Software License Agreement

RTF to PDF Converter for .NET Core - windows (x64)

and

RTF to PDF Converter for .NET Framework

Version 17

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 RTF 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 distribution ZIP file, and provides a step by step process of incorporating the TER routine into your application. To begin:

1. Add the reference for rpn.dll in your project.

Net Core: For the .NET core product, create a project reference for the included product package. The package name is found as rps.17.n.n.n.nupkg. The 'n.n.n' stands for the product minor release number. This is how your project file would apear:

```
<PackageReference Include="rps" Version="16.0.0.0"/>
```

Also, please ensure that:

a) the target framework for your project file is set to net6.0-windows. Example:

```
<TargetFramework>net6.0-windows</TargetFramework>
```

b) the platform target should be set to x64. Example:

```
<PlatformTarget>x64</PlatformTarget>
```

2. Add the 'using' or 'Import' namespace statement for the project dll, example:

using SubSystems.RP

or

Import SubSystems.RP

Files

.NET Core: The .NET Core includes a nuget package called rps.17.n.n.n.nupkg. The 'n.n.n' stands for the product minor release number.

.NET Framework: The zip folder contains the RPN.DLL, TESN30.DLL, and PDN17.DLL files necessary to incorporate this product into your application.

The zip folder 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.cs	Source code for the demo program
demo.exe	Executable demo program
demo.csproj	The project file to compile the demo.
AssemblyInfo.cs	Assembly information file

Visual Basic Interface and Demo Files:

Form1.vb	vb source file
dmo_vbn.vbproject	The project file for the visual basic demo program.
AssemblyInfo.vb	Assembly information file for the demo program.

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 RpsSetLicenseInfo static function. This should be preferably done before creating the Rpn object to avoid pop-up nag screens.

int RpsSetLicnseInfo(String LicenseKey, String LicenseNumber, String 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=Rpn.RpsSetLicenseInfo("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: *RpsSetLicenseInfo method should be called only once at the beginning of your application. Calling this method for each conversion would degrade the conversion performance.*

Sample Conversion Code

.NET Core: Include the rps.17.n.n.n.nupkg nuget package in your project. This is how your project file entry would appear:

<PackageReference Include="rps" Version="17.0.0.0"/>

This package is included in the distribution zip folder.

.NET Framework: Please ensure that RPN.DLL, TESN30.DLL, and PDN17.DLL files are available in the project directory. Set the reference for RPN.DLL in your project. The

TESN30.DLL and PDN17.DLL are referenced indirectly by RPN.DLL.

Now set the namespace for the product:

```
using SubSystems.RP;              // C# example
Imports SubSystems.RP           ' VB Example
```

Now set the product license key and create an RPN type object:

Rpn.RpsSetLicenseInfo("xxxxx-yyyyy-zzzzz","srabnnnnn-n","My company name")

```
Rpn rp = new Rpn()  // C# example
dim rp as Rpn        ' VB example

rp.InWebServer = true;   // set to true when hosting the
                         // converter in an ASP.NET app
```

Now use one of the following calls to convert from RTF to PDF formats:

1. Convert an RTF file to a PDF file.

```
rp.RpsConvertFile("test.rtf","test.pdf")
```

2. Convert an RTF string to a PDF string

```
Dim PdfString as string

PdfString= rp.RpsConvertBuffer(RtfString)
```

Control Methods

These methods allow you to convert from rtf to pdf format. Please set the namespace for the Rpn class before using these methods:

```
using SubSystems.RP;              // C# example
Imports SubSystems.RP           ' VB Example
```

In This Chapter
RpsConvertBuffer
RpsConvertFile

RpsConvertBuffer

Convert rtf to pdf using text string.

String RpsConvertBuffer(InString)

String InString; // Input string containing RTF formatted document.

Return value: This function returns a string containing the converted documented.

This pdf string can be written to a disk file by using the RpsWriteToFile method.

Also, you can extract the byte array from this string by using the RpsStrToBytes method.

A null return values indicates an error.

Examples:

Convert an RTF string to a PDF string

```
Dim PdfString as string

PdfString= rp.RpsConvertBuffer(RtfString)
```

RpsConvertFile

Convert rtf to pdf using disk files.

bool RpsConvertFile(InFile, OutFile)

string InFile; // Input file containing RTF document

string OutFile; // Output files, contains the converted document

Return value: This function returns TRUE when successful.

