



Environment variable

An **environment variable** is a user-definable value that can affect the way running processes will behave on a computer. Environment variables are part of the environment in which a process runs. For example, a running process can query the value of the TEMP environment variable to discover a suitable location to store temporary files, or the HOME or USERPROFILE variable to find the directory structure owned by the user running the process.

They were introduced in their modern form in 1979 with Version 7 Unix, so are included in all Unix operating system flavors and variants from that point onward including Linux and macOS. From PC DOS 2.0 in 1982, all succeeding Microsoft operating systems, including Microsoft Windows, and OS/2 also have included them as a feature, although with somewhat different syntax, usage and standard variable names.

Design

In all Unix and Unix-like systems, as well as on Windows, each process has its own separate set of environment variables. By default, when a process is created, it inherits a duplicate run-time environment of its parent process, except for explicit changes made by the parent when it creates the child. At the API level, these changes must be done between running fork and exec. Alternatively, from command shells such as bash, a user can change environment variables for a particular command invocation by indirectly invoking it via env or using the `ENVIRONMENT_VARIABLE=VALUE <command>` notation. A running program can access the values of environment variables for configuration purposes.

Shell scripts and batch files use environment variables to communicate data and preferences to child processes. They can also be used to store temporary values for reference later in a shell script. However, in Unix, non-exported variables are preferred for this as they do not leak outside the process.

In Unix, an environment variable that is changed in a script or compiled program will only affect that process and possibly child processes. The parent process and any unrelated processes will not be affected. Similarly, changing or removing a variable's value inside a DOS or Windows batch file will change the variable for the duration of COMMAND.COM or CMD.EXE's existence, respectively.

In Unix, the environment variables are normally initialized during system startup by the system init startup scripts, and hence inherited by all other processes in the system. Users can, and often do, augment them in the profile script for the command shell they are using. In Microsoft Windows, each environment variable's default value is stored in the Windows Registry or set in the AUTOEXEC.BAT file.

On Unix, a setuid program is given an environment chosen by its caller, but it runs with different authority from its caller. The dynamic linker will usually load code from locations

specified by the environment variables `$LD_LIBRARY_PATH` and `$LD_PRELOAD` and run it with the process's authority. If a `setuid` program did this, it would be insecure, because its caller could get it to run arbitrary code and hence misuse its authority. For this reason, `libc` unsets these environment variables at startup in a `setuid` process. `setuid` programs usually unset unknown environment variables and check others or set them to reasonable values.

In general, the collection of environment variables function as an associative array where both the keys and values are strings. The interpretation of characters in either string differs among systems. When data structures such as lists need to be represented, it is common to use a colon (common on Unix and Unix-like) or semicolon-delineated (common on Windows and DOS) list.

Syntax

The variables can be used both in scripts and on the command line. They are usually referenced by putting special symbols in front of or around the variable name.

It is conventional for environment-variable names to be chosen to be in all upper cases. In programming code generally, this helps to distinguish environment variables from other kinds of names in the code. Environment-variable names are case sensitive on Unix-like operating systems but not on DOS, OS/2, and Windows.

Unix

In most Unix and Unix-like command-line shells, an environment variable's value is retrieved by placing a `$` sign before the variable's name. If necessary, the name can also be surrounded by braces.

To display the user home directory, the user may type:

```
echo $HOME
```

In Unix and Unix-like systems, the names of environment variables are case-sensitive.

The command **env** displays all environment variables and their values. The command **printenv** can also be used to print a single variable by giving that variable name as the sole argument to the command.

DOS, OS/2 and Windows

In DOS, OS/2 and Windows command-line interpreters such as `COMMAND.COM` and `CMD.EXE`, an environment variable is retrieved by placing a `%` sign before and after it.

In DOS, OS/2 and Windows command-line interpreters as well as their API, upper or lower case is not distinguished for environment variable names.

The environment variable named `HOMEDRIVE` contains the drive letter (plus its trailing `:` colon) of the user's home directory, whilst `HOMEPATH` contains the full path of the user's home directory within that drive.

So to see the home drive and path, the user may type this:

```
ECHO %HOMEDRIVE%%HOMEPATH%
```

The command **SET** (with no arguments) displays all environment variables and their values. In Windows NT and later `set` can also be used to print all variables whose name begins with a given prefix by giving the prefix as the sole argument to the command.

