

## DOS Batch - Functions

Keep your batch script clean, call DOS functions.

<a href="#">:append</a>	appends a string to a specific line in a text file
<a href="#">:bannerPingPong</a>	moves text in varref one step left or right and updates title
<a href="#">:bannerRotate</a>	rotates text in varref one step and updates title
<a href="#">:choiceListInput</a>	lets the user choose from list of last entered values
<a href="#">:CmpFTime</a>	compares the time of two files, succeeds if condition is met, fails otherwise
<a href="#">:date2jdate</a>	converts a gregorian calender date to julian day format
<a href="#">:dayOfYear</a>	returns the day of the year, i.e. 1 for 1/1/2008, 266 for 12/31/2008
<a href="#">:DeleteIfOld</a>	deletes file or directory if older than given number of days
<a href="#">:doProgress</a>	displays the next progress tick
<a href="#">:dumpArr</a>	dumps the array content / under construction
<a href="#">:echo</a>	echoes text in a specific color
<a href="#">:echoLine</a>	outputs a formatted string, substitutes file name and line number
<a href="#">:extractFileSection</a>	extracts a section of file that is defined by a start and end mark
<a href="#">:ExtractFunction</a>	extracts a function by label
<a href="#">:ExtractSimple</a>	extracts a marked section from batch file, sustitutes variables during extraction
<a href="#">:false</a>	returns failure
<a href="#">:Format</a>	outputs columns of strings right or left aligned
<a href="#">:fprop</a>	returns a file property
<a href="#">:ftime</a>	returns the file time in julian days
<a href="#">:getColorCode</a>	converts color text to color code
<a href="#">:getDriveInfo</a>	returns array of fsutil drive information
<a href="#">:getDrives</a>	This function has been replaced by :getDriveInfo
<a href="#">:getFunctions</a>	returns a comma separated list of all functions
<a href="#">:getHostName</a>	resolves IP address to computer name
<a href="#">:getIP</a>	return THIS computers IP address
<a href="#">:getIPConfig</a>	return IPCONFIG /ALL data in array variable
<a href="#">:getNextInList</a>	return next value in list
<a href="#">:getRandomColor</a>	returns a random color
<a href="#">:GetRegKeys</a>	returns all registry sub keys of a given registry key
<a href="#">:GetRegValue</a>	returns a registry value
<a href="#">:GetRegValues</a>	returns all registry values of a given registry key
<a href="#">:getServices</a>	returns array of service information
<a href="#">:getVarsByAttr</a>	returns a comma separated list of variables based on an attribute name
<a href="#">:htmlhelp</a>	dumps html help to console
<a href="#">:Init</a>	initializes the environment for this command library
<a href="#">:initProgress</a>	initializes an internal progress counter and display the progress in percent
<a href="#">:initVarsByAttr</a>	restores the values of the persistent variables to defaults
<a href="#">:IsFileOpen</a>	succeeds if file in use, fails otherwise
<a href="#">:IsRegKey</a>	succeeds if Key exists
<a href="#">:IsRegValue</a>	succeeds if Key and Value exists
<a href="#">:IsServiceRunning</a>	returns success if service is running, otherwise failure
<a href="#">:jdate</a>	converts a date string to julian day number with respect to regional date format
<a href="#">:jdate2date</a>	converts julian days to gregorian date format
<a href="#">:kBytesFree</a>	returns the free space of a drive in kilobytes
<a href="#">:l2a</a>	converts a list to an array
<a href="#">:loadPersistentVars</a>	loads the values of the persistent variables

C:\> DosTips.com

C:\>title DosTips.com

for Windows10,Windows8,Windows7,VISTA,XP,NT,Server 2000,Server 2003,Server 2008

ENHANCED BY Google

[Home](#)

**Source Script**

- [Batch Files](#)
- [Functions](#)
- Function Library**
- [Script Snippets](#)
- [Interfacing](#)

**Tips and Tricks**

- [Copy Tips](#)
- [Menu in Batch](#)
- [String Manipulation](#)
- [String Operations](#)
- [Date and Time](#)
- [Arithmetic](#)
- [FTP Batch Script](#)
- [Command Index](#)

**Tutorials**

- [My First Batch](#)
- [Solid Framework](#)
- [Functions](#)
- [Persistency](#)

**Forum**

**General**

- [Terms of Use](#)
- [Enablers](#)
- [About Us](#)
- [Contact Us](#)

**Others**

- [Want dona](#)

<a href="#">:loadRegVars</a>	loads the values of the persistent variables
<a href="#">:lookup</a>	returns the value associated with a key in a map of key-value pairs
<a href="#">:lTrim</a>	strips white spaces (or other characters) from the beginning of a string
<a href="#">:MakeAbsolute</a>	makes a file name absolute considering a base path
<a href="#">:MakeRelative</a>	makes a file name relative to a base path
<a href="#">:NetworkDeviceGuid2Name</a>	gets a Network Device Name from its corresponding GUID
<a href="#">:NetworkDeviceName2Guid</a>	gets a Network Device GUID from its corresponding Name
<a href="#">:parse</a>	parses a string matching a formatter
<a href="#">:pwd</a>	shows a password dialog box
<a href="#">:regedit</a>	opens regedit at predefined location
<a href="#">:removeArr</a>	removes an array
<a href="#">:rTrim</a>	strips white spaces (or other characters) from the end of a string
<a href="#">:savePersistentVars</a>	save values of persistent variables into a file
<a href="#">:saveRegVars</a>	save values of persistent variables into a file
<a href="#">:set</a>	sets var[] to output of command / under construction
<a href="#">:SetRegValue</a>	sets a registry value
<a href="#">:sleep</a>	waits some seconds before returning
<a href="#">:StartsWith</a>	Tests if a text starts with a given string
<a href="#">:strLen</a>	returns the length of a string
<a href="#">:substitute</a>	substitutes a string in a text file
<a href="#">:ToANSI</a>	converts a file to ANSI
<a href="#">:toCamelCase</a>	converts a string to camel case
<a href="#">:toDec</a>	convert a hexadecimal number to decimal
<a href="#">:toHex</a>	convert a decimal number to hexadecimal, i.e. -20 to FFFFFFEC or 26 to 0000001A
<a href="#">:toLower</a>	converts uppercase character to lowercase
<a href="#">:ToUNICODE</a>	converts a file to UNICODE
<a href="#">:toUpper</a>	converts lowercase character to uppercase
<a href="#">:Trim</a>	strip white spaces (or other characters) from the beginning and end of a string
<a href="#">:trimSpaces</a>	trims spaces around string variable
<a href="#">:trimSpaces2</a>	trims spaces around string and assigns result to variable
<a href="#">:true</a>	returns success
<a href="#">:Unique</a>	returns a unique string based on a date-time-stamp, YYYYMMDDhhmmsscc
<a href="#">DOS Batch - Functions</a>	Keep your batch script clean, call DOS functions.
<a href="#">DOS Function Template</a>	A function template with all features of a true DOS function.

## :append - appends a string to a specific line in a text file

TOP  
2008-02-19Description: call:[:append](#) str file line

Script:

```

1. :append str file line -- appends a string to a specific line in a text file
2. ::          -- str [in] - string to be appended
3. ::          -- file [in] - file name to append the string to
4. ::          -- line [in] - line number to append the string to, first line is 1, omit for last line
5. :$created 20060101 :$changed 20080219 :$categories FileManipulation
6. :$source https://www.dostips.com
7. SETLOCAL
8. set ap=%~1
9. set f=%~2
10. set c=%~3
11. if not defined c (
12.     set c=0
13.     for /f %%a in ('find /v /c ""^<%f%""') do set c=%%a

```

```

14.)
15.(for /f "tokens=1,* delims:=" %%a in ('findstr /v /n "$$^<%f%"') do (
16.    if "%a"=="%c%" (echo.%b%ap%) ELSE echo.%b%
17.))>%temp%.t0815.txt"
18.move /y "%temp%.t0815.txt" "%f%"
19.EXIT /b

```

**:bannerPingPong - moves text in varref one step left or right and updates title**TOP  
2008-02-19Description: call:[:bannerPingPong](#) varref

```

1. :bannerPingPong varref -- moves text in varref one step left or right and updates title
2. :: -- varref [in,out] - variable name with banner text, format: "Banner Text-----"
3. :$created 20060101 :$changed 20080219
4. :$source https://www.dostips.com
5. SETLOCAL ENABLEDELAYEDEXPANSION
6. set s=!%~1: =-
7. if "!s:~-1!" NEQ "-" if "!s:~-1!" NEQ "+" set s=!s!-----
8. set d=!s:~-1!
9. if "!s:~0,1!" NEQ "-" set d+=
10. if "!s:~-2,1!" NEQ "-" set d=-
11. if "!d!"=="+" (set s=-!s:~0,-2!+) ELSE (set s=!s:~1,-1--)
12. TITLE !s!
13. ( ENDLOCAL & REM RETURN VALUES
14.     IF "%~1" NEQ "" SET %~1=%s%
15. )
16. EXIT /b

```

**:bannerRotate - rotates text in varref one step and updates title**TOP  
2008-02-19Description: call:[:bannerRotate](#) varref

```

1. :bannerRotate varref -- rotates text in varref one step and updates title
2. :: -- varref [in,out] - variable name with banner text, format: "Banner Text-----"
3. :$created 20060101 :$changed 20080219
4. :$source https://www.dostips.com
5. SETLOCAL ENABLEDELAYEDEXPANSION
6. set s=!%~1: =-
7. set s=!s:~1!!s:~0,1!
8. TITLE !s!
9. ( ENDLOCAL & REM RETURN VALUES
10.    IF "%~1" NEQ "" SET %~1=%s%
11. )
12. EXIT /b

```

**:choiceListInput - lets the user choose from list of last entered values**TOP  
2008-02-19Description: call:[:choiceListInput](#) ret list title max

```

1. :choiceListInput ret list title max -- lets the user choose from list of last entered values
2. :: -- ret [out] - varref returns input value
3. :: -- list [in,out] - varref with choice list, returns trimmed reordered choice list
4. :: -- title [in] - list title
5. :: -- max [in] - maximum number of list entries
6. :$created 20060101 :$changed 20080219 :$categories Input

```

```

7. :$source https://www.dostips.com
8. SETLOCAL ENABLEDELAYEDEXPANSION
9. set l=,!%~2!,&           rem.-- get the choice list
10. set t=%~3&               rem.-- get the list title
11. set c=%~4&               rem.-- get the list entry count limit
12. set m=&                 rem.-- a message
13. set l2=
14. set v=
15. :choiceListInput_LOOP1
16. echo.%t%
17. set i=0&for /f "delims=" %%a in ('"echo.%l:,=&echo.%"'') do (
18.     set /a i+=1
19.     set l2=!l2!!i!;%%~a,
20.     set v=%%~a
21.     echo. !i! - %%~a
22. )
23. if "%m%" NEQ "" echo. ^>^> %m%
24. echo. Make a choice or enter a new value [%v%]
25. set v1=%v%
26. set /p v1= :
27. echo.
28. set v2=!v1!&set v2=!v2:,!=!&set v2=@=:!&set v2=!v2:;!=!&set v2=!v2:"=!
29. rem.--reject entry with illegal character
30. if "!v1!" NEQ "!v2!" (
31.     set m=Note: ,;@" and empty string not allowed. Try again.
32.     goto:choiceListInput_LOOP1
33. )
34. rem.--if first character is minus then remove the entry
35. set remove=&if "%v1:~0,1%"=="-" set remove=y&set v1=%v1:~1%
36. set v=%v1%
37. rem.--if number chosen then find corresponding value
38. set l3=!l2:,%v%;=,@!
39. if "%l3%" NEQ "%l2%" (
40.     for /f "delims=@ tokens=2" %%a in ("!l3!") do set l3=%%a
41.     for /f "delims=," %%a in ("!l3!") do set v=%%a
42. )
43. rem.--remove value from list if exist
44. set l3=%l%
45. set l=!l:,%v%,=,!%
46. if "%remove%"=="y" (
47.     if "%1%"=="%l3%" (set m='%v%' cannot be removed from list
48. ) ELSE (set m='%' has been removed from list)
49. goto:choiceListInput_LOOP1
50. )
51. if "%1%"=="%l3%" echo. ^>^>%v% has been added to the list
52. rem.--add to the value to the end
53. set l=!l:~1!%v%
54. rem.--enforce the list entry count limit if requested
55. if "%c%" NEQ "" (
56.     set i=0&for %%a in (%l%) do set /a i+=1
57.     if /i !i! GTR !c! (
58.         for /f "delims=, tokens=1,*" %%a in ("!l!") do (
59.             set l=%b
60.             echo. ^>^>%a dropped out of the list

