Investintech.com Inc.
Software Development Kit:
PDF-to-Excel Function Library
User's Guide

May 25, 2007
http://www.investintech.com

Contents

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 Excel Conversion DLL, a Dynamic-Link Library for converting PDF files

This document provides a comprehensive description of the DLL.

About This Documentation

This documentation consists of:

- Investintech PDF To Excel Conversion DLL, describing usage and methods of DLL designed to convert PDF files into various formats (Excel, CSV).

Typographical Conventions Used in This Document

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

Font	*Used for*	*Examples*
`monospaced`	Expected input from the user	`a2ecl -pdf2text abc.pdf abc.txt`
	Paths and filenames	`c:\a2ecl\a2ecl.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.

	Customer Service	*Technical Support*
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.
	410-96 Spadina Avenue
	Toronto, ON
	M5V 2J6 Canada
Fax:	+1 416 920 5848

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

Investintech PDF2Excel Conversion DLL

What is the Investintech PDF2Excel Conversion DLL?

The Investintech PDF2Excel 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 Microsoft Excel (XLS), and Comma Separated Values (CSV) formats.

Installation Instructions for PDF2ExcelDLL

Below you will find step-by-step instructions to install PDF2ExcelDLL Conversion Librarry on your system. You may want to print this instruction sheet for reference before beginning to install PDF2ExcelDLL.

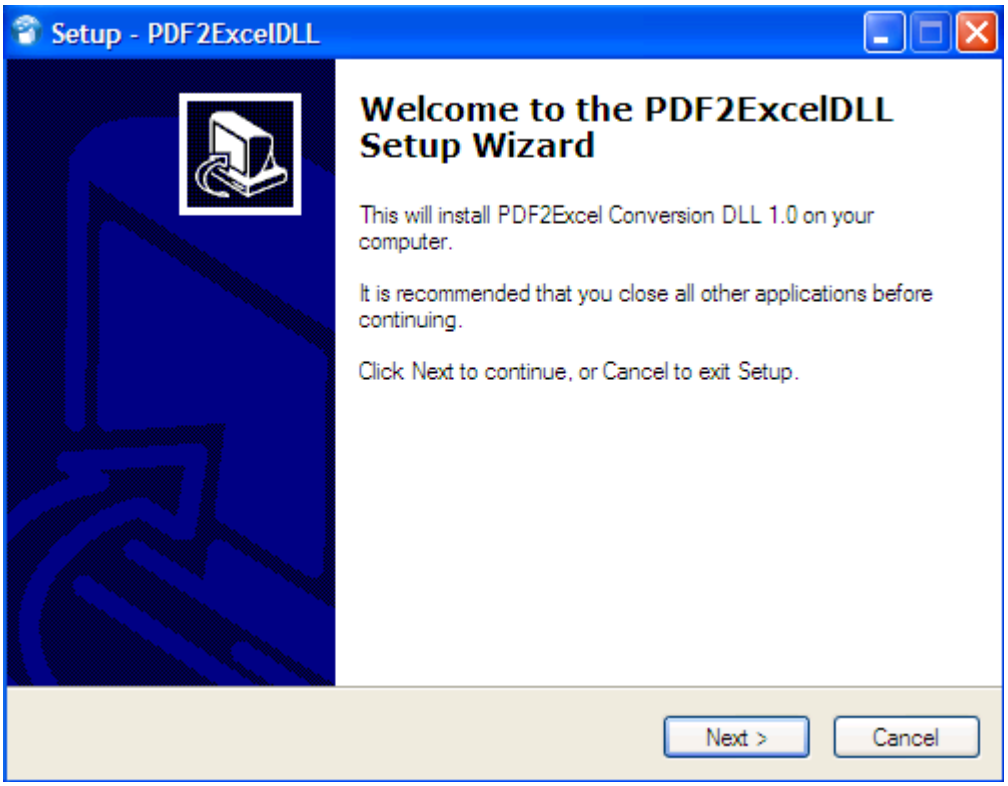
System Requirements

The minimum computer system resources required to install and use PDF2ExcelDLL are:

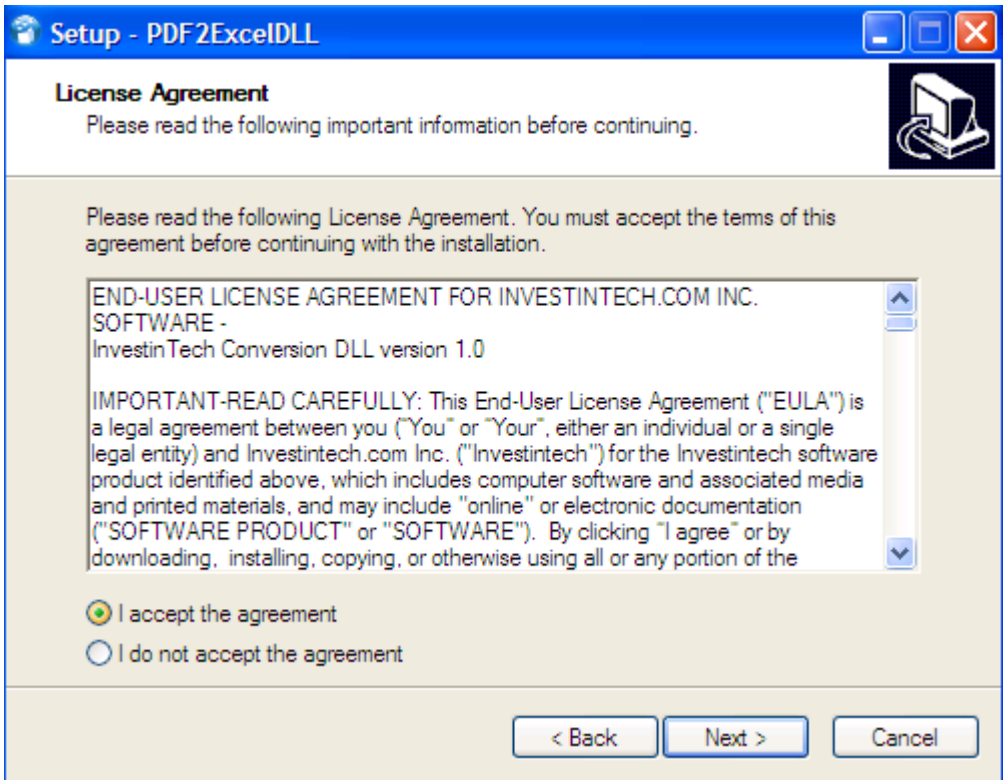
- Microsoft Windows XP
- 64 MB of RAM
- 10 MB of available hard disk space
- A software development environment, such as Microsoft Visual Studio or Borland Delphi

