



Software License Agreement

RTF to Image Converter

For Win32/64

Version 11

2007-2018

ALL RIGHTS RESERVED BY

SUB SYSTEMS, INC.

1221 New Meister Lane, #2712

Pflugerville, TX 78660

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 RTF to Image 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.

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 RTF to Image 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
RIS.H	The <i>include</i> file to include into a C/C++ application module that calls the RIS routine. It contains the constant definitions and the prototypes for the API functions.
ris32.DLL	The DLL file
ris32.LIB	Import library for the ris32 DLL
ter25.dll	Used internally by the ris32.DLL
ricc.dll	Wrapper DLL to used with an ASP page

Visual Basic Interface and Demo Files:

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

```
int RisSetLicenseInfo(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=RisSetLicenseInfo("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: *RisSetLicenseInfo* 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 `RisGetLicenseStatus` function at anytime to retrieve the license status.



Incorporating the DLL into Your Application

A C/C++ application should include the RIS.h file into the application module that needs to call the ris32.dll. It also should include the ris32.LIB as the linker library. Please refer to the demo application for an example.

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

Please also make sure that the ris32.dll and ter25.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=RisSetLicenseInfo("xxxxx-yyyyy-zzzzz",  
                          "srabnnnnn-n", "Your Company Name")
```

```
id = RisNewSession()
```

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

Here are sample code examples to convert RTF to Image format.

1. Convert an RTF file to an image file.

```
PageCount = RisLoadFile(id, "test.rtf")  
  
If (PageCount > 0) Then  
    PageNo = 1 'Get image for the first page of the document  
    result = RisImageToFile(id, "test.jpg", PageNo)  
End If
```

2. Convert an RtfString to a string containing image data:

```
Dim hMem as long  
  
Dim PageCount as long  
  
Dim ImageType as long  
  
Dim OutString as string  
  
Dim RtfString as string  
  
  
'load the document and return the number of pages
```

```

'in the document
PageCount = RisLoadBuffer(id, RtfString, Len(RtfString))

If (PageCount > 0) Then
    ' get the output image type from a file name
    ImageType = RisGetImageType(id, "test.jpg")

    ' set the output image type
    call RisSetNumProp(id, RIPROP_IMAGE_TYPE, ImageType)

    ' return the page image in global memory handle
    hMem = RisImageToBuffer(id, OutSize, PageNo)
    If (hMem > 0) Then
        ' allocate space for the output string
        OutString = Space$(OutSize + 1)
        Call RisHandleToStr(OutString, OutSize, hMem)
    End If
End if

```

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

```
RisEndSession(id)
```

Note: RTF to Image converter makes use of Windows' GdiPlus API. GdiPlus must be installed on a system to use RTF to Image Converter.



Application Interface functions

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

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

In This Chapter

[RisEndSession](#)

[RisGetImageType](#)

[RisGetLastMessage](#)

[RisGetLicnseStatus](#)

[RisGetPageImage](#)

[RisLoadBuffer](#)

[RisImageToBuffer](#)

[RisImageToFile](#)

[RisHandleToStr](#)

[RisLoadFile](#)

[RisNewSession](#)

[RisResetLastMessage](#)

[RisSetFlags](#)

[RisSetBoolProp](#)

[RisSetHdrFtrText](#)

[RisSetImageSize](#)

[RisSetNumProp](#)

[RisSetPageMargin](#)

[RisSetPaperOrient](#)

[RisSetPaperSize](#)

[RisSetTextProp](#)



RisEndSession

End a conversion session.

BOOL RisEndSession(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.



RisGetImageType

Get the image type constant for to the requested file name.

```
int RisGetImageType(id, ImageFile)
```

```
DWORD id;                // Session id
```

```
LPBYTE ImageFile;       // image file name
```

Return value: This method returns Image-type constant corresponding to the extension of the given file.

Examples:

```
' get the image type for a file name  
  
ImageType = RisGetImageType(id, "test.jpg")  
  
' set the output image type  
  
RisSetNumProp(id, RIPROP_IMAGE_TYPE, ImageType)
```



RisGetLastMessage

Get the last message.

```
int RisGetLastMessage(id, RisMessage, DebugMessage);
```

DWORD id;	Session id.
LPBYTE RisMessage;	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 HPFLAG_RETURN_MSG_ID flag. This flag is set using the RisSetFlags function.



RisGetLicnseStatus

Get the license status.

```
int RisGetLicnseStatus()
```

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 RisGetLicenseStatus function at anytime to retrieve the license status.



