Software License Agreement

WinPDF Converter

For Win32/Win64

Version 14

2004-2019

ALL RIGHTS RESERVED BY

SUB SYSTEMS, INC.

1221 New Meister Lane, #2712

Pflugerville, TX 78660

512-733-2525

Software License Agreement

The Software is protected by copyright laws and international copyright treaties, as well as other intellectual property laws and treaties. The Software is licensed, not sold. This LICENSE AGREEMENT grants you the following rights:

A. This product is licensed per developer basis only. Each developer working with this package needs to purchase a separate license.

B. The purchaser has the right to modify and link the DLL functions into their application. Such an application is free of distribution royalties with these conditions: the target application is not a stand-alone PDF Converter; the target application uses this product for one operating system platform only; and the source code (or part) of the editor is not distributed in any form.

C. The DESKTOP LICENSE allows for the desktop application development. Your desktop application using this product can be distributed royalty-free. Each desktop license allows one developer to use this product on up to two development computers. A developer must purchase additional licenses to use the product on more than two development computers.

D. The SERVER LICENSE allows for the server application development. The server licenses must be purchased separately when using this product in a server application. Additionally, the product is licensed per developer basis. Only an UNLIMITED SERVER LICENSE allows for royalty-free distribution of your server applications using this product.

E. ENTERPRISE LICENSE: The large corporations with revenue more than $50 million and large government entities must purchase an Enterprise License. An Enterprise license is also applicable if any target customer of your product using the Software have revenue more than $500 million. Please contact us at info@subsystems.com for a quote for an Enterprise License.

F. Your license rights under this LICENSE AGREEMENT are non-exclusive. All rights not expressly granted herein are reserved by Licensor.

G. You may not sell, transfer or convey the software license to any third party without Licensor's prior express written consent.

This software is designed keeping the safety and the reliability concerns as the main

considerations. Every effort has been made to make the product reliable and error free. However, Sub Systems, Inc. makes no warranties against any damage, direct or indirect, resulting from the use of the software or the manual and can not be held responsible for the same. The product is provided 'as is' without warranty of any kind, either expressed or implied, including but not limited to the implied warranties of suitability for a particular purpose. The buyer assumes the entire risk of any damage caused by this software. In no event shall Sub Systems, Inc. be liable for damage of any kind, loss of data, loss of profits, interruption of business or other financial losses arising directly or indirectly from the use of this product. Any liability of Sub Systems will be exclusively limited to refund of purchase price.

Sub Systems, Inc. offers a 30 day money back guarantee with the product. Must call for an RMA number before returning the product.

Desclaimer

This software is designed keeping the safety and the reliability concerns as the main considerations. Every effort has been made to make the product reliable and error free. However, Sub Systems, Inc. makes no warranties against any damage, direct or indirect, resulting from the use of the software or the manual and can not be held responsible for the same.

Windows 95/98/NT/2000/XP, Visual C++, MFC, and Visual Basic are the trademarks of Microsoft Corp. (for ease of reading Windows refer to MS Windows)

Delphi is the trademark of Borland International.

Getting Started

This chapter describes the contents of the software diskettes and provides a step by step process of incorporating WinPDF Converter into your application.

In This Chapter

Files
Incorporating the DLL into Your Application
Sample Conversion Code

Files

The package contains the DLL and header files. The package also includes a set of files to construct a demo program. The demo program shows by example the process of linking the DLL to your program.

DLL Demo Files:

The following demo files are included in the c_demo.zip file.

DEMO.C	Source code for the demo program
DEMO.H	Include file for the demo program
DEMO.RC	Resource source file for the demo program
DEMO.DEF	Definition file for linking the demo program
DEMO.EXE	Executable demo program
DEMO_DLG.H	Dialog Identifiers for the demo program
DEMO_DLG.DLG	Dialog templates for the demo program
DEMO_DLG.RES	Compiled dialogs for the demo program
PDC.H	The *include* file to include into a C/C++ application module that calls the PDC routine. It contains the constant definitions and the prototypes for the API functions.
PDC32.DLL	The DLL file
PDC32.LIB	Import library for the PDC32 DLL

Incorporating the DLL into Your Application

This product can be used in two ways. You can use this product with TE Developer's Kit to convert the current document into the PDF format. You can also use this product directly within your application to generate PDF output.

When using WinPDF Converter with TE Developer's Kit

Copy the pdc32.dll file to the folder which contains the ter32.dll file. Then please follow the help file for TE Developer's Kit, topic: Application Interface Functions -> PDF Output.