Starting the Installation

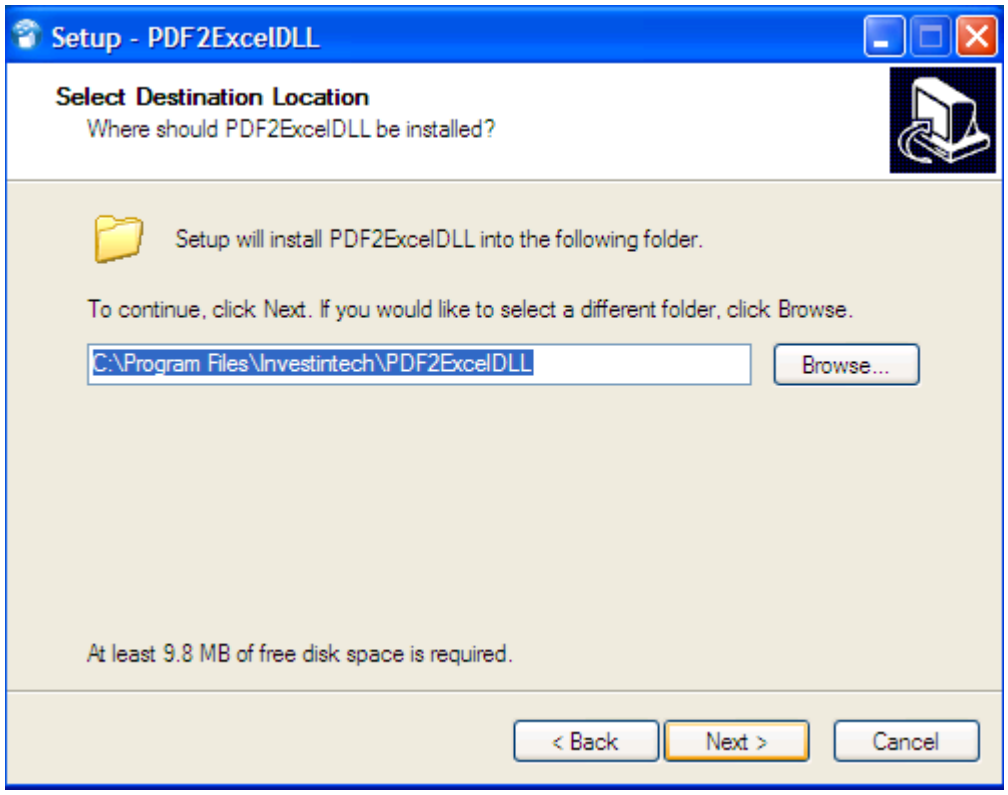
Start the installation by launching the `InstallPDF2ExcelDLL.exe` file, e.g. by double clicking it in the Windows Explorer. After that you should see a window that looks like this:



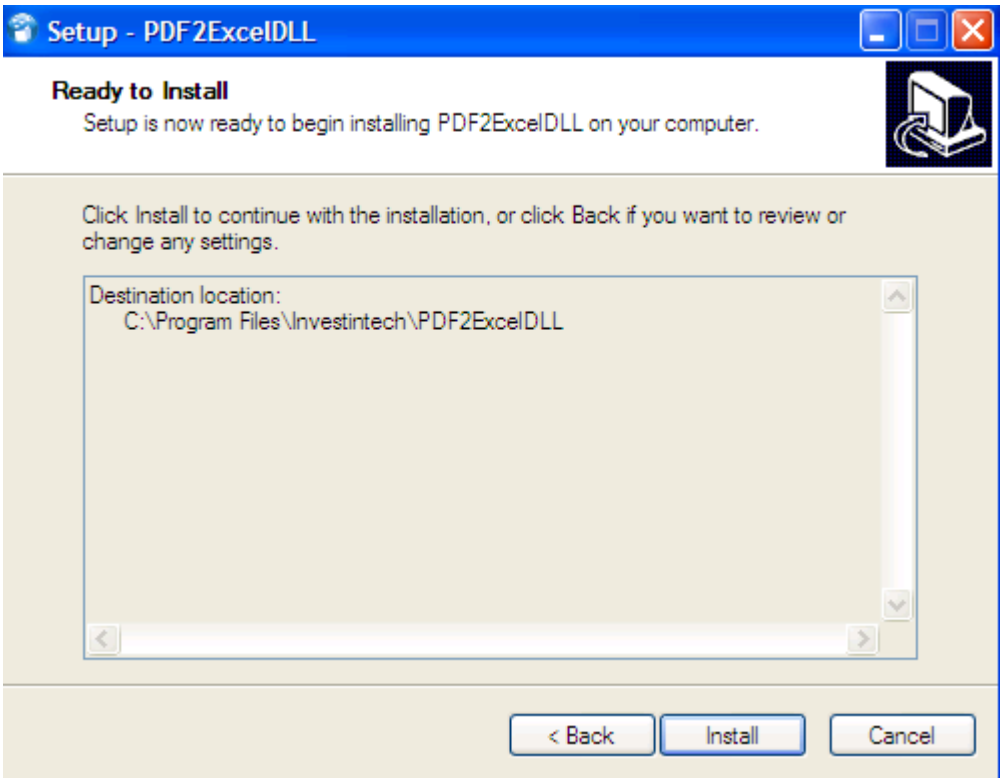
Click the '**Next**' button to continue installation. You should see the next window:



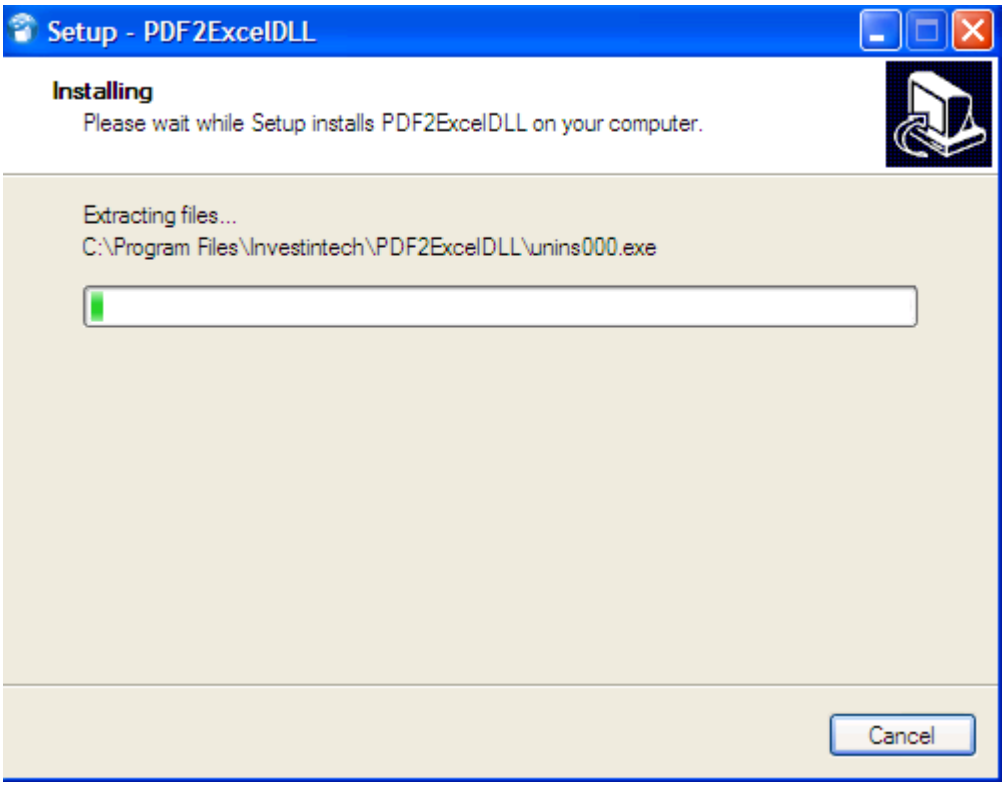
Please read the Investintech Software License Agreement carefully and thoroughly. If you wish to continue with installation, choose '**I agree with the above terms and conditions**' and click '**Next**'. Otherwise, click '**Exit**' and installation will terminate. If you clicked '**Next**' you should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



