

**Investintech.com Inc.
Software Development Kit:
PDFtoImage Function Library
User's Guide**

Novemebr 6, 2007
<http://www.investintech.com>

Copyright 2007 Investintech.com, Inc. All rights reserved

Adobe® is registered by Adobe Systems Incorporated
Acrobat® is registered by Adobe Systems Incorporated
Postscript® is registered by Adobe Systems Incorporated
Access™ is registered by Microsoft Corporation
Excel® is registered by Microsoft Corporation
Visual Basic® is registered by Microsoft Corporation
Visual C++® is registered by Microsoft Corporation
Visual C#® is registered by Microsoft Corporation
Visual J#® is registered by Microsoft Corporation
Visual Studio® is registered by Microsoft Corporation
Win32® is registered by Microsoft Corporation
Windows® is registered by Microsoft Corporation
Windows NT® is registered by Microsoft Corporation
Windows Vista™ is registered by Microsoft Corporation

Contents

Preface	1
About This Documentation.....	1
Typographical Conventions Used in This Document.....	1
Getting More Information.....	2
Customer Service and Technical Support.....	2
Fax and Mailing Address	2
Investintech PDFtoImage Conversion DLL.....	3
What is the Investintech PDFtoImage Conversion DLL?	4
Installation Instructions for PDFtoImageDLL.....	4
System Requirements.....	4
Starting the Installation	4
Using the Investintech PDFtoImage Conversion DLL in Microsoft Visual C++ .NET 2003.....	11
Implicit Linking	11
Linking .lib file with project	12
Using methods from DLL file.....	13
Using the Investintech PDFtoImage Conversion DLL in Microsoft Visual C++ .NET 2003.....	13
Investintech PDFtoImage Conversion DLL Methods	14
Interface.....	14
Parameter Type	14
File Names	15
Error Handling	15
Common Sample Source Code	15
<i>Conversion from PDF document to Image document</i>	16
Conversion to Bitmap Images.....	16
PDF_to_BMP_CBR.....	16
Prototype.....	16
Description.....	16
Calling Convention	16
Parameters.....	16
Returns	16
Example	16
PDF_to_BMP_CBT	17
Prototype.....	17
Description.....	17
Calling Convention	17
Parameters.....	17
Returns	17
Example	17
PDF_to_BMP_CLR	18
Prototype.....	18
Description.....	18
Calling Convention	18
Parameters.....	18
Returns	18
Example	18
PDF_to_BMP_CLT	18

Prototype	18
Description	19
Calling Convention	19
Parameters	19
Returns	19
Example	19
PDF_to_BMP_SBR	19
Prototype	19
Description	20
Calling Convention	20
Parameters	20
Returns	20
Example	20
PDF_to_BMP_SBT	20
Prototype	20
Description	20
Calling Convention	20
Parameters	21
Returns	21
Example	21
PDF_to_BMP_SLR	21
Prototype	21
Description	21
Calling Convention	21
Parameters	22
Returns	22
Example	22
PDF_to_BMP_SLT	22
<i>Prototype</i>	22
Description	22
Calling Convention	22
Parameters	22
Returns	23
Example	23
Conversion to GIF Images.....	23
PDF_to_GIF_CBR	23
Prototype	23
Description	23
Calling Convention	23
Parameters	23
Returns	24
Example	24
PDF_to_GIF_CBT	24
Prototype	24
Description	24
Calling Convention	24
Parameters	24
Returns	24
Example	25
PDF_to_GIF_CLR.....	25
Prototype	25
Description	25

Calling Convention	25
Parameters.....	25
Returns	25
Example	26
PDF_to_GIF_CLT	26
Prototype.....	26
Description.....	26
Calling Convention	26
Parameters.....	26
Returns	26
Example	26
PDF_to_GIF_SBR	27
Prototype.....	27
Description.....	27
Calling Convention	27
Parameters.....	27
Returns	27
Example	27
PDF_to_GIF_SBT	28
Prototype.....	28
Description.....	28
Calling Convention	28
Parameters.....	28
Returns	28
Example	28
PDF_to_GIF_SLR	29
Prototype.....	29
Description.....	29
Calling Convention	29
Parameters.....	29
Returns	29
Example	29
PDF_to_GIF_SLT.....	29
<i>Prototype</i>	29
Description.....	30
Calling Convention	30
Parameters.....	30
Returns	30
Example	30
Conversion to JPEG Images	31
PDF_to_JPG_CBR.....	31
Prototype.....	31
Description.....	31
Calling Convention	31
Parameters.....	31
Returns	31
Example	31
PDF_to_JPG_CBT.....	32
Prototype.....	32
Description.....	32
Calling Convention	32
Parameters.....	32

Returns	32
Example	32
PDF_to_JPG_CLR	33
Prototype	33
Description	33
Calling Convention	33
Parameters	33
Returns	33
Example	33
PDF_to_JPG_CLT	33
Prototype	33
Description	34
Calling Convention	34
Parameters	34
Returns	34
Example	34
PDF_to_JPG_SBR.....	34
Prototype	34
Description	35
Calling Convention	35
Parameters	35
Returns	35
Example	35
PDF_to_JPG_SBT	35
Prototype	35
Description	35
Calling Convention	35
Parameters	36
Returns	36
Example	36
PDF_to_JPG_SLR	36
Prototype	36
Description	36
Calling Convention	36
Parameters	37
Returns	37
Example	37
PDF_to_JPG_SLT	37
<i>Prototype</i>	37
Description	37
Calling Convention	37
Parameters	37
Returns	38
Example	38
Conversion to PNG Images	38
PDF_to_PNG_CBR.....	38
Prototype	38
Description	38
Calling Convention	38
Parameters	38
Returns	39
Example	39

PDF_to_PNG_CBT	39
Prototype.....	39
Description.....	39
Calling Convention.....	39
Parameters.....	39
Returns	39
Example	40
PDF_to_PNG_CLR	40
Prototype.....	40
Description.....	40
Calling Convention.....	40
Parameters.....	40
Returns	40
Example	41
PDF_to_PNG_CLT.....	41
Prototype.....	41
Description.....	41
Calling Convention.....	41
Parameters.....	41
Returns	41
Example	41
PDF_to_PNG_SBR.....	42
Prototype.....	42
Description.....	42
Calling Convention.....	42
Parameters.....	42
Returns	42
Example	42
PDF_to_PNG_SBT	42
Prototype.....	42
Description.....	43
Calling Convention.....	43
Parameters.....	43
Returns	43
Example	43
PDF_to_PNG_SLR.....	43
Prototype.....	43
Description.....	44
Calling Convention.....	44
Parameters.....	44
Returns	44
Example	44
PDF_to_PNG_SLT	44
<i>Prototype</i>	44
Description.....	44
Calling Convention.....	44
Parameters.....	45
Returns	45
Example	45
Conversion to TIFF Images	45
PDF_to_TIF_CBR	45
Prototype.....	45

Description	46
Calling Convention	46
Parameters	46
Returns	46
Example	46
PDF_to_TIF_CBT	46
Prototype	46
Description	46
Calling Convention	46
Parameters	47
Returns	47
Example	47
PDF_to_TIF_CLR	47
Prototype	47
Description	47
Calling Convention	47
Parameters	48
Returns	48
Example	48
PDF_to_TIF_CLT	48
Prototype	48
Description	48
Calling Convention	48
Parameters	48
Returns	49
Example	49
PDF_to_TIF_SBR	49
Prototype	49
Description	49
Calling Convention	49
Parameters	49
Returns	50
Example	50
PDF_to_TIF_SBT	50
Prototype	50
Description	50
Calling Convention	50
Parameters	50
Returns	50
Example	50
PDF_to_TIF_SLR	51
Prototype	51
Description	51
Calling Convention	51
Parameters	51
Returns	51
Example	51
PDF_to_TIF_SLT	52
<i>Prototype</i>	52
Description	52
Calling Convention	52
Parameters	52

Returns	52
Example	52
VB6 callable code	53
PDF_to_BMP_VB6	53
Prototype	53
Description	53
Calling Convention	53
Parameters	53
Returns	53
Example	53
PDF_to_GIF_VB6	54
Prototype	54
Description	54
Calling Convention	54
Parameters	54
Returns	54
Example	54
PDF_to_JPG_VB6	54
Prototype	54
Description	55
Calling Convention	55
Parameters	55
Returns	55
Example	55
PDF_to_PNG_VB6	55
Prototype	55
Description	55
Calling Convention	55
Parameters	56
Returns	56
Example	56
PDF_to_TIF_VB6	56
Prototype	56
Description	56
Calling Convention	56
Parameters	56
Returns	57
Example	57
InvestintechConversionDLL_IDString	57
Description	57
Parameters	57
Returns	57
Investintech PDFtoImage Conversion COM Server Methods	58
PDFtoBMP	58
Prototype	58
Description	58
Parameters	58
Returns	59
Example	59
PDFtoGIF	59
Prototype	59
Description	59

Parameters	59
Returns	59
Example	59
PDFtoJPG	60
Prototype	60
Description	60
Parameters	60
Returns	60
Example	60
PDFtoPNG.....	60
Prototype	60
Description	61
Parameters	61
Returns	61
Example	61
PDFtoTIF.....	61
Prototype	61
Description	61
Parameters	61
Returns	62
Example	62
Index	62

Preface

Welcome to Investintech.com Inc. Software Development Kit (SDK). This SDK provides technologies for converting PDF files into other file. This user's guide provides the basic information needed to use all parts of the SDK. This SDK consists of:

- Investintech PDF To Image Conversion DLL, a Dynamic-Link Library for converting PDF files to image formats

This document provides a comprehensive description of the DLL.

About This Documentation

This documentation consists of:

- Investintech PDF To Image Conversion DLL, describing usage and methods of DLL designed to convert PDF files into image formats

Typographical Conventions Used in This Document

The following table describes typographical and naming conventions used in this document:

<i>Font</i>	<i>Used for</i>	<i>Examples</i>
monospaced	Expected input from the user	pdftoimage -cbmp -iabc.pdf -oabc.txt
	Paths and filenames	C:\program files\investintech\pdftoim age\pdftoimage.exe
	Source code	MessageBox("ABC");
monospaced blue	Source code keywords	If
monospaced green	Source code comments	//this is a comment
monospaced bold	Method name	The AboutBox() method
'Single quoted bold'	Elements of the user interface like buttons, icons etc.	click on ' Start ' button, press ' Enter '
blue underlined	Internet links	http://www.investintech.com

Getting More Information

Investintech.com Inc. conducts research in data conversion technology and develops data conversion products utilizing results of the research. More information about our company is available at the company's website <http://www.investintech.com>.