TE provides an automatic interface with WinPDf Converter. You would not need to refer to this help file any further.

When using WinPDf Converter directly within your application

A C/C++ application should include the PDC.h file into the application module that needs to call the PDC32 dll. It also should include the PDC32.LIB as the linker library. Please also make sure that the pdc32.dll file is copied to a directory available at the run-time.

The PDF file generation process is simply as following:

PdcStartDoc(...)

for each page:

HDC hDC=PdcStartPage(...)

// make necessary Windows SDK functions call to draw the page using text, images, and objects functions.

PdcEndPage(..)

PdcEndDoc(..)

Please refer to the next topic for a sample code.

.

Sample Conversion Code

1. **First you would create a new document session.** Here is an example:

```
DocId=PdcStartDoc(LicKey,OutFile,"Mike","Sub Systems, Inc",
                "PDF Generation Test",
                "PDF Generation Test","test,subsys",
                "12/31/2014","12/31/2014",0);
```

The first argument is used to pass your product license key. Your license key is e-mailed to you after your order is processed. Please set the LicKey parameter to "" when using WinPDF in the evaluation mode.

This function returns the Document Id. You would use the Document Id to call other conversion functions.

Once the document id is created, you would use the PdcStartPage and PdcEndPage functions to create each page. The GUI statements to define the content of the page are included between these two function.

2. **Start a new page:**

```
HDC hDC;
PageWidth=8.5*1440;    // assume a letter size paper
                      //paper size specified in twips units
PageHeight=11*1440;
hDC=PdcStartPage(DocId,PageWidth,PageHeight)
```

3. **Use the Windows GUI functions to create the contents of the page**

```
SelectObject(hDC,GetStockObject(BLACK_PEN));
MoveToEx(hDC,2000,2900,NULL);
LineTo(hDC,10000,2900);

hPrevFont=SelectObject(hDC,hMedFont);
SetTextColor(hDC,0xFF0000);      // write in red
```

```
pText="Customer Name: ";
TextOut(hDC,2000,3000,pText,strlen(pText));
SelectObject(hDC,hPrevFont);   // restore

hPrevFont=SelectObject(hDC,hMedBoldFont);
pText="Affront Dog Collars";
TextOut(hDC,3800,3000,pText,strlen(pText));
SelectObject(hDC,hPrevFont);   // restore

MoveToEx(hDC,2000,3400,NULL);
LineTo(hDC,10000,3400);
```

You can use Windows GUI functions to draw text, line, rectangle, ellipse, images, etc. You can also use the map mode function to draw the content in a different unit.

4. End the page.

```
PdcEndPage(DocId);
```

The steps 2 to 4 can be repeated to create a multi-page report.

5. *After the PDF generation process, end the session by calling the PdcEndDoc function. This function assembles the final document and frees up the memory used by the session.*

```
PdcEndDoc(DocId)
```

Please refer to the demo.c file for a working source code example.

Application Interface functions

These API functions allow you to generate a PDF document. Your application must include the PDC.H file (c/c++).

The following is a description of the PDC API functions in an alphabetic order:

In This Chapter

PdcEndDoc
PdcEndPage
PdcGetLastMessage
PdcSetPictQuality
PdcStartDoc
PdcStartPage
PdcResetLastMessage
PdcSetFlags

PdcEndDoc

Terminate the current document.

BOOL PdcEndDoc(id)

HGLOBAL PdcEndDoc2(id, pSize)

DWORD id;	Document Id.
LPINT pSize;	Size of the text in returned global memory handle.

Description: One of these two functions is called after all pages are drawn to terminate the PDF creation process.

Return Value: The PdcEndDoc function returns TRUE when successful. This function writes the PDF text to the output file specified when calling the PdcStartDoc function.

The PdcEndDoc2 function return a global memory handle containing the PDF data. The size of the PDF text in the global memory handle is returned using the pSize parameter. This function returns NULL to indicate an error condition.

PdcEndPage

Terminate the current page.

BOOL PdcEndPage(id)

DWORD id;	Document Id.

Description: This function is called after the page drawing for the current page is completed.

Return Value: The function returns TRUE when successful.

PdcGetLastMessage

Get the last message.

int PdcGetLastMessage(id, PdcMessage, DebugMessage);

DWORD id;	Document Id.
LPBYTE PdcMessage;	Returns the default user message text in English
LPBYTE DebugMsg;	Returns any debug message associated with the last message. The debug message need not be displayed to the user.

Return Value: This function returns the last message generated by the editor. This value is valid only if saving of the messages is enabled by setting the RFLAG_RETURN_MSG_ID flag. This flag is set using the PdcSetFlags function.

