle:Integer;
DIBHandle:Integer; FileName:PAnsiChar;
Quality:Integer); stdcall;
```

Java Declaration

```
void IO_SaveJPEGImage(int SessionHandle, int
DIBHandle, String FileName, int Quality);
```

Description

This is the function performing the saving of an image to a file in JPEG format.

Parameters

SessionHandle (in) - the session handle to use
DIBHandle (in) - the handle of the DIB to save
FileName(in) - the complete file name path of the file to save
Quality (in) - the compression quality (from 0 to 100)

Return values

n/a

Note

Only true-color or grayscale image can be saved in JPEG format.

3.34 IO_SavePDFImage

C/C++ Declaration

```
__stdcall void IO_SavePDFImage(long SessionHandle,
long DIBHandle, char* FileName, long JPEGQuality);
```

C# Declaration

```
void IO_SavePDFImage(int SessionHandle, int
DIBHandle, string FileName, int JPEGQuality);
```

Visual Basic Declaration

```
Sub IO_SavePDFImage(ByVal Session As Integer,
ByVal BMPHandle As Integer, ByVal FileName As
String, ByVal Quality As Integer)
```

Visual Basic .NET Declaration

```
Sub IO_SavePDFImage(ByVal Session As Integer,
ByVal BMPHandle As Integer, ByVal FileName As
String, ByVal Quality As Integer)
```

Delphi Declaration

```
procedure IO_SavePDFImage(SessionHandle:Integer;
DIBHandle:Integer; FileName:PAnsiChar;
Quality:Integer); stdcall;
```

Java Declaration

```
void IO_SavePDFImage(int SessionHandle, int
DIBHandle, String FileName, int JPEGQuality);
```

Description

This is the function performing the saving of an image to a file in PDF format.

Parameters

SessionHandle (in) - the session handle to use
DIBHandle (in) - the handle of the DIB to save
FileName(in) - the complete file name path of the file to save
JPEGQuality(in) - the jpeg quality for color images when compressed, between 0 and 100. Use 80 as default.

Return values

n/a

3.35 IO_SaveTIFImage

C/C++ Declaration

```
__stdcall void IO_SaveTIFImage(long SessionHandle,
long DIBHandle, char* FileName, long Compression);
```

C# Declaration

```
void IO_SaveTIFImage(int SessionHandle, int
DIBHandle, string FileName, int Compression);
```

Visual Basic Declaration

```
Sub IO_SaveTIFImage(ByVal Session As Integer,
ByVal BMPHandle As Integer, ByVal FileName As
String, ByVal Compression As Integer)
```

Visual Basic .NET Declaration

```
Sub IO_SaveTIFImage(ByVal Session As Integer,
ByVal BMPHandle As Integer, ByVal FileName As
String, ByVal Compression As Integer)
```

Delphi Declaration

```
procedure IO_SaveTIFImage(SessionHandle:Integer;
DIBHandle:Integer; FileName:PAnsiChar;
```

```
Compression:Integer); stdcall;
```

Java Declaration

```
void IO_SaveTIFImage(int SessionHandle, int
DIBHandle, String FileName, int Compression);
```

Description

This is the function performing the saving of an image to a file in TIF format.

Parameters

SessionHandle (in) - the session handle to use
DIBHandle (in) - the handle of the DIB to save
FileName(in) - the complete file name path of the file to save
Compression(in) - the compression schema to use: 0=default, 1=none, 2=packbit, 3=huffman, 4=G3, 5=G4, 6=ZLIB, 7=JPEG, 8=LZW

Return values

n/a

3.36 IO_ScanImage

C/C++ Declaration

```
__stdcall long IO_ScanImage(long SessionHandle,
long WinHandle);
```

C# Declaration

```
int IO_ScanImage(int SessionHandle, int
WinHandle);
```

Visual Basic Declaration