Customer Service and Technical Support

Investintech.com Inc. strives to provide the best possible technical support to its customers and prospective customers. If you would like personal assistance, please feel free to call us or send e-mail to Customer Service or Technical Support. We are available by phone during regular business hours and in the vast majority of cases, we will return our e-mail with an answer the same day it is received.

	<i>Customer Service</i>	<i>Technical Support</i>
Telephone:	+1 416 920 5884	+1 416 920 2539
E-mail:	cs@investintech.com	techsupport@investintech.com
Hours:	Our Business Hours are Monday to Friday 9am-6pm (Eastern Time GMT-5:00).	

Fax and Mailing Address

Mailing Address: Investintech.com Inc.
600 - 425 University Avenue
Toronto, ON
M5G 1T6 Canada

Fax: +1 416 920 5848

Investintech.com Inc. Software Development Kit User's Guide

Investintech PDFtoImage Conversion DLL

What is the Investintech PDFtoImage Conversion DLL?

The Investintech PDFtoImage Conversion DLL is a collection of methods compiled, linked and stored in a dynamic-link library (DLL) file. These methods provide a set of methods for converting files from PDF to an image format.

Installation Instructions for PDFtoImageDLL

Below you will find step-by-step instructions to install PDFtoImageDLL Conversion Library on your system. You may want to print this instruction sheet for reference before beginning to install PDFtoImageDLL.

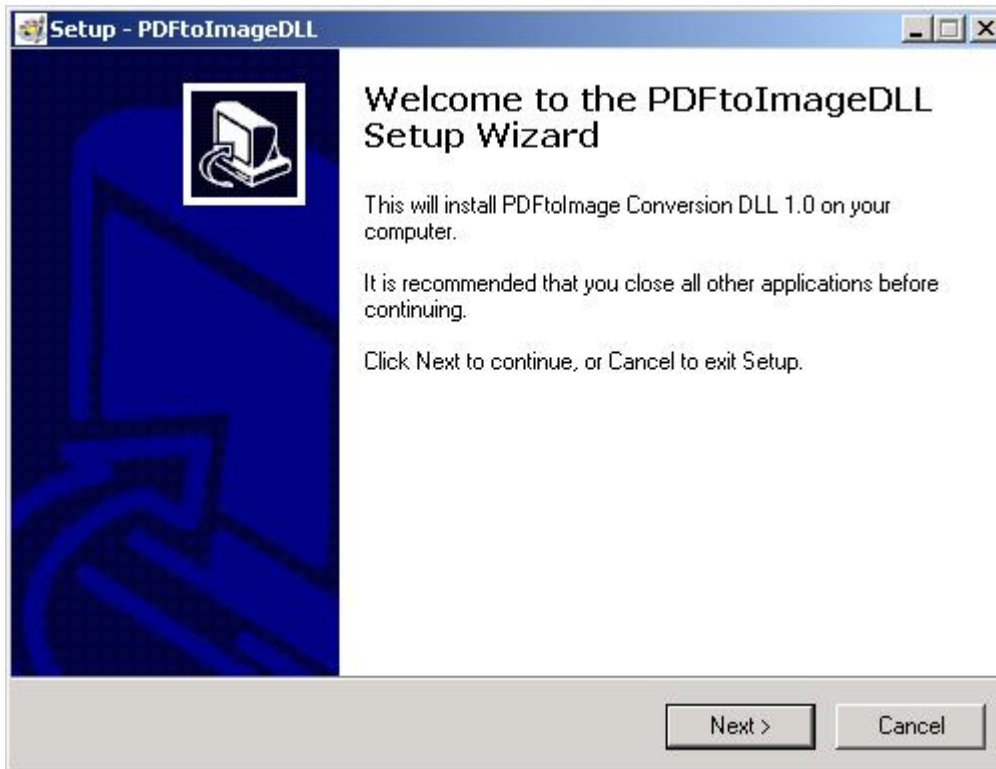
System Requirements

The minimum computer system resources required to install and use PDFtoImageDLL are:

- Microsoft Windows XP
- 64 MB of RAM
- 20 MB of available hard disk space
- A software development environment, such as Microsoft Visual Studio or Borland Delphi

Starting the Installation

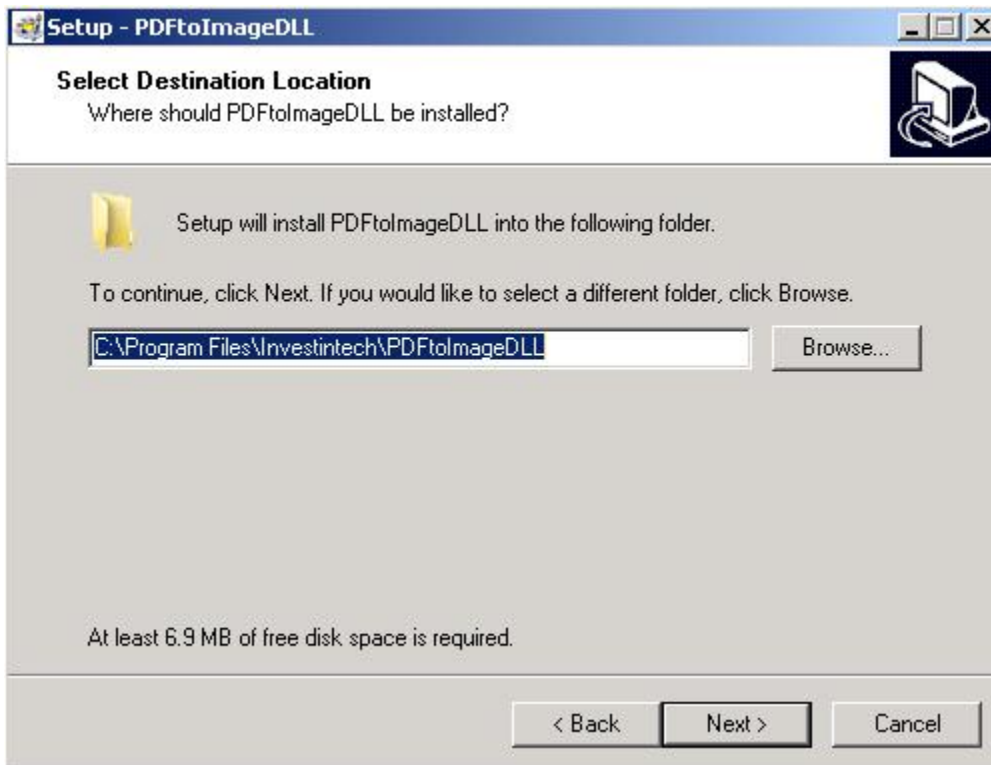
Start the installation by launching the `InstallPDFtoImageDLL.exe` file, e.g. by double clicking it in the Windows Explorer. After that you should see a window that looks like this:



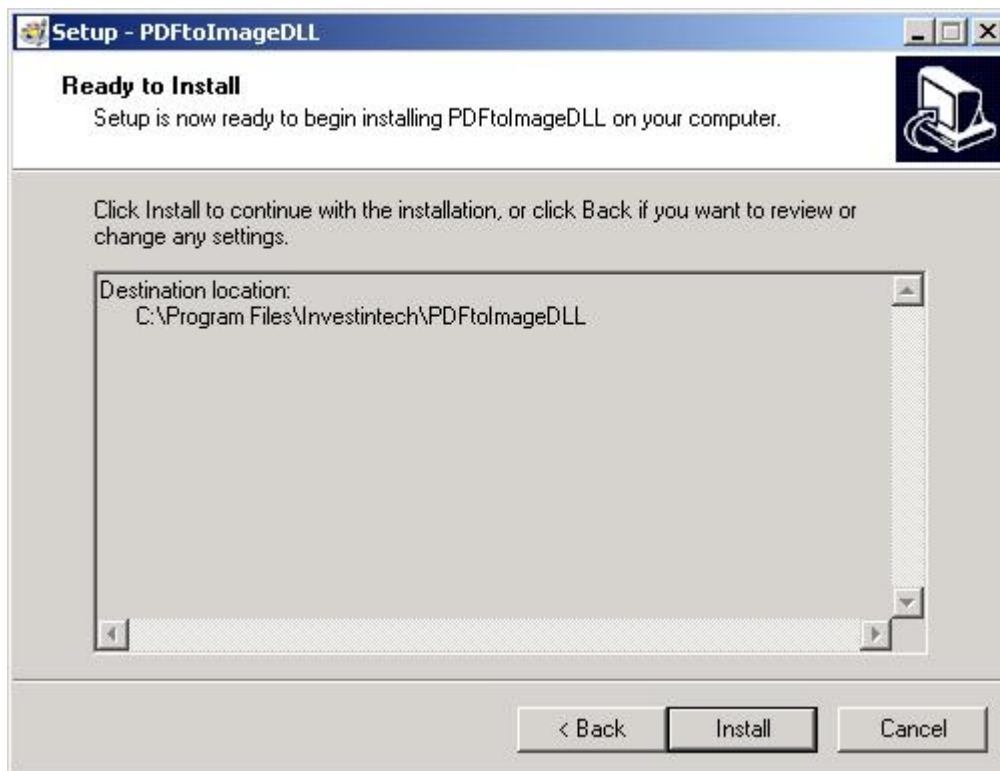
Click the 'Next' button to continue installation. You should see the next window:



