Software License Agreement

DOCX to RTF Converter

For .NET

Vesion 15

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 DOCX to 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 DOCX to RTF Converter into your application.

Files

The package contains the DRN.DLL and tesn31.dll files necessary to incorporate this product into your application.

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

int DrsSetLicnseInfo(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=Drn.DrsSetLicenseInfo("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: *DrsSetLicenseInfo 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

Please ensure that DRN.DLL and tesn31.dll files are available in the project directory. Set the reference for DRN.DLL in your project. The tesn31.dll is referenced indirectly by DRN.DLL.

Now set namespace for the product:

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

 Imports SubSystems.HR            ' VB Example
```

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

DRN.DrsSetLicenseInfo("xxxxx-yyyyy-zzzzz","srabnnnnn-n","my company name")

```
   DRN dr = new DRN()  // C# example
   dim dr as DRN          ' VB example


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

Now use one of the following calls to convert from the DOCX to the RTF format:

1. Convert an DOCX file to an RTF file.

```
   dr.DrsConvertFile("test.docx","test.rtf")
```

2. Convert an DOCX string to an RTF string

```
   Dim InData() As Byte
   Dim OutData As String


   InData = dr.DrsFileToBytes(InFile)
```

```
        If (Not IsNothing(InData)) Then
          If (InData.Length > 0) Then
              OutData = dr.DrsConvertBuffer(InData)
              If (OutData <> Nothing) Then
                ' write the memory to the output file
                result = WriteToFile(OutFile, OutData)
              End If
          End If
        End If
```

Control Methods

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

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

Imports SubSystems.DR           ' VB Example
```

DrsConvertBuffer

Convert DOCS to rtf using text string.

String DrsConvertBuffer(InString, XlateType)

byte[] InString;   // Input byte array containing DOCX document.

Return value: This function returns a string containing the converted documented.
A null return values indicates an error.
Examples:

 Convert an DOCX byte array to an RTF string

```
   Dim InData() As Byte
   Dim OutData As String


   InData = dr.DrsFileToBytes(InFile)


   If (Not IsNothing(InData)) Then
     If (InData.Length > 0) Then
        OutData = dr.DrsConvertBuffer(InData)
        If (OutData <> Nothing) Then
          ' write the memory to the output file
          result = WriteToFile(OutFile, OutData)
        End If
     End If
   End If
```


▲ ▼

DrsConvertFile

Convert docx to rtf using disk files.
bool DrsConvertFile(InFile, OutFile, XlateType)

string InFile;   // Input file containing a DOCX document

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

Return value: This function returns TRUE when successful.

1. Convert an DOCX file to an RTF file.

```
dr.DrsConvertFile("test.docx","test.rtf")
```

▲ ▼

DrsGetLastMessage

Get the last message.

int DrsGetLastMessage(DrsMessage, DebugMessage);

string DrsMessage;   // 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 DRFLAG_RETURN_MSG_ID flag. This flag is set using the DrsSetFlags method.

▲ ▼

DrsFileToBytes

Read a docx file into a byte array.

byte[] DrsFileToBytes(DocxFile)

String DocxFile;   // Input docx file name.

Return value: This function returns a byte array from the given file. This method can be used to obtain the byte array to supply to the DpsConvertBuffer method.

A null return values indicates an error.

Example:

```
InData = dr.DrsFileToBytes(InFile)
```

```
        OutData = dr.DrsConvertBuffer(InData)
```

DrsResetLastMessage

Reset the last editor message.

bool DrsResetLastMessage()

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
DrsGetLastMessage
DrsSetFlags

DrsSetFlags

Set certain flags or retrieve the values of the flags.

int DrsSetFlags(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:

    DRFLAG_RETURN_MSG_ID   Do not display the error messages. Save
                                                     the error code to be later retrieved using
                                                     the DrsGetLastMessage function.

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

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.

DrsSetPageMargin

Set the page margins for RTF output.

bool DrsSetPageMargin(left, right, top, bottom)

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 DOCX document to the RTF format. This function should be called before calling the DrsConvertFile or DrsConvertBuffer methods if you wish override the page margin values.

DrsSetPaperOrient

Set the page orientation for RTF output.

bool DrsSetPaperOrient(id, IsPortrait)

bool IsPortrait	Set to true to set to portrait orientation. Otherwise set to false.

Return Value: The function returns TRUE when successful.

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

DrsSetPaperSize

Set the page size for RTF output.

bool DrsSetPaperSize(kind, PageWidth, PageHeight)

PaperKind kind;	Use one of the PaperKind enumerations defined by .NET
int PageWidth;	The page width in twips units (1440 twips = 1 inch). This argument is used only if kind is set to PaperKind.Custom.
int PageHeight;	The page height in twips units (1440 twips = 1 inch). This argument is used only if kind is set to PaperKind.Custom.

Return Value: This method returns TRUE when successful.

Comment: This method is used to override the default letter size paper when converting an DOCX document to the RTF format. This method should be called before calling the DrsConvertFile or DrsConvertBuffer methods if you wish override the paper size.

Control Properties

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.

ProjectFolder

Set this property to the folder containing your project, such as c:\inetpub\wwwroot\MyProject. This information helps the converter locate the images which use relative path. It is also used for creating any temporary files.