In Windows PowerShell, the user may type any of the following:

```
echo $env:homedrive$env:homepath  
Write-Output $env:homedrive$env:homepath  
"$env:homedrive$env:homepath"
```

In PowerShell, upper or lower case is not distinguished for environment variable names.

The following command displays all environment variables and their values:

```
get-childitem env:
```

Assignment: Unix

The commands `env` and `set` can be used to set environment variables and are often incorporated directly into the shell.

The following commands can also be used, but are often dependent on a certain shell.

```
VARIABLE=value      # (there must be no spaces around the equals sign)  
export VARIABLE      # for Bourne and related shells
```

```
export VARIABLE=value # for ksh, bash, and related shells
```

```
setenv VARIABLE value # for csh and related shells
```

A few simple principles govern how environment variables achieve their effect.

Environment variables are local to the process in which they were set. If two shell processes are spawned and the value of an environment variable is changed in one, that change will not be seen by the other.

When a child process is created, it inherits all the environment variables and their values from the parent process. Usually, when a program calls another program, it first creates a child process by forking, then the child adjusts the environment as needed and lastly the child replaces itself with the program to be called. This procedure gives the calling program control over the environment of the called program.

In Unix shells, variables may be assigned without the **export** keyword. Variables defined in this way are displayed by the **set** command, but are *not* true environment variables, as they are stored only by the shell and are unknown to all other processes. The `printenv` command will

not display them, and child processes do not inherit them.

```
VARIABLE=value
```

The prefix syntax exports a "true" environment variable to a child process without affecting the current process:^[1]

```
VARIABLE=value program_name [arguments]
```

The persistence of an environment variable can be session-wide or system-wide.

unset is a builtin command implemented by both the Bourne shell family (sh, ksh, bash, etc.) and the C shell family (csh, tcsh, etc.) of Unix command line shells. It unsets a shell variable, removing it from memory and the shell's exported environment. It is implemented as a shell builtin, because it directly manipulates the internals of the shell.^{[2][3]} Read-only shell variables cannot be unset. If one tries to unset a read-only variable, the unset command will print an error message and return a non-zero exit code.

Assignment: DOS, OS/2 and Windows

In DOS, OS/2 and Windows command-line interpreters such as `COMMAND.COM` and `CMD.EXE`, the **SET** command is used to assign environment variables and values using the following arguments:

```
SET VARIABLE=value
```

An environment variable is removed via:

```
SET VARIABLE=
```

The **SET** command without any arguments displays all environment variables along with their values; **SET " "**, zero or more spaces, will include internal variables too. In `CMD.EXE`, it is possible to assign local variables that will not be global using the **SETLOCAL** command and **ENDLOCAL** to restore the environment.

Use the switch `/?` to display the internal documentation, or use the viewer **help**:

```
SET /?  
HELP SET  
SETLOCAL /?  
HELP SETLOCAL
```

In PowerShell, the assignment follows a syntax similar to Unix:

```
$env:VARIABLE = "VALUE"
```

Examples

Examples of environment variables include:

- **PATH**: a list of directory paths. When the user types a command without providing the full path, this list is checked to see whether it contains a path that leads to the command.
- **HOME** (Unix-like) and **USERPROFILE** (Microsoft Windows): indicate where a user's home directory is located in the file system.
- **HOME/{.AppName}** (Unix-like) and **APPDATA\{DeveloperName\AppName}** (Microsoft Windows): for storing application settings. Many applications incorrectly use **USERPROFILE** for application settings in Windows: **USERPROFILE** should only be used in dialogs that allow user to choose between paths like Documents/Pictures/Downloads/Music; for programmatic purposes, **APPDATA** (for roaming application settings shared across multiple devices), **LOCALAPPDATA** (for local application settings) or **PROGRAMDATA** (for application settings shared between multiple OS users) should be used.^[4]
- **TERM** (Unix-like): specifies the type of computer terminal or terminal emulator being used (e.g., vt100 or dumb).
- **PS1** (Unix-like): specifies how the prompt is displayed in the Bourne shell and variants.
- **MAIL** (Unix-like): used to indicate where a user's mail is to be found.
- **TEMP**: location where processes can store temporary files.