Examples:

Convert an RTF file to a PDF file.

```
rp.RpsConvertFile("test.rtf","test.pdf")
```

RpsGetLastMessage

Get the last message.

int RpsGetLastMessage(RpsMessage, DebugMessage);

string RpsMessage;	// Returns the default user message text in English
string 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 RPFLAG_RETURN_MSG_ID flag. This flag is set using the RpsSetFlags method.

RpsResetLastMessage

Reset the last editor message.

bool RpsResetLastMessage()

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

RpsGetLastMessage
RpsSetFlags

RpsSetFlags

Set certain flags or retrieve the values of the flags.

int RpsSetFlags(set, flags)

bool set;	// TRUE to set the given flags, FALSE to reset the given flags

int flags; // Flags (bits) to set or reset. Currently, the following flag values are available:

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

RpsStrToBytes

Convert a pdf string to a byte array.

byte[] RpsStrToBytes(PdfString)

String PdfString; // Input string containing PDF text.

Return value: This function returns a byte array from the given string. This is a preferred method of converting a pdf string to a byte array because it returns the raw bytes without employing an encoding method.

A null return values indicates an error.

Example:

```
Response.Clear();
Response.Charset = "";
Response.ContentType = "application/pdf";

string strFileName = "test" + ".pdf";
Response.AddHeader("Content-Disposition",
                   "inline;filename=" + strFileName);

Rpn rp = new Rpn();
rp.InWebServer=true;

String pdfString = rp.RpsConvertBuffer(RtfText);

Response.BinaryWrite(rp.RpsStrToBytes(pdfString));
```

```
Response.Flush();
Response.Close();
Response.End();
```

RpsWriteToFile

Write a pdf string to a disk file.

bool RpsWriteToFile(FileName, PdfString)

string FileName; // Output file.

string PdfString; // Pdf string to be written to the disk file

Return value: This function returns TRUE when successful.

Control Properties

The control properties can be before the conversion to affect the pdf output. The control supports the following properties:

InWebServer

This property should be set to True when this control is used in a web server. When this property is set to True, the control suppress the display of any dialog and message boxes.

Author

Set the author name for the PDF document.

Bookmark

Set to true to convert the rtf table-of-content to PDF bookmark. The default value is true.

CreDate

Set the document creation date. The date is specified in a text string.

Hyperlink

Set to true to translate rtf hyperlink fields to pdf hyperlinks. The default value is true.

Keywords

Set the keywords for the PDF document.

LicenseKey

Set the product license key for the product. *Your license key is e-mailed to you after your order is processed.*

ModDate

Set the document modification date. The date is specified in a text string.

Producer

Set the producer description for the PDF document.

Subject

Set the subject description for the PDF document.

Title

Set the title for the PDF document

CompressText

Set to true to compress the text stream in the PDF output.

PermFlags

Use this flag to specify the permissions granted when the PDF document is being viewed or manipulated. You can use one or more of the following flags using the OR operator:

pc.PERM_PRINT	Allow printing operation
pc.PERM_COPY	Allow copying operation
pc.PERM_MOD	Allow document modification

The OwnerPassword property must also be set for the PDF reader to honor the permission flags.

OwnerPassword

Optional document owner password.

When either an owner or a user password is specified, the PDF document is written out using Adobe standard encryption mechanism.

An owner password in the PDF document requires a PDF editor to prompt the user for the owner password and allow PDF modification only when the supplied owner password matches the encrypted owner password found in the file.

UserPassword

Optional user password

When either an owner or a user password is specified, the PDF document is written out using Adobe standard encryption mechanism.

A user password in the PDF document requires a PDF viewer to prompt the user for the user password and allow PDF display only when the supplied user password matches the encrypted user password (or owner password) found in the file.

EmbedFonts

Normally the converter only embeds non-standard fonts in the PDF file. This flag would instruct the converter to embed all fonts.

ExactTextPlacement

Set to True to instruct the converter to emit character width for every character thus providing for precise text layout. This option is useful when converting a file containing small fonts.

RC4_128

Set to true to enable RC4 128 bit security when a password is specified.

AES_128

Set to true to enable AES 128 bit security when a password is specified.

PdfA

Set to true to generate PDF-A compliant document.

PdfA1b

Set to true to generate PDF-A/1b compliant document

PdfUA

Set to true to generate PDF-UA (User Accessibility) compliant document

ShrinkImagesToFit

Shrink the RTF images to fit on the page or the table cell.

NoTocUpdate

Do not update original table-of-content before generating pdf.

PropPictQuality

Specify picture quality from 1 to 5, where 1=lowest, 5=highest, 3=default

UseOrigJpg

Insert the original jpeg image data into the pdf file.