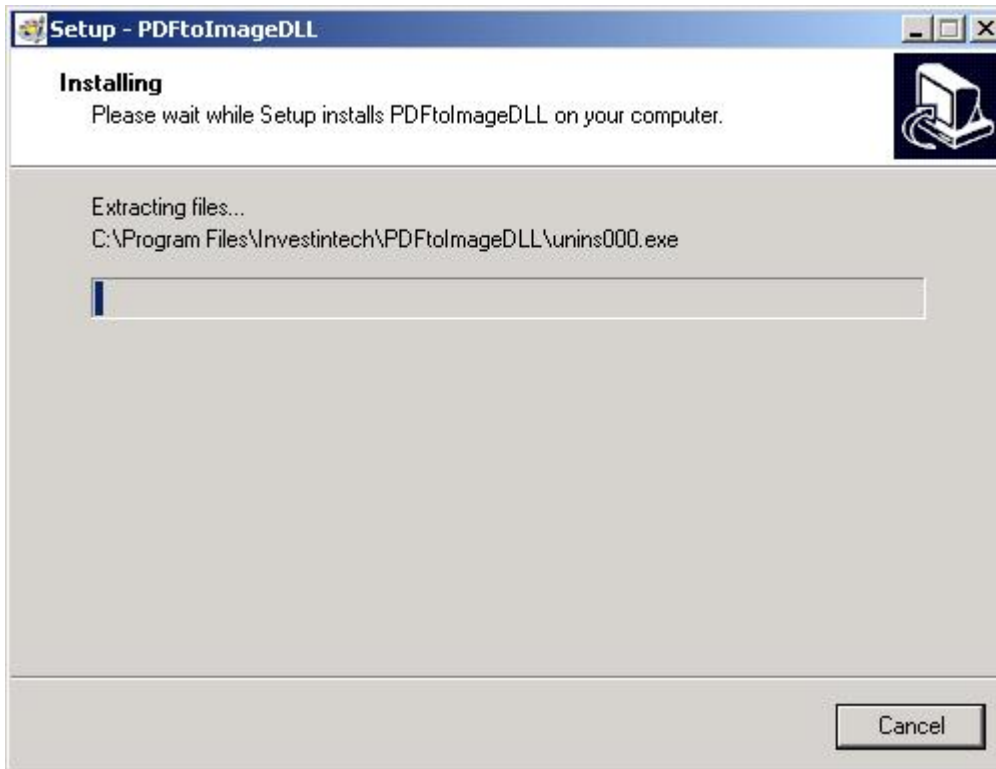
Please read the Investintech Software License Agreement carefully and thoroughly. If you wish to continue with installation, choose **‘I agree with the above terms and conditions’** and click **‘Next’**. Otherwise, click **‘Exit’** and installation will terminate. If you clicked **‘Next’** you should see the next window:



On this screen you can choose the installation directory for the PDFtoImageDLL software. The default installation folder will be contained in the `Program Files` folder on your system disk. It is possible to change the location of installation folder either by typing its path directly or by clicking the 'Browse...' button. If the destination folder (directory) you have chosen does not exist, the installation program will create it. Click the '**Next**' button to continue the installation.

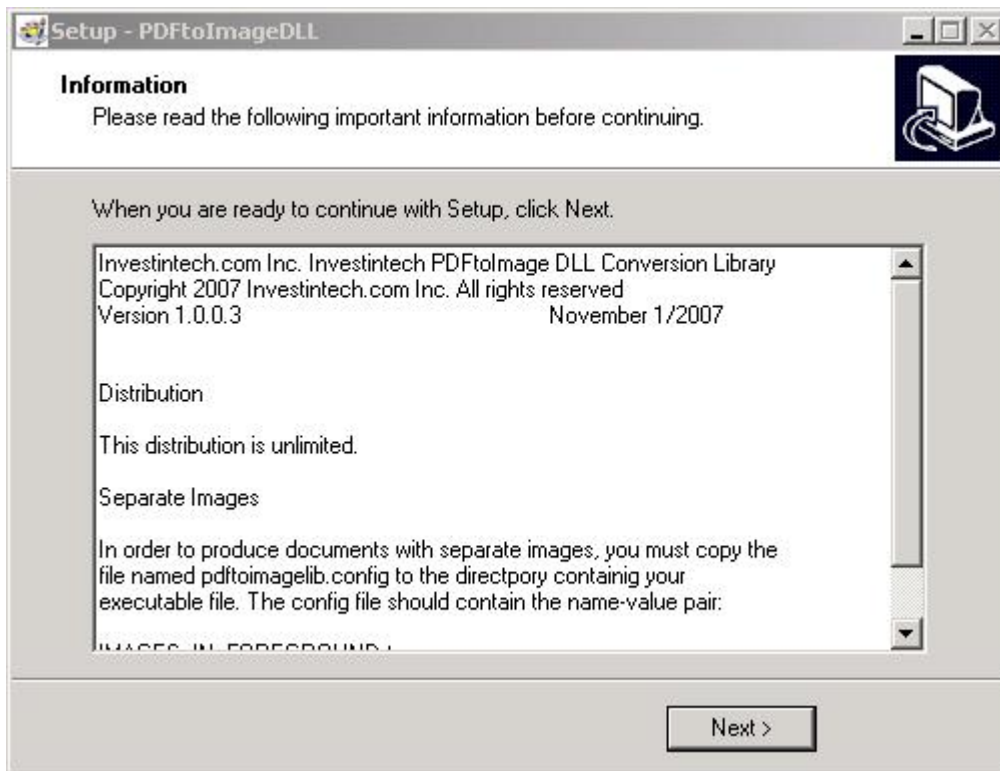


Clicking the **Install** button will start the installation.

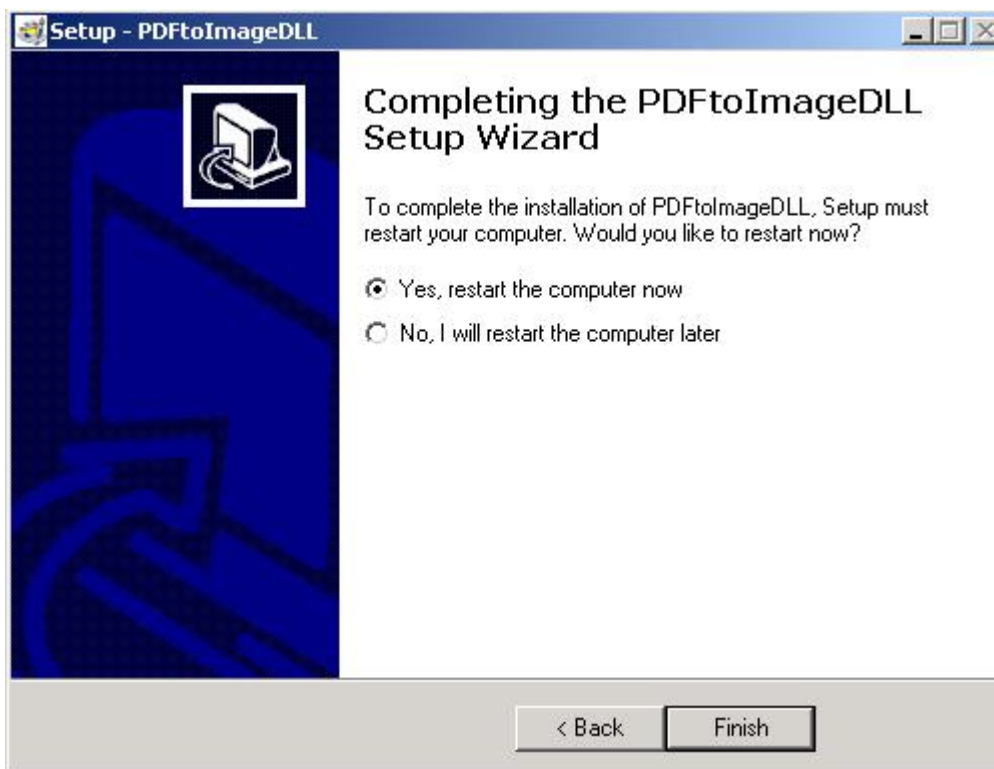


The installation will copy the files that comprise PDFtoImageDLL to the directory you specified earlier. After a very short period of time, you should see the next window:

PDFtoImageDLL – Investintech Conversion DLL



This window describes the distribution of PDFtoImageDLL being installed. Click **'Next'** to continue.



When this window appears, the installation process is complete. Click **Finish** to end the installation program.

Using the Investintech PDFtoImage Conversion DLL in Microsoft Visual C++ .NET 2003

In this section you will learn how to use the Investintech PDFtoImage Conversion DLL. All pictures and examples from now on will be based on Microsoft Visual C++ .NET 2003.

In order to use methods contained in DLL you have to link an executable file to DLL. Executable file can be either EXE or DLL file. There are two possible ways of linking: implicit and explicit linking. In this documentation implicit linking will be shown.

Implicit Linking

Implicit linking is load-time linking. The following is needed for executables to implicitly link to Investintech Conversion DLL:

- PDFtoImageDLL.h, the header file which contains declarations of exported methods
- PDFtoImageDLL.lib, an import library used by the linker

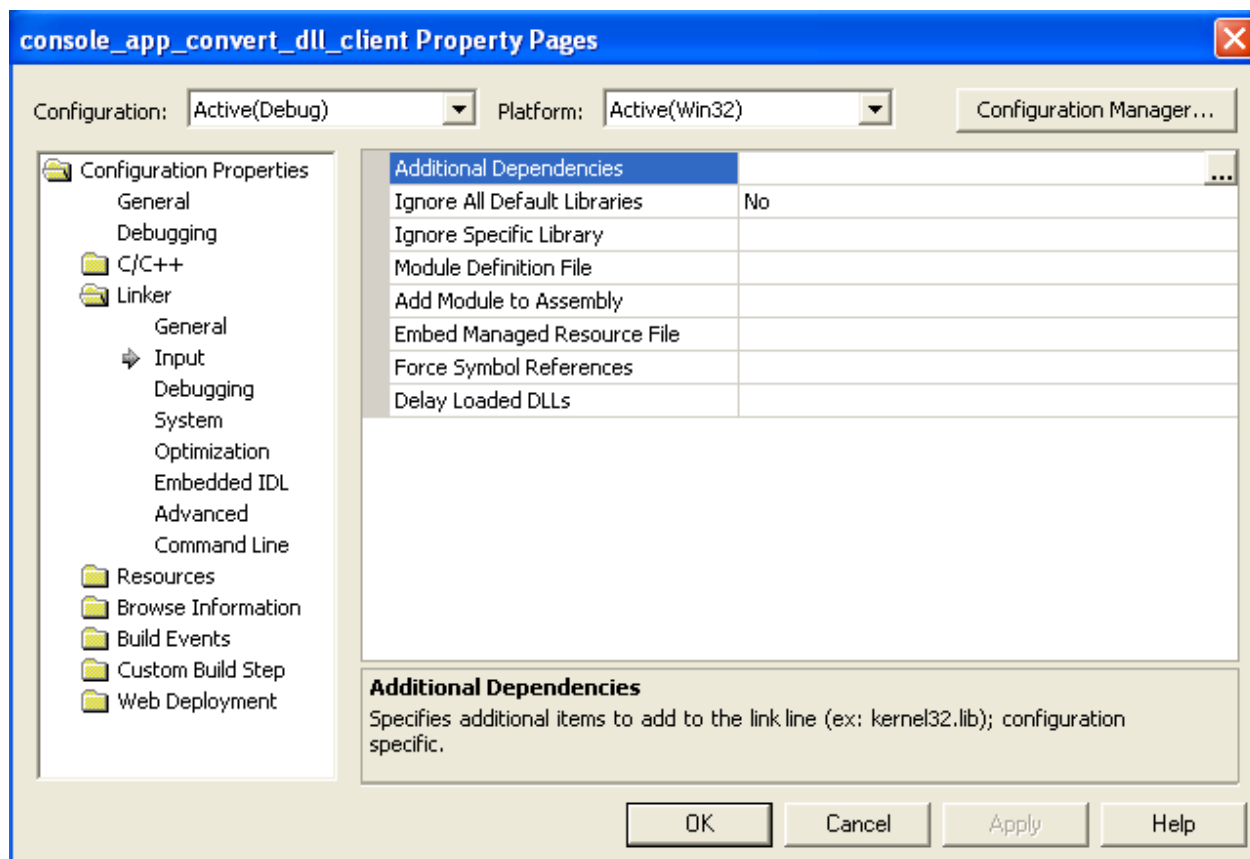
- PDFtoImageDLL.dll, the Dynamic Link Library file

All these files are provided by Investintech.com Inc.