True environment variables

Unix

\$PATH

Contains a colon-separated list of directories that the shell searches for commands that do not contain a slash in their name (commands with slashes are interpreted as file names to execute, and the shell attempts to execute the files directly). It is equivalent to the DOS, OS/2 and Windows %PATH% variable.

\$HOME

Contains the location of the user's home directory. Although the current user's home directory can also be found out through the C-functions `getpwuid` and `getuid`, **\$HOME** is often used for convenience in various shell scripts (and other contexts). Using the environment variable also gives the user the possibility to point to another directory.

\$PWD

This variable points to the current directory. Equivalent to the output of the command `pwd` when called without arguments.

\$DISPLAY

Contains the identifier for the display that X11 programs should use by default.

\$LD_LIBRARY_PATH

On many Unix systems with a dynamic linker, contains a colon-separated list of directories that the dynamic linker should search for shared objects when building a process image after `exec`, before searching in any other directories.

\$LIBPATH or \$SHLIB_PATH

Alternatives to `$LD_LIBRARY_PATH` typically used on older Unix versions.

\$LANG, \$LC_ALL, \$LC...

`$LANG` is used to set to the default locale. For example, if the locale values are `pt_BR`, then the language is set to (Brazilian) Portuguese and Brazilian practice is used where relevant. Different aspects of localization are controlled by individual `$LC_`-variables (`$LC_CTYPE`, `$LC_COLLATE`, `$LC_DATE` etc.). `$LC_ALL` can be used to force the same locale for all aspects.

\$TZ

Refers to time zone. It can be in several formats, either specifying the time zone itself or referencing a file (in `/usr/share/zoneinfo`).

\$BROWSER

Contains a colon-separated list of a user's web browser preferences, for use by programs that need to allow the user to view content at a URL. The browsers in the list are intended to be attempted from first to last, stopping after the first one that succeeds. This arrangement allows for fallback behavior in different environments, e.g., in an X11 environment, a graphical browser (such as Firefox) can be used, but in a console environment a terminal-base browser (such a Lynx) can be used. A `%s` token may be present to specify where the URL should be placed; otherwise the browser should be launched with the URL as the first argument.^{[5][6][7][8][9]}

DOS

Under DOS, the *master environment* is provided by the primary command processor, which inherits the *pre-environment* defined in `CONFIG.SYS` when first loaded. Its size can be configured through the `COMMAND /E:n` parameter between 160^[10] and 32767^[10] bytes. *Local environment* segments inherited to child processes are typically reduced down to the size of the contents they hold. Some command-line processors (like 4DOS) allow to define a minimum amount of free environment space that will be available when launching secondary shells.^[10] While the content of environment variables remains unchanged upon storage, their names (without the "%") are always converted to uppercase, with the exception of *pre-environment variables* defined via the `CONFIG.SYS` directive `SET` under DR DOS 6.0 and higher^{[11][12]} (and only with `SWITCHES=/L` (for "allow lowercase names") under DR-DOS 7.02 and higher).^{[10][13]} In principle, MS-DOS 7.0 and higher also supports lowercase variable names (`%windir%`), but provides no means for the user to define them. Environment variable names containing lowercase letters are stored in the environment just like normal environment variables, but remain invisible to most DOS software, since they are written to expect uppercase variables only.^{[10][11][12]} Some command processors limit the maximum length of a variable name to 80 characters.^[10] While principally only limited by the size of the *environment segment*, some DOS and 16-bit Windows programs^{[10][nb 1]} do not expect the contents of environment variables to exceed 128 characters. DR-DOS `COMMAND.COM` supports environment variables up to 255, 4DOS even up to 512 characters.^[10] Since `COMMAND.COM` can be configured (via `/L:128..1024`) to support command lines up to 1024 characters internally under MS-DOS 7.0 and higher, environment variables should be expected to contain at least 1024 characters as well. In some versions of DR-DOS, the environment passed to drivers, which often do not need their environment after installation, can be shrunk or relocated through `SETENV` or `INSTALL[HIGH]/LOADHIGH` options `/Z` (zero environment), `/D[:loaddrive]` (substitute drive, e.g. `B:TSR.COM`) and `/E` (relocate environment above program) in order to minimize the driver's

effectively resulting resident memory footprint.^{[14][13][11][15][16][17]}

In batch mode, non-existent environment variables are replaced by a zero-length string.

