Software License Agreement

DOC to PDF Converter

version 16

For Win32/Win64

2008-2024

ALL RIGHTS RESERVED BY

SUB SYSTEMS, INC.

3200 Maysilee Street

Austin, TX 78728

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 DOC to PDF or 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 DOC to PDF Converter into your application.

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
WPS.H	The *include* file to include into a C/C++ application module that calls the Wps routine. It contains the constant definitions and the prototypes for the API functions.
WPS32.DLL	The DLL file
WPS32.LIB	Import library for WPS32 DLL
ter31.dll	Used internally by WPS32.DLL
pdc32.dll	Used internally by WPS32.DLL
txml2.dll	Used internally by WPS32.DLL
wrs15.dll	Used Internally by WPS32.DLL
WPCC.DLL	Wrapper DLL to used with an ASP page

Visual Basic Interface and Demo Files:

WPS.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 WpsSetLicenseInfo static function. This
should be preferably done before creating the converter session to avoid pop-up nag
screens.

int WpsSetLicnseInfo(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 "smo" 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=WpsSetLicenseInfo("xxxxx-yyyyy-zzzzz","srabnnnn-n","Your Company Name")

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

Note: *WpsSetLicenseInfo 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 WpsGetLicenseStatus function at anytime to retrieve the license
status.

Incorporating the DLL into Your Application

A C/C++ application should include the WPS.h file into the application module that needs
to call the WPS32 dll. It also should include the WPS32.LIB as the linker library. Please
refer to the demo application for an example.

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

Please also make sure that the wps32.dll, pdc32.dll, txml2.dll, wrs15.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=WpsSetLicenseInfo("xxxxx-yyyyy-zzzzz","srabnnnnn-n","Your Company Name")

```
 id = WpsNewSession()
```

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

Here are sample code examples to convert Word DOC/DOCX to PDF format.

1. Convert an DOC file to an PDF file.

```
WpsConvertFile(id,"test.doc","test.pdf")
```

2. Convert an DOC string to an PDF string

```
Dim hMem as long
Dim OutSize as long
Dim DocString as string

hMem = WpsConvertBuffer(id,DocString, Len(DocString),
OutSize)
If (hMem <> 0) Then
   PdfString = Space$(OutSize + 1) ' allocate space for the
                                    output string
   WpsHandleToStr(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 WpsEndSession function. This frees up the memory used by the session.

```
WpsEndSession(id)
```


Application Interface functions

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

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

WpsConvertBuffer

Convert doc to pdf using memory buffers.

HGLOBAL WpsConvertBuffer(id, InString, InStringLen, OutStringLen)

DWORD id;   Session id.

LPBYTE InString;   Input string containing DOC 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 WpsHandleToStr 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 DocString as string


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

WpsConvertDocToPdfBytes

Convert DOC string to PDF byte array.

ByteArray WpsConvertDocToPdfBytes(InString)

string InString;   Input string containing DOC document.

Remark: This method is available only when using the wpcc.dll COM object.

Return value: This function returns byte array containing the converted documented.

A null return values indicates an error.

Examples:

```
 PdfBytes = obj.ConvertDocToPdfBytes(DocString)
```

```
Response.BinaryWrite(PdfBytes)
```

WpsConvertFile

Convert doc to pdf using disk files.

BOOL WpsConvertFile(id, InFile, OutFile)

DWORD id;   Session id.

LPBYTE InFile;   Input file containing DOC document

LPBYTE OutFile;   Output files, contains the converted document

Return value: This function returns TRUE when successful.

Examples:

```
WpsConvertFile(id,"test.doc","test.pdf")
```

WpsEndSession

End a conversion session.

BOOL WpsEndSession(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.

WpsGetLastMessage

Get the last message.

int WpsGetLastMessage(id, WpsMessage, DebugMessage);

DWORD id;   Session id.

LPBYTE WpsMessage;   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
WPFLAG_RETURN_MSG_ID flag. This flag is set using the WpsSetFlags function.

WpsGetLicenseStatus

Get the license status.

int WpsGetLicnseStatus()

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 WpsGetLicenseStatus function at anytime to retrieve the license status.

WpsHandleToStr

Convert a global memory handle to a Visual Basic string.

BOOL WpsHandleToStr(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.

WpsNewSession

Create a new conversion session.

DWORD WpsNewSession()

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

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

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

WpsResetLastMessage

Reset the last editor message.

BOOL WpsResetLastMessage(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.

▼

WpsSetFlags

Set certain flags or retrieve the values of the flags.

DWORD WpsSetFlags(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:

    WPFLAG_RETURN_MSG_ID   Do not display the error messages. Save the error code to be later retrieved using the WpsGetLastMessage 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.

▼

WpsSetBoolProp

Set a boolean property for the pdf document.

BOOL WpsSetBoolProp(id, prop, val)

DWORD id;   Session id.

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

    WPPROP_COMPRESS_TEXT   Compress text inside the pdf document. Default = false.

    WPPROP_EMBED_FONTS   Embed font data in the pdf document. Default = false.

    WPPROP_BOOKMARK   Include bookmarks. Default = true.

WPPROP_HYPERLINK	Include hyperlinks. Default = true.
WPPROP_ENABLE_GDIPLUS	Enable GDI++. The ssgp.dll file must be installed for this property to be effective.
WPPROP_RC4_128	Enable RC4 128 bit security when password is specified.
WPPROP_AES_128	Enable AES 128 bit security when password is specified.
WPPROP_NO_TOC_UPDATE	Set to true to use the original table-of-content.
WPPROP_USE_ORIG_JPG	Set to true to use the original jpeg data in the generated pdf file.
WPPROP_TAGGED_PDF	Generate tagged pdf file.
WPPROP_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.

WpsSetNumProp

Set a numeric property for the pdf document.

BOOL WpsSetNumProp(id, prop, val)

DWORD id;   Session id.

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

WPPROP_PICT_QUALITY	Set the picture quality from 1 to 5, 1=lowest, 5=highest, 3=default.
WPPROP_PERM_FLAGS	Permission flags. The permission flags is effective only when the OwnerPassword is also specified using the WpsSetTextProp 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.

WpsSetTextProp

Set a text property for the pdf document.
BOOL WpsSetTextProp(id, prop, val)

DWORD id;   Session id.

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

WPPROP_AUTHOR	Author of the document
WPPROP_TITLE	Document title
WPPROP_SUBJECT	Subject
WPPROP_KEYWORDS	Key words
WPPROP_CRE_DATE	Creation date
WPPROP_MOD_DATE	Modification date
WPPROP_USER_PASSWORD	User password
WPPROP_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 DOC to PDF Converter within an ASP page. The product includes an additional wrapper DLL called WPCC.DLL which is used to access the converter within an ASP page. Please follow the following steps:

Copy ter31.dll, pdc32.dll, txml2, wrs15.dll, wps32.dll and wpcc.dll to the Windows system directory, or any other directory available at the run-time. Now register wpcc.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 DOC string into an PDF string:

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

Dim DocString
Dim PdfBytes
Dim obj

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

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

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

if len(DocString) > 0 then
  PdfBytes = obj.ConvertDocToPdfBytes(CStr(DocString))
End If

Set obj = Nothing


%>

<html>
<head>
</head>
```

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


<%

      Response.Clear()

      Response.Charset = ""

      Response.ContentType = "application/pdf"


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

                  + "test.pdf"


      Response.BinaryWrite(PdfBytes)


      Response.Flush()

      Response.End()


%>


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

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

----

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

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