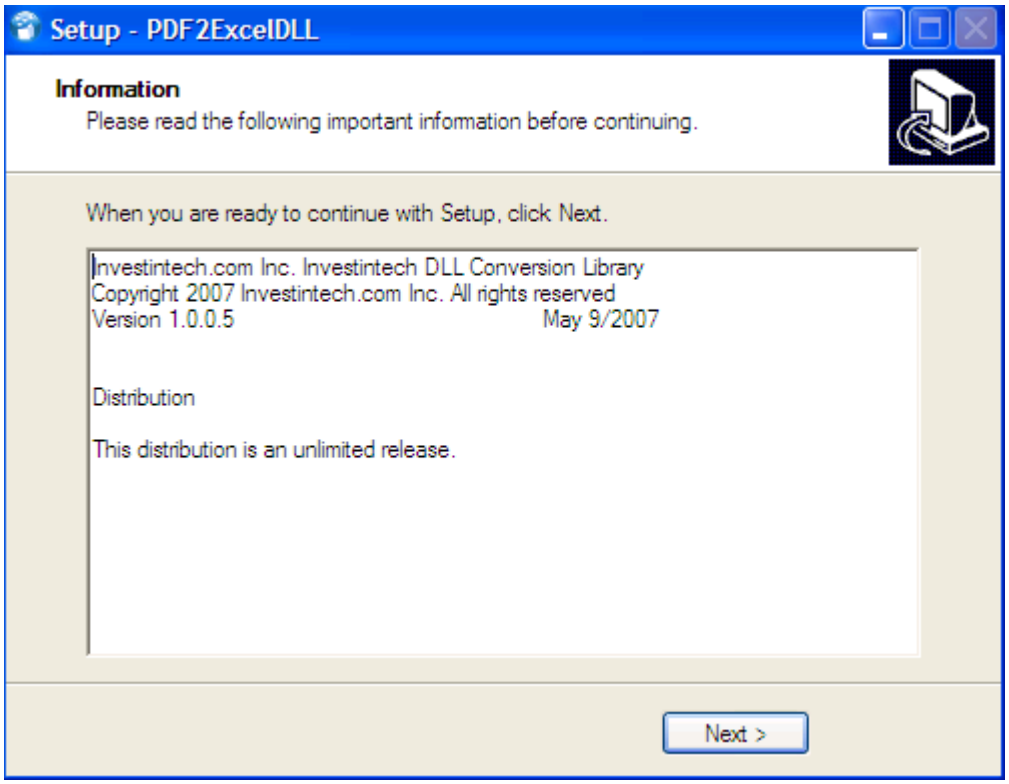
On this screen you can choose the installation directory for the PDF2ExcelDLL software. The default installation folder will be contained in the `Program Files` folder on your system disk. It is possible to change the location of installation folder either by typing its path directly or by clicking the '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.



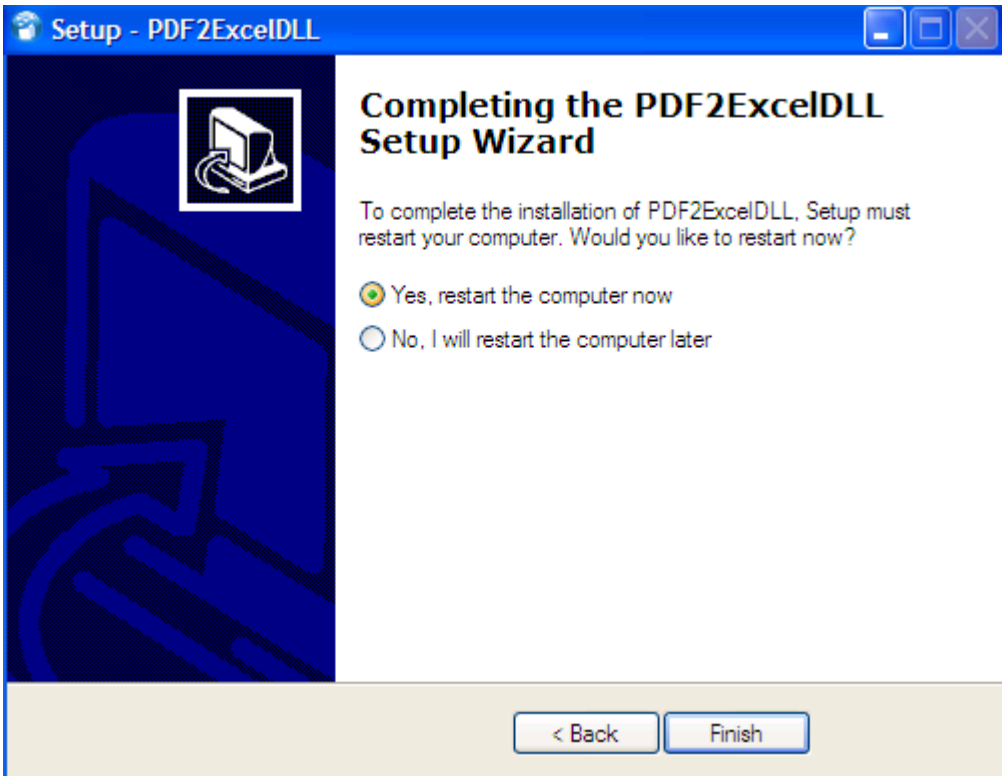
Clicking the '**Install**' button will start the installation.