```

```

61.      )
62.    )
63.  )
64.  ( ENDLOCAL & REM RETURN VALUES
65.    IF "%~1" NEQ "" (SET %~1=%v%) ELSE (echo.%v%)
66.    IF "%~2" NEQ "" (SET %~2=%1%) ELSE (echo.%1%)
67.  )
68. EXIT /b

```

**:CmpFTime - compares the time of two files, succeeds if condition is met, fails otherwise**TOP  
2008-02-19Description: call:[:CmpFTime](#) op file1 file2 attr1 attr2

```

Script: 1. :CmpFTime op file1 file2 attr1 attr2 -- compares the time of two files, succeeds if condition is met, fails otherwise
2. ::          -- op      [in]   - compare operator, see 'IF /?', i.e.EQU, NEQ, LSS, LEQ, GTR, GEQ
3. ::          -- fileL  [in]   - file name, left side of comparisation
4. ::          -- fileR  [in]   - file name, right side of comparisation
5. ::          -- attrL  [in,opt] - time field to be used for fileL, see 'dir /?', i.e. /tc, /ta, /tw, default is /tw
6. ::          -- attrR  [in,opt] - time field to be used for fileR, default is attrL
7. :$created 20060101 :$changed 20080219 :$categories DateAndTime,FileOperation
8. :$source https://www.dostips.com
9. SETLOCAL
10. set op=%~1
11. set fileL=%~2
12. set fileR=%~3
13. set attrL=%~4
14. set attrR=%~5
15. if "%op%"=="=" set op===
16. if "%attrL%"=="=" set attrL=/tw
17. if "%attrR%"=="=" set attrR=%attrL%
18. for /f "tokens=1-6 delims=/:" %%a in ("dir %attrL% /-c "%fileL%"|findstr ""^[[0-1]"") do (
19.     set TL=%%c%%a%%b%%f%%d%%e
20. )
21. for /f "tokens=1-6 delims=/:" %%a in ("dir %attrR% /-c "%fileR%"|findstr ""^[[0-1]"") do (
22.     set TR=%%c%%a%%b%%f%%d%%e
23. )
24. if "%TL%" "%op%" "%TR%" (rem.) ELSE set=2>NUL
25. EXIT /b

```

**:date2jdate - converts a gregorian calender date to julian day format**TOP  
2008-02-19Description: call:[:date2jdate](#) JD YYYY MM DD

```

Script: 1. :date2jdate JD YYYY MM DD -- converts a gregorian calender date to julian day format
2. ::          -- JD    [out] - julian days
3. ::          -- YYYY [in]  - gregorian year, i.e. 2006
4. ::          -- MM    [in]  - gregorian month, i.e. 12 for december
5. ::          -- DD    [in]  - gregorian day, i.e. 31
6. :$reference https://aa.usno.navy.mil/faq/docs/JD_Formula.html
7. :$created 20060101 :$changed 20080219 :$categories DateAndTime
8. :$source https://www.dostips.com
9. SETLOCAL
10. set "yy=%~2"&set "mm=%~3"&set "dd=%~4"
11. set /a "yy=1000%yy% %%0000,mm=100%mm% %% 100,dd=100%dd% %% 100"
12. if %yy% LSS 100 set /a yy+=2000 &rem Adds 2000 to two digit years

```

```

13. set /a JD=dd-32075+1461*(yy+4800+(mm-14)/12)/4+367*(mm-2-(mm-14)/12*12)/12-3*((yy+4900+(mm-14)/12)/100)/4
14. ENDLOCAL & IF "%~1" NEQ "" (SET %~1=%JD%) ELSE (echo.%JD%)
15. EXIT /b

```

**:dayOfYear - returns the day of the year, i.e. 1 for 1/1/2008, 266 for 12/31/2008**TOP  
2008-06-12Description: call:[:dayOfYear](#) JD DateStr

```

Script: 1. :dayOfYear JD DateStr -- returns the day of the year, i.e. 1 for 1/1/2008, 266 for 12/31/2008
2. ::          -- day      [out,opt] - variable name to store resulting day of the year
3. ::          -- DateStr [in,opt] - date string, e.g. "3/31/2006" or "Fri 03/31/2006" or "31.3.2006", or omit form current date
4. ::$reference https://groups.google.com/group/alt.msdos.batch.nt/browse_frm/thread/a0c34d593e782e94/50ed3430b6446af8#50ed3430b6446af8
5. ::$created 20080612 ::$changed 20080612 ::$categories DateAndTime
6. ::$source https://www.dostips.com
7. SETLOCAL
8. set "DateStr=%~2"&if "%~2""="" set "DateStr=%date%"
9. for /f "skip=1 tokens=2-4 delims=(-)" %%A in ('"echo.|date"') do (
10.    for /f "tokens=1-3 delims=/-" %%A in ("%DateStr:* =%") do (
11.      set %%a=%%A&set %%b=%%B&set %%c=%%C))
12. set /a "yy=10000%yy% %%10000,mm=100%mm% %% 100,dd=100%dd% %% 100"
13. set /a JD=dd-32075+1461*(yy+4800+(mm-14)/12)/4+367*(mm-2-(mm-14)/12*12)/12-3*((yy+4900+(mm-14)/12)/100)/4
14. set /a "yy=10000%yy% %%10000,mm=1,dd=1"
15. set /a JD---1+dd-32075+1461*(yy+4800+(mm-14)/12)/4+367*(mm-2-(mm-14)/12*12)/12-3*((yy+4900+(mm-14)/12)/100)/4
16. ENDLOCAL & IF "%~1" NEQ "" (SET %~1=%JD%) ELSE (echo.%JD%)
17. EXIT /b

```

**:DeleteIfOld - deletes file or directory if older than given number of days**TOP  
2008-02-19Description: call:[:DeleteIfOld](#) name days tnowDependencies: [:ftime](#)

```

Script: 1. :DeleteIfOld name days tnow -- deletes file or directory if older than given number of days
2. ::          -- name [in] - name of file or directory
3. ::          -- days [in] - number of days to expire
4. ::          -- tnow [in] - today's date in julia days
5. ::$created 20060101 ::$changed 20080219 ::$categories DateAndTime,FileOperation
6. ::$source www.DosTips.com
7. SETLOCAL
8. set "days=%~2"
9. set "tnow=%~3"
10. call:ftime tfile "%~1"
11. set /a "diff=tnow-tfile"
12. if %diff% LEQ %days% EXIT /b
13. set "attr=%~1"
14. rem ECHO.%attr%, %attr:~0,1%, %~nx1 is %diff% days old
15. if /i "%attr:~0,1%"=="d"
16.   rd /Q /S "%~1"
17. ) ELSE (
18.   del /Q "%~1"
19. )
20. EXIT /b

```

**:doProgress - displays the next progress tick**TOP  
2008-03-27

Description: call:[doProgress](#)

Script:

```

1. :doProgress -- displays the next progress tick
2. ::$created 20060101 ::$changed 20080327 ::$categories Progress
3. ::$source https://www.dostips.com
4. set /a "ProgressCnt+=1"
5. SETLOCAL ENABLEDELAYEDEXPANSION
6. set /a "per100=100*ProgressCnt/ProgressMax"
7. set /a "per10=per100/10"
8. set /a "per10m=10-per100/10-1"
9. set "P=%per100%%"
10. set "PP="
11. for /l %%N in (0,1,%per10%) do call set "PP=%>PP%*"
12. for /l %%N in (%per10%,1,9) do call set "PP=%>PP%* "
13. set "PPP="
14. for /l %%N in (0,1,%per10m%) do call set "PPP=%>PPP%*"
15. set "ProgressFormat=%ProgressFormat:[P]=!P!%"
16. set "ProgressFormat=%ProgressFormat:[PP]=!PP!%"
17. set "ProgressFormat=%ProgressFormat:[PPP]=!PPP!%"
18. title %ProgressFormat%
19. EXIT /b

```

## :dumpArr - dumps the array content / under construction

[TOP](#)  
2008-02-19

Description: call:[dumpArr](#) arr

Script:

```

1. :dumpArr arr -- dumps the array content / under construction
2. ::          -- arr [in] - array name
3. ::$created 20060101 ::$changed 20080219 ::$categories Array
4. ::$source https://www.dostips.com
5. SETLOCAL
6. call set i=%%%~1[#]%
7. for /l %%n in (1,1,%i%) do call echo."%>%~1[%n]~2%"
8. EXIT /b

```

## :echo - echoes text in a specific color

[TOP](#)  
2008-02-19

Description: call:[echo](#) col txt

Dependencies: [getColorCode](#)

Script:

```

1. :echo col txt -- echoes text in a specific color
2. ::          -- col [in] - color code, append a DOT to omit line break, call 'color /?' for color codes
3. ::          -- txt [in] - text output
4. ::$created 20060101 ::$changed 20080219 ::$categories Echo,Color
5. ::$source https://www.dostips.com
6. SETLOCAL
7. for /f "tokens=1,*" %%a in ("*") do (
8.     set col=%>%%a
9.     set txt=%>~b
10.)
11.set cr=Y
12.if "%col:~-1%"=="." (
13.    set cr=N
14.    set col=%col:~0,-1%
15.)

```

```

16. call:getColorCode "%col%" col
17. set com=%temp%\color%col%.com
18. if not exist "%com%" (
19.     echo.N %COM%
20.     echo.A 100
21.     echo.MOV BL,%col%
22.     echo.MOV BH,0
23.     echo.MOV SI,0082
24.     echo.MOV AL,[SI]
25.     echo.MOV CX,1
26.     echo.MOV AH,09
27.     echo.INT 10
28.     echo.MOV AH,3
29.     echo.INT 10
30.     echo.INC DL
31.     echo.MOV AH,2
32.     echo.INT 10
33.     echo.INC SI
34.     echo.MOV AL,[SI]
35.     echo.CMP AL,0D
36.     echo.JNZ 109
37.     echo.RET
38.     echo.
39.     echo.r cx
40.     echo.22
41.     echo.w
42.     echo.q
43.)|debug>NUL
44."%com%" %txt%
45.rem del "%com%" /q
46.if "%cr%"=="Y" echo.
47.EXIT /b

```

**:echoLine - outputs a formatted string, substitutes file name and line number**TOP  
2008-05-12Description: call:echoLine uniqueStr formatter offset

```

Script: 1. :echoLine uniqueStr formatter offset -- outputs a formatted string, substitutes file name and line number
2. :: -- uniqueStr [in]      - a unique string to identify the line
3. :: -- formatter [in,opt] - a string using __FILE__ and/or __LINE__ to be substituted and echoed
4. :: -- offset   [in,opt] - offset to be added to the line number
5. :$reference https://www.dostips.com/forum/viewtopic.php?t=369
6. :$created 20080512 ;$changed 20080512 :$categories Echo,Debug
7. :$source https://www.dostips.com
8. Setlocal Disabledelayedexpansion
9. Set "Fmt=%~2"
10.If Not Defined Fmt Set "Fmt=__FILE__(__LINE__): ERROR"
11.For /F "Delims=:" %%A In ('"Findstr /N "%~1" "%~f0""') Do Set /A "lineNr=%A+%~30/10"
12.Call Set "Fmt=%&Fmt:__LINE__=%lineNr%%"
13.Call Echo.%&Fmt:__FILE__=%~nx0%
14.EXIT /b

```

**:extractFileSection - extracts a section of file that is defined by a start and end mark**TOP  
2010-02-05

Description: call:[extractFileSection](#) StartMark EndMark FileName

Script:

```

1. :extractFileSection StartMark EndMark FileName -- extracts a section of file that is defined by a start and end mark
2. ::          -- [IN]      StartMark - start mark, use '...:S' mark to allow variable substitution
3. ::          -- [IN,OPT] EndMark   - optional end mark, default is first empty line
4. ::          -- [IN,OPT] FileName - optional source file, default is THIS file
5. :$created 20080219 :$changed 20100205 :$categories FileOperation
6. :$source https://www.dostips.com
7. SETLOCAL Disabledelayedexpansion
8. set "bmk=%~1"
9. set "emk=%~2"
10. set "src=%~3"
11. set "bExtr="
12. set "bSubs="
13. if "%src%"=="" set src=%~f0&           rem if no source file then assume THIS file
14. for /f "tokens=1,* delims=]" %%A in ('find /n /v "" "%src%"') do (
15.     if /i "%%B"=="%emk%" set "bExtr=&set "bSubs="
16.     if defined bExtr if defined bSubs (call echo.%%B) ELSE (echo.%%B)
17.     if /i "%%B"=="%bmk%" set "bExtr=Y"
18.     if /i "%%B"=="%bmk%:S" set "bExtr=Y&set "bSubs=Y"
19. )
20. EXIT /b