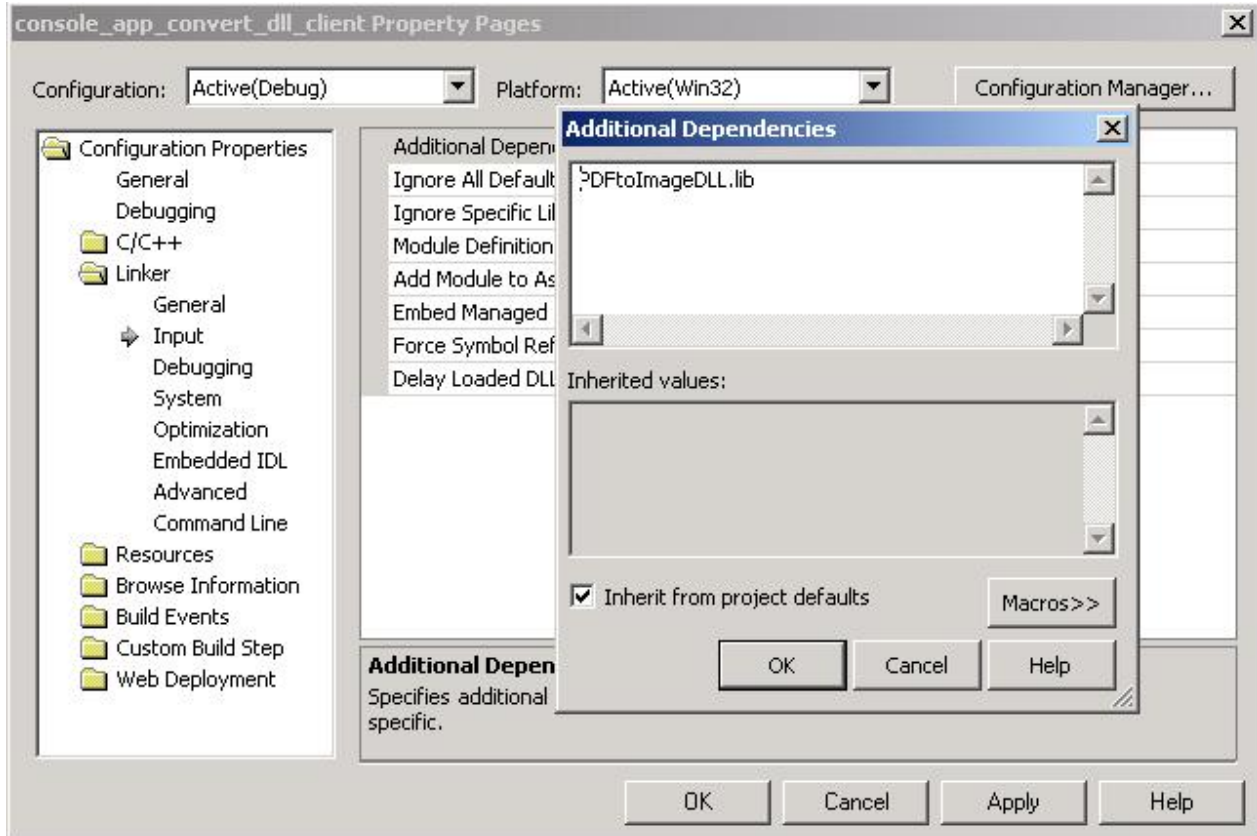
Linking .lib file with project

Below you will find step-by-step instructions on how to link PDFtoImageDLL.lib file with your project.

Start Visual Studio and select 'File > Open > Project'. Select the project in which you want to use PDFtoImageDLL.dll and click 'OK'. Select 'Project > Properties'. You should see the window like next one:



Click the 'Linker' folder and open 'Input' property page. Modify the 'Additional Dependencies' property by clicking the '...' button.



The window titled “Additional Dependencies” should open: type PDFtoImageDLL.lib and click the ‘OK’ button. Click ‘OK’ to close “Property Pages” window. You have successfully linked PDFtoImageDLL.lib with your project.

Using methods from DLL file

After linking with PDFtoImageDLL.lib file you can use methods contained in PDFtoImageDLL.dll file. Simply include the PDFtoImageDLL.h file in all source files that need to use those methods.

Using the Investintech PDFtoImage Conversion DLL in Microsoft Visual C++ .NET 2003

In this section you will learn how to use the Investintech PDFtoImage Conversion DLL. All pictures and examples from now on will be based on Microsoft Visual C++ .NET 2003.

Investintech PDFtoImage Conversion DLL Methods

In this section you will find details about each method such as: prototype; description; calling conventions; parameters; return value; and sample use of the methods.

Interface

The interface of the Investintech PDFtoImage Conversion DLL consists of its exposed methods. The interface contains an exposed method for each combination of: calling convention; parameter type; and error handling.

The calling convention is either the “cdecl” calling convention or the “stdcall” calling convention.

The “cdecl” calling convention is normally used for calling C/C++ functions. It is characterized by the following points:

- arguments are passed in order from right to left
- the calling function is responsible for popping the arguments from the stack at the conclusion of function execution
- an underscore character (_) is prefixed to function names
- no case translation is performed on the function name

The “stdcall” calling convention is typically used to call Windows operating system API functions. It is characterized by the following points:

- arguments are passed in order from right to left
- arguments are passed by value, unless a pointer or reference type is passed.
- The called function is responsible for popping its own arguments from the stack.
- an underscore (_) is prefixed to the name. The name is followed by the at sign (@) followed by the number of bytes (in decimal) in the argument list. For instance, the function declared as void foo(int bar) is decorated as follows: _foo@4
- no case-translation is performed on the function name

Parameter Type

The parameter type is either ANSI (C char *) or UNICODE (BSTR).

File Names

Files may be specified by supplying relative or absolute file names. A relative file name is relative to the application default directory containing the application executable file (e.g. “..\in_parent_dir.pdf”, “in_application_dir.pdf”, “subdir\in_sub_directory.pdf”). An absolute file name includes the directory path to the file beginning at the top of the directory tree (e.g. “c:\pdfs\absolute_file.pdf”).

It is also possible to use UNC notation (e.g. `\\myserver\public\file.pdf`) provided your user account has sufficient access permissions.

Error Handling

The error handling strategy is either to throw an exception (part of the C++ throw-catch error handling construct) or return in integer/boolean type result code (the C style error handling strategy).

The Investintech Conversion DLL identifies the characteristics of a function by an encoded suffix. The suffix is defined by the rules:

- C or S - cdecl or stdcall calling convention
- B or L - BSTR or LPST parameter type
- R or T - return error code or throw exception

Common Sample Source Code

All sample code snippets share the following initialization code snippet:

```

//.h file must be included if we want to use conversion methods
#include <afx.h>
#include <afxwin.h>           // MFC core and standard components
#include "PDFtoImageDLL.h"

CString str_inputFile = _T("c:\pdfs\in.pdf"); //input file name
CString str_outputFile = _T("out.bmp"); //output file name

//conversion from CString to BSTR
BSTR bstr_inputFile = str_inputFile.AllocSysString();

```

```
BSTR bstr_templateFile = str_templateFile.AllocString();  
BSTR bstr_outputFile = str_output.AllocSysString();
```

Conversion from PDF document to Image document

The following methods convert a PDF document file to an image format.

Conversion to Bitmap Images

PDF_to_BMP_CBR

Prototype

```
bool PDF_to_BMP_CBR(BSTR inFile, BSTR outFile)
```

Description

The PDF_to_BMP_CBR () method converts the user specified file from PDF to bitmap format. Microsoft Paint may be used to view the created image files.. The original PDF file is not modified.

Calling Convention

The function uses the “cdecl” calling convention.

Parameters

inFile Unicode string containing the absolute or relative filename of the PDF file that will be converted to bitmap format.

outFile Unicode string containing the absolute or relative filename of the directory that will contain the bitmap files created by the conversion.

Returns

true if PDF file is successfully converted to bitmap format, false otherwise.

Example

```
bool ret = PDF_to_BMP_CBR(bstr_inputFile, bstr_outputFile);
```

PDF_to_BMP_CBT

Prototype

```
bool PDF_to_BMP_CBT(BSTR inFile, BSTR outFile)
```

Description

The `PDF_to_BMP_CBT ()` method converts the user specified file from PDF to bitmap format. Microsoft Paint may be used to view the created image files. The original PDF file is not modified.

Calling Convention

The function uses the “cdecl” calling convention.

Parameters

- | | |
|----------------|--|
| inFile | Unicode string containing the absolute or relative filename of PDF file that will be converted to bitmap format. |
| outFile | Unicode string containing the absolute or relative filename of the directory that will contain the bitmap files created by the conversion. |

Returns

`true` if PDF file is successfully converted to bitmap format, otherwise throws an exception that should be handled by the calling application.

Example

```
try
{
bool ret = PDF_to_BMP_CBT(bstr_inputFile, bstr_outputFile);
}
catch(...)
{
// TODO: Handle exception
}
```

PDF_to_BMP_CLR

Prototype

bool PDF_to_BMP_CLR(LPSTR inFile, LPSTR outFile)

Description

The PDF_to_BMP_CLR () method converts the user specified file from PDF to bitmap format. Microsoft Paint may be used to view the created image files. The original PDF file is not modified.

Calling Convention

The function uses the “cdecl” calling convention.

Parameters

inFile ANSI string containing the absolute or relative filename of the PDF file that will be converted to bitmap format.

outFile ANSI string containing the absolute or relative filename of the directory that will contain the bitmap files created by the conversion.

Returns

true if PDF file is successfully converted to bitmap format, false otherwise.