RisGetPageImage

Return the image for the requested page number for the currently loaded RTF document.

```
HANDLE RisGetPageImage(PageNo)
```

```
DWORD id;           Session id.
```

```
int PageNo;         // Page number. This value should be between 1 and  
                    // the PageCount for the currently loaded RTF  
                    // document.
```

Return value: This function returns a metafile handle for a metafile image, or a bitmap handle for other type of images. A null value indicates an error condition.

This function is useful if you wish to retrieve the image handle for further processing before saving to a disk file.

Examples:

```
PageCount=RisLoadFile(id, "test.rtf");
```

```
RisSetNumProp(id, RIPROP_IMAGE_TYPE, PICT_JPG)
```

```
handle=RisGetPageImage(id, 1);
```



RisLoadBuffer

Load rtf string and determine the number of pages in the rtf document..

```
int RisLoadBuffer(id, InString, InStringLen)
```

DWORD id; Session id.

LPBYTE InString; Input string containing RTF document.

int InStringLen; length of the input document string.

Return value: This function returns the number of pages in the rtf document. A value of zero indicates an error condition.

Examples:

```
Dim RtfString as string
```

```
PageCount = RisLoadBuffer(id,RtfString, Len(RtfString))
```



RisImageToBuffer

Save the current image in a memory handle.

```
HGLOBAL RisImageToFile(id, OutFile, PageNo)
```

```
DWORD id;           // Session id
```

```
LPLONG OutSize;    // (output) size of the global memory block returned by  
this function.
```

```
int PageNo;        // Page number. This value should be between 1 and  
the PageCount for the currently loaded RTF  
document.
```

Return value: This method returns TRUE when successful.

Examples:

```
PageCount = RisLoadBuffer(id, RtfString, Len(RtfString))
```

```
If (PageCount > 0) Then
```

```
    ' set the output image type to Jpeg
```

```
    call RisSetNumProp(id, RIPROP_IMAGE_TYPE, PICT_JPG)
```

```
    ' return the page image in global memory handle
```

```
    hMem = RisImageToBuffer(id, OutSize, PageNo)
```

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

```
        ' allocate space for the output string
```

```
        OutString = Space$(OutSize + 1)
```

```
        Call RisHandleToStr(OutString, OutSize, hMem)
```

```
    End If
```

```
End if
```




RisImageToFile

Save the current image to the requested file name.

```
BOOL RisImageToFile(id, OutFile, PageNo)
```

```
DWORD id;           // Session id
```

```
LPBYTE OutFile;    // Output image file name
```

```
int PageNo;        // Page number. This value should be between 1 and  
                   // the PageCount for the currently loaded RTF  
                   // document.
```

Return value: This method returns TRUE when successful.

Examples:

```
PageCount = RisLoadFile(id, "test.rtf")  
  
If (PageCount > 0) Then  
  
    PageNo = 1 'Get image for the first page of  
              'the document  
  
    result = RisImageToFile(id, "test.jpg", PageNo)  
  
End If
```



RisHandleToStr

Convert a global memory handle to a Visual Basic string.

```
BOOL RisHandleToStr(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.



RisLoadFile

Load rtf file and determine the number of pages in the rtf document..

```
int RisLoadFile(id, InString, InStringLen)
```

DWORD id; Session id.

LPBYTE InFile; Input file containing RTF document.

int InStringLen; length of the input document string.

Return value: This function returns the number of pages in the rtf document. A value of zero indicates an error condition.

Examples:

```
Dim RtfString as string
```

```
PageCount = RisLoadFile(id, "test.rtf")
```



RisNewSession

Create a new conversion session.

DWORD RisNewSession()

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

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

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



RisSetFlags

Set certain flags or retrieve the values of the flags.

DWORD RisSetFlags(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:

HPFLAG_RETURN_MSG_ID

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



RisSetBoolProp

Set a boolean property for the conversion.

BOOL RisSetBoolProp(id, prop, val)

DWORD id;

Session id.

int prop;

One of the following property type to set:

RIPROP_AUTO_WIDTH

Adjust the image width to fit the contents.

RIPROP_SHRINK_TO_FIT

Set to TRUE to eliminate the ending white spaces to shrink the image height. This property is only effective for one page rtf documents

RIPROP_DONT_UPDATE_TOC

Don't update the table-of-content.

LPBYTE val;

The text value of the selected property.

Return value: This function returns TRUE when successful.



RisSetHdrFtrText

Set header or footer text.

BOOL RisSetHdrFtrText(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.
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.

Examples:

