

# StrPlus.DLL Help

Copyright © 1994 by Digital PowerTOOLS  
*Another Shareware Product from Digital PowerTOOLS*

## Overview of StrPlus.DLL

StrPlus.DLL Functions:

[CountWords](#)

[GetStateOfKey](#)

[GetStrPlusVersion](#)

[JustifyLine](#)

[Number Conversion Functions](#)

[Path String Functions](#)

[String Conversion Functions](#)

[SuperPrint](#)

[Token Functions](#)

[WordColor](#)

[Copyright Notice](#)

[Order Form and Software Evaluation Form](#)

[Shareware Concept](#)

[Shareware License](#)

[Shareware Registration](#)

[Custom DLL's](#)

[Registered License](#)

# Overview of StrPlus.DLL

Microsoft's VisualBasic is a powerful programming language for the Windows operating system. One of its most appealing features is its extensibility. You can add subroutines and functions directly within the language. You can use most of the 800+ Windows API functions. And, you can use virtually any Dynamic Link Library (DLL) or VBX (a specialized type of DLL, Visual Basic eXtension). The author, with 12 years of programming experience, prefers VB to Assembly, C, and C++.

Visual Basic comes with numerous, useful string functions. However, StrPlus.DLL re-creates several string commands that are included with other languages, but are missing from VB. For example, the new GetToken command parses substrings. This lets you easily count words, sort comma-delimited strings for database conversions, parse the Command\$ parameters, or create a command interpreter for your own applications. Have you ever wanted the facility of the old BASIC Data/Read commands? Many folks do. We've included a VB subroutine, ReadData, that uses GetToken to allow you to have that facility in Visual Basic. In fact, it's superior to the old Read command because it actually counts the data elements for you.

Also included are commands such as ToName (instantly converts the first letter of every word in a string to uppercase), ReverseString (reverses a string so that you can use INSTR\$ in the opposite direction), Encrypt and Decrypt (which encode and decode strings), and CountWords (a C routine written in Assembly which quickly counts the words in a document). Also included are number conversion commands such as GetOrdinalExt (which returns the Ordinal extension of a number "st" for "1", "nd" for "2", and so on), GetRomanNumber (which returns the Roman number for a given integer), and GetWordNumber (which converts a number, usually currency, to the word equivalent; for example, "1,234.56" returns "One Thousand Two Hundred Thirty-Four and 56/100" -- this is great for printing checks).

Do you find that you have to refer to color charts to find the values for colors? StrPlus.DLL includes a new command, WordColor, which can be used anywhere that you'd use RGB or QBColor. However, WordColor accepts strings instead of hard-to-remember numbers. For example, WordColor("blue") is the same as QBColor(1). All 16 basic colors and 20 system colors ("BtnHighlight", "BtnShadow", etc.) are supported.

StrPlus.DLL also comes with two string output commands. SuperPrint displays text rotated, in 3D, and with programmer-selected alignment. JustifyLine displays lines of text justified evenly between two margins. The new GetStateOfKey command returns the state of certain system keys (for those who don't have VB Pro) such as LeftShift, RightShift, NumLock, CapsLock, etc.

The demo program (written in VB 3.0 standard) demonstrates nearly all of the new features of StrPlus and reveals a few VB programming tricks too. And, since StrPlus.DLL comes with a complete Help file and a practical demonstration project in VB, you'll quickly understand how to use the new functions so that you can easily incorporate these new techniques into your own projects.

# Shareware Concept

Commercial software is copyrighted (protected by law) and is typically sold through computer and department stores and through discount mail order houses. The quality of most commercial software is usually very good to excellent. Public domain software is not copyrighted and is typically found on computer bulletin boards. The quality of public domain software ranges from poor to very good. Shareware has similarities with both commercial and public domain software. The quality is usually good to excellent (frequently surpassing commercial software in price versus features); and shareware can be obtained from computer bulletin boards, on disks in the back of some computer books, computer stores, on racks in some book stores, and from mail order houses.

Shareware is COPYRIGHTED (protected by law). However, the author typically authorizes distribution of the unregistered software to others in the hopes that the users will like the program enough to send in a registration fee which removes a 'Need To Register' message (when the registration number is sent by return mail or an upgrade is provided) and offers notification of program updates and other benefits. And ... sending the registration fee encourages the author to write MORE software.

Many shareware authors are professional programmers who write these programs in their spare time. The fundamental concept of shareware is TRY, BEFORE YOU BUY. And, not only do you get to try the program before you buy; but, you also will typically get professional quality software at well below the market price of comparable commercial packages when you do register.