Example

```
bool ret = PDF_to_BMP_CLR(str_inputFile.GetBuffer(), str_outputFile.GetBuffer());
```

PDF_to_BMP_CLT

Prototype

bool PDF_to_BMP_CLT(LPSTR inFile, LPSTR outFile)

Description

The `PDF_to_BMP_CLT()` method converts the user specified file from PDF to bitmap format. Microsoft Paint may be used to view the created image files. The original PDF file is not modified.

Calling Convention

The function uses the “cdecl” calling convention.

Parameters

inFile ANSI string containing the absolute or relative filename of the PDF file that will be converted to bitmap format.

outFile ANSI string containing the absolute or relative filename of the directory that will contain the bitmap files created by the conversion

Returns

`true` if PDF file is successfully converted to bitmap format, otherwise throws an exception that the calling application should handle.

Example

```
try
{
bool ret = PDF_to_BMP_CLT(str_inputFile.GetBuffer(), str_outputFile.GetBuffer());
}
catch(...)
{
// TODO: Handle exception
}
```

PDF_to_BMP_SBR

Prototype

```
UINT PDF_to_BMP_SBR(BSTR inFile, BSTR outFile)
```

Description

The PDF_to_BMP_SBR () method converts the user specified file from PDF to bitmap format. Microsoft Paint may be used to open the created output file. The original PDF file is not modified.

Calling Convention

The function uses the “stdcall” calling convention.

Parameters

inFile Unicode string containing the absolute or relative filename of the PDF file that will be converted to bitmap format.

outFile Unicode string containing the absolute or relative filename of the directory that will contain the bitmap files created by the conversion

Returns

Non-zero integer value if PDF file is successfully converted to bitmap format, zero integer value otherwise.

Example

```
UINT ret = PDF_to_BMP_SBR(bstr_inputFile, bstr_outputFile);
```

PDF_to_BMP_SBT

Prototype

```
UINT PDF_to_BMP_SBT(BSTR inFile, BSTR outFile)
```

Description

The PDF_to_BMP_SBT () method converts the user specified file from PDF to bitmap format. Microsoft Paint may be used to view the created image files. The original PDF file is not modified.

Calling Convention

The function uses the “stdcall” calling convention.

Parameters

- inFile** Unicode string containing the absolute or relative filename of the PDF file that will be converted to bitmap format.
- outFile** Unicode string containing the absolute or relative filename of the directory that will contain the bitmap files created by the conversion

Returns

Non-zero integer value if PDF file is successfully converted to bitmap format, otherwise throws an exception that the calling application should handle.

Example

```
try
{
    UINT ret = PDF_to_BMP_SBT(bstr_inputFile, bstr_outputFile);
}
catch(...)
{
    // TODO: Handle exception
}
```

PDF_to_BMP_SLR**Prototype**

```
UINT PDF_to_BMP_SLR(LPSTR inFile, LPSTR outFile)
```

Description

The `PDF_to_BMP_SLR()` method converts the user specified file from PDF to bitmap format. Microsoft Paint may be used to open the created output file. The original PDF file is not modified.

Calling Convention

The function uses the “stdcall” calling convention.

Parameters

inFile ANSI string containing the absolute or relative filename of the PDF file that will be converted to bitmap format.

outFile ANSI string containing the absolute or relative filename of the directory that will contain the bitmap files created by the conversion

Returns

Non-zero integer value if PDF file is successfully converted to bitmap format, zero integer value otherwise.

Example

```
UINT retCode = PDF_to_BMP_SLR(str_inputFile, str_outputFile);
```

PDF_to_BMP_SLT

Prototype

```
UINT PDF_to_BMP_SLT(LPSTR inFile, LPSTR outFile)
```

Description

The PDF_to_BMP_SLT () method converts the user specified file from PDF to bitmap format. Microsoft Paint may be used to open the created output file. The original PDF file is not modified.

Calling Convention

The function uses the “stdcall” calling convention.

Parameters

inFile ANSI string containing absolute or relative filename of the PDF file that will be converted to bitmap format.

outFile ANSI string containing the absolute or relative filename of the directory that will contain the bitmap files created by the conversion

Returns

Non-zero integer value if PDF file is successfully converted to bitmap format, otherwise throws an exception that should be handled by the calling application.

Example

```
try
{
    UINT retCode = PDF_to_BMP_SLT(str_inputFile, str_outputFile);
}
catch(...)
{
    //TODO: handle error
}
```

Conversion to GIF Images**PDF_to_GIF_CBR****Prototype**

```
bool PDF_to_GIF_CBR(BSTR inFile, BSTR outFile)
```

Description

The PDF_to_GIF_CBR () method converts the user specified file from PDF to Graphics Interchange Format. Microsoft Paint may be used to view the created image files. The original PDF file is not modified.

Calling Convention

The function uses the “cdecl” calling convention.

Parameters

inFile Unicode string containing the absolute or relative filename of the PDF file that will be converted to Graphics Interchange Format.

outFile Unicode string containing the absolute or relative filename of the directory that will contain the GIF files created by the conversion.

Returns

true if PDF file is successfully converted to Graphics Interchange Format, false otherwise.

Example

```
bool ret = PDF_to_GIF_CBR(bstr_inputFile, bstr_outputFile);
```

PDF_to_GIF_CBT

Prototype

```
bool PDF_to_GIF_CBT(BSTR inFile, BSTR outFile)
```

Description

The PDF_to_GIF_CBT () method converts the user specified file from PDF to Graphics Interchange Format. Microsoft Paint may be used to view the created image files. The original PDF file is not modified.

Calling Convention

The function uses the “cdecl” calling convention.

Parameters

inFile Unicode string containing the absolute or relative filename of PDF file that will be converted to Graphics Interchange Format.

outFile Unicode string containing the absolute or relative filename of the directory that will contain the GIF files created by the conversion.

Returns

true if PDF file is successfully converted to Graphics Interchange Format, otherwise throws an exception that should be handled by the calling application.

Example

```
try
{
bool ret = PDF_to_GIF_CBT(bstr_inputFile, bstr_outputFile);
}
catch(...)
{
// TODO: Handle exception
}
```

PDF_to_GIF_CLR**Prototype**

```
bool PDF_to_GIF_CLR(LPSTR inFile, LPSTR outFile)
```

Description

The PDF_to_GIF_CLR () method converts the user specified file from PDF to Graphics Interchange Format. Microsoft Paint may be used to view the created image files. The original PDF file is not modified.

Calling Convention

The function uses the “cdecl” calling convention.

Parameters

inFile	ANSI string containing the absolute or relative filename of the PDF file that will be converted to Graphics Interchange Format.
outFile	ANSI string containing the absolute or relative filename of the directory that will contain the GIF files created by the conversion.

Returns

true if PDF file is successfully converted to Graphics Interchange Format, false otherwise.

Example

```
bool ret = PDF_to_GIF_CLR(str_inputFile.GetBuffer(), str_outputFile.GetBuffer());
```

PDF_to_GIF_CLT

Prototype

```
bool PDF_to_GIF_CLT(LPSTR inFile, LPSTR outFile)
```

Description

The PDF_to_GIF_CLT() method converts the user specified file from PDF to Graphics Interchange Format. Microsoft Paint may be used to view the created image files. The original PDF file is not modified.

Calling Convention

The function uses the “cdecl” calling convention.

Parameters

inFile ANSI string containing the absolute or relative filename of the PDF file that will be converted to Graphics Interchange Format.

outFile ANSI string containing the absolute or relative filename of the directory that will contain the GIF files created by the conversion

Returns

true if PDF file is successfully converted to Graphics Interchange Format, otherwise throws an exception that the calling application should handle.

Example

```
try
{
bool ret = PDF_to_GIF_CLT(str_inputFile.GetBuffer(), str_outputFile.GetBuffer());
}
catch(...)
{
```

```
// TODO: Handle exception  
}
```

PDF_to_GIF_SBR

Prototype

```
UINT PDF_to_GIF_SBR(BSTR inFile, BSTR outFile)
```

Description

The PDF_to_GIF_SBR() method converts the user specified file from PDF to Graphics Interchange Format. Microsoft Paint may be used to open the created output file. The original PDF file is not modified.

Calling Convention

The function uses the “stdcall” calling convention.

Parameters

- | | |
|----------------|--|
| inFile | Unicode string containing the absolute or relative filename of the PDF file that will be converted to Graphics Interchange Format. |
| outFile | Unicode string containing the absolute or relative filename of the directory that will contain the GIF files created by the conversion |

Returns

Non-zero integer value if PDF file is successfully converted to Graphics Interchange Format, zero integer value otherwise.

Example

```
UINT ret = PDF_to_GIF_SBR(bstr_inputFile, bstr_outputFile);
```

PDF_to_GIF_SBT

Prototype

UINT PDF_to_GIF_SBT(BSTR inFile, BSTR outFile)

Description

The PDF_to_GIF_SBT () method converts the user specified file from PDF to Graphics Interchange Format . Microsoft Paint may be used to view the created image files The original PDF file is not modified.

Calling Convention

The function uses the “stdcall” calling convention.

Parameters

inFile Unicode string containing the absolute or relative filename of the PDF file that will be converted to Graphics Interchange Format.

outFile Unicode string containing the absolute or relative filename of the directory that will contain the GIF files created by the conversion

Returns

Non-zero integer value if PDF file is successfully converted to Graphics Interchange Format, otherwise throws an exception that the calling application should handle.

Example

```
try
{
  UINT ret = PDF_to_GIF_SBT(bstr_inputFile, bstr_outputFile);
}
catch(...)
{
  // TODO: Handle exception
}
```

PDF_to_GIF_SLR

Prototype

UINT PDF_to_GIF_SLR(LPSTR inFile, LPSTR outFile)

Description

The PDF_to_GIF_SLR() method converts the user specified file from PDF to Graphics Interchange Format. Microsoft Paint may be used to open the created output file. The original PDF file is not modified.

Calling Convention

The function uses the “stdcall” calling convention.

Parameters

inFile ANSI string containing the absolute or relative filename of the PDF file that will be converted to containing the absolute or relative filename of the directory that will contain the GIF files created by the conversion.

outFile ANSI string containing the absolute or relative filename of the directory that will contain the GIF files created by the conversion

Returns

Non-zero integer value if PDF file is successfully converted to Graphics Interchange Format, zero integer value otherwise.

Example

```
UINT retCode = PDF_to_GIF_SLR(str_inputFile, str_outputFile);
```

PDF_to_GIF_SLT

Prototype

UINT PDF_to_GIF_SLT(LPSTR inFile, LPSTR outFile)

Description

The PDF_to_GIF_SLT() method converts the user specified file from PDF to Graphics Interchange Format. Microsoft Paint may be used to open the created output file. The original PDF file is not modified.