```
Function IO_ScanImage(ByVal Session As Integer,
ByVal WinHandle As Integer) As Integer
```

Visual Basic .NET Declaration

```
Function IO_ScanImage(ByVal Session As Integer,
ByVal WinHandle As Integer) As Integer
```

Delphi Declaration

```
function IO_ScanImage(SessionHandle:Integer;
WinHandle:Integer):integer; stdcall;
```

Java Declaration

```
int IO_ScanImage(int SessionHandle, int
WinHandle);
```

Description

Start the image acquisition using default twain data source. The application show the scanner driver dialog where the user can select scanning parameters and can start acquisition.

Parameters

SessionHandle (in) - the session handle to use
WinHandle (in) - the handle of window where the scanner dialog has to be displayed

Return values

The DIB handle of the scanned image. If more that one images are scanned (as using an ADF), only the last image handle will be returned. All other handles will be passed to the application using callback method (see IO_SetScannerCallback)

3.37 IO_ScanImageAsync

C/C++ Declaration

```
__stdcall long IO_ScanImageAsync(long
SessionHandle, long WinHandle);
```

C# Declaration

```
int IO_ScanImageAsync(int SessionHandle, int
WinHandle);
```

Visual Basic Declaration

```
Function IO_ScanImageAsync(ByVal Session As
Integer, ByVal WinHandle As Integer) As Integer
```

Visual Basic .NET Declaration

```
Function IO_ScanImageAsync(ByVal Session As
Integer, ByVal WinHandle As Integer) As Integer
```

Delphi Declaration

```
function IO_ScanImageAsync(SessionHandle:Integer;
WinHandle:Integer):integer; stdcall;
```

Java Declaration

```
int IO_ScanImageAsync(int SessionHandle, int
WinHandle);
```

Description

Start the image acquisition using default twain data source. The application show the scanner driver dialog where the user can select scanning parameters and can start acquisition.

Parameters

SessionHandle (in) - the session handle to use
WinHandle (in) - the handle of window where the scanner dialog has to be displayed

Return values

None

3.38 IO_ScanImageAsyncWithParameters

C/C++ Declaration

```
__stdcall long
IO_ScanImageAsyncWithParameters(long
SessionHandle, long BitsPerPixel, long Resolution,
long Contrast, long Brightness, long Width, long
Height, long ADF, long Duplex);
```

C# Declaration

```
IO_ScanImageAsyncWithParameters(int SessionHandle,
int BitsPerPixel, int Resolution, int Contrast,
```

```
int Brightness, double Width, double Height,int
ADF, int Duplex);
```

Visual Basic Declaration

```
Function IO_ScanImageAsyncWithParameters(ByVal
Session As Integer, ByVal BitsPerPixel As Integer,
ByVal Resolution As Integer, ByVal Contrast As
Integer, ByVal Brightness As Integer, ByVal Width
As Double, ByVal Height As Double, ByVal ADF As
Integer, ByVal Duplex As Integer) As Integer
```

Visual Basic .NET Declaration

```
Function IO_ScanImageAsyncWithParameters(ByVal
Session As Integer, ByVal BitsPerPixel As Integer,
ByVal Resolution As Integer, ByVal Contrast As
Integer, ByVal Brightness As Integer, ByVal Width
As Double, ByVal Height As Double, ByVal ADF As
Integer, ByVal Duplex As Integer) As Integer
```

Delphi Declaration

```
function
IO_ScanImageAsyncWithParameters(SessionHandle:Inte
ger;
BitsPerPixel,Resolution,Contrast,Brightness:Intege
r; Width,Height:Double;
ADF,Duplex:Integer):integer; stdcall;
```

Java Declaration

