Contents

Overview

API Reference, List of Functions

Registering STRING.DLL

You can register using CIS; type "go swreg" and enter the ID 3005. The cost for a single place license is US$12.00; If you'd like a license for several runtime copies, please contact me either via:

- **CIS mail: 100342,412 or**
- **Internet: biinside@bgb.ch**

You may also register by mail. Send a letter with US$30.00 enclosed; NO CHECKS ACCEPTED! to:

INside, att. STRING

P.O. Box 965

3000 Bern 9

Switzerland

Please allow 2 weeks for mail delivery and at least 2 days for CIS registration.

Legal Stuff

The program, its data, documentation and other items and services (hereafter referred to as "products") provided with STRING.DLL are provided on an "as is" basis. INside Corp. AND ITS LICENSOR MAKE NO WARRANTIES, EXPRESS OR IMPLIED, INCLUDING WITHOUT LIMITATION THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE, REGARDING THIS PRODUCT FOR ANY PURPOSE. THE EXCLUSION OF IMPLIED WARRANTIES IS NOT PERMITTED IN SOME JURISDICTIONS. THE ABOVE EXCLUSION MAY NOT APPLY TO YOU.

IN NO EVENT WILL INside Corp., ITS LICENSOR OR THEIR OFFICERS, EMPLOYEES OR AGENTS (COLLECTIVELY, INside Corp. AND ITS LICENSOR) BE LIABLE TO YOU FOR ANY CONSEQUENTIAL, INCIDENTIAL, OR INDIRECT DAMAGES ARISING OUT OF OR IN CONNECTION WITH THE USE OF THIS PROGRAM, EVEN IF INside Corp. OR ITS LICENSOR HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. BECAUSE SOME JURISDICTIONS DO NOT ALLOW THE EXCLUSION OR LIMITATION OF LIABILITY FOR CONSEQUENTIAL DAMAGES, THE ABOVE LIMITATIONS MAY NOT APPLY TO YOU.

THE LIABILITY OF INside Corp. OR ITS LICENSOR TO YOU FOR ACTUAL DAMAGES FROM ANY CAUSE WHATSOEVER, AND REGARDLESS OF THE FORM OF THE ACTION (WHETHER IN CONTRACT, TORT (INCLUDING NEGLIGENCE), PRODUCT LIABILITY OR OTHERWISE), WILL BE LIMITED TO US$50.

IN ANY CASE OF LAWSUIT, COMPETENCY OF COURT IS BERN, SWITZERLAND

Release History

Version / Release	Changes
1.0 / A	Jan 94: DOB; Initially, only the ReverseStr function was needed.
1.0 / B	Feb 94: The FindRightChar function was implemented.
1.1 / A	Jun 94: All the other functions followed: FindFirstNIS, FindNOO, FindFOS and LexSortStr.
1.2 / A	Jul 94: Implemetation of CountWords function.
1.2 / B	Jul 94: Patched CountWords function

STRINGVer (hWnd)

Parameters:

[**hWnd**] = Handle of parent window to display the message box.

Description:

Shows a message box on top of the specified parent window containing information about your license of STRING.DLL version and release number.

Sample Code:

```
'Place this code in the general declarations section of a form:

Declare Function STRINGVer Lib "STRING.DLL" (ByVal hwnd As Integer) As Integer

'Place this code in a Button on a form:

Sub Command1_Click ()

Dim res As Integer

res = STRINGVer(form1.hWnd)

End Sub
```

ReverseStr (lpstrDATA)

Parameters:

[**lpstrDATA**] = Fixed length string variable defined in Visual Basic with a size of exactly 65000 characters. It's not really 64K (65535 Characters) because VB needs some space to allocate the structure.

Description:

This function reverses 65000 Characters and returns them in the same string. A string reading "ABCDE" would return reading "EDCBA".

Sample code:

```
'Place this code in a basic module:

Global lpstrDATA as String * 65000

Declare Function ReverseStr Lib "STRING.DLL" (ByVal lpstrDATA As String) As Integer

'Place this code in a Button on a form:

Sub Command1_Click

Dim ret as Integer

lpstrDATA = Space$(65000) 'Allocates Memory

lpstrDATA = "Hello World"

ret = ReverseStr(lpstrDATA)

MsgBox "Hello World has become:'" & Trim$(lpstrDATA) & "'"", 64, "ReverseStr Example"

End Sub
```

FindRightChar (lpstrDATA, iChar)

Parameters:

[**lpstrDATA**] = Fixed length string variable defined in Visual Basic with a size of exactly 65000 characters. It's not really 64K (65535 Characters) because VB needs some space to allocate the structure.