```

## **:ExtractFunction - extracts a function by label**

[TOP](#)  
2008-02-19

Description: call:[ExtractFunction](#) func

Script:

```

1. :ExtractFunction func -- extracts a function by label
2. ::          -- func [in] - name of the function to be extracted
3. :$created 20060101 :$changed 20080219
4. :$source https://www.dostips.com
5. SETLOCAL ENABLEDELAYEDEXPANSION
6. set func=%~f0
7. set /a b=2000000000
8. set /a e=2000000000
9. for /f "tokens=1,* delims=:" %%a in ('"findstr /n /b /c:"%~1" "%func%""') do set /a b=%%a
10. for /f "tokens=1,* delims=:" %%a in ('"findstr /n /b /c:"EXIT /b" "%func%""') do (
11.     if /i %b% LSS %%a if /i %%a LSS !e! set /a e=%%a
12. )
13. SETLOCAL DISABLEDELAYEDEXPANSION& rem --disabling preserves examination marks in %%b
14. for /f "tokens=1,* delims=[]" %%a in ('"find /n /v "" "%func%""') do (
15.     if /i %b% LEQ %%a if /i %%a LEQ %e% echo.%%b
16. )
17. EXIT /b

```

## **:ExtractSimple - extracts a marked section from batch file, substitutes variables during extraction**

[TOP](#)  
2009-11-13

Description: call:[ExtractSimple](#) StartMark EndMark

Script:

```

1. :ExtractSimple StartMark EndMark -- extracts a marked section from batch file, substitutes variables during extraction
2. ::          -- [IN]      StartMark - start mark
3. ::          -- [IN,OPT] EndMark   - optional end mark, default is first empty line
4. :$created 20091113 :$changed 20091113 :$categories FileOperation
5. :$source https://www.dostips.com
6. SETLOCAL

```

```

7. set "cpy="
8. for /f "tokens=1* delims=]" %%A in ('find /v /n "" "%~f0"') do (
9.     if "%~B"=="%~1" set "cpy=Y"
10.    if "%~B"=="%~2" set "cpy="
11.    if defined cpy call echo.%%B
12. )
13. EXIT /b

```

**:false - returns failure**
[TOP](#)  
2008-02-19
Description: [call:false](#)

Script:

```

1. :false -- returns failure
2. :$created 20060101 :$changed 20080219
3. :$source https://www.dostips.com
4. set=2>NUL
5. EXIT /b

```

**:Format - outputs columns of strings right or left aligned**
[TOP](#)  
2009-11-30
Description: [call:Format](#) fmt str1 str2 ...

Script:

```

1. :Format fmt str1 str2 ... -- outputs columns of strings right or left aligned
2. :: -- fmt [in] - format string specifying column width and alignment, i.e. "[-10][10][10]"
3. :$created 20060101 :$changed 20091130 :$categories Echo
4. :$source https://www.dostips.com
5. SETLOCAL
6. set "fmt=%~1"
7. set "line="
8. set "spac=          "
9. set "i=1"
10. for /f "tokens=1,2 delims=[ " %%a in ('echo.%fmt:]=&echo.%'') do (
11.     set /a i+=1
12.     call call set "subst=%%%~%%i%%spac%%%%~%%i%%"
13.     if %%b0 GEQ 0 (call set "subst=%&subst:~0,%&b%%"
14. ) ELSE (call set "subst=%&subst:~%%b%%")
15.     call set "const=%a"
16.     call set "line=%&1ine%%&cconst:~1%%&subst%%"
17. )
18. echo.%line%
19. EXIT /b

```

**:fprop - returns a file property**
[TOP](#)  
2008-02-19
Description: [call:fprop](#) filename prop ret

Script:

```

1. :fprop filename prop ret -- returns a file property
2. :: -- filename [in] - file name
3. :: -- prop      [in] - property, i.e.: d p n x a f s t z
4. :: -- ret       [out] - return value
5. :$created 20060101 :$changed 20080219 :$categories FileInfo
6. :$source https://www.dostips.com
7. SETLOCAL
8. set ret=
9. for %%a in ("%~1") do set ret=%~%~2a

```

```

10. ( ENDLOCAL & REM RETURN VALUES
11.     IF "%~3" NEQ "" (SET %~3=%ret%) ELSE echo.%ret%
12. )
13. EXIT /b

```

**:ftime - returns the file time in julian days**TOP  
2009-03-22Description: call:[ftime](#) JD filename attrDependencies: [jdate](#)

```

Script:   1. :ftime JD filename attr -- returns the file time in julian days
         2. ::-- JD      [out] - valref file time in julian days
         3. ::-- attr   [in,opt] - time field to be used, creation/last-access/last-write, see 'dir /?', i.e. /tc, /ta, /tw, default is /tw
         4. :$created 20060101 :$changed 20090322 :$categories DateAndTime
         5. :$source https://www.dostips.com
         6. SETLOCAL
         7. set file=%~2
         8. set attr=%~3
         9. if not defined attr (call:jdate JD "- %~t2"
        10. ) ELSE (for /f %%a in ('"dir %attr% /c "%file%"|findstr "^[0-9]"') do call:jdate JD "%%a")
        11. ( ENDLOCAL & REM RETURN VALUES
        12.     IF "%~1" NEQ "" (SET %~1=%JD%) ELSE (echo.%JD%)
        13. )
        14. EXIT /b

```

**:getColorCode - converts color text to color code**TOP  
2008-02-19Description: call:[getColorCode](#) col ret

```

Script:   1. :getColorCode col ret -- converts color text to color code
         2. ::-- col [in] - color text BackgroundForeground, i.e.: BlueLYellow for 1E
         3. ::-- ret [out] - return variable to return color code in
         4. :$created 20060101 :$changed 20080219 :$categories Color,Echo
         5. :$source https://www.dostips.com
         6. SETLOCAL
         7. set col=%~1
         8. set col=%col:Gray=8%
         9. set col=%col:LBlue=9%
        10. set col=%col:LGreen=A%
        11. set col=%col:LAqua=B%
        12. set col=%col:LRed=C%
        13. set col=%col:LPurple=D%
        14. set col=%col:LYellow=E%
        15. set col=%col:LWhite=F%
        16. set col=%col:Black=0%
        17. set col=%col:Blue=1%
        18. set col=%col:Green=2%
        19. set col=%col:Aqua=3%
        20. set col=%col:Red=4%
        21. set col=%col:Purple=5%
        22. set col=%col:Yellow=6%
        23. set col=%col:White=7%
        24. ENDLOCAL & IF "%~2" NEQ "" (SET %~2=%col%) ELSE (echo.%col%)
        25. EXIT /b

```

**:getDriveInfo - returns array of fsutil drive information**Description: call:[:getDriveInfo](#) var

```
Script:
1. :getDriveInfo var -- returns array of fsutil drive information
2. :: -- var [out] - return variable, as array
3. :$created 20060101 :$changed 20091115 :$categories DriveInfo,Array
4. :$source https://www.dostips.com
5. if "%~1" NEQ "" for /f "delims==" %%A in ('set %~1[ 2^>NUL') do @set %%A=
6. SETLOCAL ENABLEDELAYEDEXPANSION
7. set "data="
8. set /a n=-1
9. for /f "tokens=1 delims=" %%A in ('fsutil fsinfo drives^|find "\"') do (
10.    set "dr=%%A"
11.    set "dr=!dr:~-3,-1!"
12.    set /a n+=1
13.    set "data=!data!&set %~1[!n!].Drive=!dr!"
14.    for /f "tokens=1,* delims=- %%a in ('"fsutil fsinfo drivetype !dr!"') do set "data=!data!&set %~1[!n!].DriveType=%%b"
15.    for /f "tokens=1,* delims=:= %%a in ('"fsutil volume diskfree !dr!"') do (
16.      set "v= %%a"
17.      set "v=!v: =!"
18.      set "d= %%b"
19.      set "d=!d:~2!"
20.      set "data=!data!&set %~1[!n!].!v!=!d!"
21.    )
22.  )
23. set "data=rem.%data:)=^)%""
24. ( ENDLOCAL & REM RETURN VALUES
25.   %data%
26.   SET "%~1[#]=%n%"
27. )
28. EXIT /b
```

**:getDrives - This function has been replaced by :getDriveInfo**Description: call:[:getDrives](#) varDependencies: [:getDriveInfo](#)

```
Script:
1. :getDrives var -- This function has been replaced by :getDriveInfo
2. :$created 20060101 :$changed 20091115
3. :$source https://www.dostips.com
4. call:getDriveInfo %~1
5. EXIT /b
```

**:getFunctions - returns a comma separated list of all functions**Description: call:[:getFunctions](#) ret

```
Script:
1. :getFunctions ret -- returns a comma separated list of all functions
2. :: -- ret [out] - reference to return variable
3. :$created 20060101 :$changed 20080219
4. :$source https://www.dostips.com
5. SETLOCAL
6. set ret=
7. for /f %%a in ('"findstr ^:[a-z].*-- "%~f0" "') do call set ret=%ret%%a,
```

```

8. ( ENDLOCAL & REM RETURN VALUES
9.   IF "%~1" NEQ "" (SET %~1=%ret%) ELSE ECHO.%ret%
10. )
11. EXIT /b

```

**:getHostName - resolves IP address to computer name**TOP  
2008-02-19Description: call:[:getHostName](#) ip retDependencies: [:getIP](#)

Script:

```

1. :getHostName ip ret -- resolves IP address to computer name
2. ::          -- ip [in,opt] - ip, default is THIS computer's IP
3. ::          -- ret [out,opt] - computer name
4. :$created 20060101 :$changed 20080219 :$categories Network
5. :$source https://www.dostips.com
6. SETLOCAL
7. set ip=%~1
8. if "%ip%"=="" call:getIP "" ip
9. set name=
10. for /f "tokens=2" %%a in ('"ping /a /n 1 %ip%|find "Pinging" 2>NUL"') do set name=%%a
11. ENDLOCAL & IF "%~2" NEQ "" (SET %~2=%name%) ELSE (echo.%name%)
12. EXIT /b

```

**:getIP - return THIS computers IP address**TOP  
2008-02-19Description: call:[:getIP](#) host ret

Script:

```

1. :getIP host ret -- return THIS computers IP address
2. ::          -- host [in,opt] - host name, default is THIS computer
3. ::          -- ret [out,opt] - IP
4. :$created 20060101 :$changed 20080219 :$categories Network
5. :$source https://www.dostips.com
6. SETLOCAL
7. set host=%~1
8. set ip=
9. if "%host%"=="" ( for /f "tokens=2,* delims=:." %%a in ('"ipconfig|find "IP Address""') do set ip=%%b
10. ) ELSE ( for /f "tokens=2 delims=[]" %%a in ('"ping /a /n 1 %host%|find "Pinging" 2>NUL"') do set ip=%%a)
11. ENDLOCAL & IF "%~2" NEQ "" (SET %~2=%ip%) ELSE (echo.%ip%)
12. EXIT /b

```

**:getIPConfig - return IPCONFIG /ALL data in array variable**TOP  
2009-11-11Description: call:[:getIPConfig](#) arr

Script:

```

1. :getIPConfig arr -- return IPCONFIG /ALL data in array variable
2. ::          -- arr [out] - target array variable for IPCONFIG data
3. :$created 20091111 :$changed 20091111 :$categories Network,Array
4. :$source https://www.dostips.com
5. if "%~1" NEQ "" for /f "delims==" %%A in ('set %~1[ 2^>NUL') do @set "%%%A="
6. SETLOCAL ENABLEDELAYEDEXPANSION
7. set "data="
8. set /a n=0
9. for /f "tokens=1,* delims=:" %%A in ('ipconfig /all^|find ":"') do (
10.   set "v=%~A"
11.   if "!v:~0,8!" NEQ "      " (