```
IO_ScanImageAsyncWithParameters(int SessionHandle,
int BitsPerPixel, int Resolution, int Contrast,
int Brightness, double Width, double Height,int
ADF, int Duplex);
```

Description

Start the image acquisition using default twain data source. The application setup the scanner driver with passed parameters so that no driver dialog is displayed.

Parameters

SessionHandle (in) - the session handle to use

BitsPerPixel (in) - the bits per pixel of required image. Valid parameter are 1= monochrome, 8=grayscale, 24=true-color
Resolution (in) - the resolution in DPI of required image. See your scanner specification to know valid values.
Contrast (in) - the contrast of required image. Valid values are from –1000 to 1000. Default value is 0.
Brightness (in) - the brightness of required image. Valid values are from –1000 to 1000. Default value is 0.
Width (in) - the width in points of image to acquire (1 point=1/72 inch)
Height (in) - the height in points of image to acquire (1 point=1/72 inch)
ADF (in) – option to enable/disable ADF usage. Valid values are 0=use flatbed and 1=use ADF
Duplex (in) – option to enable/disable duplex scanning. Valid values are 0=simplex 1=duplex.

Return values

None

3.39 IO_ScanImageWithParameters

C/C++ Declaration

```
__stdcall long IO_ScanImageWithParameters(long
SessionHandle, long BitsPerPixel, long Resolution,
long Contrast, long Brightness, long Width, long
Height, long ADF, long Duplex);
```

C# Declaration

```
int IO_ScanImageWithParameters(int SessionHandle,
int BitsPerPixel, int Resolution, int Contrast,
int Brightness, double Width, double Height,int
ADF, int Duplex);
```

Visual Basic Declaration

```
Function IO_ScanImageWithParameters(ByVal Session
As Integer, ByVal BitsPerPixel As Integer, ByVal
Resolution As Integer, ByVal Contrast As Integer,
ByVal Brightness As Integer, ByVal Width As
Double, ByVal Height As Double, ByVal ADF As
Integer, ByVal Duplex As Integer) As Integer
```

Visual Basic .NET Declaration

```
Function IO_ScanImageWithParameters(ByVal Session
As Integer, ByVal BitsPerPixel As Integer, ByVal
Resolution As Integer, ByVal Contrast As Integer,
ByVal Brightness As Integer, ByVal Width As
Double, ByVal Height As Double, ByVal ADF As
Integer, ByVal Duplex As Integer) As Integer
```

Delphi Declaration

```
function
IO_ScanImageWithParameters(SessionHandle:Integer;
BitsPerPixel,Resolution,Contrast,Brightness:Intege
r; Width,Height:Double;
ADF,Duplex:Integer):integer; stdcall;
```

Java Declaration

```
int IO_ScanImageWithParameters(int SessionHandle,
int BitsPerPixel, int Resolution, int Contrast,
int Brightness, double Width, double Height,int
ADF, int Duplex);
```

Description

Start the image acquisition using default twain data source. The application setup the scanner driver with passed parameters so that no driver dialog is displayed.

Parameters

SessionHandle (in) - the session handle to use
BitsPerPixel (in) - the bits per pixel of required image. Valid parameter are 1= monochrome, 8=grayscale, 24=true-color
Resolution (in) - the resolution in DPI of required image. See your scanner specification to know valid values.
Contrast (in) - the contrast of required image. Valid values are from –1000 to 1000. Default value is 0.
Brightness (in) - the brightness of required image. Valid values are from –1000 to 1000. Default value is 0.
Width (in) - the width in points of image to acquire (1 point=1/72 inch)
Height (in) - the height in points of image to acquire (1 point=1/72 inch)
ADF (in) – option to enable/disable ADF usage. Valid values are

0=use flatbed and 1=use ADF
Duplex (in) – option to enable/disable duplex scanning. Valid values are 0=simplex 1=duplex.

Return values

The DIB handle of the scanned image. If more that one images are scanned (as using an ADF), only the last image handle will be returned. All other handles will be passed to the application using callback method (see IO_SetScannerCallback)

3.40 IO_SelectScanner

C/C++ Declaration