Calling Convention

The function uses the “stdcall” calling convention.

Parameters

inFile ANSI string containing absolute or relative filename of the PDF file that will be converted to Graphics Interchange Format.

outFile ANSI string containing the absolute or relative filename of the directory that will contain the GIF files created by the conversion

Returns

Non-zero integer value if PDF file is successfully converted to Graphics Interchange Format, otherwise throws an exception that should be handled by the calling application.

Example

```
try
{
UINT retCode = PDF_to_GIF_SLT(str_inputFile, str_outputFile);
}
catch(...)
{
//TODO: handle error
}
```


Conversion to JPEG Images

PDF_to_JPG_CBR

Prototype

```
bool PDF_to_JPG_CBR(BSTR inFile, BSTR outFile)
```

Description

The PDF_to_JPG_CBR () method converts the user specified file from PDF to JPEG format. Microsoft Paint may be used to view the created image files. The original PDF file is not modified.

Calling Convention

The function uses the “cdecl” calling convention.

Parameters

- | | |
|----------------|--|
| inFile | Unicode string containing the absolute or relative filename of the PDF file that will be converted to JPEG format. |
| outFile | Unicode string containing the absolute or relative filename of the directory that will contain the JPEG files created by the conversion. |

Returns

true if PDF file is successfully converted to JPEG format, false otherwise.

Example

```
bool ret = PDF_to_JPG_CBR(bstr_inputFile, bstr_outputFile);
```

PDF_to_JPG_CBT

Prototype

```
bool PDF_to_JPG_CBT(BSTR inFile, BSTR outFile)
```

Description

The PDF_to_JPG_CBT () method converts the user specified file from PDF to JPEG format. Microsoft Paint may be used to view the created image files. The original PDF file is not modified.

Calling Convention

The function uses the “cdecl” calling convention.

Parameters

inFile Unicode string containing the absolute or relative filename of PDF file that will be converted to JPEG format.

outFile Unicode string containing the absolute or relative filename of the directory that will contain the JPEG files created by the conversion.

Returns

true if PDF file is successfully converted to JPEG format, otherwise throws an exception that should be handled by the calling application.

Example

```
try
{
bool ret = PDF_to_JPG_CBT(bstr_inputFile, bstr_outputFile);
}
catch(...)
{
// TODO: Handle exception
}
```

PDF_to_JPG_CLR

Prototype

```
bool PDF_to_JPG_CLR(LPSTR inFile, LPSTR outFile)
```

Description

The PDF_to_JPG_CLR () method converts the user specified file from PDF to JPEG format. Microsoft Paint may be used to view the created image files. The original PDF file is not modified.

Calling Convention

The function uses the “cdecl” calling convention.

Parameters

inFile ANSI string containing the absolute or relative filename of the PDF file that will be converted to JPEG format.

outFile ANSI string containing the absolute or relative filename of the directory that will contain the JPEG files created by the conversion.

Returns

true if PDF file is successfully converted to JPEG format, false otherwise.

Example

```
bool ret = PDF_to_JPG_CLR(str_inputFile.GetBuffer(), str_outputFile.GetBuffer());
```

PDF_to_JPG_CLT

Prototype

```
bool PDF_to_JPG_CLT(LPSTR inFile, LPSTR outFile)
```

Description

The PDF_to_JPG_CLT() method converts the user specified file from PDF to JPEG format. Microsoft Paint may be used to view the created image files. The original PDF file is not modified.

Calling Convention

The function uses the “cdecl” calling convention.

Parameters

inFile ANSI string containing the absolute or relative filename of the PDF file that will be converted to JPEG format.

outFile ANSI string containing the absolute or relative filename of the directory that will contain the JPEG files created by the conversion

Returns

true if PDF file is successfully converted to JPEG format, otherwise throws an exception that the calling application should handle.

Example

```
try
{
bool ret = PDF_to_JPG_CLT(str_inputFile.GetBuffer(), str_outputFile.GetBuffer());
}
catch(...)
{
// TODO: Handle exception
}
```

PDF_to_JPG_SBR

Prototype

UINT PDF_to_JPG_SBR(BSTR inFile, BSTR outFile)

Description

The PDF_to_JPG_SBR() method converts the user specified file from PDF to JPEG format. Microsoft Paint may be used to open the created output file. The original PDF file is not modified.

Calling Convention

The function uses the “stdcall” calling convention.

Parameters

inFile Unicode string containing the absolute or relative filename of the PDF file that will be converted to JPEG format.

outFile Unicode string containing the absolute or relative filename of the directory that will contain the JPEG files created by the conversion

Returns

Non-zero integer value if PDF file is successfully converted to JPEG format, zero integer value otherwise.

Example

```
UINT ret = PDF_to_JPG_SBR(bstr_inputFile, bstr_outputFile);
```

PDF_to_JPG_SBT**Prototype**

```
UINT PDF_to_JPG_SBT(BSTR inFile, BSTR outFile)
```

Description

The PDF_to_JPG_SBT() method converts the user specified file from PDF to JPEG format . Microsoft Paint may be used to view the created image files The original PDF file is not modified.

Calling Convention

The function uses the “stdcall” calling convention.

Parameters

- inFile** Unicode string containing the absolute or relative filename of the PDF file that will be converted to JPEG format.
- outFile** Unicode string containing the absolute or relative filename of the directory that will contain the JPEG files created by the conversion

Returns

Non-zero integer value if PDF file is successfully converted to JPEG format, otherwise throws an exception that the calling application should handle.

Example

```
try
{
UINT ret = PDF_to_JPG_SBT(bstr_inputFile, bstr_outputFile);
}
catch(...)
{
// TODO: Handle exception
}
```

PDF_to_JPG_SLR

Prototype

UINT PDF_to_JPG_SLR(LPSTR inFile, LPSTR outFile)

Description

The PDF_to_JPG_SLR () method converts the user specified file from PDF to JPEG format. Microsoft Paint may be used to open the created output file. The original PDF file is not modified.

Calling Convention

The function uses the “stdcall” calling convention.

Parameters

- inFile** ANSI string containing the absolute or relative filename of the PDF file that will be converted to containing the absolute or relative filename of the directory that will contain the JPEG files created by the conversion.
- outFile** ANSI string containing the absolute or relative filename of the directory that will contain the JPEG files created by the conversion

Returns

Non-zero integer value if PDF file is successfully converted to JPEG format, zero integer value otherwise.

Example

```
UINT retCode = PDF_to_JPG_SLR(str_inputFile, str_outputFile);
```

PDF_to_JPG_SLT**Prototype**

```
UINT PDF_to_JPG_SLT(LPSTR inFile, LPSTR outFile)
```

Description

The PDF_to_JPG_SLT () method converts the user specified file from PDF to JPEG format. Microsoft Paint may be used to open the created output file. The original PDF file is not modified.

Calling Convention

The function uses the “stdcall” calling convention.

Parameters

- inFile** ANSI string containing absolute or relative filename of the PDF file that will be converted to JPEG format.
- outFile** ANSI string containing the absolute or relative filename of the directory that will contain the JPEG files created by the conversion

Returns

Non-zero integer value if PDF file is successfully converted to JPEG format, otherwise throws an exception that should be handled by the calling application.

Example

```
try
{
UINT retCode = PDF_to_JPG_SLT(str_inputFile, str_outputFile);
}
catch(...)
{
//TODO: handle error
}
```

Conversion to PNG Images

PDF_to_PNG_CBR

Prototype

```
bool PDF_to_PNG_CBR(BSTR inFile, BSTR outFile)
```

Description

The PDF_to_PNG_CBR () method converts the user specified file from PDF to Portable Network Graphics format. Microsoft Paint may be used to view the created image files. The original PDF file is not modified.

Calling Convention

The function uses the “cdecl” calling convention.

Parameters

inFile Unicode string containing the absolute or relative filename of the PDF file that will be converted to PNG format.

outFile Unicode string containing the absolute or relative filename of the directory that will contain the PNG files created by the conversion.

Returns

true if PDF file is successfully converted to PNG format, false otherwise.

Example

```
bool ret = PDF_to_PNG_CBR(bstr_inputFile, bstr_outputFile);
```

PDF_to_PNG_CBT

Prototype

```
bool PDF_to_PNG_CBT(BSTR inFile, BSTR outFile)
```

Description

The PDF_to_PNG_CBT () method converts the user specified file from PDF to Portable Network Graphics format. Microsoft Paint may be used to view the created image files. The original PDF file is not modified.

Calling Convention

The function uses the “cdecl” calling convention.

Parameters

inFile Unicode string containing the absolute or relative filename of PDF file that will be converted to PNG format.

outFile Unicode string containing the absolute or relative filename of the directory that will contain the PNG files created by the conversion.

Returns

true if PDF file is successfully converted to PNG format, otherwise throws an exception that should be handled by the calling application.

Example

```
try
{
bool ret = PDF_to_PNG_CBT(bstr_inputFile, bstr_outputFile);
}
catch(...)
{
// TODO: Handle exception
}
```

PDF_to_PNG_CLR

Prototype

```
bool PDF_to_PNG_CLR(LPSTR inFile, LPSTR outFile)
```

Description

The PDF_to_PNG_CLR () method converts the user specified file from PDF to Portable Network Graphics format. Microsoft Paint may be used to view the created image files. The original PDF file is not modified.

Calling Convention

The function uses the “cdecl” calling convention.

Parameters

- | | |
|----------------|--|
| inFile | ANSI string containing the absolute or relative filename of the PDF file that will be converted to PNG format. |
| outFile | ANSI string containing the absolute or relative filename of the directory that will contain the PNG files created by the conversion. |

Returns

true if PDF file is successfully converted to PNG format, false otherwise.

Example

```
bool ret = PDF_to_PNG_CLR(str_inputFile.GetBuffer(), str_outputFile.GetBuffer());
```

PDF_to_PNG_CLT**Prototype**

```
bool PDF_to_PNG_CLT(LPSTR inFile, LPSTR outFile)
```

Description

The PDF_to_PNG_CLT() method converts the user specified file from PDF to Portable Network Graphics format. Microsoft Paint may be used to view the created image files. The original PDF file is not modified.

Calling Convention

The function uses the “cdecl” calling convention.

Parameters

inFile	ANSI string containing the absolute or relative filename of the PDF file that will be converted to PNG format.
outFile	ANSI string containing the absolute or relative filename of the directory that will contain the PNG files created by the conversion

Returns

true if PDF file is successfully converted to PNG format, otherwise throws an exception that the calling application should handle.

