



Software License Agreement

DOCX to PDF Converter

For Win32/Win64

Vesion 14

2008-2024

ALL RIGHTS RESERVED BY

SUB SYSTEMS, INC.

3200 Maysilee Street

Austin, TX 78728

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 DOCX to 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.
- H. The license remains valid for 12 months after the issue date. The subsequent year license renewal cost is discounted by 20 percent from the license acquisition cost. The license includes standard technical support, patches and new releases.

I. You may not disable, deactivate or remove any license enforcement mechanism used by the software.

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.



Getting Started

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

In This Chapter

[Files](#)

[License Key](#)

[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
DPS.H	The <i>include</i> file to include into a C/C++ application module that calls the Dps routine. It contains the constant definitions and the prototypes for the API functions.
DPS32.DLL	The DLL file
DPS32.LIB	Import library for the DPS32 DLL
ter31.dll	Used internally by the DPS32.DLL
txml2.dll	Used internal by the DPS32.DLL
pd32.dll	Used internally by the DPS32.DLL
DPCC.DLL	Wrapper DLL to used with an ASP page

Visual Basic Interface and Demo Files:

DPS.BAS	Function declaration file.
DMO_VB.FRM	Demo form file.
DMO_VB.BAS	Demo variable declaration file.
DMO_VB.VPB	Demo project file.



License Key

Your License Key and License number are e-mailed to you after your order is processed.

You would set the license information using the `DpsSetLicenseInfo` static function. This should be preferably done before creating the converter session to avoid pop-up nag screens.

```
int DpsSetLicenseInfo(LPBYTE LicenseKey, LPBYTE LicenseNumber, LPBYTE  
CompanyName);
```

LicenseKey: Your license key is available in the product delivery email sent to you upon the purchase of the product. It consists of a string in the form of "xxxxx-yyyyy-zzzzz".

LicenseNumber: Your license number is also available in the product delivery email. The license number string starts with a "srab" or "sno" prefix.

CompanyName: Your company name as specified in your order.

Return Value: This method returns 0 when successful. A non-zero return value indicates an error condition. Here are the possible return values:

- 0 License application successful.
- 1 Invalid License Key.
- 2 Invalid License Number.
- 3 Ran out of available licenses. Please consider purchasing additional licenses.

Example:

```
result=DpsSetLicenseInfo("xxxxx-yyyyy-zzzzz", "srabnnnnn-n", "Your Company Name")
```

Replace the 'xxxxx-yyyyy-zzzzz' by your license key, replace "srabnnnnn-n" with your license number, and "Your Company Name" with your company name as specified in your order.

Note: *DpsSetLicenseInfo* method should be called only once at the beginning of your application. Calling this method for each conversion would degrade the conversion performance.

Also, you can use the `DpsGetLicenseStatus` function at anytime to retrieve the license status.



Incorporating the DLL into Your Application

A C/C++ application should include the `DPS.h` file into the application module that needs to call the `DPS32.dll`. It also should include the `DPS32.LIB` as the linker library. Please refer to the demo application for an example.

A Visual Basic application needs to include the `DPS.BAS` file in the project. Please refer to the `DMO_VB` project for an example.

Please also make sure that the dps32.dll, pdc32.dll, txml2.dll and ter31.dll files are copied to a directory available at run-time.



Sample Conversion Code

First you would create a new conversion session:

```
dim id as long
```

Set the product [license key](#) and create a session id:

```
result=DpsSetLicenseInfo("xxxxx-yyyyy-zzzzz","srabnnnnn-n","Your Company Name")
```

```
id = DpsNewSession()
```

You would use the session id to call other conversion functions.

Here are sample code examples to convert DOCX to PDF format.

1. Convert an DOCx file to an PDF file.

```
DpsConvertFile(id,"test.docx","test.pdf")
```

2. Convert an DOCx string to an PDF string

```
Dim hMem as long
```

```
Dim OutSize as long
```

```
Dim DocXstring as string
```

```
hMem = DpsConvertBuffer(id,DocXstring, Len(DocXstring),  
OutSize)
```

```
If (hMem <> 0) Then
```

```
    PdfString = Space$(OutSize + 1) ' allocate space for the  
                                     output string
```

```
    DpsHandleToStr(PdfString, OutSize, hMem) ' copy pdf from  
                                             hMem global handle to the PdfString variable.
```

```
End If
```

After the conversion process, end the session by calling the DpsEndSession function. This frees up the memory used by the session.

LPINT OutStringLen; The variable to receive the length of the converted document.

Return value: This function returns a global memory handle containing the converted documented. You can either use the DpsHandleToStr or GlobalLock functions to access the data string contained in this global memory handle. GlobalLock is a Windows SDK function.

A null return values indicates an error.

Examples:

```
Dim hMem as long
Dim OutSize as long
Dim DocXstring as string

hMem = DpsConvertBuffer(id,DocXstring, Len(DocXstring),
OutSize)
If (hMem <> 0) Then
    PdfString = Space$(OutSize + 1) ' allocate space for the
                                   output string
    DpsHandleToStr(PdfString, OutSize, hMem) ' copy pdf from
                                             hMem global handle to the PdfString variable.
End If
```



DpsConvertFile

Convert docx to pdf using disk files.