The installation will copy the files that comprise PDF2ExcelDLL to the directory you specified earlier. After a very short period of time, usually about 30 seconds, you should see the next window:



This windows describes the distribution of PDF2ExcelDLL 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 PDF2Excel Conversion DLL in Microsoft Visual C++ .NET 2003

In this section you will learn how to use the Investintech PDF2Excel 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:

- PDF2Excel`DLL.h`, the header file which contains declarations of exported methods
- PDF2Excel`DLL.lib`, an import library used by the linker

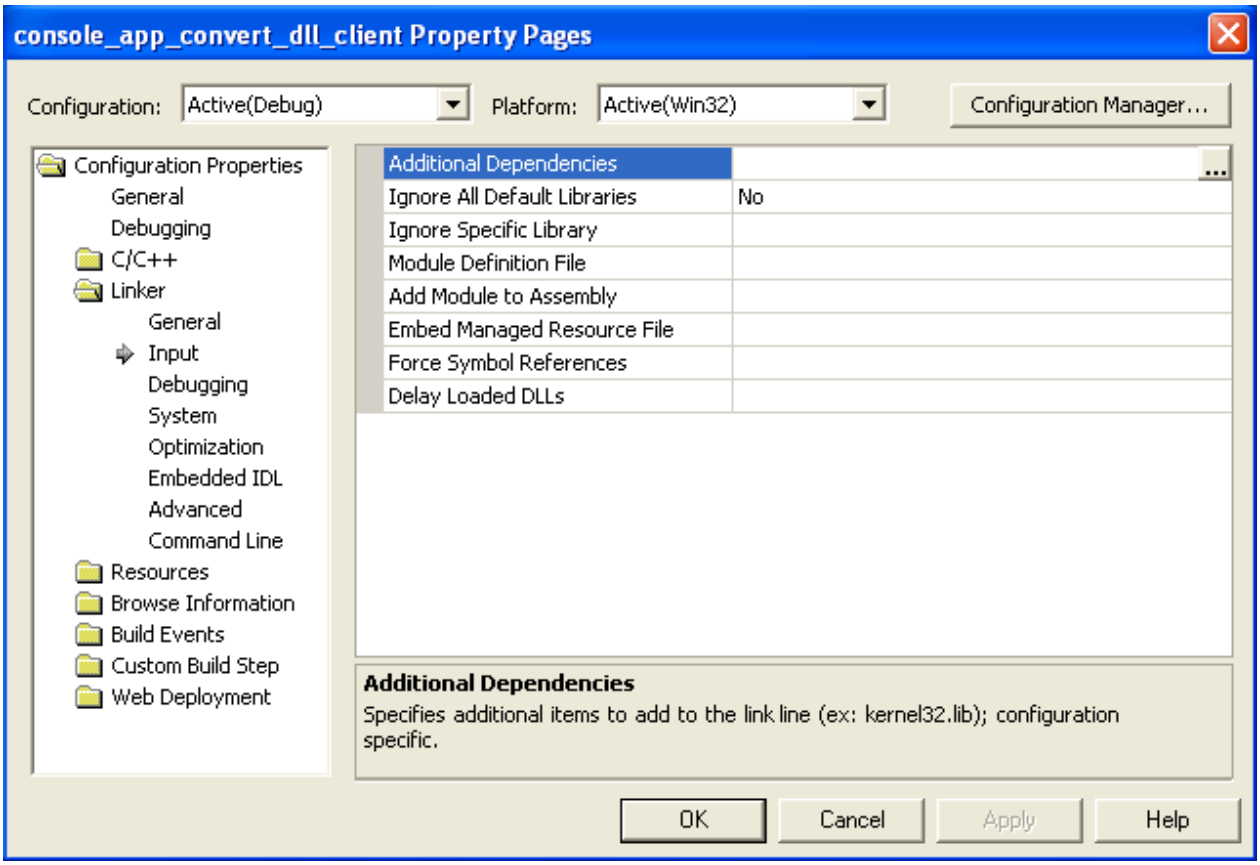
- PDF2ExcelDLL`.dll`, the Dynamic Link Library file

All these files are provided by Investintech.com Inc.

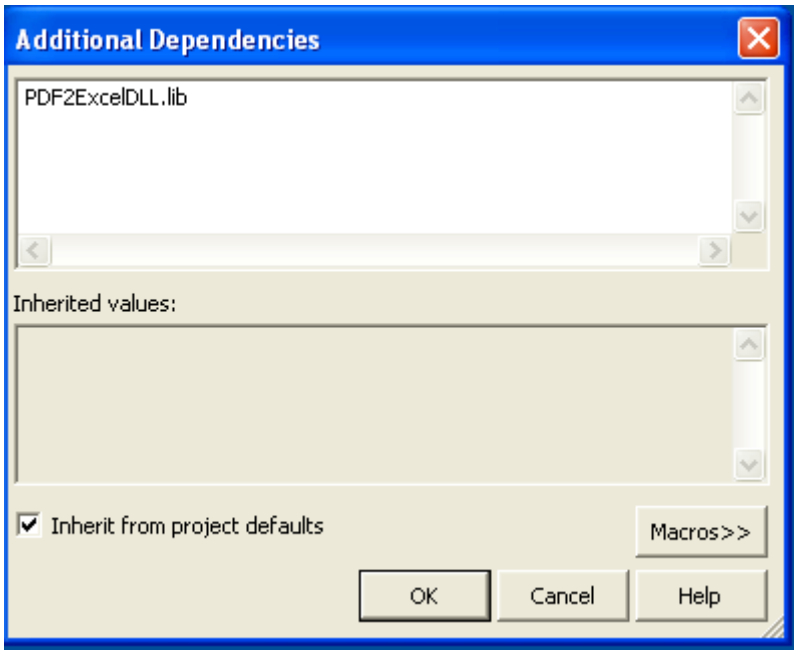
Linking .lib file with project

Below you will find step-by-step instructions on how to link `PDF2ExcelDLL.lib` file with your project.

Start Visual Studio and select '**File** > **Open** > **Project**'. Select the project in which you want to use `PDF2ExcelDLL.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 PDF2ExcelDLL.lib and click the '**OK**' button. Click '**OK**' to close "Property Pages" window. You have successfully linked `PDF2ExcelDLL.lib` with your project.

Using methods from DLL file

After linking with PDF2ExcelDLL.lib file you can use methods contained in InvestintechConversionDLL.dll file. Simply include the PDF2ExcelDLL.h file in all source files that need to use those methods.

Using the Investintech PDF2Excel Conversion DLL in Microsoft Visual C++ .NET 2003

In this section you will learn how to use the Investintech PDF2Excel Conversion DLL. All pictures and examples from now on will be based on Microsoft Visual C++ .NET 2003.