Example

```
try
{
bool ret = PDF_to_PNG_CLT(str_inputFile.GetBuffer(), str_outputFile.GetBuffer());
}
catch(...)
{
```

```
// TODO: Handle exception  
}
```

PDF_to_PNG_SBR

Prototype

UINT PDF_to_PNG_SBR(BSTR inFile, BSTR outFile)

Description

The PDF_to_PNG_SBR() method converts the user specified file from PDF to Portable Network Graphics format. Microsoft Paint may be used to open the created output file. The original PDF file is not modified.

Calling Convention

The function uses the “stdcall” calling convention.

Parameters

inFile Unicode string containing the absolute or relative filename of the PDF file that will be converted to PNG format.

outFile Unicode string containing the absolute or relative filename of the directory that will contain the PNG files created by the conversion

Returns

Non-zero integer value if PDF file is successfully converted to PNG format, zero integer value otherwise.

Example

```
UINT ret = PDF_to_PNG_SBR(bstr_inputFile, bstr_outputFile);
```

PDF_to_PNG_SBT

Prototype

UINT PDF_to_PNG_SBT(BSTR inFile, BSTR outFile)

Description

The `PDF_to_PNG_SBT()` method converts the user specified file from PDF to Portable Network Graphics format . Microsoft Paint may be used to view the created image files The original PDF file is not modified.

Calling Convention

The function uses the “stdcall” calling convention.

Parameters

inFile Unicode string containing the absolute or relative filename of the PDF file that will be converted to PNG format.

outFile Unicode string containing the absolute or relative filename of the directory that will contain the PNG files created by the conversion

Returns

Non-zero integer value if PDF file is successfully converted to PNG format, otherwise throws an exception that the calling application should handle.

Example

```
try
{
    UINT ret = PDF_to_PNG_SBT(bstr_inputFile, bstr_outputFile);
}
catch(...)
{
    // TODO: Handle exception
}
```

PDF_to_PNG_SLR**Prototype**

```
UINT PDF_to_PNG_SLR(LPSTR inFile, LPSTR outFile)
```

Description

The PDF_to_PNG_SLR() method converts the user specified file from PDF to Portable Network Graphics format. Microsoft Paint may be used to open the created output file. The original PDF file is not modified.

Calling Convention

The function uses the “stdcall” calling convention.

Parameters

inFile ANSI string containing the absolute or relative filename of the PDF file that will be converted to containing the absolute or relative filename of the directory that will contain the PNG files created by the conversion.

outFile ANSI string containing the absolute or relative filename of the directory that will contain the PNG files created by the conversion

Returns

Non-zero integer value if PDF file is successfully converted to PNG format, zero integer value otherwise.

Example

```
UINT retCode = PDF_to_PNG_SLR(str_inputFile, str_outputFile);
```

PDF_to_PNG_SLT

Prototype

```
UINT PDF_to_PNG_SLT(LPSTR inFile, LPSTR outFile)
```

Description

The PDF_to_PNG_SLT () method converts the user specified file from PDF to Portable Network Graphics format. Microsoft Paint may be used to open the created output file. The original PDF file is not modified.

Calling Convention

The function uses the “stdcall” calling convention.

Parameters

- inFile** ANSI string containing absolute or relative filename of the PDF file that will be converted to PNG format.
- outFile** ANSI string containing the absolute or relative filename of the directory that will contain the PNG files created by the conversion

Returns

Non-zero integer value if PDF file is successfully converted to PNG format, otherwise throws an exception that should be handled by the calling application.

Example

```
try
{
    UINT retCode = PDF_to_PNG_SLT(str_inputFile, str_outputFile);
}
catch(...)
{
    //TODO: handle error
}
```

Conversion to TIFF Images**PDF_to_TIF_CBR****Prototype**

```
bool PDF_to_TIF_CBR(BSTR inFile, BSTR outFile)
```

Description

The `PDF_to_TIF_CBR()` method converts the user specified file from PDF to Tagged Image File Format. Microsoft Paint may be used to view the created image files. The original PDF file is not modified.

Calling Convention

The function uses the “cdecl” calling convention.

Parameters

inFile Unicode string containing the absolute or relative filename of the PDF file that will be converted to TIFF format.

outFile Unicode string containing the absolute or relative filename of the directory that will contain the TIFF files created by the conversion.

Returns

`true` if PDF file is successfully converted to TIFF format, `false` otherwise.

Example

```
bool ret = PDF_to_TIFF_CBR(bstr_inputFile, bstr_outputFile);
```

PDF_to_TIF_CBT

Prototype

```
bool PDF_to_TIF_CBT(BSTR inFile, BSTR outFile)
```

Description

The `PDF_to_TIF_CBT()` method converts the user specified file from PDF to Tagged Image File Format. Microsoft Paint may be used to view the created image files. The original PDF file is not modified.

Calling Convention

The function uses the “cdecl” calling convention.

Parameters

inFile	Unicode string containing the absolute or relative filename of PDF file that will be converted to TIFF.
outFile	Unicode string containing the absolute or relative filename of the directory that will contain the TIFF files created by the conversion.

Returns

true if PDF file is successfully converted to TIFF format, otherwise throws an exception that should be handled by the calling application.

Example

```
try
{
bool ret = PDF_to_TIF_CBT(bstr_inputFile, bstr_outputFile);
}
catch(...)
{
// TODO: Handle exception
}
```

PDF_to_TIF_CLR**Prototype**

```
bool PDF_to_TIF_CLR(LPSTR inFile, LPSTR outFile)
```

Description

The PDF_to_TIF_CLR() method converts the user specified file from PDF to Tagged Image File Format. Microsoft Paint may be used to view the created image files. The original PDF file is not modified.

Calling Convention

The function uses the “cdecl” calling convention.

Parameters

inFile ANSI string containing the absolute or relative filename of the PDF file that will be converted to TIFF.

outFile ANSI string containing the absolute or relative filename of the directory that will contain the TIFF files created by the conversion.

Returns

true if PDF file is successfully converted to TIFF format, false otherwise.

Example

```
bool ret = PDF_to_TIF_CLR(str_inputFile.GetBuffer(), str_outputFile.GetBuffer());
```

PDF_to_TIF_CLT

Prototype

```
bool PDF_to_TIF_CLT(LPSTR inFile, LPSTR outFile)
```

Description

The PDF_to_TIF_CLT () method converts the user specified file from PDF to Tagged Image File Format. Microsoft Paint may be used to view the created image files. The original PDF file is not modified.

Calling Convention

The function uses the “cdecl” calling convention.

Parameters

inFile ANSI string containing the absolute or relative filename of the PDF file that will be converted to TIFF.

outFile ANSI string containing the absolute or relative filename of the directory that will contain the TIFF files created by the conversion

Returns

true if PDF file is successfully converted to TIFF format, otherwise throws an exception that the calling application should handle.

Example

```
try
{
bool ret = PDF_to_TIF_CLT(str_inputFile.GetBuffer(), str_outputFile.GetBuffer());
}
catch(...)
{
// TODO: Handle exception
}
```

PDF_to_TIF_SBR**Prototype**

```
UINT PDF_to_TIF_SBR(BSTR inFile, BSTR outFile)
```

Description

The PDF_to_TIF_SBR() method converts the user specified file from PDF to Tagged Image File Format. Microsoft Paint may be used to open the created output file. The original PDF file is not modified.

Calling Convention

The function uses the “stdcall” calling convention.

Parameters

inFile	Unicode string containing the absolute or relative filename of the PDF file that will be converted to PNG format.
outFile	Unicode string containing the absolute or relative filename of the directory that will contain the TIFF files created by the conversion

Returns

Non-zero integer value if PDF file is successfully converted to TIFF format, zero integer value otherwise.

Example

```
UINT ret = PDF_to_TIF_SBR(bstr_inputFile, bstr_outputFile);
```

PDF_to_TIF_SBT

Prototype

```
UINT PDF_to_TIF_SBT(BSTR inFile, BSTR outFile)
```

Description

The PDF_to_TIF_SBT () method converts the user specified file from PDF to Tagged Image File Format . Microsoft Paint may be used to view the created image files The original PDF file is not modified.

Calling Convention

The function uses the “stdcall” calling convention.

Parameters

// inFile Unicode string containing the absolute or relative filename of the PDF file that will be converted to PNG format.

outFile Unicode string containing the absolute or relative filename of the directory that will contain the TIFF files created by the conversion

Returns

Non-zero integer value if PDF file is successfully converted to TIFF format, otherwise throws an exception that the calling application should handle.

Example

```
try
```

```

{
UINT ret = PDF_to_TIF_SBT(bstr_inputFile, bstr_outputFile);
}
catch(...)
{
// TODO: Handle exception
}

```

PDF_to_TIF_SLR

Prototype

```
UINT PDF_to_TIF_SLR(LPSTR inFile, LPSTR outFile)
```

Description

The PDF_to_TIF_SLR() method converts the user specified file from PDF to Tagged Image File Format. Microsoft Paint may be used to open the created output file. The original PDF file is not modified.

Calling Convention

The function uses the “stdcall” calling convention.

Parameters

inFile ANSI string containing the absolute or relative filename of the PDF file that will be converted to containing the absolute or relative filename of the directory that will contain the TIFF files created by the conversion.

outFile ANSI string containing the absolute or relative filename of the directory that will contain the TIFF files created by the conversion

Returns

Non-zero integer value if PDF file is successfully converted to PNG format, zero integer value otherwise.

Example

```
UINT retCode = PDF_to_TIF_SLR(str_inputFile, str_outputFile);
```

PDF_to_TIF_SLT

Prototype

UINT PDF_to_TIF_SLT(LPSTR inFile, LPSTR outFile)

Description

The PDF_to_TIF_SLT () method converts the user specified file from PDF to Tagged Image File Format. Microsoft Paint may be used to open the created output file. The original PDF file is not modified.

Calling Convention

The function uses the “stdcall” calling convention.

Parameters

inFile	ANSI string containing absolute or relative filename of the PDF file that will be converted to TIFF.
outFile	ANSI string containing the absolute or relative filename of the directory that will contain the TIFF files created by the conversion

Returns

Non-zero integer value if PDF file is successfully converted to TIFF, otherwise throws an exception that should be handled by the calling application.

Example

```
try
{
    UINT retCode = PDF_to_TIF_SLT(str_inputFile, str_outputFile);
}
catch(...)
{
    //TODO: handle error
}
```

VB6 callable code

VB6 client applications must use these functions in order to avoid runtime errors related to stack and string references.