```
__stdcall void IO_SelectScanner(long
SessionHandle, long WinHandle);
```

C# Declaration

```
void IO_SelectScanner(int SessionHandle, int
WinHandle);
```

Visual Basic Declaration

```
Sub IO_SelectScanner(ByVal Session As Integer,
ByVal WinHandle As Integer)
```

Visual Basic .NET Declaration

```
Sub IO_SelectScanner(ByVal Session As Integer,
ByVal WinHandle As Integer)
```

Delphi Declaration

```
procedure IO_SelectScanner(SessionHandle:Integer;
WinHandle:Integer); stdcall;
```

Java Declaration

```
void IO_SelectScanner(int SessionHandle, int
WinHandle);
```

Description

Display the TWAIN device selection dialog, allowing user to select

default source to use to acquire images.

Parameters

SessionHandle (in) - the session handle to use
WinHandle (in) - the window handle used as parent of selection dialog

Return values

n/a

3.41 IO_SetParameter

C/C++ Declaration

```
__stdcall void IO_SetParameter(long SessionHandle,
long ParameterIndex, long ParameterValue);
```

C# Declaration

```
IO_SetParameter(int SessionHandle, int
ParameterIndex, int ParameterValue);
```

Visual Basic Declaration

```
Sub IO_SetParameter(ByVal Session As Integer,
ByVal ParameterIndex As Integer, ByVal
ParameterValue As Integer)
```

Visual Basic .NET Declaration

```
Sub IO_SetParameter(ByVal Session As Integer,
ByVal ParameterIndex As Integer, ByVal
ParameterValue As Integer)
```

Delphi Declaration

```
procedure IO_SetParameter(SessionHandle:Integer;
ParameterIndex:Integer; ParameterValue:Integer);
stdcall;
```

Java Declaration

```
IO_SetParameter(int SessionHandle, int
ParameterIndex, int ParameterValue);
```

Description

Set the silent mode parameter.

Parameters

SessionHandle (in) - the session handle to use
ParameterIndex(in) - the only index to use is 20
ParameterValue(in) - Silent Mode 0 (disable, default), 1 (enabled)

Return values

None

3.42 IO_SetPixel

C/C++ Declaration

```
__stdcall void IO_SetPixel(long SessionHandle,
long DIBHandle, long X, long Y,long C);
```

C# Declaration

```
IO_SetPixel(int SessionHandle, int DIBHandle, int
X, int Y, int C);
```

Visual Basic Declaration

```
Sub IO_SetPixel(ByVal Session As Integer,  Byval
DIBHandle As Integer, Byval X As Integer, Byval Y
As Integer, Byval C As Integer)
```

Visual Basic .NET Declaration

```
Sub IO_SetPixel(ByVal Session As Integer, Byval
DIBHandle As Integer, Byval X As Integer, Byval Y
As Integer, Byval C As Integer)
```

Delphi Declaration

```
procedure IO_SetPixel(SessionHandle:Integer;
DIBHandle,X,Y,C:Integer):Integer; stdcall;
```

Java Declaration

```
IO_SetPixel(int SessionHandle, int DIBHandle, int
```

```
X, int Y, int C);
```

Description

Gets the color of a specific pixel element using x and y coordinates

Parameters

SessionHandle (in) - the session handle to use
intImageHandle: integer value corresponding to the image handle.
intX: integer value corresponding to the column.
intY: integer value corresponding to the line.
int C: Numerical value showing the indicated pixel color

Return values

None

3.43 IO_SetScannerCallback

C/C++ Declaration

```
__stdcall void IO_SetScannerCallback(long
SessionHandle, void *CallBack);
```

C# Declaration

```
void IO_SetScannerCallback(int SessionHandle,
Delegate CallbackProc);
```

Visual Basic Declaration

```
Sub IO_SetScannerCallback(ByVal Session As
Integer, ByVal CallbackProc As Integer)
```

Visual Basic .NET Declaration