Standard environment variables or reserved environment variables include:

%APPEND% (supported since DOS 3.3)

This variable contains a semicolon-delimited list of directories in which to search for files. It is usually changed via the `APPEND /E` command, which also ensures that the directory names are converted into uppercase. Some DOS software actually expects the names to be stored in uppercase and the length of the list not to exceed 121^[10] characters, therefore the variable is best not modified via the `SET` command. Long filenames containing spaces or other special characters must not be quoted (").

%CONFIG% (supported since MS-DOS 6.0 and PC DOS 6.1, also supported by ROM-DOS^[18])

This variable holds the symbolic name of the currently chosen boot configuration. It is set by the DOS BIOS (`IO.SYS`, `IBMBIO.COM`, etc.) to the name defined by the corresponding `CONFIG.SYS` directive `MENUITEM` before launching the primary command processor. Its main purpose is to allow further special cases in `AUTOEXEC.BAT` and similar batchjobs depending on the selected option at boot time. This can be emulated under DR-DOS by utilizing the `CONFIG.SYS` directive `SET` like `SET CONFIG=1`.

%CMDLINE% (introduced with 4DOS, also supported since MS-DOS 7.0)

This variable contains the fully expanded text of the currently executing command line. It can be read by applications to detect the usage of and retrieve long command lines, since the traditional method to retrieve the command line arguments through the `PSP` (or related `API` functions) is limited to 126 characters and is no longer available when `FCBs` get expanded or the default `DTA` is used. While 4DOS supports longer command lines, `COMMAND.COM` still only supports a maximum of 126 characters at the prompt by default (unless overridden with `/U:128..255` to specify the size of the command line buffer), but nevertheless internal command lines can become longer through f.e. variable expansion (depending on `/L:128..1024` to specify the size of the internal buffer). In addition to the command-line length byte in the `PSP`, the `PSP` command line is normally limited by `ASCII-13`, and command lines longer than 126 characters will typically be truncated by having an `ASCII-13` inserted at position 127,^[12] but this cannot be relied upon in all scenarios.^{[12][nb 2]} The variable will be suppressed for external commands invoked with a preceding `@`-symbol like in `@XCOPY ...` for backward compatibility and in order to minimize the size of the environment when loading non-relocating terminate-and-stay-resident programs. Some beta versions of Windows Chicago used `%CMDLINE%` to store only the remainder of the command line exceeding 126 characters instead of the complete command line.^{[11][12]}

%COMSPEC% (supported since DOS 2.0)

This variable contains the full 8.3 path to the command processor, typically `C:\COMMAND.COM` or `C:\DOS\COMMAND.COM`. It must not contain long filenames, but under DR-DOS it may contain file and directory passwords. It is set up by the primary command processor to point to itself (typically reflecting the settings of the `CONFIG.SYS` directive `SHELL`), so that the resident portion of the command processor can reload its transient portion from disk after the execution of larger programs. The value can be changed at runtime to reflect changes in the configuration, which would require the command processor to reload itself from other locations. The variable is also used when launching secondary shells.

%COPYCMD% (supported since MS-DOS 6.2 and PC DOS 6.3, also supported by ROM-DOS^[18])

Allows a user to specify the /Y switch (to assume "Yes" on queries) as the default for the COPY, XCOPY, and MOVE commands. A default of /Y can be overridden by supplying the /-Y switch on the command line. The /Y switch instructs the command to replace existing files without prompting for confirmation.

%DIRCMD% (supported since MS-DOS 5.0 and PC DOS 5.0, also supported by ROM-DOS^[18])

Allows a user to specify customized default parameters for the DIR command, including file specifications. Preset default switches can be overridden by providing the negative switch on the command line. For example, if %DIRCMD% contains the /W switch, then it can be overridden by using DIR /-W at the command line. This is similar to the environment variable %\$DIR% under DOS Plus^[19] and a facility to define default switches for DIR through its /C or /R switches under DR-DOS COMMAND.COM.^[11] %DIRCMD% is also supported by the external SDIR.COM/DIR.COM Stacker commands under Novell DOS 7 and higher.^[11]

%LANG% (supported since MS-DOS 7.0)

This variable is supported by some tools to switch the locale for messages in multilingual issues.