[**iChar**] = Integer value of the character to be located.

Description:

Finds the first occurence of iChar within lpstrDATA. As opposed to Visual Basics Instr function, this function returns the position of the first Character from the right (still **counting from the left**, though!).

Sample code:

```
'Place this code in a basic module:

Global lpstrDATA as String * 65000

Declare Function FindRightChar Lib "STRING.DLL" (ByVal lpstrDATA As String, ByVal iChar As Integer) As Integer

'Place this code in a Button on a form:

Sub Command1_Click

Dim ret, iChar as Integer

lpstrDATA = Space$(65000) 'Allocates Memory

lpstrDATA = "Hello World"

iChar = Asc("W")

ret = FindRightChar(lpstrDATA, iChar)

MsgBox "'W' lies at position "+str$(ret), 64, "FindRightChar   Example"

End Sub
```

FindFirstNIS (lpstrDATA, lpstrSUBSET)

Parameters:

[**lpstrDATA**] = Fixed length string variable defined in Visual Basic with a size of exactly 65000 characters. It's not really 64K (65535 Characters) because VB needs some space to allocate the structure.

[**lpstrSUBSET**] = Fixed length string variable defined in Visual Basic with a size of exactly 256 characters.

Description:

Finds the first character within lpstrDATA not belonging to the character subset specified in lpstrSUBSET and returns the index where to find it. FindFirstNIS means: FindFirst[N]ot[I]n[S]ubset

Sample code:

```
'Place this code in a basic module:

Global lpstrDATA as String * 65000

Declare Function FindFirstNIS Lib "STRING.DLL" (ByVal lpstrDATA As String, ByVal lpstrSUBSET As String) As Integer

'Place this code in a Button on a form:

Sub Command1_Click

Dim ret as Integer

Dim lpstrSUBSET as String * 256

lpstrDATA = Space$(65000) 'Allocates Memory

lpstrSUBSET = Space$(256)

lpstrDATA = "              Hello World"

lpstrSUBSET = " "

ret = FindFirstNIS(lpstrDATA, lpstrSUBSET)

MsgBox "The first character not belonging to the subset is: "+Mid$(lpstrDATA,ret+1,1), 64, "FindFirstNIS Example"

End Sub
```

LexSortStr (lpstrSTRING1, lpstrSTRING2, iNofC, iCase)

Parameters:

[**lpstrSTRING1**] = Fixed length string variable defined in Visual Basic with a size of exactly **1024** characters.

[**lpstrSTRING2**] = Fixed length string variable defined in Visual Basic with a size of exactly **1024** characters.

[**iNofC**] = Integer number describing how many characters within the strings are supposed to be included in the comparison. ([N]umber [o]f [C]haracters).

[**iCase**] = Integer number describing whether the strings are supposed to be compared case sensitive or not: **0 = Not case sensitive; 1 = Case sensitive**

Description:

Compares two strings and returns a value telling you the result of the comparison. The values are to be interpreted as follows:

ret = 0 : The two strings are to be ordered the SAME

ret > 0 : lpstrSTRING1 is ordered AFTER lpstrSTRING2

ret < 0 : lpstrSTRING1 is ordered BEFORE lpstrSTRING2

Sample code:

```
'Place this code in the general declarations section of a form:

Declare Function LexSortStr Lib "STRING.DLL" (ByVal lpstrSTRING1 As String, ByVal lpstrSTRING2 As String, ByVal iNofC As Integer, ByVal iCase As Integer) As Integer

'Place this code in a Button on a form:

Sub Command1_Click

Dim ret As Integer

Dim iCase As Integer 'Case sensitivity

Dim iNofC As Integer 'Number of Characters to check

Dim lpstrSTRING1 As String * 1024

Dim lpstrSTRING2 As String * 1024

'allocate memory:

lpstrSTRING1 = Space$(1024)

lpstrSTRING1 = Space$(1024)

lpstrSTRING1 = InputBox("Enter the first string to compare:")

lpstrSTRING2 = InputBox("Enter the second string to compare:")

iCase = 1 '1=case sensitive comparison, 0=non case sensitive

iNofC = 10 'compare the first 10 characters of the string

ret = LexSortStr(lpstrSTRING1, lpstrSTRING2, iNofC, iCase)

Select Case ret
```

```
Case Is < 0
   MsgBox "'" + Trim$(lpstrSTRING1) + "' BEFORE '" + Trim$(lpstrSTRING2) + "'", 64, "LexSortStr Example"
Case 0
   MsgBox "'" + Trim$(lpstrSTRING1) + "' THE SAME as '" + Trim$(lpstrSTRING2) + "'", 64, "LexSortStr Example"
Case Is > 0
   MsgBox "'" + Trim$(lpstrSTRING1) + "' AFTER '" + Trim$(lpstrSTRING2) + "'", 64, "LexSortStr Example"
End Select
End Sub
```