Investintech PDF2Excel 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 PDF2Excel 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 characteristrics 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 "PDF2ExcelDLL.h"

CString str_inputFile = _T("c:\pdfs\in.pdf"); //input file name
CString str_templateFile = _T(“c:\pdfs\template.ta2e”);
CString str_outputFile = _T(“out.xls”); //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 Excel Workbook

The following methods convert a PDF document file to an Excel workbook file.

PDF_to_Excel_CBR

Prototype

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

Description

The PDF_to_Excel_CBR**()** method converts the user specified file from PDF to Microsoft Excel 2.1 format. Any version of Microsoft Excel can open the created output file. 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 XLS format.
`outFile`	Unicode string containing the absolute or relative filename of the XLS file that will contain the result of the conversion.

Returns

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

Example

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

PDF_to_Excel_CBT

Prototype

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

Description

The PDF_to_Excel_CBT**()** method converts the user specified file from PDF to Microsoft Excel 2.1 format. Any version of Microsoft Excel can open the created output file. 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 XLS format.
`outFile`	Unicode string containing the absolute or relative filename of XLS file containing the result of conversion.

Returns

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

Example

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

PDF_to_Excel_CLR

Prototype

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

Description

The PDF_to_Excel_CLR**()** method converts the user specified file from PDF to Microsoft Excel 2.1 format. Any version of Microsoft Excel can open the created output file. 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 XLS format.
outFile	ANSI string containing the absolute or relative filename of the XLS file containing the result of conversion.

Returns

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

Example

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

PDF_to_Excel_CLT

Prototype

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

Description

The PDF_to_Excel_CLT**()** method converts the user specified file from PDF to Microsoft Excel 2.1 format. Any version of Microsoft Excel can open the created output file. 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 XLS format.
outFile	ANSI string containing the absolute or relative filename of the XLS file containing the result of conversion.

Returns

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

Example

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

PDF_to_Excel_SBR

Prototype

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

Description

The PDF_to_Excel_SBR**( )** method converts the user specified file from PDF to Microsoft Excel 2.1 format. Any version of Microsoft Excel can 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 XLS format.
`outFile`	Unicode string containing the absolute or relative filename of the XLS file containing the result of conversion.

Returns

`Non-zero integer value` if PDF file is successfully converted to XLS, zero integer value otherwise.

Example

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

PDF_to_Excel_SBT

Prototype

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

Description

The PDF_to_Excel_SBT**()** method converts the user specified file from PDF to Microsoft Excel 2.1 format. Any version of Microsoft Excel can 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 XLS format.
outFile	Unicode string containing the absolute or relative filename of the XLS file containing the result of conversion.

Returns

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

Example

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

PDF_to_Excel_SLR

Prototype

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

Description

The PDF_to_Excel_SLR**()** method converts the user specified file from PDF to Microsoft Excel 2.1 format. Any version of Microsoft Excel can 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 XLS format.
`outFile`	ANSI string containing the absolute or relative filename of the XLS file containing the result of conversion.

Returns

`Non-zero integer value` if PDF file is successfully converted to XLS, zero integer value otherwise.

Example

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

PDF_to_Excel_SLT

Prototype

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

Description

The PDF_to_Excel_SLT**()** method converts the user specified file from PDF to Microsoft Excel 2.1 format. Any version of Microsoft Excel can 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 XLS format.
`outFile`	ANSI string containing the absolute or relative filename of the XLS file containing the result of conversion.

Returns

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

Example

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

Conversion from PDF document to Excel Workbook using a template file

The following methods convert a PDF document file to an Excel workbook file using a template file to describe column positions in the PDF file.

PDF_to_Excel_Template_CBR

Prototype

```
bool PDF_to_Excel_Template_CBR(BSTR inFile, BSTR outFile, BSTR
templateFile)
```

Description

The PDF_to_Excel_Template_CBR**()** method converts the user specified file from PDF to Microsoft Excel 2.1 format. It uses the column positions described in the template file to convert data on the PDF file. Any version of Microsoft Excel can open the created output file. 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 XLS format.
outFile	Unicode string containing the absolute or relative filename of the XLS file containing the result of conversion.
templateFile	Unicode string containing the absolute or relative filename of the template file used to parse the PDF file.

Returns

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

Example

```
bool ret = PDF_to_Excel_Template_CBR(str_inputFile, str_outputFile, str_templateFile);
```

PDF_to_Excel_Template_CBT

Prototype

```
bool PDF_to_Excel_Template_CBT(BSTR inFile, BSTR outFile, BSTR templateFile)
```

Description

The PDF_to_Excel_Template_CBT**()** method converts the user specified file from PDF to Microsoft Excel 2.1 format. It uses the column positions described in the template file to convert data on the PDF file. Any version of Microsoft Excel can open the created output file. 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 XLS format.
outFile	Unicode string containing the absolute or relative filename of the XLS file containing the result of conversion.
templateFile	Unicode string containing the absolute or relative filename of the template file used to parse the PDF file.

Returns

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

Example

```
try
{
  bool ret = PDF_to_Excel_Template_CBT(str_inputFile, str_outputFile, str_templateFile);
}
catch(...)
{
  //TODO: handle error
}
```

PDF_to_Excel_Template_CLR

Prototype

```
bool PDF_to_Excel_Template_CLR(LPSTR inFile, LPSTR outFile, LPSTR templateFile)
```

Description

The PDF_to_Excel_Template_CLR**()** method converts the user specified file from PDF to Microsoft Excel 2.1 format. It uses the column positions described in the template file to convert data on the PDF file. Any version of Microsoft Excel can open the created output file. 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 XLS format.
`outFile`	ANSI string containing the absolute or relative filename of the XLS file containing the result of conversion.
`templateFile`	ANSI string containing the absolute or relative filename of the template file used to parse the PDF file.

Returns

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

Example

```
bool ret = PDF_to_Excel_Template_CLR(str_inputFile, str_outputFile, str_templateFile);
```

PDF_to_Excel_Template_CLT

Prototype

```
bool PDF_to_Excel_Template_CLT(LPSTR inFile, LPSTR outFile, LPSTR templateFile)
```

Description

The PDF_to_Excel_Template_CLT**()** method converts the user specified file from PDF to Microsoft Excel 2.1 format. It uses the column positions described in the template file to convert data on the PDF file. Any version of Microsoft Excel can open the created output file. 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 XLS format.
`outFile`	ANSI string containing the absolute or relative filename of the XLS file containing the result of conversion.
`templateFile`	ANSI string containing the absolute or relative filename of the template file used to parse the PDF file.

