Software License Agreement

HTML - RTF Converter

For Win32/Win64

Vesion 20

2004-2024

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 HTML to/from RTF 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 HTML - RTF 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
HRS.H	The *include* file to include into a C/C++ application module that calls the HRS routine. It contains the constant definitions and the prototypes for the API functions.
HRS32.DLL	The DLL file
HRS32.LIB	Import library for the HRS32 DLL
ter31.dll	Used internally by the HRS32.DLL
hts26.dll	Used internally by the HRS32.DLL
HRCC.DLL	Wrapper DLL to used with an ASP page

Visual Basic Interface and Demo Files:

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

int HrsSetLicenseInfo(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=HrsSetLicenseInfo("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: *HrsSetLicenseInfo 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 HrsGetLicenseStatus function at anytime to retrieve the license status.

Incorporating the DLL into Your Application

A C/C++ application should include the HRS.h file into the application module that needs to call the HRS32 dll. It also should include the HRS32.LIB as the linker library. Please refer to the demo application for an example.

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

Please also make sure that the hrs32.dll, hts26.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=HrsSetLicenseInfo("xxxxx-yyyyy-zzzzz","srabnnnnn-
n","Your Company Name")
```

```
id = HrsNewSession()
```

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

Here are sample code examples to convert between HTML and RTF formats.

1. Convert an HTML file to an RTF file.

```
HrsConvertFile(id,"test.htm","test.rtf",XLATE_HTML_TO_RTF)
```

2. Convert an RTF file to an HTML file.

```
HrsConvertFile(id,"test.rtf","test.htm",XLATE_RTF_TO_HTML)
```

3. Convert an HTML string to an RTF string

```
Dim hMem as long
Dim OutSize as long
Dim RtfString as string

hMem = HrsConvertBuffer(id,HtmlString, Len(HtmlString),
OutSize,
                        XLATE_HTML_TO_RTF)
If (hMem <> 0) Then
    RtfString = Space$(OutSize + 1) ' allocate space for the
                                      output string
    HrsHandleToStr(RtfString, OutSize, hMem) ' copy rtf from
```

```
                        hMem global handle to the RtfString variable.
        End If
```

4. Convert an RTF string to an HTML string

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


    hMem = HrsConvertBuffer(id,RtfString, Len(RtfString), OutSize,
                            XLATE_RTF_TO_HTML)
    If (hMem <> 0) Then
        HtmlString = Space$(OutSize + 1) ' allocate space for the
                                            output string
        HrsHandleToStr(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 HrsEndSession function. This frees up the memory used by the session.

```
    HrsEndSession(id)
```

Application Interface functions

These API functions allow you to convert between html and rtf formats. Your application must include the HRS.H file (c/c++), or HRS.BAS (VB) files. These files declare these functions.

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

HrsConvertBuffer

Convert html to/from rtf using memory buffers.

HGLOBAL HrsConvertBuffer(id, InString, InStringLen, OutStringLen, XlateType)

DWORD id;   Session id.

LPBYTE InString;   Input string containing HTML or RTF document.

int InStringLen;   length of the input document string.

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

int XlateType;   Translation type

       XLATE_HTML_TO_RTF   HTML to RTF conversion. The input-string must contain an HTML document.

       XLATE_RTF_TO_HTML   RTF to HTML conversion. The input-string must contain an RTF document.

Return value: This function returns a global memory handle containing the converted documented. You can either use the HrsHandleToStr 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:

  1. Convert an HTML string to an RTF string

```
Dim hMem as long
Dim OutSize as long
Dim RtfString as string


id = HrsNewSession()


hMem = HrsConvertBuffer(id, HtmlString, Len(HtmlString),
OutSize,
                         XLATE_HTML_TO_RTF)
If (hMem <> 0) Then
    RtfString = Space$(OutSize + 1) ' allocate space for the
                                     output string
    HrsHandleToStr(RtfString, OutSize, hMem) ' copy rtf from
            hMem global handle to the RtfString variable.
End If
```

2. Convert an RTF string to an HTML string

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


id = HrsNewSession()


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

▲ ▼

HrsConvertFile

Convert html to/from rtf using disk files.

BOOL HrsConvertFile(id, InFile, OutFile, XlateType)

DWORD id;   Session id.

LPBYTE InFile;   Input file containing HTML or RTF document

LPBYTE OutFile;   Output files, contains the converted document

int XlateType;   Translation type

      XLATE_HTML_TO_RTF   HTML to RTF conversion. The input-file must contain an HTML document.

      XLATE_RTF_TO_HTML   RTF to HTML conversion. The input-file must contain an RTF document.

Return value: This function returns TRUE when successful.

Examples:

1. Convert an HTML file to an RTF file.