FindNOO (lpstrDATA, cCHARA)

Parameters:

[**lpstrDATA**] = Fixed length string variable defined in Visual Basic with a size of exactly 65000 characters. It's not really 64K (65535 Characters) because VB needs some space to allocate the structure.

[**cCHARA**] = A single character being searched for.

Description:

Finds the number of occurences of character cCHARA within lpstrDATA. FindNOO means: Find[N]umber[O]f[O]ccurences

Limitations:

- A double character will only be counted once. i.e. if you search for "b" in "Yabba" only ONE occurence will be counted!
- If the character being searched for appears at the beginning of the string, it will not be counted (heaven knows why! It sounds simple but... it will be fixed in a later release). i.e. if you search for "Y" in "Yabba" it won't return an occurence.

Sample code:

```
'Place this code in a basic module:

Global lpstrDATA as String * 65000

Declare Function FindNOO Lib "STRING.DLL" (ByVal lpstrDATA As String, ByVal cCHARA As String) As Integer

'Place this code in a Button on a form:

Sub Command1_Click

Dim ret As Integer

Dim cCHARA As String * 1

lpstrDATA = Space$(65000)

cCHARA = InputBox("Enter character to count:")

lpstrDATA = "abcdefghiklmnopqrstuvwxyzabcdefghiklmnopqrstuvwxyz"

ret = FindNOO(lpstrDATA, cCHARA)

MsgBox cCHARA + " was found " + Str$(ret) + " times", 64, "FindNOO Example"

End Sub
```

FindFOS (lpstrDATA, lpstrSUBSET)

Parameters:

[**lpstrDATA**] = Fixed length string variable defined in Visual Basic with a size of exactly 65000 characters. It's not really 64K (65535 Characters) because VB needs some space to allocate the structure.

[**lpstrSUBSET**] = Fixed length string variable defined in Visual Basic with a size of exactly 256 characters.

Description:

Finds the first occurence of any character contained in the string subset lpstrSUBSET within lpstrDATA and returns this characters position. FindFOS means: Find[F]irst[O]f[S]ubset

Limitations:

- lpstrDATA (the string being searched) may not contain any blanks. If it does contain blanks, the position will be wrong.

Sample code:

```
'Place this code in a basic module:

Global lpstrDATA as String * 65000

Declare Function FindFOS Lib "STRING.DLL" (ByVal lpstrDATA As String, ByVal lpstrSUBSET As String) As Integer

'Place this code in a Button on a form:

Sub Command1_Click

Dim ret As Integer

Dim lpstrSUBSET As String * 256

lpstrDATA = Space$(65000)

lpstrSUBSET = Space$(256)

lpstrSUBSET = InputBox("Enter subset to search for:")

lpstrDATA = "Hello, this is a test!"

ret = FindFOS(lpstrDATA, lpstrSUBSET)

MsgBox "The first character from '" + Trim$(lpstrSUBSET) + "' was found at position " + Str$(ret), 64, "FindFOS Example"

End Sub
```

CountWords (lpstrDATA)

Parameters:

[**lpstrDATA**] = Fixed length string variable defined in Visual Basic with a size of exactly 65000 characters. It's not really 64K (65535 Characters) because VB needs some space to allocate the structure.

Description:

Counts the words contained within lpstrDATA.

Sample code:

```
'Place this code in a basic module:

Global lpstrDATA as String * 65000

Declare Function CountWords Lib "STRING.DLL" (ByVal lpstrDATA As String) As Long

'Place this code in a Button on a form:

Sub Command1_Click

Dim ret As Long

lpstrDATA = Space$(65000)

lpstrDATA = "Hello, this is a test!"

ret = CountWords(lpstrDATA)

MsgBox "The number of words is: '" + Trim$(Str$(ret)), 64, "CountWords Example"

End Sub
```