Returns

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

Example

```
try
{
 bool ret = PDF_to_Excel_Template_CLT(str_inputFile, str_outputFile, str_templateFile);
```

```
}
catch(...)
{
  //TODO: handle error
}
```

PDF_to_Excel_Template_SBR

Prototype

```
UINT PDF_to_Excel_Template_SBR(BSTR inFile, BSTR outFile, BSTR templateFile)
```

Description

The PDF_to_Excel_Template_SBR**()** method converts the user specified file from PDF to Microsoft Excel 2.1 format. It uses the column positions described in the template file to convert data on the PDF file. Any version of Microsoft Excel can 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 XLS format.
`outFile`	Unicode string containing the absolute or relative filename of the XLS file containing the result of conversion.
`templateFile`	Unicode string containing the absolute or relative filename of the template file used to parse the PDF file.

Returns

`Non-zero integer value` if PDF file is successfully converted to XLS, zero integer value otherwise.

Example

```
UINT ret = PDF_to_Excel_Template_SBR(str_inputFile, str_outputFile, str_templateFile);
```

PDF_to_Excel_Template_SBT

Prototype

```
UINT PDF_to_Excel_Template_SBT(BSTR inFile, BSTR outFile, BSTR templateFile)
```

Description

The PDF_to_Excel_Template_SBT**()** method converts the user specified file from PDF to Microsoft Excel 2.1 format. It uses the column positions described in the template file to convert data on the PDF file. Any version of Microsoft Excel can 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 XLS format.
`outFile`	Unicode string containing the absolute or relative filename of the XLS file containing the result of conversion.
`templateFile`	Unicode string containing the absolute or relative filename of the template file used to parse the PDF file.

Returns

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

Example

```
try
{
  UINT ret = PDF_to_Excel_Template_SBT(bstr_inputFile, bstr_outputFile, bstr_templateFile);
}
catch(...)
{
  //TODO: handle error
}
```

PDF_to_Excel_Template_SLR

Prototype

```
UINT PDF_to_Excel_Template_SLR(LPSTR inFile, LPSTR outFile, LPSTR templateFile)
```

Description

The PDF_to_Excel_Template_SLR**()** method converts the user specified file from PDF to Microsoft Excel 2.1 format. It uses the column positions described in the template file to convert data on the PDF file. Any version of Microsoft Excel can 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 XLS format.
`outFile`	ANSI string containing the absolute or relative filename of the XLS file containing the result of conversion.
`templateFile`	Unicode string containing the absolute or relative filename of the template file used to parse the PDF file.

Returns

`Non-zero integer value` if PDF file is successfully converted to XLS, zero integer value otherwise.

Example

```
UINT ret = PDF_to_Excel_Template_SLR(str_inputFile, str_outputFile, str_templateFile);
```

PDF_to_Excel_Template_SLT

Prototype

```
UINT PDF_to_Excel_Template_SLT(LPSTR inFile, LPSTR outFile, LPSTR templateFile)
```

Description

The PDF_to_Excel_Template_SLT**()** method converts the user specified file from PDF to Microsoft Excel 2.1 format. It uses the column positions described in the template file to convert data on the PDF file. Any version of Microsoft Excel can 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 XLS format.
outFile	ANSI string containing the absolute or relative filename of the XLS file containing the result of conversion.
templateFile	ANSI string containing the absolute or relative filename of the template file used to parse the PDF file.

Returns

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

Example

```
try
{
  UINT ret = PDF_to_Excel_Template_SLT(str_inputFile, str_outputFile, str_templateFile);
}
catch(...)
{
  //TODO: handle error
}
```

Convert from PDF document to CSV file

The following methods convert a PDF document file to a Comma Separated Value text file.

PDF_to_CSV_CBR

Prototype

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

Description

The PDF_to_CSV_CBR**()** method converts the user specified file from PDF to Comma Separated Value format. The created output file may be opened with Microsoft Excel or a text editor. 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 CSV format.
`outFile`	Unicode string containing the absolute or relative filename of the CSV file that will contain the result of the conversion.

Returns

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

Example

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

PDF_to_CSV_CBT

Prototype

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

Description

The PDF_to_CSV_CBT**()** method converts the user specified file from PDF to Comma Separated Value format. The created output file may be opened with Microsoft Excel or a text editor. 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 CSV format.
outFile	Unicode string containing the absolute or relative filename of CSV file containing the result of conversion.

Returns

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

Example

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

PDF_to_CSV_CLR

Prototype

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

Description

The PDF_to_CSV_CLR**()** method converts the user specified file from PDF to Comma Separated Value format. The created output file may be opened with Microsoft Excel or a text editor. 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 CSV format.
`outFile`	ANSI string containing the absolute or relative filename of the CSV file that will contain the result of the conversion.

Returns

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

Example

```
bool ret = PDF_to_CSV_CLR(str_inputFile, str_outputFile);
```

PDF_to_CSV_CLT

Prototype

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

Description

The PDF_to_CSV_CLT**()** method converts the user specified file from PDF to Comma Separated Value format. The created output file may be opened with Microsoft Excel or a text editor. 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 PDF file that will be converted to CSV format.
outFile	ANSI string containing the absolute or relative filename of CSV file containing the result of conversion.

Returns

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

Example

```
try
{
  bool ret  = PDF_to_CSV_CLT(str_inputFile, str_outputFile);
}
catch(…)
{
        // TODO: Handle exception
}
```

PDF_to_CSV_SBR

Prototype

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

Description

The PDF_to_CSV_SBR**()** method converts the user specified file from PDF to Comma Separated Value format. The created output file may be opened with Microsoft Excel or a text editor. 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 CSV format.

outFile	Unicode string containing the absolute or relative filename of the CSV file that will contain the result of the conversion.

Returns

`Non-zero integer value` if PDF file is successfully converted to CSV, zero integer value otherwise.

Example

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

PDF_to_CSV_SBT

Prototype

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

Description

The PDF_to_CSV_SBT**()** method converts the user specified file from PDF to Comma Separated Value format. The created output file may be opened with Microsoft Excel or a text editor. 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 PDF file that will be converted to CSV format.
outFile	Unicode string containing the absolute or relative filename of CSV files containing the result of conversion.

Returns

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

Example

```
try
{
```

```
  bool ret  = PDF_to_CSV_SBT(bstr_inputFile, bstr_outputFile);
}
catch(…)
{
        // TODO: Handle exception
}
```

PDF_to_CSV_SLR

Prototype

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

Description

The PDF_to_CSV_SLR**()** method converts the user specified file from PDF to Comma Separated Value format. The created output file may be opened with Microsoft Excel or a text editor. 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 CSV format.
`outFile`	ANSI string containing the absolute or relative filename of the CSV file that will contain the result of the conversion.

Returns

`Non-zero integer value` if PDF file is successfully converted to CSV, zero integer value otherwise.