```

```

12.     rem it's a new section
13.     set /a n+=1
14.     set "data=!data!&set %~1[!n!].DisplayName=%~A"
15. ) ELSE (
16.     set "v=!v:~8!"
17.     set "v=!v:..=!"
18.     set "v=!v: =!"
19.     set "x=%~B"
20.     set "data=!data!&set %~1[!n!].!v!=!x:~1!"
21. )
22.)
23.set "data=rem.%data:)=^)%" 
24.( ENDLOCAL & REM RETURN VALUES
25.    %data%
26.    SET "%~1[#]=%n%" 
27.)
28.EXIT /b

```

**:getNextInList - return next value in list**[TOP](#)  
2009-05-31Description: call:[:getNextInList](#) current list

```

Script: 1. :getNextInList current list -- return next value in list
2. ::                      -- current [in,out] - iterator variable holding the current value to be replaced with the next value
3. ::                      -- list      [in]      - choice list, must start and end with the delimiter
4. :$created 20091207 :$changed 20091207 :$categories List,Iterate,String
5. :$source https://www.dostips.com
6. Setlocal Enabledelayedexpansion
7. set "lst=%~2"
8. set "lst=%lst:~-1%%1st%%1st:~1%"
9. call set "v=%~1lst:*%1st:~-1%~1!%1st:~-1=%"
10. set "v=!v:%1st:~-1=%&rem.%!" 
11. ( ENDLOCAL & REM RETURN VALUES
12.     IF "%~1" NEQ "" (SET %~1=%v%) ELSE (echo.%new%)
13. )
14. GOTO:EOF
15.
16.
17. :count needle haystack ret -- counts the number of occurrences of needle in haystack
18. ::                      -- needle   [in] - string to find
19. ::                      -- haystack [in] - string to search in
20. ::                      -- ret      [out] - valref for return value
21. :$created 20060101 :$changed 20090531 :$categories String
22. :$source https://www.dostips.com
23. SETLOCAL
24. set "haystack=%~2"
25. set /a ret=0
26. FOR /f %%a in ('"(echo.%~1&echo.%%)|find /c /v "")') DO set /a ret=%a-1
27. ( ENDLOCAL & REM RETURN VALUES
28.     IF "%~3" NEQ "" (SET %~3=%ret%) ELSE ECHO.%ret%
29. )
30. EXIT /b

```

**:getRandomColor - returns a random color**[TOP](#)  
2008-02-19

Description: call:[:getRandomColor](#) ret

Script:

```

1. : getRandomColor ret -- returns a random color
2. :: -- ret [out,opt] - return variable to return color code in
3. :$created 20060101 :$changed 20080219 :$categories Color
4. :$source https://www.dostips.com
5. SETLOCAL
6. set HEX=0123456789ABCDEF
7. set /a r1=%random% %% 16
8. set /a r2=%random% %% 16
9. call set rndcolor=%HEX:~%r1%,1%%%HEX:~%r2%,1%
10. ENDLOCAL & IF "%~1" NEQ "" (SET %~1=%rndcolor%) ELSE (echo.%rndcolor%)
11. EXIT /b

```

## :GetRegKeys - returns all registry sub keys of a given registry key

[TOP](#)  
2009-11-23

Description: call:[:GetRegKeys](#) Key Data Type

Script:

```

1. :GetRegKeys Key Data Type -- returns all registry sub keys of a given registry key
2. :: -- Key [in] - registry key
3. :: -- Arr [out] - array of sub keys
4. :$created 20091123 :$changed 20091123 :$categories Registry, Array
5. :$source https://www.dostips.com
6. if "%~2" NEQ "" for /f "delims==" %%A in ('set %~2[ 2^>NUL') do set "%A="
7. set /a %~2[#]=-1
8. for /f "tokens=*" %%A in ('reg query "%~1"^(findstr /b /c:"HKEY_")' do (
9.     set /a %~2[#]+=1
10.    call set %~2[%~2[#]]=%A
11. )
12. EXIT /b

```

## :GetRegValue - returns a registry value

[TOP](#)  
2008-02-19

Description: call:[:GetRegValue](#) Key Value Data Type

Script:

```

1. :GetRegValue Key Value Data Type -- returns a registry value
2. :: -- Key [in] - registry key
3. :: -- Value [in] - registry value
4. :: -- Data [out] - return variable for Data
5. :: -- Type [out] - return variable for Type, i.e.: REG_SZ, REG_MULTI_SZ, REG_DWORD_BIG_ENDIAN, REG_DWORD, REG_BINARY, REG_DWORD_LITTLE_ENDIAN, REG_NONE, REG_EXPAND_SZ
6. :$created 20060101 :$changed 20080219 :$categories Registry
7. :$source https://www.dostips.com
8. SETLOCAL ENABLEDELAYEDEXPANSION
9. set Key=%~1
10. set Val=%~2
11. if "%Val%"=="" (set v=/ve) ELSE set v=/v "%Val%"
12. set Data=
13. set Type=
14. for /f "tokens=2,* delims= " %%a in ('reg query "%Key%" %v%|findstr /b "....%match%"') do (
15.     set Type=%%a
16.     set Data=%%b
17. )
18. ( ENDLOCAL & REM RETURN VALUES
19.     IF "%~3" NEQ "" (SET %~3=%Data%) ELSE echo.%Data%
20.     IF "%~4" NEQ "" (SET %~4=%Type%)

```

```
21.)  
22.EXIT /b
```

**:GetRegValues - returns all registry values of a given registry key**TOP  
2009-11-23Description: call:[:GetRegValues](#) Key Data Type

Script:

```
1. :GetRegValues Key Data Type -- returns all registry values of a given registry key
2. ::          -- Key [in] - registry key
3. ::          -- Stc [out] - struct of registry values
4. :$created 20091123 :$changed 20091123 :$categories Registry, Array
5. :$source https://www.dostips.com
6. if "%~2" NEQ "" for /f "delims==" %%A in ('"set %~2. 2>NUL"') do set "%A="
7. for /f "tokens=1,2,* skip=4 delims= " %%A in ('reg query "%~1"') do (
8.     for /f "tokens=* %%X in ("%%A") do ( rem this removes the leading tab
9.         set "%~2.%~X=%C"
10.    )
11. )
12.EXIT /b
```

**:getServices - returns array of service information**TOP  
2009-11-13Description: call:[:getServices](#) var

Script:

```
1. :getServices var -- returns array of service information
2. ::          -- var [out] - return variable, as array
3. :$created 20060101 :$changed 20091113 :$categories ServiceControl,Array
4. :$source https://www.dostips.com
5. set /a %~1[%#]=-1
6. for /f "tokens=* %%x in ('sc queryex state= all type= service') do (
7.     for /f "tokens=1,* delims= " %%a in ("%%x") do (
8.         if "%%a"=="SERVICE_NAME" set /a %~1[%#]+=1
9.         call set %~1[%~1[%#]]%%.%%a=%%b
10.    )
11. )
12.EXIT /b
```

**:getVarsByAttr - returns a comma separated list of variables based on an attribute name**TOP  
2008-02-19Description: call:[:getVarsByAttr](#) attr ret

Script:

```
1. :getVarsByAttr attr ret -- returns a comma separated list of variables based on an attribute name
2. ::          -- attr [in] - attribute name
3. ::          -- ret [out] - reference to return variable
4. :$created 20060101 :$changed 20080219
5. :$source https://www.dostips.com
6. SETLOCAL
7. set varattr=%~1
8. set value=
9. FOR /F "usebackq tokens=1,2 delims=.=." %%a IN (`set`) DO if "%%a"=="%varattr%" (set value=!value!%%b,)
10. ( ENDLOCAL & REM RETURN VALUES
11.     IF "%~2" NEQ "" SET %~2=%value%
12. )
13.EXIT /b
```

**:htmlhelp - dumps html help to console**Description: call:[:htmlhelp](#)Dependencies: [:ExtractFunction](#)

Script:

```

1. :htmlhelp -- dumps html help to console
2. :$created 20060101 :$changed 20080219
3. :$source https://www.dostips.com
4. SETLOCAL ENABLEEXTENSIONS
5. SETLOCAL ENABLEDELAYEDEXPANSION
6. rem echo.^<html^>^<body^>
7. echo.^<center^>
8. echo.^<table border=0^>
9. for /f "tokens=1,* delims=-" %%b in ('"findstr ^^:[^:]*-- "%~f0" "') do (
10.   for /f "tokens=1,* delims= " %%m in ("%%b") do (
11.     set func=%%m
12.     set args=%%n
13.   )
14.   set sign=%%b
15.   set desc=%%c
16.   echo.^<tr^>^<td^>^<a href="#"_!func:~1!^>!func!^</a^>^</td^>^<td^> !desc!^</td^>^</tr^>
17. )
18. echo.^</table^>
19. echo.^</center^>
20.
21. for /f "tokens=*" %%a in ('"findstr ^^:.*-- ^:^$ "%~f0" "') do (
22.   set line=%%a
23.   if "!line:~1,1!"==":" (           rem process function arguments section
24.     for /f "tokens=1,2,* delims=-[] " %%b in ("!line:*--!=") do (
25.       set pname=%%b
26.       set ptype=%%c
27.       set pdesc=%%d
28.     )
29.     echo.^<tr^>
30.     echo.^<td^>!pname!^</td^>
31.     echo.^<td^>!ptype!^</td^>
32.     echo.^<td^>!pdesc!^</td^>
33.     echo.^</tr^>
34.   ) ELSE if "!line:~1,1!"=="$" (      rem process the end section
35.     if "!func!" NEQ "" (
36.       set depe=
37.       for /f "tokens=*" %%F in ('%~f0 :ExtractFunction !func!|findstr /i "call\:[a-z]"') do (
38.         set line=%%F"
39.         set line=!line:*call:=!
40.         for /f "delims=&|>< " %%G in (!line!) do set depe=!depe!%%G &set line=%%G
41.         rem echo.!func! - !line!>&2
42.       )
43.       rem echo.!func! >&2
44.       rem echo.^<pre>&call:ExtractFunction !func!&echo.^</pre>
45.       echo.^</table^>
46.       echo.^<p^>Dependencies:
47.       for %%F in (!depe!) do (echo.^<a href="#"_%%F'^>:%%F^</a^> )
48.       if not defined depe echo.none
49.       echo.^</p^>
50.       set func=&rem this makes sure we process the END only once

```

```

51.      )
52. ) ELSE (           rem process function title section
53.   for /f "tokens=1,* delims=-" %%b in ("!line!") do (
54.     for /f "tokens=1,* delims= " %%m in ("%%b") do (
55.       set func=%%m
56.       set args=%%n
57.     )
58.     TITLE processing !func!
59.     set sign=%%b
60.     set desc=%%c
61.   )
62.   echo.^<br/^>
63.   echo.^<H2>^<a name='_!func:~1!'^>!func! -- !desc!^</a>^</H2>
64.   echo.^<span style='float:right;'> ^<a href="#!func:~1!'^>View Script^</a>^&nbsp;^&nbsp;^</span>
65.   echo.^<p>Call ^<b>^<a href="#!func:~1!'^>!func!^</a>^</b> ^<i>!args!^</i>^</p>
66.   echo.^<table border=1>^<tr>^<td>Parameter^</td>^<td>Type^</td>^<td>Desc^</td>^</tr>
67. )
68.)
69. TITLE processing done
70. rem echo.^</body>^</html>
71. EXIT /b

```

**:Init - initializes the environment for this command library**TOP  
2008-02-19Description: call:[:Init](#)

```

Script: 1. :Init -- initializes the environment for this command library
2. :$created 20060101 :$changed 20080219
3. :$source https://www.dostips.com
4. set ".=%~f0"          & rem set full name, i.e. type %%
5. set ":=call %%."      & rem set cl call prefix, i.e. %::func
6. set "?=&&echo.TRUE|!echo.FALSE" & rem set true/false check, i.e. %::true %?
7. set "*=%:::set"        & rem set array setter for console output, i.e. %% d=dir /b
8.
9. if not exist "%temp%" set "temp=%tmp%"
10. if not exist "%temp%" set "temp=c:\temp"
11. if not exist "%temp%" md "%temp%"
12. EXIT /b