```
Sub IO_SetScannerCallback(ByVal Session As
Integer, byval CallbackProc As [Delegate])
```

Delphi Declaration

```
procedure
IO_SetScannerCallback(SessionHandle:Integer;
Callback:Pointer); stdcall;
```

Java Declaration

```
void IO_SetScannerCallback(int SessionHandle,
Delegate CallbackProc);
```

Description

Set the scanner callback method required to receive images from scanners when multiple images are acquired. The user has to provide a callback function, called by the library with the DIB handle as parameter. The application has to free the DIB handle after usage. The callback function has to be declared as: void MyCallaback(long DIBHandle)

Parameters

SessionHandle (in) - the session handle to use
CallbackProc (in) - the callback procedure or null if not callback is required

Return values

n/a

3.44 IO_ShowImage

C/C++ Declaration

```
__stdcall void IO_ShowImage(long SessionHandle,
long DIBHandle, long DC, long X, long Y);
```

C# Declaration

```
void IO_ShowImage(int SessionHandle, int
DIBHandle, int DC, int X, int Y);
```

Visual Basic Declaration

```
Sub IO_ShowImage(ByVal Session As Integer, ByVal
BMPHandle As Integer, ByVal DC As Integer, ByVal X
As Integer, ByVal Y As Integer)
```

Visual Basic .NET Declaration

```
Sub IO_ShowImage(ByVal Session As Integer, ByVal
```

```
BMPHandle As Integer, ByVal DC As Integer, ByVal X
As Integer, ByVal Y As Integer)
```

Delphi Declaration

```
procedure IO_ShowImage(SessionHandle:Integer;
DIBHandle:Integer; DC:Integer; X,Y:Integer);
stdcall;
```

Java Declaration

```
void IO_ShowImage(int SessionHandle, int
DIBHandle, int DC, int X, int Y);
```

Description

Display an image in a draw context.

Parameters

SessionHandle (in) - the session handle to use
DIBHandle(in) - the DIB handle of the image to use.
DC(in) - the draw context where to display the image.
X (in) - the x coordinates where to display the image
Y (in) - the y coordinates where to display the image

Return values

n/a

3.45 IO_ShowStretchedImage

C/C++ Declaration

```
__stdcall void IO_ShowStretchedImage(long
SessionHandle, long DIBHandle, long DC, long X,
long Y, long W, long H);
```

C# Declaration

```
void IO_ShowStretchedImage(int SessionHandle, int
DIBHandle, int DC, int X, int Y, int W, int H);
```

Visual Basic Declaration

```
Sub IO_ShowStretchedImage(ByVal Session As
```

```
Integer, ByVal BMPHandle As Integer, ByVal DC As
Integer, ByVal X As Integer, ByVal Y As Integer,
ByVal W As Integer, ByVal H As Integer)
```

Visual Basic .NET Declaration

```
Sub IO_ShowStretchedImage(ByVal Session As
Integer, ByVal BMPHandle As Integer, ByVal DC As
Integer, ByVal X As Integer, ByVal Y As Integer,
ByVal W As Integer, ByVal H As Integer)
```

Delphi Declaration

```
procedure
IO_ShowStretchedImage(SessionHandle:Integer;
DIBHandle:Integer; DC:Integer; X,Y,W,H:Integer);
stdcall;
```

Java Declaration

```
void IO_ShowStretchedImage(int SessionHandle, int
DIBHandle, int DC, int X, int Y, int W, int H);
```

Description

Display an image in a draw context, stretching it.

Parameters

SessionHandle (in) - the session handle to use
DIBHandle(in) - the DIB handle of the image to use.
DC(in) - the draw context where to display the image.
X (in) - the x coordinates where to display the image
Y (in) - the y coordinates where to display the image
W (in) - the width of the image stretched
H(in) - the height of the image stretched

Return values

n/a

3.46 IO_ScanImageTSPFile

C/C++ Declaration