Example

```
  UINT ret = PDF_to_CSV_SLR(str_inputFile, str_outputFile);
```

PDF_to_CSV_SLT

Prototype

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

Description

The PDF_to_CSV_SLT**()** method converts the user specified file from PDF to Comma Separated Value format. The created output file may be opened with Microsoft Excel or a text editor. 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 PDF file that will be converted to CSV format.
outFile	ANSI string containing the absolute or relative filename of CSV files containing the result of conversion.

Returns

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

Example

```
try
{
  bool ret  = PDF_to_CSV_SLT(str_inputFile, str_outputFile);
}
catch(…)
{
        // TODO: Handle exception
}
```

Convert from PDF document to CSV file using template file

The following methods convert a PDF document file to an Excel workbook file using a template file to describe column positions in the PDF file.

PDF_to_CSV_Template_CBR

Prototype

```
bool PDF_to_CSV_Template_CBR(BSTR inFile, BSTR outFile, BSTR
templateFile)
```

Description

The PDF_to_CSV_Template_CBR**()** method converts the user specified file from PDF to Comma Separated Value format. It uses the column positions described in the template file to convert data on the PDF file. The output file may be opened with a text editor or any version of Microsoft Excel. 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 CSV format.
`outFile`	Unicode string containing the absolute or relative filename of the CSV file containing the result of conversion.
`templateFile`	Unicode string containing the absolute or relative filename of the template file used to parse the PDF file.

Returns

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

Example

```
bool ret = PDF_to_CSV_Template_CBR(bstr_inputFile, bstr_outputFile, bstr_templateFile);
```

PDF_to_CSV_Template_CBT

Prototype

```
bool PDF_to_CSV_Template_CBT(BSTR inFile, BSTR outFile, BSTR
templateFile)
```

Description

The PDF_to_CSV_Template_CBT**()** method converts the user specified file from PDF to Comma Separated Value format. It uses the column positions described in the template file to convert data on the PDF file. The output file may be opened with a text editor or any version of Microsoft Excel. 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 XLS format.
`outFile`	Unicode string containing the absolute or relative filename of the XLS file containing the result of conversion.
`templateFile`	Unicode string containing the absolute or relative filename of the template file used to parse the PDF file.

Returns

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

Example

```
try
{
  bool ret = PDF_to_CSV_Template_CBT(bstr_inputFile, bstr_outputFile, bstr_templateFile);
}
catch(...)
{
  //TODO: handle error
}
```

PDF_to_CSV_Template_CLR

Prototype

```
bool PDF_to_CSV_Template_CLR(LPSTR inFile, LPSTR outFile, LPSTR
templateFile)
```

Description

The PDF_to_CSV_Template_CLR**()** method converts the user specified file from PDF to Comma Separated Value format. It uses the column positions described in the template file to convert data on the PDF file. The output file may be opened with a text editor or any version of Microsoft Excel. 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 CSV format.
`outFile`	ANSI string containing the absolute or relative filename of the CSV file containing the result of conversion.
`templateFile`	ANSI string containing the absolute or relative filename of the template file used to parse the PDF file.

Returns

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

Example

```
bool ret = PDF_to_CSV_Template_CLR(str_inputFile, str_outputFile, str_templateFile);
```

PDF_to_CSV_Template_CLT

Prototype

```
bool PDF_to_CSV_Template_CLT(LPSTR inFile, LPSTR outFile, LPSTR
templateFile)
```

Description

The PDF_to_CSV_Template_CLT**()** method converts the user specified file from PDF to Comma Separated Value format. It uses the column positions described in the template file to convert data on the PDF file. The output file may be opened with a text editor or any version of Microsoft Excel. 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 XLS format.
`outFile`	ANSI string containing the absolute or relative filename of the XLS file containing the result of conversion.
`templateFile`	ANSI string containing the absolute or relative filename of the template file used to parse the PDF file.

Returns

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

Example

```
try
{
  bool ret = PDF_to_CSV_Template_CLT(str_inputFile, str_outputFile, str_templateFile);
}
catch(...)
{
  //TODO: handle error
}
```

PDF_to_CSV_Template_SBR

Prototype

```
UINT PDF_to_CSV_Template_SBR(BSTR inFile, BSTR outFile, BSTR
templateFile)
```

Description

The PDF_to_CSV_Template_SBR**()** method converts the user specified file from PDF to Comma Separated Value format. It uses the column positions described in the template file to convert data on the PDF file. The output file may be opened with a text editor or any version of Microsoft Excel. 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 CSV format.
outFile	Unicode string containing the absolute or relative filename of the CSV file containing the result of conversion.
templateFile	Unicode string containing the absolute or relative filename of the template file used to parse the PDF file.

Returns

`Non-zero integer value` if PDF file is successfully converted to XLS, zero integer value otherwise.

Example

```
UINT ret = PDF_to_CSV_Template_SBR(bstr_inputFile, bstr_outputFile, bstr_templateFile);
```

PDF_to_CSV_Template_SBT

Prototype

```
UINT PDF_to_CSV_Template_SBT(BSTR inFile, BSTR outFile, BSTR
templateFile)
```

Description

The PDF_to_CSV_Template_SBT**()** method converts the user specified file from PDF to Comma Separated Value format. It uses the column positions described in the template file to convert data on the PDF file. The output file may be opened with a text editor or any version of Microsoft Excel. 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 XLS format.
outFile	Unicode string containing the absolute or relative filename of the XLS file containing the result of conversion.
templateFile	Unicode string containing the absolute or relative filename of the template file used to parse the PDF file.

Returns

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

Example

```
try
{
  UINT ret = PDF_to_CSV_Template_CBT(str_inputFile, str_outputFile, str_templateFile);
}
catch(...)
{
  //TODO: handle error
}
```

PDF_to_CSV_Template_SLR

Prototype

```
UINT PDF_to_CSV_Template_SLR(LPSTR inFile, BSTR outFile, BSTR
templateFile)
```

Description

The PDF_to_CSV_Template_SLR**()** method converts the user specified file from PDF to Comma Separated Value format. It uses the column positions described in the template file to convert data on the PDF file. The output file may be opened with a text editor or any version of Microsoft Excel. 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 CSV format.
`outFile`	ANSI string containing the absolute or relative filename of the CSV file containing the result of conversion.
`templateFile`	ANSI string containing the absolute or relative filename of the template file used to parse the PDF file.

Returns

`Non-zero integer value` if PDF file is successfully converted to CSV, zero integer value otherwise.

Example

```
UINT ret = PDF_to_CSV_Template_SLR(str_inputFile, str_outputFile, bstr_templateFile);
```

PDF_to_CSV_Template_SLT

Prototype

```
UINT PDF_to_CSV_Template_SLT(LPSTR inFile, LPSTR outFile, LPSTR templateFile)
```

Description

The PDF_to_CSV_Template_SLT**()** method converts the user specified file from PDF to Comma Separated Value format. It uses the column positions described in the template file to convert data on the PDF file. The output file may be opened with a text editor or any version of Microsoft Excel. 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 XLS format.
`outFile`	ANSI string containing the absolute or relative filename of the XLS file containing the result of conversion.
`templateFile`	ANSI string containing the absolute or relative filename of the template file used to parse the PDF file.