```

**:initProgress - initializes an internal progress counter and display the progress in percent**TOP  
2008-03-27Description: call:[:initProgress](#) max formatDependencies: [:doProgress](#)

```

Script: 1. :initProgress max format -- initializes an internal progress counter and display the progress in percent
2. ::          -- max    [in]    - progress counter maximum, equal to 100 percent
3. ::          -- format [in,opt] - title string formatter, default is '[P] completed.'
4. :$created 20060101 :$changed 20080327 :$categories Progress
5. :$source https://www.dostips.com
6. set /a "ProgressCnt=-1"
7. set /a "ProgressMax=%~1"
8. set "ProgressFormat=%~2"
9. if not defined ProgressFormat set "ProgressFormat=[PPPP]"
10. set "ProgressFormat=%ProgressFormat:[PPP]=[P] completed.%"

```

```
11. call:doProgress
12. EXIT /b
```

**:initVarsByAttr - restores the values of the persistent variables to defaults**
[TOP](#)  
2008-02-19
Description: call:[:initVarsByAttr](#)Dependencies: [:getVarsByAttr](#)

Script:

```
1. :initVarsByAttr -- restores the values of the persistent variables to defaults
2. :$created 20060101 :$changed 20080219 :$categories Persistency
3. :$source https://www.dostips.com
4. SETLOCAL ENABLEDELAYEDEXPANSION
5. call:getVarsByAttr persist vars
6. ( ENDLOCAL
7.   for %%a in (%vars%) do SET %%a=!persist.%%a!& rem AA=persist.AA
8. )
9. EXIT /b
```

**:IsFileOpen - succeeds if file in use, fails otherwise**
[TOP](#)  
2008-02-19
Description: call:[:IsFileOpen](#) filename

Script:

```
1. :IsFileOpen filename -- succeeds if file in use, fails otherwise
2. :: -- filename [in] - name of file to be tested
3. :$created 20060101 :$changed 20080219 :$categories FileOperation
4. :$source https://www.dostips.com
5. echo.N|copy /y NUL "%~1">>NUL
6. EXIT /b
```

**:IsRegKey - succeeds if Key exists**
[TOP](#)  
2008-02-19
Description: call:[:IsRegKey](#) Key

Script:

```
1. :IsRegKey Key -- succeeds if Key exists
2. :: -- Key [in] - registry key
3. :$created 20060101 :$changed 20080219 :$categories Registry
4. :$source https://www.dostips.com
5. reg query "%~1">>NUL 2>NUL
6. EXIT /b
```

**:IsRegValue - succeeds if Key and Value exists**
[TOP](#)  
2008-02-19
Description: call:[:IsRegValue](#) Key Value

Script:

```
1. :IsRegValue Key Value -- succeeds if Key and Value exists
2. :: -- Key [in] - registry key
3. :: -- Value [in] - registry value, return default value if omitted
4. :$created 20060101 :$changed 20080219 :$categories Registry
5. :$source https://www.dostips.com
6. SETLOCAL
7. if "%~2""="" (set v=/ve) ELSE set v=/v "%~2"
8. reg query "%~1" %v%>NUL 2>NUL
9. EXIT /b
```

**:IsServiceRunning - returns success if service is running, otherwise failure**
[TOP](#)  
2008-02-19

Description: call:[IsServiceRunning](#) service

Script:

```

1. :IsServiceRunning service -- returns success if service is running, otherwise failure
2. ::                                -- service [in] - name of the service to check
3. :$created 20060101 :$changed 20080219 :$categories ServiceControl
4. :$source https://www.dostips.com
5. SETLOCAL ENABLEDELAYEDEXPANSION
6. sc query "%~1" | findstr "STATE.*.*4.*RUNNING">>NUL
7. EXIT /b

```

## **:jdate - converts a date string to julian day number with respect to regional date format**

[TOP](#)  
2009-03-28

Description: call:[jdate](#) JD DateStr

Script:

```

1. :jdate JD DateStr -- converts a date string to julian day number with respect to regional date format
2. ::                                -- JD      [out,opt] - julian days
3. ::                                -- DateStr [in,opt] - date string, e.g. "03/31/2006" or "Fri 03/31/2006" or "31.3.2006"
4. :$reference https://groups.google.com/group/alt.msdos.batch.nt/browse_frm/thread/a0c34d593e782e94/50ed3430b6446af8#50ed3430b6446af8
5. :$created 20060101 :$changed 20090328 :$categories DateAndTime
6. :$source https://www.dostips.com
7. SETLOCAL
8. set DateStr=%~2&if "%~2""="" set DateStr=%date%
9. for /f "skip=1 tokens=2-4 delims=(-)" %%a in ('echo.%date%') do (
10.    for /f "tokens=1-3 delims=/-. " %%A in ("%DateStr:* =%") do (
11.      set %%a=%%A&set %%b=%%B&set %%c=%%C)
12. set /a "yy=10000%yy% %%10000,mm=100%mm% %% 100,dd=100%dd% %% 100"
13. if %yy% LSS 100 set /a yy+=2000 &rem Adds 2000 to two digit years
14. set /a JD=dd-32075+1461*(yy+4800+(mm-14)/12)/4+367*(mm-2-(mm-14)/12*12)/12-3*((yy+4900+(mm-14)/12)/100)/4
15. ENDLOCAL & IF "%~1" NEQ "" (SET %~1=%JD%) ELSE (echo.%JD%)
16. EXIT /b

```

## **:jdate2date - converts julian days to gregorian date format**

[TOP](#)  
2008-02-19

Description: call:[jdate2date](#) JD YYYY MM DD

Script:

```

1. :jdate2date JD YYYY MM DD -- converts julian days to gregorian date format
2. ::                                -- JD      [in] - julian days
3. ::                                -- YYYY [out] - gregorian year, i.e. 2006
4. ::                                -- MM      [out] - gregorian month, i.e. 12 for december
5. ::                                -- DD      [out] - gregorian day, i.e. 31
6. :$reference https://aa.usno.navy.mil/faq/docs/JD_Formula.html
7. :$created 20060101 :$changed 20080219 :$categories DateAndTime
8. :$source https://www.dostips.com
9. SETLOCAL ENABLEDELAYEDEXPANSION
10. set /a L= %~1+68569,      N= 4*L/146097, L= L-(146097*N+3)/4, I= 4000*(L+1)/1461001
11. set /a L= L-1461*I/4+31, J= 80*L/2447, K= L-2447*I/80,      L= J/11
12. set /a J= J+2-12*L,      I= 100*(N-49)+I+L
13. set /a YYYY= I, MM=100+J, DD=100+K
14. set MM=%MM:~-2%
15. set DD=%DD:~-2%
16. ( ENDLOCAL & REM RETURN VALUES
17.   IF "%~2" NEQ "" (SET %~2=%YYYY%) ELSE echo.%YYYY%
18.   IF "%~3" NEQ "" (SET %~3=%MM%) ELSE echo.%MM%
19.   IF "%~4" NEQ "" (SET %~4=%DD%) ELSE echo.%DD%

```

```
20.)
21. EXIT /b
```

**:kBytesFree - returns the free space of a drive in kilobytes**
[TOP](#)  
2008-02-19
Description: call:[:kBytesFree](#) drive ret

Script:

```
1. :kBytesFree drive ret -- returns the free space of a drive in kilobytes
2. ::                      -- drive [in]      - drive letter, default is c:
3. ::                      -- ret   [out,opt] - varref returns number of bytes free
4. :$created 20060101 :$changed 20080219 :$categories FileInfo
5. :$source https://www.dostips.com
6. SETLOCAL
7. SET "drive=%~1"
8. IF "%drive%"=="." SET "drive=C:"
9. FOR /f "tokens=3 delims=" %%a in ('dir %drive%|find "bytes free"') do set free=%%a
10. SET "free=%free:,=%"      &rem eleminate commas in number
11. SET "free=%free:.=%"      &rem eleminate dots in number
12. SET "free=%free:~0,-3%"  &rem divide by 1000 by cutting off the last 3 digits
13. ( ENDLOCAL & REM RETURN VALUES
14.     IF "%~2" NEQ "" (SET "%~2=%free%") ELSE echo.%free%
15. )
16. EXIT /b
```

**:l2a - converts a list to an array**
[TOP](#)  
2008-02-19
Description: call:[:l2a](#) list arr

Script:

```
1. :l2a list arr -- converts a list to an array
2. ::                      -- list [in]      - list name
3. ::                      -- arr   [in]      - array name
4. :$created 20060101 :$changed 20080219 :$categories Array,List
5. :$source https://www.dostips.com
6. call set l=%~1%
7. set /a i=0
8. for /f "delims=" %%a in ('echo.%l:,=&echo.%') do (
9.     call set %~2[%~1%%]=%%a
10.    set /a i+=1
11. )
12. set %~2=%i%
13. EXIT /b
```

**:loadPersistentVars - loads the values of the persistent variables**
[TOP](#)  
2008-02-19
Description: call:[:loadPersistentVars](#) filenameDependencies: [:initVarsByAttr](#)

Script:

```
1. :loadPersistentVars filename -- loads the values of the persistent variables
2. ::                      -- filename [in]      - batch file name to restore from
3. :$created 20060101 :$changed 20080219 :$categories Persistence
4. :$source https://www.dostips.com
5. call:initVarsByAttr
6. if exist "%FilePersist%" call "%FilePersist%"
7. EXIT /b
```

## :loadRegVars - loads the values of the persistent variables

Description: call:[:loadRegVars](#) key

Dependencies: [:getVarsByAttr](#), [:GetRegValue](#)

Script:

```

1. :loadRegVars key -- loads the values of the persistent variables
2. ::          -- key [in] - registry key to restore from
3. :$created 20060101 :$changed 20080219 :$categories Registry,Persistency
4. :$source https://www.dostips.com
5. SETLOCAL ENABLEDELAYEDEXPANSION
6. call:getVarsByAttr reg vars
7. ( ENDLOCAL
8.   for %%a in (%vars%) do SET %%a=!reg.%%a!& rem AA=reg.AA
9.   for %%a in (%vars%) do call:GetRegValue "%~1" "%~a" %%a
10. )
11. EXIT /b

```

## :lookup - returns the value associated with a key in a map of key-value pairs

Description: call:[:lookup](#) key map ret

Script:

```

1. :lookup key map ret -- returns the value associated with a key in a map of key-value pairs
2. ::          -- key [in] - key to lookup
3. ::          -- map [in] - map with key-value pairs, i.e. key1-val1;key2-val2;key3-val3
4. ::          -- ret [out] - valref for return value
5. :$created 20060101 :$changed 20080219 :$categories Lookup
6. :$source https://www.dostips.com
7. SETLOCAL
8. SET map=%~2
9. CALL SET ret=%map:*%~1-=%
10. SET ret=%ret:;=&:%
11. IF "%ret%" NEQ "%ret:--%" set ret=
12. ( ENDLOCAL & REM RETURN VALUES
13.   IF "%~3" NEQ "" (SET %~3=%ret%) ELSE ECHO.%ret%
14. )
15. EXIT /b

```

## :lTrim - strips white spaces (or other characters) from the beginning of a string

Description: call:[:lTrim](#) string char

Script:

```

1. :lTrim string char -- strips white spaces (or other characters) from the beginning of a string
2. ::          -- string [in,out] - string variable to be trimmed
3. ::          -- char [in,opt] - character to be trimmed, default is space
4. :$created 20060101 :$changed 20080227 :$categories StringManipulation
5. :$source https://www.dostips.com
6. SETLOCAL ENABLEDELAYEDEXPANSION
7. call set "string=%~1%"
8. set "charlist=%~2"
9. if not defined charlist set "charlist="
10. for /f "tokens=* delims=%charlist%" %%a in ("%string%") do set "string=%%a"
11. ( ENDLOCAL & REM RETURN VALUES
12.   IF "%~1" NEQ "" SET "%~1=%string%"
13. )
14. EXIT /b

```

**:MakeAbsolute - makes a file name absolute considering a base path**Description: call:[:MakeAbsolute](#) file base

```

Script: 1. :MakeAbsolute file base -- makes a file name absolute considering a base path
        2. :: -- file [in,out] - variable with file name to be converted, or file name itself for result in stdout
        3. :: -- base [in,opt] - base path, leave blank for current directory
        4. :$created 20060101 :$changed 20080219 :$categories Path
        5. :$source https://www.dostips.com
        6. SETLOCAL ENABLEDELAYEDEXPANSION
        7. set "src=%~1"
        8. if defined %1 set "src=!%~1!"
        9. set "bas=%~2"
       10. if not defined bas set "bas=%cd%"
       11. for /f "tokens=*" %%a in ("%bas%.%src%") do set "src=%%~fa"
       12. ( ENDLOCAL & REM RETURN VALUES
       13.   IF defined %1 (SET %~1=%src%) ELSE ECHO.%src%
       14. )
       15. EXIT /b
  