%LANGSPEC% (supported since MS-DOS 7.0)

This variable is supported by some tools to switch the locale for messages in multilingual issues.

%NO_SEP% (supported since PC DOS 6.3 and DR-DOS 7.07)

This variable controls the display of thousands-separators in messages of various commands. Issued by default, they can be suppressed by specifying SET NO_SEP=ON or SET NO_SEP=1 under PC DOS. DR-DOS additionally allows to override the system's thousands-separator displayed as in f.e. SET NO_SEP=..^[10]

%PATH% (supported since DOS 2.0)

This variable contains a semicolon-delimited list of directories in which the command interpreter will search for executable files. Equivalent to the Unix \$PATH variable (but some DOS and Windows applications also use the list to search for data files similar to \$LD_LIBRARY_PATH on Unix-like systems). It is usually changed via the PATH (or PATH /E under MS-DOS 6.0) command, which also ensures that the directory names are converted into uppercase. Some DOS software actually expects the names to be stored in uppercase and the length of the list not to exceed 123^[10] characters,^[nb 1] therefore the variable should better not be modified via the SET command.^[10] Long filenames containing spaces or other special characters must not be quoted ("). By default, the current directory is searched first, but some command-line processors like 4DOS allow "." (for "current directory") to be included in the list as well in order to override this search order; some DOS programs are incompatible with this extension.^[10]

%PROMPT% (supported since DOS 2.0)

This variable contains a \$-tokenized string defining the display of the prompt. It is usually changed via the PROMPT command.

%TEMP% (and %TMP%)

These variables contain the path to the directory where temporary files should be stored. Operating system tools typically only use %TEMP%, whereas third-party programs also use %TMP%. Typically %TEMP% takes precedence over %TMP%.

The DR-DOS family supports a number of additional *standard environment variables* including:

%BETA%

This variable contains an optional message displayed by some versions (including DR DOS 3.41) of COMMAND.COM at the startup of secondary shells.^[20]

%DRDOSCFG%/ %NWDOSCFG%/ %OPENDOSCFG%

This variable contains the directory^[21] (without trailing "\") where to search for .INI and .CFG configuration files (that is, DR-DOS application specific files like TASKMGR.INI, TASKMAX.INI, VIEWMAX.INI, FASTBACK.CFG etc., class specific files like COLORS.INI, or global files like DRDOS.INI, NWDOS.INI, OPENDOS.INI, or DOS.INI), as used by the INSTALL and SETUP commands and various DR-DOS programs like DISKOPT, DOSBOOK, EDIT, FBX, FILELINK, LOCK, SECURITY.OVL/NWLOGIN.EXE, SERNO, TASKMAX, TASKMGR, VIEWMAX, or UNDELETE.^[11] It must not contain long filenames.

%DRCOMSPEC%

This variable optionally holds an alternative path to the command processor taking precedence over the path defined in the %COMSPEC% variable, optionally including file and directory passwords. Alternatively, it can hold a special value of "ON" or "1" in order to enforce the usage of the %COMSPEC% variable even in scenarios where the %COMSPEC% variable may point to the wrong command-line processor, for example, when running some versions of the DR-DOS SYS command under a foreign operating system.^[22]

%DRSYS%

Setting this variable to "ON" or "1" will force some versions of the DR-DOS SYS command to work under foreign operating systems instead of displaying a warning.^[22]

%FBP_USER%

Specifies the user name used by the FastBack command FBX and {user}.FB configuration files under Novell DOS 7.^[11]

%HOMEDIR%

This variable may contain the home directory under DR-DOS (including DR DOS 5.0 and 6.0).^{[11][20]}

%INFO%

In some versions of DR-DOS COMMAND.COM this variable defines the string displayed by the \$I token of the PROMPT command.^[20] It can be used, for example, to inform the user how to exit secondary shells.

%LOGINNAME%

In some versions of DR-DOS COMMAND.COM this variable defines the user name displayed by the \$U token of the PROMPT command, as set up by f.e. login scripts for Novell NetWare.^{[10][11][20]} See also the similarly named pseudo-variable %LOGIN_NAME%.