Returns

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

Example

```
try
{
  bool ret = PDF_to_CSV_Template_SLT(str_inputFile, str_outputFile, str_templateFile);
}
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_Excel_VB6

Prototype

```
UINT PDF_to_Excel_VB6(BSTR inFile, BSTR outFile)
```

Description

The PDF_to_Excel_VB6**()** method converts the user specified file from PDF to Microsoft Excel 2.1 format. Any version of Microsoft Excel can 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 XLS format.
outFile	Unicode string containing the absolute or relative filename of the XLS file containing the result of conversion.

Returns

`Non-zero integer value` if PDF file is successfully converted to XLS, zero integer value otherwise.

Example

```
UINT ret = PDF_to_Excel_VB6(bstr_inputFile, bstr_outputFile);
```

PDF_to_Excel_Template_VB6

Prototype

```
UINT PDF_to_Excel_Template_VB6(BSTR inFile, BSTR outFile, BSTR templateFile)
```

Description

The PDF_to_Excel_Template_VB6**()** method converts the user specified file from PDF to Microsoft Excel 2.1 format. It uses the column positions described in the template file to convert data on the PDF file. Any version of Microsoft Excel can 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 XLS format.
`outFile`	Unicode string containing the absolute or relative filename of the XLS file containing the result of conversion.
`templateFile`	Unicode string containing the absolute or relative filename of the template file used to parse the PDF file.

Returns

`Non-zero integer value` if PDF file is successfully converted to XLS, zero integer value otherwise.

Example

```
try
{
```

```
  UINT ret = PDF_to_Excel_Template_VB6(str_inputFile, str_outputFile, str_templateFile);
}
catch(...)
{
  //TODO: handle error
}
```

`PDF_to_CSV_VB6`

Prototype

```
UINT PDF_to_CSV_VB6(BSTR inFile, BSTR outFile)
```

Description

The PDF_to_CSV_VB6**()** method converts the user specified file from PDF to Comma Separated Value format. The created output file may be opened with Microsoft Excel or a text editor. 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 CSV format.
`outFile`	Unicode string containing the absolute or relative filename of the CSV file that will contain the result of the conversion.

Returns

`Non-zero integer value` if PDF file is successfully converted to CSV, zero integer value otherwise.

Example

```
  UINT ret = PDF_to_CSV_VB6(bstr_inputFile, bstr_outputFile);
```

PDF_to_CSV_Template_VB6

Prototype

```
UINT PDF_to_CSV_Template_VB6(BSTR inFile, BSTR outFile, BSTR templateFile)
```

Description

The PDF_to_CSV_Template_VB6**()** method converts the user specified file from PDF to Comma Separated Value format. It uses the column positions described in the template file to convert data on the PDF file. The output file may be opened with a text editor or any version of Microsoft Excel. 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 CSV format.
outFile	Unicode string containing the absolute or relative filename of the CSV file containing the result of conversion.
templateFile	Unicode string containing the absolute or relative filename of the template file used to parse the PDF file.

Returns

`Non-zero integer value` if PDF file is successfully converted to XLS, zero integer value otherwise.

Example

```
UINT ret = PDF_to_CSV_Template_VB6(bstr_inputFile, bstr_outputFile, bstr_templateFile);
```

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 PDF-To-Excel Conversion COM Server Methods

This section provides a description of the conversion methods exposed by the PDF-To-Excel COM Server. The COM Server defines the CoClass CPDF2Excel that implements the interface IPDF2Excel. The interface consists of the following methods.

PDF2Excel

Prototype

```
HRESULT  PDF2Excel([in] BSTR sourceFile, [in] BSTR destFile,
[out,retval] VARIANT_BOOL *successFlag)
```

Description

The PDF2Excel**()** method converts the user specified file from PDF to Excel format. The output file may be opened with any version of Microsoft Excel. 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 Excel format.
`destFile`	Unicode string containing the absolute or relative filename of the Excel file that contains the result of 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 = PDF2Excel(BSTR_sourceFile BSTR_destinationFile, &boolFlag);
```

PDF2CSV

Prototype

```
HRESULT  PDF2CSV([in] BSTR sourceFile, [in] BSTR destFile,
[out,retval] VARIANT_BOOL *successFlag)
```

Description

The PDF2CSV**()** method converts the user specified file from PDF to Comma Separated Value format. The output file may be opened with a text editor or any version of Microsoft Excel. 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 CSV format.
destFile	Unicode string containing the absolute or relative filename of the CSV file that contains the result of conversion.
success	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 = PDF2CSV(BSTR_sourceFile BSTR_destinationFile, &boolFlag);
```

TemplatePDF2Excel

Prototype

```
HRESULT  TemplatePDF2Excel([in] BSTR sourceFile, [in] BSTR
destFile, [in] BSTR templateFile, [out,retval] VARIANT_BOOL
*successFlag)
```

Description

The TemplatePDF2Excel**()** method converts the user specified file from PDF to Excel format. It uses the column positions described in the template file to convert data on the PDF file. The

output file may be opened with any version of Microsoft Excel. 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 Excel format.
destFile	Unicode string containing the absolute or relative filename of the Excel file that contains the result of conversion.
templateFile	Unicode string containing the absolute or relative filename of the template file that describes the position of columns in a table of data to be converted.
success	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 = TemplatePDF2Excel(BSTR_sourceFile BSTR_destinationFile,
BSTR_templateFile, &boolFlag);
```

TemplatePDF2CSV

Prototype

```
HRESULT  TemplatePDF2CSV([in] BSTR sourceFile, [in] BSTR
destFile, [in] BSTR templateFile, [out,retval] VARIANT_BOOL
*sucessFlag)
```

Description

The TemplatePDF2CSV**()** method converts the user specified file from PDF to Comma Separated Value format. It uses the column positions described in the template file to convert data on the PDF file. The output file may be opened with a text editor or any version of Microsoft Excel. 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 CSV format.
`destFile`	Unicode string containing the absolute or relative filename of the CSV file that contains the result of conversion.
`templateFile`	Unicode string containing the absolute or relative filename of the template file that describes the position of columns in a table of data to be converted.
`Success`	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 = TemplatePDF2CSV(BSTR_sourceFile, BSTR_destinationFile,
BSTR_templateFile, &boolFlag);
```

Index

ERROR: undefined
OFFENDING COMMAND: ceo

STACK:

/Ref_part3
/Dest
-mark-