```

**:MakeRelative - makes a file name relative to a base path**Description: call:[:MakeRelative](#) file base

```

Script: 1. :MakeRelative file base -- makes a file name relative to a base path
        2. :: -- file [in,out] - variable with file name to be converted, or file name itself for result in stdout
        3. :: -- base [in,opt] - base path, leave blank for current directory
        4. :$created 20060101 :$changed 20080219 :$categories Path
        5. :$source https://www.dostips.com
        6. SETLOCAL ENABLEDELAYEDEXPANSION
        7. set src=%~1
        8. if defined %1 set src=!%~1!
        9. set bas=%~2
       10. if not defined bas set bas=%cd%
       11. for /f "tokens=*" %%a in ("%src%") do set src=%%~fa
       12. for /f "tokens=*" %%a in ("%bas%") do set bas=%%~fa
       13. set mat=&rem variable to store matching part of the name
       14. set upp=&rem variable to reference a parent
       15. for /f "tokens=*" %%a in ('echo.%bas:\=\^&echo.%') do (
       16.   set sub=!sub!%%a\
       17.   call set tmp=%src:!sub!=%
       18.   if "!tmp!" NEQ "!src!" (set mat=!sub!)ELSE (set upp=!upp...\)
       19. )
       20. set src=%upp%!src:mat%=
       21. ( ENDLOCAL & REM RETURN VALUES
       22.   IF defined %1 (SET %~1=%src%) ELSE ECHO.%src%
       23. )
       24. EXIT /b
  
```

**:NetworkDeviceGuid2Name - gets a Network Device Name from its corresponding GUID**Description: call:[:NetworkDeviceGuid2Name](#) guid name

```

Script: 1. :NetworkDeviceGuid2Name guid name -- gets a Network Device Name from its corresponding GUID
        2. :: -- guid [in]      - Network Device GUID
  
```

```

3. :: -- name [out,opt] - valref to return Name in
4. :$created 20060101 :$changed 20080219
5. :$source https://www.dostips.com
6. SETLOCAL
7. set guid=%~1
8. set key=HKLMSYSTEM\CurrentControlSet\Control\Network\{4D36E972-E325-11CE-BFC1-08002BE10318}
9. set ret=
10. for /f "tokens=*" %%k in ('"reg query "%key%" /s|find /I "}Connection" "') do (
11.     for /f "tokens=2,*" %%m in ('"reg query "%k" /v "Name"|findstr /b "....Name" "') do (
12.         for /f "tokens=7 delims=\\" %%g in ("%%k") do (
13.             if "%%g"=="%guid%" set ret=%%m
14.         ))
15. ENDLOCAL&if "%~2" NEQ "" (SET %~2=%ret%) ELSE ECHO.%ret%
16. EXIT /b

```

**:NetworkDeviceName2Guid - gets a Network Device GUID from its corresponding Name**TOP  
2008-02-19Description: call:[:NetworkDeviceName2Guid](#) name guid

Script:

```

1. :NetworkDeviceName2Guid name guid -- gets a Network Device GUID from its corresponding Name
2. :: -- name [in]      - Network Device Name
3. :: -- guid [out,opt] - valref to return GUID in
4. :$created 20060101 :$changed 20080219
5. :$source https://www.dostips.com
6. SETLOCAL
7. set name=%~1
8. set key=HKLMSYSTEM\CurrentControlSet\Control\Network\{4D36E972-E325-11CE-BFC1-08002BE10318}
9. set ret=
10. for /f "tokens=*" %%k in ('"reg query "%key%" /s|find /I "}Connection" "') do (
11.     for /f "tokens=2,*" %%m in ('"reg query "%k" /v "Name"|findstr /b "....Name" "') do (
12.         for /f "tokens=7 delims=\\" %%g in ("%%k") do (
13.             if "%%n"=="%name%" set ret=%%g
14.         ))
15. ENDLOCAL&if "%~2" NEQ "" (SET %~2=%ret%) ELSE ECHO.%ret%
16. EXIT /b

```

**:parse - parses a string matching a formatter**TOP  
2008-02-19Description: call:[:parse](#) fmt str

Script:

```

1. :parse fmt str -- parses a string matching a formatter
2. :: -- fmt [in] - formatter containing return variable names and delimiters
3. :: -- str [in] - string to be parsed
4. :$created 20060101 :$changed 20080219
5. :$source https://www.dostips.com
6. SETLOCAL
7. echo...under construction
8. EXIT /b

```

**:pwd - shows a password dialog box**TOP  
2008-02-26Description: call:[:pwd](#) var title color

Script:

```

1. :pwd var title color -- shows a password dialog box
2. :: -- var   [in]      - return variable

```

```

3. ::          -- title [in,opt] - dialog title, default is "Password"
4. ::          -- color [in,opt] - color default is AB (Light Green on Light Aqua)
5. :$created 20060101 :$changed 20080226 :$categories Input,Password
6. :$source https://www.dostips.com
7. SETLOCAL
8. set "tit=%~2"
9. set "col=%~3"
10. set "pwd="
11. if not defined col set "col=EF"
12. if not defined tit set "tit=Password:"
13. set "f=%temp%\%~n%0.tmp~0.tmp"
14. start "%tit%" /wait cmd /c "mode con cols=24 lines=1&color %col%&set /p "in=&call echo.%~1n%>%f%"
15. for /f "usebackq tokens=*" %%a in ("%f%") do set "pwd=%%a"
16. del /q "%f%"
17. ENDLOCAL&if "%~1" NEQ "" (SET %~1=%pwd%) ELSE ECHO.%txt%
18. EXIT /b

```

**:regedit - opens regedit at predefined location**TOP  
2008-03-14Description: call:[:regedit](#) key

Script:

```

1. :regedit key -- opens regedit at predefined location
2. ::          -- [IN] key - registry key to be shown after opening regedit
3. :$created 20080314 :$changed 20080314 :$categories Regedit,Registry
4. :$source https://www.dostips.com
5. SETLOCAL
6. REG ADD "HKCU\Software\Microsoft\Windows\CurrentVersion\Applets\Regedit" /v "LastKey" /f /d "%~1">NUL
7. start /b regedit
8. EXIT /b

```

**:removeArr - removes an array**TOP  
2008-02-19Description: call:[:removeArr](#) arr

Script:

```

1. :removeArr arr -- removes an array
2. ::          -- arr [in] - array name
3. :$created 20060101 :$changed 20080219 :$categories Array
4. :$source https://www.dostips.com
5. for /f "delims==" %%a in ('"set %~1[ 2>NUL"') do set "%%a="
6. EXIT /b

```

**:rTrim - strips white spaces (or other characters) from the end of a string**TOP  
2008-02-19Description: call:[:rTrim](#) string char max

Script:

```

1. :rTrim string char max -- strips white spaces (or other characters) from the end of a string
2. ::          -- string [in,out] - string variable to be trimmed
3. ::          -- char   [in,opt] - character to be trimmed, default is space
4. ::          -- max    [in,opt] - maximum number of characters to be trimmed from the end, default is 32
5. :$created 20060101 :$changed 20080219 :$categories StringManipulation
6. :$source https://www.dostips.com
7. SETLOCAL ENABLEDELAYEDEXPANSION
8. call set string=%~1%
9. set char=%~2
10. set max=%~3

```

```

11. if "%char%"=="" set char= &rem one space
12. if "%max%"=="" set max=32
13. for /l %%a in (1,1,%max%) do if "!string:~1!"=="%char%" set string=!string:~0,-1!
14. ( ENDLOCAL & REM RETURN VALUES
15.   IF "%~1" NEQ "" SET %~1=%string%
16. )
17. EXIT /b

```

**:savePersistentVars - save values of persistent variables into a file**TOP  
2008-02-19Description: call:[:savePersistentVars](#) filenameDependencies: [:getVarsByAttr](#)

```

1. :savePersistentVars filename -- save values of persistent variables into a file
2. ::          -- filename [in] - file name used for storage
3. :$created 20060101 :$changed 20080219 :$categories Persistence
4. :$source https://www.dostips.com
5. SETLOCAL ENABLEDELAYEDEXPANSION
6. type nul>"%~1"
7. call:getVarsByAttr persist vars
8. for %%a in (%vars%) do (echo.SET %%a=!%%a!>>"%~1")
9. EXIT /b

```

**:saveRegVars - save values of persistent variables into a file**TOP  
2008-02-19Description: call:[:saveRegVars](#) keyDependencies: [:getVarsByAttr](#), [:SetRegValue](#)

```

1. :saveRegVars key -- save values of persistent variables into a file
2. ::          -- key [in] - registry key used for storage
3. :$created 20060101 :$changed 20080219 :$categories Registry,Persistence
4. :$source https://www.dostips.com
5. SETLOCAL ENABLEDELAYEDEXPANSION
6. call:getVarsByAttr reg vars
7. for %%a in (%vars%) do call:SetRegValue "%~1" "%%a" "!%%a!"
8. EXIT /b

```

**:set - sets var[] to output of command / under construction**TOP  
2008-02-19Description: call:[:set](#) var=commandDependencies: [:removeArr](#)

```

1. :set var=command -- sets var[] to output of command / under construction
2. ::          -- var      [out] varref of target variable
3. ::          -- command [in]  command to be parsed, i.e. ipconfig /all
4. :$created 20060101 :$changed 20080219
5. :$source https://www.dostips.com
6. for /f "tokens=1,* delims==" %%x in ("*") do (
7.   call:removeArr "%%x"
8.   if "%%x" NEQ "" (
9.     for /f "tokens=1,2 delims=]" %%a in ('"%%y|find /n /v """) do (
10.       set "%%x%%a]=%%b"
11.       set "%%x[#]=%%a"
12.     )&call set "%%x[#]=%%%%x[#]:~1%%"
13.   )

```

```

14.     call set "%~x=%~%~x[1]%""
15. )
16. EXIT /b

```

**:SetRegValue - sets a registry value**
[TOP](#)  
2008-02-19

Description: call:[:SetRegValue](#) Key Value Data Type

Script:

```

1. :SetRegValue Key Value Data Type -- sets a registry value
2. ::          -- Key      [in]  - registry key
3. ::          -- Value    [in]  - registry value
4. ::          -- Data     [in]  - (optional) data
5. ::          -- Type     [in]  - (optional) data type, i.e.: REG_SZ, REG_MULTI_SZ, REG_DWORD_BIG_ENDIAN, REG_DWORD, REG_BINARY, REG_DWORD_LITTLE_ENDIAN, REG_NONE, REG_EXPAND_SZ
6. :$created 20060101 :$changed 20080219 :$categories Registry
7. :$source https://www.dostips.com
8. SETLOCAL
9. set Key=%~1
10. set Val=%~2
11. set Data=%~3
12. set Type=%~4
13. if "%Val%" NEQ "" set Val=/v "%Val%"
14. if "%Data%" NEQ "" set Data=/d "%Data%"
15. if "%Type%" NEQ "" set Type=/t %Type%
16. reg add "%Key%" %Val% %Type% %Data% /f>NUL
17. EXIT /b

```

**:sleep - waits some seconds before returning**
[TOP](#)  
2008-02-19

Description: call:[:sleep](#) seconds

Script:

```

1. :sleep seconds -- waits some seconds before returning
2. ::          -- seconds [in]  - number of seconds to wait
3. :$created 20060101 :$changed 20080219
4. :$source https://www.dostips.com
5. FOR /1 %%a in (%~1,-1,1) do (ping -n 2 -w 1 127.0.0.1>NUL)
6. EXIT /b

```

**:StartsWith - Tests if a text starts with a given string**
[TOP](#)  
2008-03-20

Description: call:[:StartsWith](#) text string

Script:

```

1. :StartsWith text string -- Tests if a text starts with a given string
2. ::          -- [IN] text   - text to be searched
3. ::          -- [IN] string - string to be tested for
4. :$created 20080320 :$changed 20080320 :$categories StringOperation,Condition
5. :$source https://www.dostips.com
6. SETLOCAL
7. set "txt=%~1"
8. set "str=%~2"
9. if defined str call set "s=%str%%~txt:*%str%=%%"
10. if /i "%txt%" NEQ "%s%" set=2>NUL
11. EXIT /b