```
id = HrsNewSession()

HrsConvertFile(id, "test.htm", "test.rtf",
               XLATE_HTML_TO_RTF)
```

2. Convert an RTF file to an HTML file.

```
id = HrsNewSession()

HrsConvertFile(id, "test.rtf", "test.htm",
               XLATE_RTF_TO_HTML)
```

HrsEndSession

End a conversion session.

BOOL HrsEndSession(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.

HrsGetLastMessage

Get the last message.

int HrsGetLastMessage(id, HrsMessage, DebugMessage);

DWORD id;   Session id.

LPBYTE HrsMessage;   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 HRFLAG_RETURN_MSG_ID flag. This flag is set using the HrsSetFlags function.

HrsGetLicenseStatus

Get the license status.

int HrsGetLicenseStatus()

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 HrsGetLicenseStatus function at anytime to retrieve the license status.

HrsGetPictCount

Return total number of pictures found during conversion.

int HrsGetPictCount(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 HrsGetPictName function.

Return Value: When converting from rtf to html formats, it returns the number of picture files referenced in the output html file. When converting from html to rtf format, the function returns the number pictures found in the input html file.

HrsGetPictName

Return name of the picture file found during conversion.

BOOL HrsGetPictName(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 HrsGetPictCount 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)
HrsGetPictName(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.

HrsHandleToStr

Convert a global memory handle to a Visual Basic string.

BOOL HrsHandleToStr(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.

HrsNewSession

Create a new conversion session.

DWORD HrsNewSession()

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

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

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

HrsResetLastMessage

Reset the last editor message.

BOOL HrsResetLastMessage(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.

HrsSetDownloadDir

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

BOOL HrsSetDownloadDir(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 rtf to html conversion.

Return Value: This function returns TRUE when successful.

HrsSetFlags

Set certain flags or retrieve the values of the flags.

DWORD HrsSetFlags(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:

    HRFLAG_IN_SERVER   This flag must be set when hosting the converter in an ASP or other server applications.

    HRFLAG_RETURN_MSG_ID   Do not display the error messages. Save the error code to be later retrieved using the HrsGetLastMessage function.

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

    HRFLAG_EMBED_PICTURE   Embed the pictures into the output RTF file (instead of inserting picture links).

    HRFLAG_LIST_TO_TEXT   Convert rtf lists to text numbers in html output.

    HRFLAG_NO_SPAN_TAG   Do not write the 'span' tag to html output.

    HRFLAG_NO_FONT   Do not write the 'font' tag to html output.

    HRFLAG_FLASH_SUPPORT   Create html tags to support Flash.

    HRFLAG_XLATE_SINGLE_QUOTE   Write single quote character as ' to html output.

    HRFLAG_PNG_RGB   Save PNG pictures in high-quality RGB colors.

    HRFLAG_ENABLE_GDIPLUS   Enable GDIPlus support for better image processing. For this flag to be effective, the ssgp.dll file (available with the product) must also be installed and the OS should have the GDI++ support package installed.

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.

HrsSetHdrFtrText

Set header or footer text.

BOOL HrsSetHdrFtrText(id, HdrFtrType, TextType, text)

DWORD id;   Session id.

int HdrFtrType;   Select header or footer to set:

    HF_FIRST_HDR   Header text to print on the first page.

    HF_FIRST_FTR   Footer text to print on the first page..

    HF_HDR   Regular header for all pages. When the first page header is also set, then the regular header text is printed on all pages except the first page.

    HF_FTR   Regular footer for all pages. When the first page footer is also set, then the regular footer text is printed on all pages except the first page.

int TextType;   Text type:

    HFTYPE_TEXT   Plain text.

    HFTYPE_RTF   RTF text.

    HFTYPE_HTML   Html text.

LPBYTE text;   Header or footer text. The header/footer text must be specified as plain text or RTF text depending upon the value passed for the 'TextType' parameter.

Comment: The function should be called before calling the conversion functions to set the header or footer text. You can call this function multiple times to set various types of header or footer.

Return value: This function returns TRUE when successful.