```
__stdcall long IO_ScanImageTSPFile(long
SessionHandle, char* TSPFile);
```

C# Declaration

```
int IO_ScanImageTSPFile(int SessionHandle,string
TSPFile);
```

Visual Basic Declaration

```
Function IO_ScanImageTSPFile (ByVal Session As
Long, ByVal TSPFile As String) As Long
```

Visual Basic .NET Declaration

```
Function IO_ScanImageTSPFile(ByVal Session As
Integer, ByVal TSPFile As String) As Integer
```

Delphi Declaration

```
function
IO_ScanImageTSPFile(SessionHandle:Integer;TSPFile:
PAnsiChar):integer; stdcall;
```

Java Declaration

```
int IO_ScanImageTSPFile(int SessionHandle,string
TSPFile);
```

Description

Start the image acquisition using the .TSP parameters file of Recogniform PaperCapture

Parameters

SessionHandle (in) - the session handle to use
TSPFile(in) - the full path of the .tsp file containing the scanning parameters.

Return values

n/a

3.47 IO_ScanImageTSPRaw

C/C++ Declaration

```
__stdcall long IO_ScanImageTSPRaw(long
SessionHandle, char* TSPRaw);
```

C# Declaration

```
int IO_ScanImageTSPRaw(int SessionHandle, string
TSPRaw);
```

Visual Basic Declaration

```
Function IO_ScanImageTSPRaw (ByVal Session As
Long, ByVal TSPRaw As String) As Long
```

Visual Basic .NET Declaration

```
Function IO_ScanImageTSPRaw(ByVal Session As
Integer, ByVal TSPRaw As String) As Integer
```

Delphi Declaration

```
function
IO_ScanImageTSPRaw(SessionHandle:Integer;TSPRaw:PA
nsiChar):integer; stdcall;
```

Java Declaration

```
int IO_ScanImageTSPRaw(int SessionHandle, String
TSPRaw);
```

Description

Start the image acquisition using the raw of the parameters.

Parameters

SessionHandle (in) - the session handle to use
TSPRaw(in) - the full path of the .tsp file containing the scanning parameters.

Return values

n/a

Note

Use the carriage return to separate the parameter

3.48 IO_ScanImageAsyncTSPFile

C/C++ Declaration

```
__stdcall long IO_ScanImageAsyncTSPFile(long
SessionHandle, char* TSPFile);
```

C# Declaration

```
int IO_ScanImageAsyncTSPFile(int Session,string
TSPFile);
```

Visual Basic Declaration

```
Function IO_ScanImageAsyncTSPFile (ByVal Session
As Long, ByVal TSPFile As String) As Long
```

Visual Basic .NET Declaration

```
Function IO_ScanImageAsyncTSPFile(ByVal Session As
Integer, ByVal TSPFile As String) As Integer
```

Delphi Declaration

```
function
IO_ScanImageAsyncTSPFile(SessionHandle:Integer;TSP
File:PAnsiChar):integer; stdcall;
```

Java Declaration

```
int IO_ScanImageAsyncTSPFile(int Session,String
TSPFile);
```

Description

Start the image acquisition in asynchronous mode using the .TSP parameters file of Recogniform PaperCapture

Parameters

SessionHandle (in) - the session handle to use
TSPFile(in) - the full path of the .tsp file containing the scanning parameters.

Return values

n/a

3.49 IO_ScanImageAsyncTSPRaw

C/C++ Declaration

```
__stdcall long IO_ScanImageAsyncTSPRaw(long
SessionHandle, char* TSPRaw);
```

C# Declaration

```
int IO_ScanImageAsyncTSPRaw(int Session, string
TSPRaw);
```

Visual Basic Declaration

```
Function IO_ScanImageAsyncTSPRaw(ByVal Session As
Long, ByVal TSPRaw As String) As Long
```

Visual Basic .NET Declaration