```

**:strLen - returns the length of a string**
[TOP](#)  
2010-11-16

Description: call:[strLen](#) string len

Script:

```

1. :strLen string len -- returns the length of a string
2. ::                      -- string [in] - variable name containing the string being measured for length
3. ::                      -- len   [out] - variable to be used to return the string length
4. :: Many thanks to 'sowgtsoi', but also 'jeb' and 'amel27' dosTips forum users helped making this short and efficient
5. :$created 20081122 :$changed 20101116 :$categories StringOperation
6. :$source https://www.dostips.com
7. (  SETLOCAL ENABLEDELAYEDEXPANSION
8.   set "str=A%~1!"&rem keep the A up front to ensure we get the length and not the upper bound
9.   rem it also avoids trouble in case of empty string
10.  set "len=0"
11.  for /L %%A in (12,-1,0) do (
12.    set /a "len|=1<<%%A"
13.    for %%B in (!len!) do if "!str:~%%B,1!"=="" set /a "len&=~1<<%%A"
14.  )
15. )
16. ( ENDLOCAL & REM RETURN VALUES
17.  IF "%~2" NEQ "" SET /a %~2=%len%
18. )
19. EXIT /b

```

## :substitute - substitutes a string in a text file

[TOP](#)  
2010-11-22

Description: call:[substitute](#) OldStr NewStr File

Script:

```

1. :substitute OldStr NewStr File -- substitutes a string in a text file
2. ::                      -- OldStr [in] - string to be replaced
3. ::                      -- NewStr [in] - string to replace with
4. ::                      -- File   [in] - file to be parsed
5. :$created 20060101 :$changed 20101122 :$categories FileManipulation
6. :$source https://www.dostips.com
7. SETLOCAL DISABLEDELAYEDEXPANSION
8. for /f "tokens=1,* delims=]" %%A in ('"type %3|find /n /v """) do (
9.   set "line=%%B"
10.  if defined line (
11.    call set "line=echo.%~1=%%B%~2%%"
12.    for /f "delims=" %%X in ('"echo."%%1line%%%"') do %%~X
13.  ) ELSE echo.
14. )
15. EXIT /b

```

## :ToANSI - converts a file to ANSI

[TOP](#)  
2008-02-19

Description: call:[ToANSI](#) src dst

Script:

```

1. :ToANSI src dst -- converts a file to ANSI
2. ::                      -- src [in]      - source file name to be converted
3. ::                      -- trg [in,opt] - target file name, leave blank for in-place conversion
4. :$created 20060101 :$changed 20080219 :$categories Encoding
5. :$source https://www.dostips.com
6. SETLOCAL
7. set src=%~1
8. set trg=%~2
9. set tmp=%temp%.%~nx1

```

```

10. if "%trg%"==""
11. cmd /A /c type "%src%">>"%tmp%"
12. move /y "%tmp%" "%trg%"
13. EXIT /b

```

**:toCamelCase - converts a string to camel case**TOP  
2008-02-19Description: call:[:toCamelCase](#) str

```

Script: 1. :toCamelCase str -- converts a string to camel case
2. ::          -- str [in,out] - valref of string variable to be converted
3. :$created 20080219 :$changed 20080219 :$categories StringManipulation
4. :$source https://www.dostips.com
5. if not defined %~1 EXIT /b
6. REM make all lower case
7. for %%a in ("A=a" "B=b" "C=c" "D=d" "E=e" "F=f" "G=g" "H=h" "I=i"
8.         "J=j" "K=k" "L=l" "M=m" "N=n" "O=o" "P=p" "Q=q" "R=r"
9.         "S=s" "T=t" "U=u" "V=v" "W=w" "X=x" "Y=y" "Z=z"
10.        "Ä=ä" "Ö=ö" "Ü=ü") do (
11.    call set "%~1=%~1:%%~a%%"
12. )
13. call set "%~1= %%~1%"
14. REM make first character upper case
15. for %%a in (" a=A" " b=B" " c=C" " d=D" " e=E" " f=F" " g=G" " h=H" " i=I"
16.         " j=J" " k=K" " l=L" " m=M" " n=N" " o=O" " p=P" " q=Q" " r=R"
17.         " s=S" " t=T" " u=U" " v=V" " w=W" " x=X" " y=Y" " z=Z"
18.         " ä=Ä" " ö=ö" " ü=Ü") do (
19.    call set "%~1=%~1:%%~a%%"
20. )
21. call set "%~1=%~1: =%"
22. EXIT /b

```

**:toDec - convert a hexadecimal number to decimal**TOP  
2009-12-03Description: call:[:toDec](#) hex dec

```

Script: 1. :toDec hex dec -- convert a hexadecimal number to decimal
2. ::          -- hex [in]      - hexadecimal number to convert
3. ::          -- dec [out,opt] - variable to store the converted decimal number in
4. :$created 20091203 :$changed 20091203 :$categories Arithmetic,Encoding
5. :$source https://www.dostips.com
6. SETLOCAL
7. set /a dec=0x%~1
8. ( ENDLOCAL & REM RETURN VALUES
9.   IF "%~2" NEQ "" (SET %~2=%dec%) ELSE ECHO.%dec%
10. )
11. EXIT /b

```

**:toHex - convert a decimal number to hexadecimal, i.e. -20 to FFFFFFFEC or 26 to 0000001A**TOP  
2011-03-30Description: call:[:toHex](#) dec hex

```

Script: 1. :toHex dec hex -- convert a decimal number to hexadecimal, i.e. -20 to FFFFFFFEC or 26 to 0000001A
2. ::          -- dec [in]      - decimal number to convert
3. ::          -- hex [out,opt] - variable to store the converted hexadecimal number in

```

```

4. ::Thanks to 'dbenham' dostips forum users who inspired to improve this function
5. :$created 20091203 :$changed 20110330 :$categories Arithmetic,Encoding
6. :$source https://www.dostips.com
7. SETLOCAL ENABLEDELAYEDEXPANSION
8. set /a dec=%~1
9. set "hex="
10. set "map=0123456789ABCDEF"
11. for /L %%N in (1,1,8) do (
12.     set /a "d=dec&15,dec>>=4"
13.     for %%D in (!!) do set "hex=!map:~%%D,1!!hex!"
14. )
15. rem !!!! REMOVE LEADING ZEROS by activating the next line, e.g. will return 1A instead of 0000001A
16. rem for /f "tokens=* delims=0" %%A in ("%hex%") do set "hex=%%A"&if not defined hex set "hex=0"
17. ( ENDLOCAL & REM RETURN VALUES
18.     IF "%~2" NEQ "" (SET %~2=%hex%) ELSE ECHO.%hex%
19. )
20. EXIT /b

```

**:toLower - converts uppercase character to lowercase**[TOP](#)  
2008-02-19Description: call:[:toLower](#) str

Script:

```

1. :toLower str -- converts uppercase character to lowercase
2. ::          -- str [in,out] - valref of string variable to be converted
3. :$created 20060101 :$changed 20080219 :$categories StringManipulation
4. :$source https://www.dostips.com
5. if not defined %~1 EXIT /b
6. for %%a in ("A=a" "B=b" "C=c" "D=d" "E=e" "F=f" "G=g" "H=h" "I=i"
7.     "J=j" "K=k" "L=l" "M=m" "N=n" "O=o" "P=p" "Q=q" "R=r"
8.     "S=s" "T=t" "U=u" "V=v" "W=w" "X=x" "Y=y" "Z=z" "Äää"
9.     "Öö" "Üü") do (
10.     call set %~1=%~1;%~1:%%~a%
11. )
12. EXIT /b

```

**:ToUNICODE - converts a file to UNICODE**[TOP](#)  
2008-02-19Description: call:[:ToUNICODE](#) src dst

Script:

```

1. :ToUNICODE src dst -- converts a file to UNICODE
2. ::          -- src [in]      - source file name to be converted
3. ::          -- trg [in,opt] - target file name, leave blank for in-place conversion
4. :$created 20060101 :$changed 20080219 :$categories Encoding
5. :$source https://www.dostips.com
6. SETLOCAL
7. set src=%~1
8. set trg=%~2
9. set tmp=%temp%\%~nx1
10. if "%trg%"=="" set trg=%src%
11. cmd /U /c type "%src%">"%tmp%"
12. move /y "%tmp%" "%trg%"
13. EXIT /b

```

**:toUpper - converts lowercase character to uppercase**[TOP](#)  
2008-02-19

Description: call:[toUpper](#) str

Script:

```

1. :toUpper str -- converts lowercase character to uppercase
2. ::           -- str [in,out] - valref of string variable to be converted
3. :$created 20060101 :$changed 20080219 :$categories StringManipulation
4. :$source https://www.dostips.com
5. if not defined %~1 EXIT /b
6. for %%a in ("a=A" "b=B" "c=C" "d=D" "e=E" "f=F" "g=G" "h=H" "i=I"
7.     "j=J" "k=K" "l=L" "m=M" "n=N" "o=O" "p=P" "q=Q" "r=R"
8.     "s=S" "t=T" "u=U" "v=V" "w=W" "x=X" "y=Y" "z=Z" "ä=Ä"
9.     "ö=Ö" "ü=Ü") do (
10.    call set %~1=%~1:%%~a%
11. )
12. EXIT /b

```

## :Trim - strip white spaces (or other characters) from the beginning and end of a string

[TOP](#)  
2008-02-19

Description: call:[Trim](#) string char max

Dependencies: [:lTrim](#), [:rTrim](#)

Script:

```

1. :Trim string char max -- strip white spaces (or other characters) from the beginning and end of a string
2. ::           -- string [in,out] - string variable to be trimmed
3. ::           -- char   [in,opt] - character to be trimmed, default is space
4. ::           -- max    [in,opt] - maximum number of characters to be trimmed from the end, default is 32
5. :$created 20060101 :$changed 20080219 :$categories StringManipulation
6. :$source https://www.dostips.com
7. call:lTrim "%~1" "%~2"
8. call:rTrim "%~1" "%~2" "%~3"
9. EXIT /b

```

## :trimSpaces - trims spaces around string variable

[TOP](#)  
2008-02-23

Description: call:[trimSpaces](#) varref

Dependencies: [:trimSpaces2](#)

Script:

```

1. :trimSpaces varref -- trims spaces around string variable
2. ::           -- varref [in,out] - variable to be trimmed
3. :$created 20060101 :$changed 20080223 :$categories StringManipulation
4. :$source https://www.dostips.com
5. call call:trimSpaces2 %~1 %%:~1%
6. EXIT /b

```

## :trimSpaces2 - trims spaces around string and assigns result to variable

[TOP](#)  
2008-02-19

Description: call:[trimSpaces2](#) retval string

Script:

```

1. :trimSpaces2 retval string -- trims spaces around string and assigns result to variable
2. ::           -- retval [out] variable name to store the result in
3. ::           -- string [in]  string to trim, must not be in quotes
4. :$created 20060101 :$changed 20080219 :$categories StringManipulation
5. :$source https://www.dostips.com
6. for /f "tokens=1%" %%A in ("**) do set "%~A=%~B"
7. EXIT /b

```

## :true - returns success

[TOP](#)  
2008-02-19

Description: call:[true](#)

Script:

```
1. :true -- returns success
2. :$created 20060101 :$changed 20080219
3. :$source https://www.dostips.com
4. rem
5. EXIT /b
```

## :Unique - returns a unique string based on a date-time-stamp, YYYYMMDDhhmmsscc

[TOP](#)  
2008-02-19

Description: call:[Unique](#) ret

Script:

```
1. :Unique ret -- returns a unique string based on a date-time-stamp, YYYYMMDDhhmmsscc
2. ::          -- ret      [out,opt] - unique string
3. :$created 20060101 :$changed 20080219 :$categories StringOperation,DateAndTime
4. :$source https://www.dostips.com
5. SETLOCAL
6. for /f "skip=1 tokens=2-4 delims=(-)" %%a in ('echo.|date') do (
7.   for /f "tokens=1-3 delims=- " %%A in ("%date:* =%") do (
8.     set %%a=%%A&set %%b=%%B&set %%c=%%C)
9. set /a "yy=1000%yy% %%1000,mm=100%mm% %% 100,dd=100%dd% %% 100"
10. for /f "tokens=1-4 delims=:. " %%A in ("%time: =%") do @set UNIQUE=%yy%%mm%%dd%%A%%B%%C%%D
11. ENDLOCAL & IF "%~1" NEQ "" (SET %~1=%UNIQUE%) ELSE echo.%UNIQUE%
12. EXIT /b
```