The FUTURE of shareware depends on YOU. If you continue using a shareware package, you should register it.

You have 30 days to evaluate StrPlus.DLL once you install it on your system. After that, you should either register it or remove it from your system.

## **Custom DLL's**

Digital PowerTOOLS is dedicated to helping VB and C programmers. Moreover, we will consider custom DLL and/or VBX projects which have potential mass market appeal. And, of course, we certainly consider enhancements to our released products. Write to:

Digital PowerTOOLS  
P.O. Box 97794  
Jackson, MS 39288-7794

# Copyright Notice

## **StrPlus.DLL, Copyright © 1994 by Digital PowerTOOLS**

StrPlus.DLL is distributed under the shareware concept. Shareware Registration is mandatory for continued use of this product.

And, as shareware, you may evaluate StrPlus.DLL for a period of no more than 30 days. After this time you must either register it or remove it from your system. Failure to comply with this condition is a violation of United States and international copyright law.

This program is fully functional; however, you will receive the latest version upon registering (with the startup registration reminder removed). Please register if you use this program for more than the 30-day trial period. See Shareware Registration for more specific details on registering this product.

# Shareware License



## Shareware Distribution

You have the right to freely distribute the unregistered version of this software which includes uploading to BBS's provided that: (1) no files in the original package are altered in any way and that the software is not sold for profit -- cost of duplication and shipping may, of course, be requested in distribution to end users, (2) all the files must be distributed together, and (3) individual copies of this software may not be sold for more than \$10.00.

You must stop selling/distributing copies of this version of this software upon written notice from the author or Digital PowerTOOLS. The terms of this distribution license are subject to change with the release of new versions of this software. This software may not be represented as anything other than shareware and the shareware concept must be explained in any advertisement or catalog that quotes a price and on any packaging used to display the disk. The explanation of the shareware concept must make it clear that with shareware an additional registration fee is required if users continue to utilize the software.



## Shareware Registration

You have the right to use this unregistered software for a period not to exceed 30 days from installation. After this trial period, you should either register it or remove it from your system. Failure to comply with this software license is a violation of U.S. Copyright laws. See Software Registration for more details.



## Site Licenses

Write to receive the latest details on site licenses.



## Rack Sales & CD Distribution

Write to request the latest details on rack sales and other For-Profit distribution. You may not include this shareware on a CD-ROM or other type of package produced by any user group, commercial library, or any for-profit or nonprofit organization; and you may not distribute same with any other product or as an incentive to purchase any other product without the express written permission of Digital PowerTOOLS. This software may not be sold on retail racks or CD-ROM disks, by either ASP or non-ASP vendors, without specific written permission from the Digital PowerTOOLS.

# Shareware Registration

This program is Shareware. You may evaluate it for a period of no more than 30 days. After this time you should either register it or remove it from your system. Failure to comply with this condition is a violation of United States and international copyright law. If you do find that this program is applicable to your computer usage, please send in your registration fee.

Take it for a spin around the block. Kick the tires. Look under the hood. If you like it and use it, register the application. If it doesn't suit your needs, take it off your system. If you do enjoy the features that 3DfxPlus.DLL gives you, try other products from Digital PowerTOOLS too.

(notice the subliminal messages) This program is provided as Shareware and may be freely distributed (according to the terms under the Shareware License). If you find the program useful (register today), we ask that you send in the nominal registration fee of \$10.00 to the address below. Many hours of work went into this project. And, this registration fee includes FREE shipping of the latest version on disk (specify your size preference).

Registering this product will also encourage us to write more software under the Shareware concept (register today) for the PC. If you like it, register it (register today). If you don't, don't -- and remove it from your system. Either way, though, we'd like hear your suggestions for improvement (register today) -- send in the Software Evaluation form (EvalFrm.Wri). And ... registered users will have any questions answered (register today). The future of Shareware depends on YOU ... register today.

Be sure to include the PROGRAM TITLE and VERSION NUMBER that you have. Also specify the disk format that you prefer (3.5-HD or 5.25-HD) of the latest version. Be sure to include your name, company, and mailing address. You may also use our order form (OrderFrm.Wri).

Thank you (yadot retsiger). You may also contact the author with private mail through America Online, ID# Vernon486. Mail check, money order, cash and/or suggestions to (Make checks payable to Digital PowerTOOLS):

Digital PowerTOOLS P.O. Box 97794 Jackson, MS 39288-7794

# Order Form



## **Order Form**

This product includes a file in Windows Write format entitled OrderFrm.Wri which you can use to print an order form for registering StrPlus.DLL.