```
Function IO_ScanImageAsyncTSPRaw(ByVal Session As
Integer, ByVal TSPRaw As String) As Integer
```

Delphi Declaration

```
function
IO_ScanImageAsyncTSPRaw(SessionHandle:Integer;TSPR
aw:PAnsiChar):integer; stdcall;
```

Java Declaration

```
int IO_ScanImageAsyncTSPRaw(int Session, String
TSPRaw);
```

Description

Start the image acquisition in asynchronous mode using the raw of the parameters.

Parameters

SessionHandle (in) - the session handle to use
TSPRaw(in) - the full path of the .tsp file containing the scanning parameters.

Return values

n/a

Note

Use the carriage return to separate the parameter

3.50 IO_MakeSearchablePDF

C/C++ Declaration

```
__stdcall void IO_MakeSearchablePDF(long
SessionHandle, char* TifFileIn,char*
XMLFileIn,char* PDFFileOut,char* PDFInfo);
```

C# Declaration

```
void IO_MakeSearchablePDF(int Session,string
TifFileIn,string XMLFileIn,string
PDFFileOut,string PDFInfo);
```

Visual Basic Declaration

```
Sub IO_MakeSearchablePDF(ByVal Session As Long,
ByVal TifFileIn As String, ByVal XMLFileIn As
String, ByVal PDFFileOut As String, ByVal PDFInfo
As String)
```

Visual Basic .NET Declaration

```
Sub IO_MakeSearchablePDF(ByVal Session As Integer,
ByVal TifFileIn As String, ByVal XMLFileIn As
String, ByVal PDFFileOut As String, ByVal PDFInfo
As String)
```

Delphi Declaration

```
procedure
IO_MakeSearchablePDF(SessionHandle:Integer;TifFile
```

```
In,XMLFileIn,PDFFileOut,PDFInfo:PAnsiChar);
stdcall;
```

Java Declaration

```
void IO_MakeSearchablePDF(int Session,String
TifFileIn,String XMLFileIn,String
PDFFileOut,String PDFInfo);
```

Description

Make an PDF image adding on its searchable hidden text.

Parameters

SessionHandle (in) - the session handle to use
XMLFileIn(in) - the full path of the XML containing text to insert..
PDFFileOut(in) - the full path of the output file name.
PDFInfo (in) - the pdf info

Return values

n/a

3.51 IO_GetScannerStatus

C/C++ Declaration

```
__stdcall long IO_GetScannerStatus(long
SessionHandle);
```

C# Declaration

```
int IO_GetScannerStatus(int Session);
```

Visual Basic Declaration

```
Function IO_GetScannerStatus(ByVal Session As
Long) As Long
```

Visual Basic .NET Declaration

```
Function IO_GetScannerStatus(ByVal Session As
Integer) As Integer
```

Delphi Declaration

```
function
IO_GetScannerStatus(SessionHandle:Integer):integer
; stdcall;
```

Java Declaration

```
int IO_GetScannerStatus(int Session);
```

Description

Test the scanner status. This function is useful when the scanning is asynchronous or if the user interface is not used

Parameters

SessionHandle (in) - the session handle to use

Return values
Code of status

- 1 = Pre-Session
- 2 = Source Manager Loaded
- 3 = Source Manager Opened
- 4 = Source Opened
- 5 = Source Enabled
- 6 = Transfer Ready
- 7 = Transferring

3.52 IO_GetScannerName

C/C++ Declaration

```
__stdcall long IO_GetScannerName(long
SessionHandle, char* ScannerName);
```

C# Declaration

```
int IO_GetScannerName(int SessionHandle,
Stringbuilder ScannerName);
```

Visual Basic Declaration

```
Function IO_GetScannerName(ByVal Session As
Integer, ByVal Key As String) As Integer
```

Visual Basic .NET Declaration

```
Function IO_GetScannerName(ByVal Session As
Integer,  ByVal Key As Stringbuilder) As Integer
```

Delphi Declaration