PdcSetPictQuality

Specify the picture quality when generating the pdf.

int PdcSetPictQuality(id, quality);

DWORD id;	Document Id.
int quality;	Specify the picture quality from 1 (lowest) to 5 (highest). default = 3.

Return Value: This function returns true when successful.

PdcStartDoc

Begin the PDF generation process.

int PdcStartDoc(LicKey,OutFile, author, producer, title, subject, keywords, date, ModDate, flags)

int PdcStartDoc2(LicKey,OutFile, author, producer, title, subject, keywords, date, ModDate, flags, PermFlags, OwnerPassword, UserPassword)

LPBYTE LicKey;	// Your license key is e-mailed to you after your order is processed.
LPBYTE OutFile;	// The name of the PDF output file.
LPBYTE author;	// The optional author name for the document. Pass a blank string as default.
LPBYTE producer;	// The optional producer name for the document. Pass a blank string as default.
LPBYTE title;	// The optional title for the document. Pass a blank string as default.
LPBYTE subject;	// The optional subject string for the document . Pass a blank string as default.
LPBYTE keywords;	// The optional keywords for the document. Pass a blank string as default.
LPBYTE date;	// The optional creation date for the document. Pass a blank string as default.
LPBYTE ModDate;	// The optional modification date for the document. Pass a blank string as default.
DWORD flags;	// The following flag constants are available:

PDFFLAG_COMPRESS_TEXT	Compress text in the PDF file.
PDFFLAG_EMBED_FONTS	Always embed fonts.
PDFFLAG_NO_BOOKMARK	Don't create bookmarks
PDFFLAG_NO_BOOKMARK_PANE	Don't open the bookmark pane
PDFFLAG_PDFA	Generate pdf/a compliant document.
PDFFLAG_PDFA_1B	Generate pdf/a-1b compliant document.
PDFFLAG_TAGGED	Generate tagged document.

	PDFFLAG_AES_128	Use AES-128 encryption
	PDFFLAG_EXACT_TEXT_PLACEMENT	Generate pdf code to place each character at exact location.

DWORD PermFlags; // Use this flag to specify the permissions granted when the PDF document is being viewed or manipulated without using the owner password. You can use one or more of the following flags using the OR operator:

PERM_PRINT	Allow printing operation
PERM_COPY	Allow copying operation
PERM_MOD	Allow document modification

LPWORD OwnerPassword; // Optional document owner password.

When either an owner or a user password is specified, the PDF document is written out using Adobe standard encryption mechanism.

An owner password in the PDF document requires a PDF editor to prompt the user for the owner password and allow PDF modification only when the supplied owner password matches the encrypted owner password found in the file.

LPWORD UserPassword; // Optional user password

When either an owner or a user password is specified, the PDF document is written out using Adobe standard encryption mechanism.

A user password in the PDF document requires a PDF viewer to prompt the user for the user password and allow PDF display only when the supplied user password matches the encrypted user password (or owner password) found in the file.

Description: This function begins PDF document creation.

Return Value: This function returns a non-zero Document Id for the output process. The Document Id value is needed for passing to other Pdf APIs. This function returns 0 to indicate an error condition.

 

PdcStartPage

Start a new page.

HDC PdcStartPage(id, PageWidth, PageHeight)

DWORD id;	Document Id.
int PageWidth;	Page width in twips units (1 inch = 1440 twips).
int PageHeight;	Page height in twips units.

Description: This function is called to start a new page.

Return Value: The function returns the device context to draw the contents of the page. A null value would indicate an error.

PdcResetLastMessage

Reset the last editor message.

BOOL PdcResetLastMessage(id)

DWORD id; Document Id.

Description: This function can be called before calling any other function to reset the last error message.

Return Value: The function returns TRUE when successful.

See Also

PdcGetLastMessage
PdcSetFlags

PdcSetFlags

Set certain flags or retrieve the values of the flags.

DWORD PdcSetFlags(id, set, flags)

DWORD id;	Document Id.
BOOL set;	TRUE to set the given flags, FALSE to reset the given flags
DWORD flags;	Flags (bits) to set or reset. Currently, the following flag values are available:

RFLAG_RETURN_MSG_ID	Do not display the error messages. Save the error code to be later retrieved using the PdcGetLastMessage function.
RFLAG_FIRST_MESSAGE_ONLY	Display the first message. Other messages would be suppressed.

Return value: This function returns the new value of all the flags. Call this function with the 'flags' parameter set to zero to retrieve flag values without modifying it.