## **Software Evaluation Form**

This package includes a file in Windows Write format entitled EvalFrm.Wri which you can use to print a software evaluation form.



## **Address**

Send to:

Digital PowerTOOLS  
P.O. Box 97794  
Jackson, MS 39288-7794



# Registered License

Upon registration of this product, you may include StrPlus.DLL in your own application projects to end users. You may not offer StrPlus.DLL as an independent application or utility; and, you may not offer StrPlus.DLL in a manner that would directly compete with the product.

StrPlus.DLL is provided as a utility to enhance the functionality of your own applications . As such, you are not permitted to reveal how to use the functions of StrPlus.DLL to your end users without the express written permission of Digital PowerTOOLS.

It is our sincere hope that as a registered user of this product you find StrPlus.DLL to be a welcome addition to your library of VB (or C) utilities. If you have questions about usage or suggestions for improvement, please write to let us know. We appreciate your business!

## Function: **GetStrPlusVersion**

Declare Function GetStrPlusVersion% Lib "3DfxPlus.DLL" ()

- \* This function returns the version of StrPlus.
- \* The returned value is 100 times the actual version number.
- \* For instance, a returned value of 150 represents version 1.50.

## Function: JustifyLine

Declare Sub JustifyLine Lib "3DfxPlus.DLL" (ByVal hDC%, ByVal StartXpixel%, ByVal StartYpixel%, ByVal TheWidthPixels%, ByVal lpString\$)

- \* Use this function to justify a line of text evenly between two margins.
- \* You should use the command in the Paint Event

- \* hDC% will invariably be: me.hDC
- \* StartXpixel% is the horizontal starting point in pixels
- \* StartYpixel% is the vertical starting point in pixels
- \* TheWidthPixels% is the number of pixels wide in which to do the formatting
- \* lpString\$ is the string to output

- \* the TextJust.Frm demonstrates the use of this command

## Function: SuperPrint

Declare Sub SuperPrint Lib "3DfxPlus.DLL" (ByVal hDC%, ByVal XstartPixel%, ByVal YstartPixel%, ByVal TheString\$, ByVal FontName\$, ByVal TheStyle\$, ByVal The3DStyle\$, ByVal AlignmentType\$, ByVal PointSize%, ByVal TheColor&, ByVal ShadowColor&, ByVal RotationAngle%)

- \* this command outputs text to a Form or PictureBox with 3D shading, alignment, and Rotation options

- \* use this command in the Paint Event of your Forms or PictureBoxes

- \* hDC% will invariably be: me.hDC

- \* XstartPixel% is the horizontal starting point in pixels

- \* YstartPixel% is the vertical starting point in pixels

- \* TheString\$ is the string to be output

- \* FontName\$ is the facename to use ("Arial", "Times New Roman", etc.)

- \* TheStyle\$ is one of the following: "bold", "italic", "BoldItalic", or "plain"

- \* The3DStyle\$ is one of the following: "raised", "sunken", or "plain"

- \* AlignmentType\$ is one of the following: "left", "center", or "right"

- \* PointSize% is the output point size to use (8, 10, 12, etc.)

- \* TheColor& is the output color to use: for example, RGB(0,0,255) or QBColor (1)

- \* ShadowColor& is the color of the shadow: for example, RGB (255,255,255) or QBColor (15)

- \* RotationAngle is the angle of rotation to use: any value from 0 thru 359

- \* the TextOut.Frm demonstrates how to use this command

# Functions: Path Strings

These functions extract portions of a full path string.

**Declare Sub ExtFromPath Lib "StrPlus.DLL" (ByVal IpFullPath\$, ByVal IpString\$)**

- \* Extracts the file extension from a full path.
- \* IpFullPath\$ is the full path including the file name.
- \* IpString\$ should be set to 255 blank spaces before calling the function.
- \* For example, IpString\$=space\$(255).
- \* Upon completion, IpString\$ will contain the file extension.

**Declare Sub FileNameOnlyFromPath Lib "StrPlus.DLL" (ByVal IpFullPath\$, ByVal IpString\$)**

- \* Extracts the filename (without the extension) from a full path.
- \* IpFullPath\$ is the full path including the file name.
- \* IpString\$ should be set to 255 blank spaces before calling the function.
- \* For example, IpString\$=space\$(255).
- \* Upon completion, IpString\$ will contain the filename (without the extension).

