



# Software License Agreement

**DOCX to PDF Converter**

**For Win32/Win64**

Version 7

2008-2017

*ALL RIGHTS RESERVED BY*

*SUB SYSTEMS, INC.*

**4380 Caldwell Palm Circle**

**Round Rock, TX 78665**

**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 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 \$500 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](mailto: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.



## 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
ter24.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 ter24.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.*

```
DpsEndSession(id)
```



## Application Interface functions

These API functions allow you to convert from docx to pdf format. Your application must include the DPS.H file (c/c++), or DPS.BAS (VB) files. These files declare these functions.

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

### In This Chapter

[DpsConvertBuffer](#)

[DpsConvertFile](#)

[DpsEndSession](#)

[DpsGetLastMessage](#)

[DpsGetLicenseStatus](#)

[DpsHandleToStr](#)

[DpsNewSession](#)

[DpsResetLastMessage](#)

[DpsSetFlags](#)

[DpsSetBoolProp](#)

[DpsSetNumProp](#)

[DpsSetTextProp](#)



## DpsConvertBuffer

**Convert docx to pdf using memory buffers.**

HGLOBAL DpsConvertBuffer(id, InString, InStringLen, OutStringLen)

DWORD id;	Session id.
LPBYTE InString;	Input string containing DOCx document.
int InStringLen;	length of the input document string.
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")
```



## DpsEndSession

**End a conversion session.**

BOOL DpsEndSession(id)

DWORD id;

Session id.

**Description:** This function is called at the end of the conversion process to free up the session related resources.

**Return Value:** The function returns TRUE when successful.



## DpsGetLastMessage

**Get the last message.**

```
int DpsGetLastMessage(id, DpsMessage, DebugMessage);
```

DWORD id;	Session id.
LPBYTE DpsMessage;	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 DPFLAG\_RETURN\_MSG\_ID flag. This flag is set using the DpsSetFlags function.



## DpsGetLicenseStatus

**Get the license status.**

```
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

**Create a new conversion session.**

DWORD DpsNewSession()

**Description:** This function needs to be called before calling any other conversion function. This function creates a new conversion session.

The DpsEndSession must be called at the end to free up the session resources. All other conversion functions are called between the calls to the DpsNewSession and DpsEndSession functions.

**Return Value:** The function returns a non-zero session-id when successful. A zero value indicates a fail return.



## DpsResetLastMessage

**Reset the last editor message.**

BOOL DpsResetLastMessage(id)

DWORD id;    Session 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

[DpsGetLastMessage](#)

[DpsSetFlags](#)



## DpsSetFlags

**Set certain flags or retrieve the values of the flags.**

DWORD DpsSetFlags(id, set, flags)

DWORD id;                                      Session 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:

DPFLAG_RETURN_MSG_ID	Do not display the error messages. Save the error code to be later retrieved using the DpsGetLastMessage function.
----------------------	--

**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.



## DpsSetBoolProp

**Set a boolean property for the pdf document.**

BOOL DpsSetBoolProp(id, prop, val)

DWORD id;

Session id.

int prop;

One of the following property type to set:

DPPROP_COMPRESS_TEXT	Compress text inside the pdf document. Default = false.
DPPROP_EMBED_FONTS	Embed font data in the pdf document. Default = false.
DPPROP_BOOKMARK	Include bookmarks. Default = true.
DPPROP_HYPERLINK	Include hyperlinks. Default = true.
DPPROP_RC4_128	Enable RC4 128 bit security when a password is specified.
DPPROP_AES_128	Enable AES 128 bit security when a password is specified.
DPPROP_PDFA_1B	Generate PDF-A/1b compliant document
DPPROP_PDFA	Generate PDF-A compliant document
DPPROP_NO_TOC_UPDATE	Do not update the table-of-content before generating the pdf document.
DPPROP_USE_ORIG_JPG	Insert the original jpeg data into the pdf document.
DPPROP_TAGGED_PDF	Generate tagged pdf.
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 ter24.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 StringToByteArray(S)
```

```
    Dim i, ByteArray
```

```
    For i=1 To Len(S)
```

```
        ByteArray = ByteArray & ChrB(Asc(Mid(S,i,1)))
```

```
    Next
```

```
    StringToByteArray = ByteArray
```

```
End Function
```

```
Response.Clear()
```

```
Response.Charset = ""
```

```
Response.ContentType = "application/pdf"
```

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



```
Response.BinaryWrite(StringToByteArray(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.