PDF_to_BMP_VB6

Prototype

```
UINT PDF_to_BMP_VB6(BSTR inFile, BSTR outFile)
```

Description

The PDF_to_BMP_VB6() method converts the user specified file from PDF to bitmap format. Microsoft Paint may be used to open the created output file. The original PDF file is not modified.

Calling Convention

The function uses the “stdcall” calling convention.

Parameters

- | | |
|-----------------------------|---|
| <code>inFile</code> | Unicode string containing the absolute or relative filename of the PDF file that will be converted to bitmap format. |
| <code>outFile</code> | Unicode string containing the absolute or relative filename of the directory that will contain the bitmap files created by the conversion |

Returns

Non-zero integer value if PDF file is successfully converted to bitmap format, zero integer value otherwise.

Example

```
UINT ret = PDF_to_BMP_VB6(bstr_inputFile, bstr_outputFile);
```

PDF_to_GIF_VB6

Prototype

UINT PDF_to_GIF_VB6(BSTR inFile, BSTR outFile)

Description

The PDF_to_GIF_VB6() method converts the user specified file from PDF to Graphics Interchange Format. Microsoft Paint may be used to open the created output file. The original PDF file is not modified.

Calling Convention

The function uses the “stdcall” calling convention.

Parameters

inFile Unicode string containing the absolute or relative filename of the PDF file that will be converted to Graphics Interchange Format.

outFile Unicode string containing the absolute or relative filename of the directory that will contain the GIF files created by the conversion

Returns

Non-zero integer value if PDF file is successfully converted to Graphics Interchange Format, zero integer value otherwise.

Example

```
UINT ret = PDF_to_GIF_VB6(bstr_inputFile, bstr_outputFile);
```

PDF_to_JPG_VB6

Prototype

UINT PDF_to_JPG_VB6(BSTR inFile, BSTR outFile)

Description

The PDF_to_JPG_VB6() method converts the user specified file from PDF to JPEG format. Microsoft Paint may be used to open the created output file. The original PDF file is not modified.

Calling Convention

The function uses the “stdcall” calling convention.

Parameters

inFile Unicode string containing the absolute or relative filename of the PDF file that will be converted to JPEG format.

outFile Unicode string containing the absolute or relative filename of the directory that will contain the JPEG files created by the conversion

Returns

Non-zero integer value if PDF file is successfully converted to JPEG format, zero integer value otherwise.

Example

```
UINT ret = PDF_to_JPG_VB6(bstr_inputFile, bstr_outputFile);
```

PDF_to_PNG_VB6**Prototype**

```
UINT PDF_to_PNG_VB6(BSTR inFile, BSTR outFile)
```

Description

The PDF_to_PNG_VB6() method converts the user specified file from PDF to Portable Network Graphics format. Microsoft Paint may be used to open the created output file. The original PDF file is not modified.

Calling Convention

The function uses the “stdcall” calling convention.

Parameters

- inFile** Unicode string containing the absolute or relative filename of the PDF file that will be converted to PNG format.
- outFile** Unicode string containing the absolute or relative filename of the directory that will contain the PNG files created by the conversion

Returns

Non-zero integer value if PDF file is successfully converted to PNG format, zero integer value otherwise.

Example

```
UINT ret = PDF_to_PNG_VB6(bstr_inputFile, bstr_outputFile);
```

PDF_to_TIF_VB6

Prototype

```
UINT PDF_to_TIF_VB6(BSTR inFile, BSTR outFile)
```

Description

The PDF_to_TIF_VB6() method converts the user specified file from PDF to Tagged Image File Format. Microsoft Paint may be used to open the created output file. The original PDF file is not modified.

Calling Convention

The function uses the “stdcall” calling convention.

Parameters

- inFile** Unicode string containing the absolute or relative filename of the PDF file that will be converted to PNG format.
- outFile** Unicode string containing the absolute or relative filename of the directory that will contain the TIFF files created by the conversion

Returns

Non-zero integer value if PDF file is successfully converted to TIFF format, zero integer value otherwise.

Example

```
UINT ret = PDF_to_TIF_VB6(bstr_inputFile, bstr_outputFile);
```

InvestintechConversionDLL_IDString

```
char* InvestintechConversionDLL_IDString()
```

Description

The `InvestintechConversionDLL_IDString()` method returns vendor's ID string.

Parameters

None.

Returns

"Investintech Conversion DLL Version 3.00 by Investintech.com Inc." string.

Investintech PDFtoImage Conversion COM Server Methods

This section provides a description of the conversion methods exposed by the PDFtoImage COM Server. The COM Server defines the CoClass CPDF2XML that implements the interface IPDF2XML. The interface consists of the following methods.

```
[id(1), helpstring("method PDFtoBMP")] HRESULT PDFtoBMP([in] BSTR sourceFile,
[in] BSTR destinationFile, [out,retval] VARIANT_BOOL* successFlag);
    [id(2), helpstring("method PDFtoGIF")] HRESULT PDFtoGIF([in] BSTR
sourceFile, [in] BSTR destinationFile, [out,retval] VARIANT_BOOL*
successFlag);
    [id(3), helpstring("method PDFtoJPG")] HRESULT PDFtoJPG([in] BSTR
sourceFile, [in] BSTR destinationFile, [out,retval] VARIANT_BOOL*
successFlag);
    [id(4), helpstring("method PDFtoPNG")] HRESULT PDFtoPNG([in] BSTR
sourceFile, [in] BSTR destinationFile, [out,retval] VARIANT_BOOL*
successFlag);
    [id(5), helpstring("method PDFtoTIF")] HRESULT PDFtoTIF([in] BSTR
sourceFile, [in] BSTR destinationFile, [out,retval] VARIANT_BOOL*
successFlag);
```

PDFtoBMP

Prototype

```
HRESULT PDFtoBMP([in] BSTR sourceFile, [in] BSTR destFile, [out,retval]
VARIANT_BOOL *successFlag)
```

Description

The PDFtoBMP() method converts the user specified file from PDF to bitmap format. Microsoft Paint may be used to view the created files. The original PDF file is not modified.

Parameters

- | | |
|--------------------|--|
| sourceFile | Unicode string containing the absolute or relative filename of the PDF file that will be converted to XML format. |
| destFile | Unicode string containing the absolute or relative filename of the directory that will contain the bitmap files created by the conversion. |
| SuccessFlag | Pointer to boolean variable that will contain the result of the operation: true is the conversion operation is successful; else false. |

Returns

HRESULT enumerated value indicating the result of the COM operation.

Example

```
HRESULT result = PDFtoBMP(BSTR_sourceFile BSTR_destinationFile, &boolFlag);
```

PDFtoGIF**Prototype**

```
HRESULT PDFtoGIF([in] BSTR sourceFile, [in] BSTR destFile, [out,retval] VARIANT_BOOL *successFlag)
```

Description

The PDFtoGIF () method converts the user specified file from PDF to Graphics Interchange Format. Microsoft Paint may be used to view the created image files. The original PDF file is not modified.

Parameters

- sourceFile** Unicode string containing the absolute or relative filename of the PDF file that will be converted to Graphics Interchange Format.
- destFile** Unicode string containing the absolute or relative filename of the directory that will contain the GIF files created by the conversion.
- SuccessFlag** Pointer to boolean variable that will contain the result of the operation: true is the conversion operation is successful; else false.

Returns

HRESULT enumerated value indicating the result of the COM operation.

Example

```
HRESULT result = PDFtoGIF(BSTR_sourceFile BSTR_destinationFile, &boolFlag);
```

PDFtoJPG

Prototype

HRESULT PDFtoJPG([in] BSTR sourceFile, [in] BSTR destFile, [out,retval] VARIANT_BOOL *successFlag)

Description

The PDFtoJPG () method converts the user specified file from PDF to JPEG format. Microsoft Paint may be used to view the created image files. The original PDF file is not modified.

Parameters

sourceFile Unicode string containing the absolute or relative filename of the PDF file that will be converted to JPEG format.

destFile Unicode string containing the absolute or relative filename of the directory that will contain the JPEG files created by the conversion.

SuccessFlag Pointer to boolean variable that will contain the result of the operation: true is the conversion operation is successful; else false.

Returns

HRESULT enumerated value indicating the result of the COM operation.

Example

```
HRESULT result = PDFtoJPG(BSTR_sourceFile BSTR_destinationFile, &boolFlag);
```

PDFtoPNG

Prototype

HRESULT PDFtoPNG([in] BSTR sourceFile, [in] BSTR destFile, [out,retval] VARIANT_BOOL *successFlag)

Description

The PDFtoPNG() method converts the user specified file from PDF to Portable Network Graphics format. Microsoft Paint may be used to view the created image files. The original PDF file is not modified.

Parameters

- sourceFile** Unicode string containing the absolute or relative filename of the PDF file that will be converted to PNG format.
- destFile** Unicode string containing the absolute or relative filename of the directory that will contain the PNG files created by the conversion.
- SuccessFlag** Pointer to boolean variable that will contain the result of the operation: true is the conversion operation is successful; else false.

Returns

HRESULT enumerated value indicating the result of the COM operation.

Example

```
HRESULT result = PDFtoPNG(BSTR_sourceFile BSTR_destinationFile, &boolFlag);
```

PDFtoTIF**Prototype**

```
HRESULT PDFtoTIF([in] BSTR sourceFile, [in] BSTR destFile, [out,retval] VARIANT_BOOL *successFlag)
```

Description

The PDFtoTIF() method converts the user specified file from PDF to Tagged Image File Format. Microsoft Paint may be used to view the created image files. The original PDF file is not modified.

Parameters

- sourceFile** Unicode string containing the absolute or relative filename of the PDF file

that will be converted to TIFF.

destFile Unicode string containing the absolute or relative filename of the directory that will contain the TIFF files created by the conversion.

SuccessFlag Pointer to boolean variable that will contain the result of the operation: true is the conversion operation is successful; else false.

Returns

HRESULT enumerated value indicating the result of the COM operation.

Example

```
HRESULT result = PDFtoTIF(BSTR_sourceFile BSTR_destinationFile, &boolFlag);
```

Index

A

Able2Extract Command Line	
Installation Instructions	4
System Requirements	4

C

Conversion DLL	3
Customer Service	2

I

Investintech.com Inc.	2
PDFtoImageSDK DLL	4
Implicit Linking	11
InvestintechSDK_IDString	57
Methods	14, 58
PDFtoImageSDK.dll	12
PDFtoImageSDK.h	11
PDFtoImageSDK.lib	11

T

Technical Support	2
-------------------	---