Software License Agreement

DOC to RTF Converter

For Win32/Win64

Vesion 16

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 HTML or DOCX to HTML 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 HTML 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
WHS.H	The *include* file to include into a C/C++ application module that calls the WHS routine. It contains the constant definitions and the prototypes for the API functions.
WHS32.DLL	The DLL file
WHS32.LIB	Import library for the WHS32 DLL
ter31.dll	Used internally by the WHS32.DLL
txml2.dll	Used internal by the WHS32.DLL
hts26.dll	Used internally by the WHS32.DLL
txml2.dll	Used internally by the WHS32.DLL
wrs15.dll	Used internally by the WHS32.DLL
WHCC.DLL	Wrapper DLL to used with an ASP page

Visual Basic Interface and Demo Files:

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

int WhsSetLicnseInfo(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=WhsSetLicenseInfo("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: *WhsSetLicenseInfo 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 WhsGetLicenseStatus function at anytime to retrieve the license status.

Incorporating the DLL into Your Application

A C/C++ application should include the WHS.h file into the application module that needs to call the WHS32 dll. It also should include the WHS32.LIB as the linker library. Please refer to the demo application for an example.

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

Please also make sure that the whs32.dll, hts26.dll,txml2.dll, wrs15.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=WhsSetLicenseInfo("xxxxx-yyyyy-zzzzz",
                  "srabnnnnn-n","Your Company Name")


  id = WhsNewSession()
```

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

Here are sample code examples to convert from DOC or DOCX format to HTML format.

1. Convert a DOC file to an HTML file.

```
WhsConvertFile(id,"test.doc","test.htm")
```

2. Convert an DOC string to an HTML string

```
Dim hMem as long
Dim OutSize as long
Dim HtmlString as string

hMem = WhsConvertBuffer(id,DocString, Len(DocString),
OutSize)
If (hMem <> 0) Then
```

```
        HtmlString = Space$(OutSize + 1) ' allocate space for the
                                  output string
        WhsHandleToStr(HtmlString, OutSize, hMem) ' copy html from
                hMem global handle to the HtmlString variable.
     End If
```

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

```
  WhsEndSession(id)
```

▲ ▼

Application Interface functions

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

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

▲ ▼

WhsConvertBuffer

Convert doc to html using memory buffers.

HGLOBAL WhsConvertBuffer(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 WhsHandleToStr 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:

Convert an DOC string to an HTML string

```
   Dim hMem as long
   Dim OutSize as long
   Dim HtmlString as string


   id = WhsNewSession()


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


WhsConvertFile

Convert doc to html using disk files.

BOOL WhsConvertFile(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:

Convert an DOC file to an HTML file.

```
id = WhsNewSession()

WhsConvertFile(id, "test.doc", "test.htm")
```

WhsEndSession

End a conversion session.

BOOL WhsEndSession(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.

WhsGetLastMessage

Get the last message.

int WhsGetLastMessage(id, WhsMessage, DebugMessage);

DWORD id;   Session id.

LPBYTE WhsMessage;	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 WHFLAG_RETURN_MSG_ID flag. This flag is set using the WhsSetFlags function.

WhsGetLicnseStatus

Get the license status.

int WhsGetLicnseStatus()

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 WhsGetLicenseStatus function at anytime to retrieve the license status.

WhsGetPictCount

Return total number of pictures found during conversion.

int WhsGetPictCount(id)

DWORD id;   Session id.

Description: Thjs function is called after converting a file to get the number of pictures encountered during conversion. The picture names can be retrieved by using the

WhsGetPictName function.

Return Value: When converting from doc to html formats, it returns the number of picture files referenced in the output html file.

WhsGetPictName

Return name of the picture file found during conversion.

BOOL WhsGetPictName(id, idx, PictFileName)

DWORD id;   Session id.

int idx;   The index of the picture file. The index value must be between 0 and (Total pictures - 1). The total picture count can be retrieved using the WhsGetPictCount function.

LPBYTE PictFileName;   The location to receive the picture file name.

VB Note: This variable is defined as 'string'. This parameter must be assigned 300 spaces before calling this function:

```
PictFileName= space$(300)

WhsGetPictName(id,idx,PictFileName)
```

Description: Thjs function is called after converting a file to get the individual picture file name or picture url.

Return Value: This function returns TRUE when successful.

WhsHandleToStr

Convert a global memory handle to a Visual Basic string.

BOOL WhsHandleToStr(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.

WhsNewSession

Create a new conversion session.

DWORD WhsNewSession()

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

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

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

WhsResetLastMessage

Reset the last editor message.

BOOL WhsResetLastMessage(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
WhsGetLastMessage
WhsSetFlags

WhsSetDownloadDir

Set the directory to be used for downloading pictures from Internet, and the directory where the picture files are generated during doc to html conversion.

BOOL WhsSetDownloadDir(id, DownloadDir);

DWORD id;   Session id.

LPBYTE DownloadDir;   The folder to be used to download pictures form Internet.

Comments: Normally the converter uses the current working directory to download pictures from Internet for temporary processing. However, when the converter is hosted in an ASP application, the default current directory is the Window system directory. In this case, you would use this function to specify a different download directory, such as c:\inetpub\wwwroot\MyProject. A directory specified with this function should have write access so the temporary files can be created.

The specified folder is also used to place the picture files generated during doc to html conversion.

Return Value: This function returns TRUE when successful.

WhsSetFlags

Set certain flags or retrieve the values of the flags.

DWORD WhsSetFlags(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:

   WHFLAG_RETURN_MSG_ID   Do not display the error messages. Save the error code to be later retrieved using the WhsGetLastMessage function.

   WHFLAG_SEGMENT_ONLY   Do not include html header and body tags. Generate just html segment.

WHFLAG_LIST_TO_TEXT	Convert doc lists to text numbers in html output.
WHFLAG_NO_SPAN_TAG	Do not write the 'span' tag to html output.
WHFLAG_NO_FONT	Do not write the 'font' tag to html output.
WHFLAG_FLASH_SUPPORT	Create html tags to support Flash.
WHFLAG_XLATE_SINGLE_QUOTE	Write single quote character as ' to html output.
WHFLAG_PNG_RGB	Save PNG pictures in high-quality RGB colors.

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.

ASP Interface

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

Copy ter31.dll, txml2.dll, hts26.dll, txml2.dll, wrs15.dll, whs32.dll and whcc.dll to the Windows system directory, or any other directory available at the run-time. Now register whcc.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 HTML string:

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

Dim DocString
Dim HtmlString
Dim obj

Set obj = Server.CreateObject("whcc.converter")
```

```
call obj.SetDownloadDir("c:\inetpub\wwwroot\DmoWhc") ' folder
                                    where the picture files are created
call obj.SetFlags(1,obj.VAL_WHFLAG_RETURN_MSG_ID)    ' quiet
mode
call obj.SetFlags(1,obj.VAL_WHFLAG_NO_PICT_PATH)     ' dont add
                                    file path to the 'img' tag


DocString=""
HtmlString=""


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


' do conversion
HtmlString = obj.ConvertBuffer(CStr(DocString))



Set obj = Nothing



%>


' show the converted html
<%= HtmlString %>
```

IE would display the converted html data.

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

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