**Declare Sub FullFileNameFromPath Lib "StrPlus.DLL" (ByVal IpFullPath\$, ByVal IpString\$)**

- \* Extracts the filename (including the extension) from a full path.
- \* IpFullPath\$ is the full path including the file name.
- \* IpString\$ should be set to 255 blank spaces before calling the function.
- \* For example, IpString\$=space\$(255).
- \* Upon completion, IpString\$ will contain the filename (including the extension).

**Declare Sub DirFromPath Lib "StrPlus.DLL" (ByVal IpFullPath\$, ByVal IpString\$)**

- \* Extracts the directory (including leading & trailing backslashes) from a full path.
- \* IpFullPath\$ is the full path including the file name.
- \* IpString\$ should be set to 255 blank spaces before calling the function.
- \* For example, IpString\$=space\$(255).
- \* Upon completion, IpString\$ will contain the directory.

**Declare Sub DriveFromPath Lib "StrPlus.DLL" (ByVal IpFullPath\$, ByVal IpString\$)**

- \* Extracts the drive letter (including the semi-colon) from a full path.
- \* IpFullPath\$ is the full path including the file name.
- \* IpString\$ should be set to 255 blank spaces before calling the function.
- \* For example, IpString\$=space\$(255).
- \* Upon completion, IpString\$ will contain the drive letter.

# Functions: Number Conversion

Declare Sub GetOrdinalExt Lib "StrPlus.DLL" (ByVal TheNumber%, ByVal IpReturnString\$)

- \* this command returns the ordinal extension for a given integer
- \* "1" returns "st", "2" returns "nd", "3" returns "rd", and so on
- \* TheNumber% is the value to obtain the ordinal extension for
- \* IpReturnString\$ should be set 3 blank spaces before calling this command.
- \* for example, IpReturnString\$=Space\$(3)

Declare Sub GetRomanNumber Lib "StrPlus.DLL" (ByVal TheNumber%, ByVal IpReturnString\$)

- \* this command returns the Roman number of a given integer
- \* "1994" returns "MCMXCIV"
- \* TheNumber% is the value to obtain the Roman number for
- \* IpReturnString\$ should be set 255 blank spaces before calling this command.
- \* for example, IpReturnString\$=Space\$(255)

Declare Sub GetWordNumber Lib "StrPlus.DLL" (ByVal IpAmount\$, ByVal IpReturnString\$)

- \* this command returns the word equivalent of a given number, usually currency
- \* "123.45" returns "One Hundred Twenty-Three and 45/100"
- \* IpAmount\$ is the number strings to convert (transform to a string before calling routine)
- \* for example, IpAmount\$=abs(val(money#))
- \* IpReturnString\$ should be set 255 blank spaces before calling this command.
- \* for example, IpReturnString\$=Space\$(255)

# Functions: String Conversion

Declare Sub ReverseString Lib "StrPlus.DLL" (ByVal IpString\$)

- \* reverses a string
- \* "abcdefg" becomes "gfedcba"
- \* this is a convenient facility for using the VB command INSTR\$ to search backwards through a string
- \* IpString\$ is the string to reverse

Declare Sub ToName Lib "StrPlus.DLL" (ByVal IpString\$)

- \* the VB UCASE\$ command converts a string to all uppercase and the VB LCASE\$ command converts a string to all lowercase
- \* ToName makes the first letter of every word in a string upper case
- \* for example, "john q. public" becomes "John Q. Public"
- \* IpString\$ is the string to convert to first letter capitalization
- \* the conversion stops when a CHR\$(0) or CHR\$(13) is encountered
- \* the maximum string length is 1024 characters
- \* ToName is most useful with data entry TextBoxes

Declare Sub Encrypt Lib "StrPlus.DLL" (ByVal IpString\$, ByVal KeyCode%)

- \* encodes a programmer-supplied string
- \* IpString\$ is the string to encode
- \* KeyCode% is the Key for encoding (acceptable range is 1 - 64)
- \* this command is useful with small text strings in INI and database files
- \* maximum string length is 240 characters
- \* you MUST record the Key value because it is needed to decode the string

Declare Sub Decrypt Lib "StrPlus.DLL" (ByVal IpString\$, ByVal KeyCode%)

- \* decodes a string previously encoded with Encrypt
- \* IpString\$ is the string to decode
- \* KeyCode% is the SAME Key that was used with Encrypt to encode the string

## Function: GetStateOfKey

Declare Function GetStateOfKey% Lib "StrPlus.DLL" (ByVal KeyName\$)