```
RisSetHdrFtrText(id, HF_FIRST_HDR, HFTYPE_TEXT,
                 "This is first page header.");
RisSetHdrFtrText(id, HF_FIRST_FTR, HFTYPE_TEXT,
                 "This is first page footer.");
RisSetHdrFtrText(id, HF_HDR, HFTYPE_TEXT,
                 "This is regular page header.");
```



```
RisSetHdrFtrText(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
```



RisSetImageSize

Set the image size.

BOOL RisSetImageSize(id, ImageWidthTwips, ImageHeightTwips)

DWORD id; Session id.

int ImageWidthTwips; The image width in twips units (1440 twips = 1 inch).

int ImageHeightTwips; The image height in twips units (1440 twips = 1 inch).

Return Value: The function returns TRUE when successful.

Comment: This function is used to override the default image size when converting an RTF document to the PDF format. This function should be called before calling the RisLoadFile or RisLoadBuffer if you wish override the image size. The default image size is derived from the paper-size specification embedded in the rtf file.

This method works the same as the RisSetPaperSize method when called with the 'size' parameter set to 0.



RisSetNumProp

Set a numeric property for the conversion.

BOOL RisSetNumProp(id, prop, val)

DWORD id;

Session id.

int prop;

One of the following property type to set:

RIPROP_IMAGE_RES

Use this property to specify the resolution of an image. The default value is 96 dpi.

RIPROP_META_RES

Use this property to specify the resolution of a metafile image. The default value is 300

RIPROP_SIZE_PERCENT

Use this property to change the size of the output image. The default value for this property is 100. You can specify a value small than 100 to obtain a smaller image. Similarly you can specify a value greater than 100 to obtain a larger image.

RIPROP_IMAGE_TYPE

Use this property to request a particular type of image (default is Bitmap file):

PICT_BMP

Bitmap image

PICT_EMF

Enhanced metafile

PICT_TIF

Tiff image

PICT_JPG

Jpeg image

PICT_PNG

PNG image

PICT_GIF

GIF image

int val;

The numeric value of the selected property.

Return value: This function returns TRUE when successful.



RisSetPageMargin

Set the page margins for PDF output.

BOOL RisSetPageMargin(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 RTF document to the PDF format. This function should be called before calling the RisConvertFile or RisConvertBuffer if you wish override the page margin values.



RisSetPaperOrient

Set the page orientation for PDF output.

BOOL RisSetPaperOrient(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 RTF document to the PDF format. This function should be called before calling the RisConvertFile or RisConvertBuffer if you wish override the paper orientation.



RisSetPaperSize

Set the page size for PDF output.

BOOL RisSetPaperSize(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 RTF document to the PDF format. This function should be called before calling the RisConvertFile or RisConvertBuffer if you wish override the paper size.



RisSetTextProp

Set a text property for the conversion.

BOOL RisSetTextProp(id, prop, val)

DWORD id; Session id.

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

RIPROP_DOWNLOAD_DIR	Folder to store temporary rtf pictures files during conversion.
---------------------	---

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 RTF to Image Converter within an ASP page. The product includes an additional wrapper DLL called `ricc.dll` which is used to access the converter within an ASP page. Please follow the following steps:

Copy `ter25.dll` and `ris32.dll` and `ricc.dll` to the Windows system directory, or any other directory available at the run-time. Now register `ricc.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 image:

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

<%

Option Explicit

Dim sRTF

Dim obj

Dim PageCount

Dim result

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

sRTF = "{\rtf1 This is a test.}"

if len(sRTF) > 0 then

    PageCount = obj.LoadBuffer(CStr(sRTF))

    result =

        obj.ImageToFile("c:\inetpub\wwwroot\demo\test.jpg",1)

End If

Set obj = Nothing
```



```
%>

<rtf>

<head>

</head>

<body>

<p> Some text before </p>



<p> Some text after </p>

</body>

</rtf>
```

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

The method names used by the `ricc.dll` are the same as the functions mentioned in the Application Interface functions. However the 'Ris' prefix is not used by the RICC method names. For example, the `RisConvertFile` function is named as `ConvertFile` within the `ricc.dll` file.

Also, the constants values are prefixed with an 'VAL_' prefix. For example, the constant `RIPROP_IMAGE_TYPE` becomes `VAL_RIPROP_IMAGE_TYPE`.