

# WinCpAPI Official Documentation

©2014 bluePCdev. All rights reserved.

<http://www.bluepcdev.ml> Version 0.2 beta

## Disclaimer

This unit is designed to work on Windows Vista, 7, 8, 8.1 for Free Pascal Console Programs Only. This unit does NOT support graph windows, or Lazarus GUI programs. (Due to issues with SAPI and FPC in Windows XP, this unit may not function on an XP System). This unit is provided as is, and I am not responsible for any bugs within your program caused by it. If you find any bugs, please submit them in the contact page at <http://www.bluepcdev.ml> with a detailed description of the issue, as well as any error codes.

## Usage

Simply place the WinCpAPI.ppu and WinCpAPI.o files in the same directory as your program's source code (\*.pas) and add WinCpAPI to your uses statement. Example: `uses WinCpApi, crt;` You can now use these procedures within your program. The program executable will not be dependent of the unit files, therefore your program will work without the .o and .ppu files present after it is compiled.

## Procedures

*cmd(Command:String);*

~The cmd procedure will execute a single DOS command (via command prompt) within the Pascal Program. For example:

`cmd('MKDIR Folder');` Will create a folder called "Folder" in the current directory, then continue to run the program's code without delay.

*resize(columns:integer,rows:integer);*

~The resize procedure will resize the program window, as well as the crt output area to the number of columns and rows specified. For example:

`resize(120,60);` Will resize the window to hold 120 columns and 60 rows of text. Keep in mind that because the crt unit creates a text window, the rows will be wider than the columns.

*colscr(background:byte,text:byte);*

~The colscr Procedure will change the output background, and text colors in a single statement. The first parameter should be the text background color, and the second should be the font color. For example: `colscr(9,2);` or `colscr(blue,green);` will both change the text background to blue and font to green. Note: Since color names don't always work, numeric codes are preferable.

*SpeakText(txt:string);*

~The SpeakText Procedure makes use of SAPI in Windows. It will use the system's speakers to speak text out loud. For example: `speaktext('Greetings, user.');` Will give the user a friendly verbal greeting. Note: This Procedure can glitch in certain situations. Test your program thoroughly.

*WriteTxt(tfile:text,x:integer,y:integer);*

~The Writetxt Procedure will write the contents of the text file tfile to the screen starting from the coordinates (x,y). It will open and close the file on it's own so use this outside of, or without reset/close statements. This Procedure is ideal for using ascii art from a text file for animations without placing the art in the code itself. For example `writetxt(tfile,7,7);` will print the contents of tfile to the screen, with the first character of the file at (7,7).

*PlaySound(path:string);*

~The PlaySound Procedure simplifies the ability to play .wav audio files within a program. The parameter should simply be a string containing the audio file's path (or name if it is in the same directory). For example `PlaySound(music.wav);` will play the file "music.wav" if it is in the same folder as the program's exe file. Consequentially, assuming the file exists in the given path `PlaySound(C:/Users/Mike/Music/Song.wav);` will do the same. Playing the sound file will not pause the program, and will continue until the program is terminated or the track ends unless PlaySound is invoked again. Ideal for sound effects or credits.

*PlayLoop(path:string);*

~The PlayLoop Procedure will function exactly the same as PlaySound, except it will replay the file if it reaches the end of the track before it is stopped. Ideal for background music.