Software License Agreement

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

and

RTF to DOCX Converter for .NET Framework

version 14

2012-2024

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 RTF to DOCX 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 RTF to DOCX converter 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 rds.14.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="rds" Version="14.0.0.0"/>
```

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.RD

or

Import SubSystems.RD

▼

Files

NET Core: The .NET Core includes a nuget package called rds.14.n.n.n.nupkg. The 'n.n.n' stands for the product minor release number.

.NET Framework: The zip folder contains the RDN.DLL, tesn31.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 RdsSetLicenseInfo static function. This should be preferably done before creating the Rdn object to avoid pop-up nag screens.

int RdsSetLicnseInfo(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=Rdn.RdsSetLicenseInfo("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: *RdsSetLicenseInfo 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 rds.14.n.n.n.nupkg nuget package in your project. This is how

your project file entry would appear:

<PackageReference Include="rds" Version="14.0.0.0"/>

This package is included in the distribution zip folder.

.NET Framework: Please ensure that RDN.DLL and tesn31.DLL files are available in the project directory. Set the reference for RDN.DLL in your project. The tesn31.DLL is referenced indirectly by RDN.DLL.

Now set the namespace for the product:

```
  using SubSystems.RD;          // C# example

  Imports SubSystems.RD         ' VB Example
```

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

```
Rdn.RdsSetLicenseInfo("xxxxx-yyyyy-zzzzz","srabnnnnn-n","My company name")

  Rdn rd = new Rdn()  // C# example

  dim rd as Rdn        ' VB example


  rd.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 DOCX formats:

1. Convert an RTF file to a DOCX file.

```
  rd.RdsConvertFile("test.rtf","test.docx")
```

2. Convert an RTF string to a DOCX string

```
  byte[] DocxData


  DocxData= rd.RdsConvertBuffer(RtfString)
  rd.RdsWriteToFile("test.docx",DocxData)
```

Control Methods

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

using SubSystems.RD;   // C# example

Imports SubSystems.RD   ' VB Example

RdsConvertBuffer

Convert rtf to docx using text string.

byte[] RdsConvertBuffer(InString)

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

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

This docx string can be written to a disk file by using the RdsWriteToFile method.

A null return values indicates an error.

Examples:

Convert an RTF string to a DOCX byte array:

byte[] DocxData

DocxData = rp.RdsConvertBuffer(RtfString)

RdsConvertFile

Convert rtf to docx using disk files.

bool RdsConvertFile(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 DOCX file.

 rp.RdsConvertFile("test.rtf","test.docx")

RdsGetLastMessage

Get the last message.

int RdsGetLastMessage(RdsMessage, DebugMessage);

string RdsMessage;   // 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 RdsSetFlags method.

RdsResetLastMessage

Reset the last editor message.

bool RdsResetLastMessage()

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.

RdsSetFlags

Set certain flags or retrieve the values of the flags.

int RdsSetFlags(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:

    RDFLAG_RETURN_MSG_ID   Do not display the error messages. Save the error code to be later retrieved using the RdsGetLastMessage function.

    RDFLAG_EMBED_PICTURE   Embed any 'linked' picture data into the generated docx file.

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.

RdsWriteToFile

Write a docx byte array to a disk file.

bool RdsWriteToFile(FileName, DocxData)

string FileName;   // Output file.

byte[] DocxString;   // A byte array containing the docx data to write to the output file.

Return value: This function returns TRUE when successful.

Control Properties

The control properties can be before the conversion to affect the docx 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.