```
HrsSetHdrFtrText(id, HF_FIRST_HDR, HFTYPE_TEXT,
                  "This is first page header.");
HrsSetHdrFtrText(id, HF_FIRST_FTR, HFTYPE_TEXT,
                  "This is first page footer.");
HrsSetHdrFtrText(id,HF_HDR, HFTYPE_TEXT,
                  "This is regular page header.");
HrsSetHdrFtrText(id,HF_FTR, HFTYPE_RTF,"{\\rtf1 \\qc
     Page: {\\field{\\fldinst PAGE}{\\fldrslt 12}} of
     {\\field{\\fldinst NUMPAGES}{\\fldrslt 12}}
     \\par}" );  // rtf example to insert page: n of m string
```

HrsSetPageMargin

Set the page margins for RTF output.

BOOL HrsSetPageMargin(id, left, right, top, bottom)

DWORD id;	Session id.
int left;	Left margin in twip units (1440 twips = 1 inch)
int right;	Right margin in twip units
int top;	Top margin in twip units
int bottom	Bottom margin in twip units

Return Value: The function returns TRUE when successful.

Comment: This function is used to override the default page margins when converting an HTML document to the RTF format. This function should be called before calling the HrsConvertFile or HrsConvertBuffer if you wish override the page margin values.

HrsSetPaperOrient

Set the page orientation for RTF output.

BOOL HrsSetPaperOrient(id, orient)

DWORD id;   Session id.

int orient;   Orientation: DMORIENT_PORTRAIT or
                                   DMORIENT_LANDSCAPE

Return Value: The function returns TRUE when successful.

Comment: This function is used to override the default portrait orientation when converting an HTML document to the RTF format. This function should be called before calling the HrsConvertFile or HrsConvertBuffer if you wish override the paper orientation.

HrsSetPaperSize

Set the page size for RTF output.

BOOL HrsSetPaperSize(id, PageSize, PageWidth, PageHeight)

DWORD id;   Session id.

int PageSize;   Use one of the following Windows SDK defined
                                   constants:

Constant	Value
DMPAPER_LETTER	1
DMPAPER_LEGAL	5
DMPAPER_LEDGER	4
DMPAPER_TABLOID	3
DMPAPER_STATEMENT	6
DMPAPER_EXECUTIVE	7
DMPAPER_A3	8
DMPAPER_A4	9
DMPAPER_A5	11
DMPAPER_B4	12
DMPAPER_B5	13

If you need to use a paper size not listed above, please set the PageSize argument to zero and specify the page width and height using the next two arguments.

int PageWidth;   The page width in twips units (1440 twips = 1 inch). This argument is ignored if the PageSize is set to one of the

defined page sizes listed above.

int PageHeight;	The page height in twips units (1440 twips = 1 inch). This argument is ignored if the PageSize is set to one of the defined page sizes listed above

Return Value: The function returns TRUE when successful.

Comment: This function is used to override the default letter size paper when converting an HTML document to the RTF format. This function should be called before calling the HrsConvertFile or HrsConvertBuffer if you wish override the paper size.

ASP Interface

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

Copy ter31.dll, hts26.dll, hrs32.dll and hrcc.dll to the Windows system directory, or any other directory available at the run-time. Now register hrcc.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 RTF string into an HTML string:

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

Dim sRTF
Dim sHTML
Dim obj

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

' Set a flag to indicate that the control is being hosted in
' a server application
call obj.SetFlags(1,obj.VAL_HRFLAG_IN_SERVER)



sRTF = "{\rtf1 This \b is \b0 a test of \i RTF \i0 to \i HTML
\i0 Conversion.}"
```

```
if len(sRTF) > 0 then

  call obj.SetFlags(1,obj.VAL_HRFLAG_SEGMENT_ONLY)

  sHTML =
obj.ConvertBuffer(CStr(sRTF),obj.VAL_XLATE_RTF_TO_HTML)

End If


Set obj = Nothing
%>


<html>
<head>
</head>


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


<%= sHTML %>


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

On the client side, the asp page produces the following HTML code in IE

```
<html>

<head>
</head>


<body>


<p> Some text before </p>


<div align="left">This <font size="3"><span style="font-size:
12pt"><b>is </b>a test of <i>RTF </i>to <i>HTML
</i>Conversion.</span></font></div>



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

```
  </body>

  </html>
```

IE display the following formatted text

Some text before

This is a test of RTF to HTML Conversion.

Some text after

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

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