BOOL DpsConvertFile(id, InFile, OutFile)

DWORD id; Session id.

LPBYTE InFile; Input file containing DOCx document

LPBYTE OutFile; Output files, contains the converted document

Return value: This function returns TRUE when successful.

Examples:

```
DpsConvertFile(id, "test.docx", "test.pdf")
```


int DpsGetLicenseStatus()

Return Value:

- 0 License application successful.
- 1 Invalid License Key.
- 2 Invalid License Number.
- 3 Ran out of available licenses. Please consider purchasing additional licenses.
- 4 The evaluation period has expired.

You can use the DpsGetLicenseStatus function at anytime to retrieve the license status.



DpsHandleToStr

Convert a global memory handle to a Visual Basic string.

BOOL DpsHandleToStr(string, length, hMem)

LPBYTE string; pointer to a visual basic string

long length length of the string

HGLOBAL hMem; Global memory handle

Description: This function can be used to copy the contents of a global memory handle to a given visual basic string. The calling routine must expand the string to appropriate length before calling this function.

Example:

```
string=space(length)
HandleToStr(string,length,hMem)
```

The input global memory handle is freed up after copying its contents to the string.

Return Value: This function returns TRUE if successful.



DpsNewSession

DPPROP_EXACT_TEXT_PLACEMENT Use exact text placement. This flag causes the editor to emit character width for every character. Default = false.

BOOL val; Specify TRUE or FALSE value for the specified flag.

Return value: This function returns TRUE when successful.



DpsSetNumProp

Set a numeric property for the pdf document.

BOOL DpsSetNumProp(id, prop, val)

DWORD id; Session id.

int prop; One of the following property type to set:

DPPROP_PICT_QUALITY Specify the picture quality from 1 to 5, where 1=lowest, 5=highest, 3=default.

DPPROP_PERM_FLAGS Permission flags. The permission flags is effective only when UserPassword or OwnerPassword is also specified using the DpsSetTextProp method.
One or more of the following permission flags can be specified using the 'val' parameter:

PERM_PRINT Allow printing of the document

PERM_MOD Allow modification

PERM_COPY Allow copying of the document to clipboard.

You can specify more than one permission flags using the 'or' operator.

int val; The numeric value of the selected property.

Return value: This function returns TRUE when successful.



DpsSetTextProp

Set a text property for the pdf document.

BOOL DpsSetTextProp(id, prop, val)

DWORD id;

Session id.

int prop;

One of the following property type to set:

DPPROP_AUTHOR	Author of the document
DPPROP_TITLE	Document title
DPPROP_SUBJECT	Subject
DPPROP_KEYWORDS	Key words
DPPROP_CRE_DATE	Creation date
DPPROP_MOD_DATE	Modification date
DPPROP_USER_PASSWORD	User password
DPPROP_OWNER_PASSWORD	Owner password

LPBYTE val;

The text value of the selected property.

Return value: This function returns TRUE when successful.



ASP Interface

This chapter describes the usage of the DOCX to PDF Converter within an ASP page. The product includes an additional wrapper DLL called DPCC.DLL which is used to access the converter within an ASP page. Please follow the following steps:

Copy ter31.dll, txml2.dll, pdc32.dll, DPS32.dll and dpcc.dll to the Windows system directory, or any other directory available at the run-time. Now register dpcc.dll using the regsvr32 system utility. The other dlls do not need registration. Now you are ready to use

this product within an ASP page.

Here is an example ASP page to show a conversion of DOCX string into an PDF string:

```
<%@ LANGUAGE = "VBSCRIPT"%>
<%
Option Explicit

Dim DocxString
Dim sPdf
Dim obj

Set obj = Server.CreateObject("dpcc.converter")

call obj.SetFlags(1,obj.VAL_DPFLAG_RETURN_MSG_ID)' quiet mode

DocxString=""
sPdf=""

' read the test.docx file into a string
DocxString =
    obj.FileToString("c:\inetpub\wwwroot\DmoDpc\test.docx")

if len(DocxString) > 0 then
    sPdf = obj.ConvertBuffer(CStr(DocxString))
End If

Set obj = Nothing

%>

<html>
<head>
</head>

<body>
<p> Some text before </p>

<%
```

```

Function StringToArray(S)
    Dim i, ByteArray
    For i=1 To Len(S)
        ByteArray = ByteArray & ChrB(Asc(Mid(S,i,1)))
    Next
    StringToArray = ByteArray
End Function

Response.Clear()
Response.Charset = ""
Response.ContentType = "application/pdf"

Response.AddHeader "Content-Disposition",
"inline;filename=" + "test.pdf"

Response.BinaryWrite(StringToArray(sPDF))

Response.Flush()
Response.End()

%>

<p> Some text after </p>
</body>
</html>

```

When the above asp file is loaded, IE displays generated pdf document.

The method names used by the dpcc.dll are the same as the functions mentioned in the Application Interface functions. However the 'Dps' prefix is not used by the dpcc method names. For example, the DpsConvertFile function is named as ConvertFile within the dpcc.dll file.

Also, the constants values are prefixed with an 'VAL_' prefix. For example, the constant DPFLAG_SEGMENT_ONLY becomes VAL_DPFLAG_SEGMENT_ONLY.