



# Software License Agreement

**DOC to RTF Converter**

**For Win32/Win64**

Vesion 16

2009-2026

*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 with these conditions: the target application is not a stand-alone DOC 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. 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](mailto: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 DOC to RTF 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
WRS.H	The <i>include</i> file to include into a C/C++ application module that calls the DRS routine. It contains the constant definitions and the prototypes for the API functions.
wrs16.dll	The DLL file
wrs16.LIB	Import library for the wrs16. DLL
wrcc.dll	Wrapper DLL to used with an ASP page

#### Visual Basic Interface and Demo Files:

WRS.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 is e-mailed to you after your order is processed.* You would set the license key using the WrsSetLicenseKey static function. This should be preferably done before creating the conversion session to avoid pop-up nag screens.

```
WrsSetLicenseKey("xxxxx-yyyyy-zzzzz")
```

Replace the 'xxxxx-yyyyy-zzzzz' by your license key.

You do not need to call this function when using the product in the evaluation mode.



## Incorporating the DLL into Your Application

A C/C++ application should include the wrs.h file into the application module that needs to call the wrs14.dll. It also should include the wrs16.LIB as the linker library. Please refer to the demo application for an example.

A Visual Basic application needs to include the wrs.bas file in the project. Please refer to the DMO\_VB project for an example.

Please also make sure that the wrs16.dll file is 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:

```
WrsSetLicenseKey("xxxxxx-yyyyy-zzzzz")
```

```
id = WrsNewSession()
```

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

Here are sample code examples to convert from DOC to RTF format.

### 1. Convert an DOC file to an RTF file.

```
WrsConvertFile(id,"test.doc","test.rtf")
```

### 2. Convert an DOC string to an RTF string

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

```

hMem = WrsConvertBuffer(id, DocString, Len(DocString),
                        OutSize)

If (hMem <> 0) Then
    RtfString = Space$(OutSize + 1) ' allocate space for
                                   ' the output string
    Call WrsHandleToStr(RtfString, OutSize, hMem)
End If

```

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

```
WrsEndSession(id)
```



## Application Interface functions

These API functions allow you to convert between DOC and RTF formats. Your application must include the wrs.h file (c/c++), or wrs.bas (VB) files. These files declare these functions.

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

### In This Chapter

[WrsConvertBuffer](#)

[WrsConvertFile](#)

[WrsConvertFileToBuffer](#)

[WrsEndSession](#)

[WrsGetLastMessage](#)

[WrsHandleToStr](#)

[WrsNewSession](#)

[WrsResetLastMessage](#)

[WrsSetFlags](#)



## WrsConvertBuffer

### Convert DOC to RTF using memory buffers.

HGLOBAL WrsConvertBuffer(id, InString, InStringLen, OutStringLen)

DWORD id;	Session id.
LPBYTE InString;	Input string containing DOC document.
int InStringLen;	length of the input document string.
LPINT OutStringLen;	The variable to receive the length of the converted document.

**Return value:** This function returns a global memory handle containing the converted documented. You can either use the WrsHandleToStr 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 DOC string to an RTF string

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

hMem = WrsConvertBuffer(id, DocString, Len(DocString),
                        OutSize)

If (hMem <> 0) Then
    RtfString = Space$(OutSize + 1) ' allocate space for
                                   ' the output string
    Call WrsHandleToStr(RtfString, OutSize, hMem)
End If
```



## WrsConvertFile

Convert DOC to RTF using disk files.

BOOL WrsConvertFile(id, InFile, OutFile)

DWORD id;                      Session id.

LPBYTE InFile;                Input file containing DOC document

LPBYTE OutFile;               Output files, contains the converted RTF document

**Return value:** This function returns TRUE when successful.

**Examples:**

```
id = WrsNewSession()
```

```
WrsConvertFile(id, "test.doc", "test.rtf")
```



## WrsConvertFileToBuffer

**Convert the DOC file to RTF and return the RTF in a memory buffer.**

HGLOBAL WrsConvertBufferToBuffer(id, InFile, OutStringLen)

DWORD id;                      Session id.

LPBYTE InFile;                Input file name containing the doc data.

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

**Return value:** This function returns a global memory handle containing the converted documented. You can either use the WrsHandleToStr 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 DOC file to an RTF string

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

```

hMem = WrsConvertFileToBuffer(id, DocFile, OutSize)
If (hMem <> 0) Then
    RtfString = Space$(OutSize + 1) ' allocate space for
                                     ' the output string
    Call WrsHandleToStr(RtfString, OutSize, hMem)
End If

```



## WrsEndSession

**End a conversion session.**

BOOL WrsEndSession(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.



## WrsGetLastMessage

**Get the last message.**

int WrsGetLastMessage(id, WrsMessage, DebugMessage);

DWORD id;                                  Session id.

LPBYTE WrsMessage;                      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



WRFLAG\_RETURN\_MSG\_ID flag. This flag is set using the WrsSetFlags function.



## WrsHandleToStr

**Convert a global memory handle to a Visual Basic string.**

BOOL WrsHandleToStr(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.



## WrsNewSession

**Create a new conversion session.**

DWORD WrsNewSession()

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

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

A new session should be created for each conversion.

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



## WrsResetLastMessage

**Reset the last editor message.**

BOOL WrsResetLastMessage(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.

### See Also

[WrsGetLastMessage](#)

[WrsSetFlags](#)



## WrsSetFlags

**Set certain flags or retrieve the values of the flags.**

DWORD WrsSetFlags(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:

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



## ASP Interface

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

Copy wrs16.dll and wrcc.dll to the Windows system directory, or any other directory available at the run-time. Now register wrcc.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 DOC string into an RTF string:

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

Dim DocString
Dim sRtf
Dim obj

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

call obj.SetFlags(1,obj.VAL_WRFLAG_RETURN_MSG_ID)' quiet mode

DocString=""
sRtf=""

DocString =
obj.FileToString("c:\inetpub\wwwroot\DmoWrc\test.doc") 'read
' the test.docx file into a string

if len(DocString) > 0 then
    sRtf = obj.ConvertBuffer(CStr(DocString))
End If

Set obj = Nothing

%>

<html>
<head>
</head>

<body>
```

<p> Generated RTF code:</p>

<%

=sRtf

%>

<p> End of generated RTF code </p>

</body>

</html>

---

IE displays the following text

Generated RTF Code:

{\rtf.....}.

End of generated RTF code

---

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

Also, the constants values are prefixed with an 'VAL\_' prefix. For example, the constant WRFLAG\_SEGMENT\_ONLY becomes VAL\_WRFLAG\_SEGMENT\_ONLY.