## DOS Function Template - A function template with all features of a true DOS function

[TOP](#)  
2008-01-01

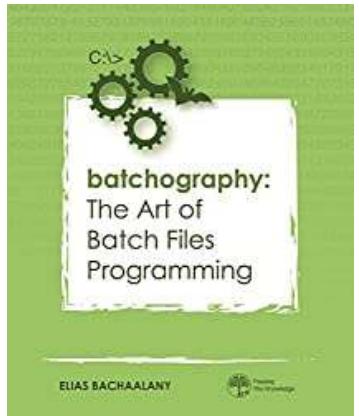
Description: The function template can be used as starting point for a new function. Complete the template into a useful function like this:

- Rename the function
- Add proper description for the function and its arguments
- Fill in the function body
- Return desired values

Script:

Download: [BatchFunctionTmpl.bat](#)

```
1. :myFunctionName    -- function description here
2. ::                  -- %~1: argument description here
3. SETLOCAL
4. REM.--function body here
5. set LocalVar1=...
6. set LocalVar2=...
7. (ENDLOCAL & REM -- RETURN VALUES
8.   IF "%~1" NEQ "" SET %~1=%LocalVar1%
9.   IF "%~2" NEQ "" SET %~2=%LocalVar2%
10.)
11.GOTO:EOF
```





**DosTips.com**  
A Forum all about DOS Batch

🔍 ⚙️

≡ Quick links ? FAQ

📝 Register >Login
[Home](#) DosTips.com < Board index < DosTips - Dos Batch < DOS Batch Forum

[Unanswered topics](#) [Active topics](#)

## DOS Batch Forum

**Moderator:** DosItHelp

[New Topic](#) Search this forum... 🔍 ⚙️

 8311 topics ➡ 1 2 3 4 5 ... 167 ➡

Announcements		Replies	Views	Last post
Topics		Replies	Views	Last post
<a href="#">Survey Popup - RESULTS</a>	by admin » 22 Aug 2017 22:26	5	23265	by ispy <span style="color: #007bff;">🔗</span> 20 Feb 2021 14:18
<a href="#">How to get help for a batch script - quickly!</a>	by foxidrive » 08 Dec 2014 05:51	2	31680	by aGerman <span style="color: #007bff;">🔗</span> 17 Sep 2016 05:48
<a href="#">Check output of called Script</a>	by ace_Friends22 » 27 May 2021 04:28	3	93	by penpen <span style="color: #007bff;">🔗</span> 01 Jun 2021 18:36
<a href="#">progress bar while doing command</a>	by AR Coding » 31 May 2021 15:15	4	53	by AR Coding <span style="color: #007bff;">🔗</span> 01 Jun 2021 18:13
<a href="#">Animated screen in Batch</a>	by Emo » 30 May 2021 22:40	4	58	by Hackoo <span style="color: #007bff;">🔗</span> 01 Jun 2021 10:44
<a href="#">prompting multiple db password for sql scripts</a>	by Aks_dolly » 01 Jun 2021 06:02	0	13	by Aks_dolly <span style="color: #007bff;">🔗</span> 01 Jun 2021 06:02
<a href="#">Commands don't execute from batch file but from CMD</a>	by falcios » 01 Jun 2021 04:10	1	15	by falcios <span style="color: #007bff;">🔗</span> 01 Jun 2021 04:55
<a href="#">Color Change and Return to Default</a>	by falcios » 31 May 2021 13:35	3	32	by AR Coding <span style="color: #007bff;">🔗</span> 31 May 2021 20:18
<a href="#">copy doc files</a>	by mike99 » 27 May 2021 11:38	1	175	by Lordoa <span style="color: #007bff;">🔗</span> 31 May 2021 13:06
<a href="#">Unexpected DOS shell expansion of %~n1 and %~x1</a>	by sandbelt » 28 May 2021 15:51	4	107	by jeb <span style="color: #007bff;">🔗</span> 31 May 2021 04:14
<a href="#">Batch Conversion issue</a>	by Emo » 23 Mar 2021 09:21	2	1122	by Emo <span style="color: #007bff;">🔗</span> 30 May 2021 22:29
<a href="#">Need some help in getting this work</a>	by Lordoa » 30 May 2021 05:07	0	307	by Lordoa <span style="color: #007bff;">🔗</span> 30 May 2021 05:07
<a href="#">Need help</a>	by Vats » 29 May 2021 00:11	4	108	by Vats <span style="color: #007bff;">🔗</span> 30 May 2021 04:39
<a href="#">For /F to process multiple lines</a>				

<a href="#">for /r to process multiple lines</a> by AR Coding » 27 May 2021 08:52		2	91	by AR Coding  27 May 2021 16:46
<a href="#">package manager</a> by npocmaka_ » 27 May 2021 01:10		2	144	by npocmaka_  27 May 2021 14:14
<a href="#">SNAKE.BAT 4.1 - An arcade style game using pure batch</a> by dbenham » 13 Jul 2013 13:43	<a href="#">1</a> ... <a href="#">4</a> <a href="#">5</a> <a href="#">6</a> <a href="#">7</a> <a href="#">8</a>	108	269965	by qwerty_kid  26 May 2021 12:56
<a href="#">Print an array elements based on an input param</a> by Newbee » 23 May 2021 13:51		2	176	by Newbee  24 May 2021 22:15
<a href="#">If not exist command</a> by falcios » 23 May 2021 23:43		1	156	by Squashman  24 May 2021 08:42
<a href="#">Sending command to a single in a text file</a> by Ryobi » 22 May 2021 13:44		2	154	by Squashman  23 May 2021 22:07
<a href="#">JREPL.BAT v8.6 - regex text processor with support for text highlighting and alternate character sets</a> by dbenham » 14 Nov 2014 14:52	<a href="#">1</a> ... <a href="#">29</a> <a href="#">30</a> <a href="#">31</a> <a href="#">32</a> <a href="#">33</a>	485	469815	by Hazado  21 May 2021 11:13
<a href="#">Need help clearing a log while it is being written to (CMD)</a> by Shahar5 » 19 May 2021 03:36		4	262	by miskox  21 May 2021 03:19
<a href="#">Batch coloring not working after git output</a> by Bleifuss » 18 May 2021 01:33		2	198	by jeb  19 May 2021 00:31
<a href="#">How to add files larger than 100 kb to the 7z archive?</a> by PiotrMP006 » 17 May 2021 00:49		1	232	by Squashman  17 May 2021 07:41
<a href="#">load a txt file and break up a string</a> by rasil » 15 May 2021 13:38		1	206	by T3RRY  16 May 2021 03:05
<a href="#">Run program with different methods in loops</a> by liargof » 15 May 2021 01:45		3	308	by Squashman  15 May 2021 21:12
<a href="#">CmdBkg - use bitmap as background to console window</a> by misol101 » 12 Sep 2016 12:56	<a href="#">1</a> <a href="#">2</a> <a href="#">3</a> <a href="#">4</a> <a href="#">5</a> <a href="#">6</a>	75	39565	by aGerman  14 May 2021 12:33
<a href="#">Entering an "ENTER" or New_Line to load multiple lines to the Clipboard</a> by Joe_Ragnar » 12 May 2021 16:22		2	258	by Joe_Ragnar  13 May 2021 09:35
<a href="#">Batch File not running in the AM hours</a> by jasieltego » 12 May 2021 15:29		1	213	by ShadowThief  13 May 2021 07:51
<a href="#">Batch Rename of files in a folder</a> by pillaisg » 10 May 2021 11:01		1	284	by Squashman  10 May 2021 12:39
<a href="#">Batch script to delete .txt files in all subfolders except one</a> by busy.bee.forest » 05 May 2021 13:34		4	494	by busy.bee.forest  07 May 2021 06:54
<a href="#">A tool for managing multiple running dos batches</a> by DosAppreciator » 05 May 2021 09:10		8	550	by DosAppreciator  06 May 2021 09:53
<a href="#">How to get MD5 hash only (Batch Script)</a> by sonelkim » 05 May 2021 17:52		1	299	by Squashman  05 May 2021 20:39
<a href="#">How do i use RoboCopy where the destination is pinpointed with the name of a file?</a> by fruitsalad23 » 05 May 2021 12:06		1	344	by aGerman  05 May 2021 15:25
<a href="#">How to make a game saving system?</a> by Thug Scripting » 03 May 2021 22:32		3	404	by T3RRY  04 May 2021 17:26
<a href="#">Replace a specific character in CSV</a>		2	500	by Mahlzeit20

<a href="#">by Mahlzeit20</a> » 19 Apr 2021 15:24		3	598	04 May 2021 07:07
<a href="#">Cmdwiz - 54 operation cmd helper tool (now with Unicode)</a> <a href="#">by misol101</a> » 10 Sep 2016 19:46	<a href="#">1</a> <a href="#">2</a> <a href="#">3</a> <a href="#">4</a>	51	46633	by Illegal Services <a href="#">🔗</a> 03 May 2021 05:56
<a href="#">tv</a> <a href="#">by tea</a> » 13 Dec 2019 20:18		9	15867	by krypton <a href="#">🔗</a> 29 Apr 2021 04:28
<a href="#">Taskkill %0</a> <a href="#">by Arne Schönbohm</a> » 26 Apr 2021 07:52		6	664	by Compo <a href="#">🔗</a> 27 Apr 2021 10:10
<a href="#">output computer, AD User</a> <a href="#">by logicassault</a> » 23 Apr 2021 12:20		4	585	by logicassault <a href="#">🔗</a> 26 Apr 2021 11:22
<a href="#">bat file dont run</a> <a href="#">by Sperhak</a> » 24 Apr 2021 04:39		3	505	by Squashman <a href="#">🔗</a> 25 Apr 2021 21:10
<a href="#">Start batch files at startup...</a> <a href="#">by rasil</a> » 21 Apr 2021 09:58		2	519	by Hackoo <a href="#">🔗</a> 24 Apr 2021 03:39
<a href="#">Delete specified multiple lines of a not in use batch file through another batch file?</a> <a href="#">by D4rks1d3r</a> » 03 Apr 2021 15:15		4	1264	by Squashman <a href="#">🔗</a> 22 Apr 2021 14:16
<a href="#">Please review my code</a> <a href="#">by timothymycompc</a> » 20 Apr 2021 20:57		1	477	by Maylow <a href="#">🔗</a> 21 Apr 2021 10:55
<a href="#">Windows (and Office?) product key finder</a> <a href="#">by miskox</a> » 12 Apr 2021 15:36		13	1399	by Compo <a href="#">🔗</a> 20 Apr 2021 07:42
<a href="#">Get all column data in csv file (from column 100 to column 110, 150)</a> <a href="#">by Comet297</a> » 14 Apr 2021 02:42		12	982	by Comet297 <a href="#">🔗</a> 20 Apr 2021 02:47
<a href="#">Proof of concept: Model, View, Controller and Interfaces</a> <a href="#">by Maylow</a> » 19 Apr 2021 19:02		0	550	by Maylow <a href="#">🔗</a> 19 Apr 2021 19:02
<a href="#">Clobbered cmd.exe!</a> <a href="#">by dbenham</a> » 11 Apr 2021 19:44		7	1126	by jfl <a href="#">🔗</a> 16 Apr 2021 15:36
<a href="#">Set Column for every file to txt</a> <a href="#">by AlphaInc.</a> » 15 Apr 2021 09:03		3	559	by AlphaInc. <a href="#">🔗</a> 16 Apr 2021 09:14
<a href="#">Safely parse nearly any CSV with pure batch</a> <a href="#">by dbenham</a> » 30 May 2019 23:07		2	3659	by SIMMS7400 <a href="#">🔗</a> 14 Apr 2021 07:04
<a href="#">MS-DOS 6.22 variable trimming</a> <a href="#">by cajunjon</a> » 01 Apr 2021 11:00		3	984	by jeb <a href="#">🔗</a> 12 Apr 2021 09:17
<a href="#">Simple replace text in txt file script</a> <a href="#">by Pieceinc</a> » 09 Apr 2021 03:31		2	785	by Pieceinc <a href="#">🔗</a> 12 Apr 2021 04:43

[New Topic](#)   

8311 topics   [1](#) [2](#) [3](#) [4](#) [5](#) ... [167](#)  
[Return to Board Index](#)
[Jump to](#) 

#### FORUM PERMISSIONS

You **cannot** post new topics in this forum  
 You **cannot** reply to topics in this forum  
 You **cannot** edit your posts in this forum  
 You **cannot** delete your posts in this forum  
 You **cannot** post attachments in this forum