```
function IO_GetScannerName(SessionHandle:Integer;
ScannerName:PAnsiChar):integer; stdcall;
```

Java Declaration

```
int IO_GetScannerName(int SessionHandle, String
ScannerName);
```

Description

Get scanner name

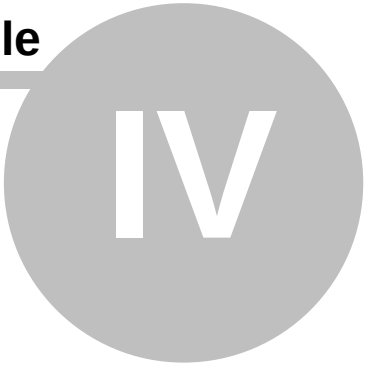
Parameters

SessionHandle (in) - the session handle to use
ScannerName (out)- buffer containing the scanner name

Return values

Return the buffer length

Sample



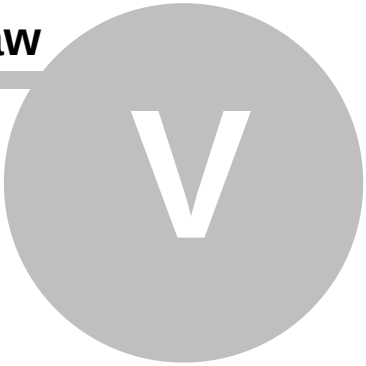
4 Sample

4.1 Code Sample

This sample load the test.tif file from SDK installation version and copy it to clipboard.

```
#include "stdafx.h"
#include "recoio.c"
int     APIENTRY     WinMain(HINSTANCE     hInstance,HINSTANCE
hPrevInstance, LPSTR lpCmdLine,int nCmdShow)
{
 // Load dynamically the library
 LoadIOLibrary();
 // Init the IO session
 int Session= IO_Init("demo", "demo");
 long hBitmap;
 hBitmap = 0;
 // Load a sample TIF file from disk
 hBitmap =
 IO_LoadTIFImage(Session,".\\test.tif",0);
 if (hBitmap>0)
 {
  // Open the Clipboard
  ::OpenClipboard(NULL);
  // Copy the DIB to the clipboard
  ::SetClipboardData(CF_DIB,(void*)hBitmap);
  // Close the Clipboard
  ::CloseClipboard();
  MessageBox(NULL, "Image copied in clipboard !", "INFO",
MB_OK);
 }
 // Show an error message
 else MessageBox(NULL, "Unable to load the image", "ERROR",
MB_OK);
 // Close the session
 IO_Done(Session);
 // Unload the library
 FreeIOLibrary();
 return 0;
}
```

IO_ScanImageTSPRaw



5 IO_ScanImageTSPRaw

C/C++ Declaration

```
__stdcall long IO_ScanImageTSPRaw(long
SessionHandle, char* TSPRaw);
```

C# Declaration

```
int IO_ScanImageTSPRaw(int SessionHandle, string
TSPRaw);
```

Visual Basic Declaration

```
Function IO_ScanImageTSPRaw (ByVal Session As
Long, ByVal TSPRaw As String) As Long
```

Visual Basic .NET Declaration

```
Function IO_ScanImageTSPRaw(ByVal Session As
Integer, ByVal TSPRaw As String) As Integer
```

Delphi Declaration

```
function
IO_ScanImageTSPRaw(SessionHandle:Integer;TSPRaw:PA
nsiChar):integer; stdcall;
```

Java Declaration

```
int IO_ScanImageTSPRaw(int SessionHandle, String
TSPRaw);
```

Description

Start the image acquisition using the raw of the parameters.

Parameters

SessionHandle (in) - the session handle to use
TSPRaw(in) - the full path of the .tsp file containing the scanning parameters.

Return values

n/a

Note

Use the carriage return to separate the parameter

Annotation