\* returns the state of certain system keys

\* KeyName\$ is the name of the key to obtain to state of:

"ScrollLock"

"NumLock"

"CapsLock"

"Rshift"

"Lshift"

"Control"

"Alt"

\* for example, with x%=GetStateOfKey ("NumLock"), x% will be TRUE if the NumLock key is on, or FALSE otherwise



## Function: CountWords

Declare Function CountWords% Lib "StrPlus.DLL" (ByVal IpString\$)

- \* counts the number of words in string
- \* IpString\$ is the string to count the words of
- \* stops counting when a CHR\$(13) or CHR\$(0) is encountered
- \* maximum string length is 1024 characters
- \* the WordCnt form demonstrates how to use this command to count the words in an entire document (or selected text only)
- \* because this routine is written in Assembly language it is fast (faster than GetToken) counting 1800 words/second on a 386 DX at 33mHz (much faster on a 486 or Pentium machine)

# Functions: Tokens

Though these two commands, `GetTokenCount` and `GetToken`, may not appear so upon first glance, these are the MOST POWERFUL commands in the `StrPlus.DLL` utility. `GetToken` can be used to parse substrings out of `Command$`, create a command interpreter for your application, separate comma-delimited strings for database conversion, count words in a string, simulate the old BASIC `Data/Read` commands, parse words in an adventure game, and much more.

In fact, `Token.Frm` demonstrates an included VB subroutine (`ReadData`) that uses `GetToken` to simulate the old BASIC `Data` and `Read` statements. Although this convenience saves considerable time in writing a program, it does mean that each data element is included twice in your EXE file. This is probably why MicroSoft omitted `Data` and `Read` from VB.

Declare Function `GetTokenCount%` Lib "StrPlus.DLL" (ByVal `lpString$`, ByVal `lpDelimiters$`)

- \* this command counts the number of tokens in a specified string
- \* `lpString$` is the string to search
- \* `lpDelimiters$` contains the list of delimiting characters (for example, ",;. ")
- \* setting `lpDelimiters$` to `CHR$(32)+CHR$(9)+CHR$(13)+CHR$(10)` effectively allows `GetTokenCount%` to count the words in a string

Declare Sub `GetToken` Lib "StrPlus.DLL" (ByVal `lpString$`, ByVal `lpDelimiters$`, ByVal `TokenNumber%`, ByVal `lpReturn$`)

- \* this command parses substrings out of a larger control string
- \* `lpString$` is the control string
- \* `lpDelimiters$` is the list of delimiting characters (to use as separators)
- \* `TokenNumber%` is the token number to parse out (acceptable range is 1 to `GetTokenCount%(lpString$,lpDelimiters$)`)
- \* `lpReturn$` should be set to 255 blank spaces before calling the routine; upon return `lpReturn$` will contain the parsed substring. For example, `lpReturn$=Space$(255)`

# Function: WordColor

Declare Function WordColor& Lib "StrPlus.DLL" (ByVal ColorWord\$)

- \* this command converts specified strings to long color values
- \* it can be used anywhere that RGB and QBColor are used
- \* ColorWord\$ is the name of the color; all 16 basic colors and 20 system colors are supported
- \* no longer do you have to refer to color number charts for when using the common colors in your programs

ColorWord\$()	QBColor()	RGB()
"Black"	0	0,0,0
"Blue"	1	0,0,191
"Green"	2	0,191,0
"Cyan"	3	0,191,191
"Red"	4	191,0,0
"Magenta"	5	191,0,191
"DarkYellow"	6	191,191,0
"LightGray"	7	191,191,191
"DarkGray"	8	64,64,64
"BrightBlue"	9	0,0,255
"BrightGreen"	10	0,255,0
"BrightCyan"	11	0,255,255
"BrightRed"	12	255,0,0
"BrightMagenta"	13	255,0,255
"BrightYellow"	14	255,255,0
"BrightWhite"	15	255,255,255

The supported system colors will vary depending on the user's Windows color configuration setup. The acceptable strings are:

"ActiveBorder"  
"ActiveCaption"  
"AppWorkSpace"  
"BackGround"  
"BtnFace"  
"BtnHighlight"  
"BtnShadow"  
"BtnText"  
"CaptionText"  
"GrayText"  
"Highlight"  
"HighlightText"  
"InactiveBorder"  
"InactiveCaption"  
"InactiveCaptionText"  
"Menu"  
"MenuText"  
"Window"  
"WindowFrame"  
"WindowText"