%MDOS_EXEC%

This variable can take the values "ON" or "OFF" under Multiuser DOS. If enabled, the operating system permits applications to shell out to secondary shells with the DOS Program Area (DPA) freed in order to have maximum DOS memory available for secondary applications instead of running them in the same domain as under DOS.^{[23][24]}

%NOCHAR%

This variable can be used to define the character displayed by some commands in

messages for "No" in [Y,N] queries, thereby overriding the current system default (typically "N" in English versions of DR-DOS). If it contains a string, only the first character, uppercased, will be taken. Some commands also support a command line parameter /Y to automatically assume "Yes" on queries, thereby suppressing such prompts. If, however, the parameter /Y:yn is used to specify the "Yes"/"No" characters (thereby overriding any %NOCHAR% setting), queries are not suppressed. See also the related CONFIG.SYS directive NOCHAR and the environment variable %YESCHAR%.^[22]

%NOSOUND%

Setting this variable to "ON" or "1" will disable default beeps issued by some DR-DOS commands in certain situations such as to inform the user of the completion of some operation, that user interaction is required, or when a wrong key was pressed. Command line options to specifically enable certain beeps will override this setting.

%OS%

This variable contains the name of the operating system in order to distinguish between different DOS-related operating systems of Digital Research-origin in batch jobs and applications.^[21] Known values include "DOSPLUS" (DOS Plus 1.2 in DOS emulation), "CPCDOS 4.1" (DOS Plus 1.2 in CP/M emulation), "DRDOS" (DR DOS 3.31-6.0, DR DOS Panther, DR DOS StarTrek, DR-DOS 7.02^[21]-7.05), "EZDOS" (EZ-DOS 3.41), "PALMDOS" and "NetWare PalmDOS" (PalmDOS 1.0), "NWDOS" (Novell DOS 7), "NWDOS7" (Novell DOS 7 Beta), "OPENDOS" (Caldera OpenDOS 7.01, Caldera DR-OpenDOS 7.02), "CDOS" (Concurrent DOS, Concurrent DOS XM), "CPCDOS" (Concurrent PC DOS), "CDOS386" (Concurrent DOS 386), "DRMDOS" (DR Multiuser DOS), "MDOS" (CCI Multiuser DOS),^[23] "IMSMDOS" (IMS Multiuser DOS), "REAL32" (REAL/32).^{[11][25]} MS-DOS INTERSVR looks for a value of "DRDOS" as well.^[25] See also the identically named environment variable %OS% later introduced in the Windows NT family.

%PEXEC%

In some versions of DR-DOS this variable defines the command executed by the \$X token of the PROMPT command before COMMAND.COM displays the prompt after returning from external program execution.^{[11][21]}

%SWITCHAR%

This variable defines the SwitChar to be used for argument parsing by some DR-DOS commands. If defined, it overrides the system's current SwitChar setting. The only accepted characters are "/" (DOS style), "-" (Unix style) and "[" (CP/M style). See also the related CONFIG.SYS directive SWITCHAR (to set the system's SwitChar setting) and the %/% system information variable in some issues of DR-DOS COMMAND.COM (to retrieve the current setting for portable batchjobs).

%TASKMGRWINDIR%

This variable specifies the directory, where the Windows SYSTEM.INI to be used by the DR-DOS TASKMGR multitasker is located, overriding the default procedure to locate the file.^[11]

%VER%

This variable contains the version of the operating system in order to distinguish between different versions of DR-DOS in batch jobs and in the display of the VER command.^[21] It is also used for the \$V token of the PROMPT command and affects the value returned by the system information variable %OS_VERSION%. Known values include "1.0" (PalmDOS 1.0), "1.2" (DOS Plus 1.2 in DOS emulation), "2.0" (Concurrent DOS 386 2.0), "3.0" (Concurrent DOS 386 3.0), "3.31" (DR DOS 3.31), "3.32" (DR DOS 3.32), "3.33" (DR DOS 3.33), "3.34" (DR DOS 3.34), "3.35" (DR DOS 3.35), "3.40" (DR DOS 3.40), "3.41"

(DR DOS 3.41, EZ-DOS 3.41), "3.41T" (DR DOS 3.41T), "4.1" (Concurrent PC DOS 4.1), "5.0" (DR DOS 5.0, DR Multiuser DOS 5.0), "5.1" (Novell DR Multiuser DOS 5.1), "6.0" (DR Concurrent DOS XM 6.0, DR DOS 6.0), "6.2" (DR Concurrent DOS XM 6.2), "7" (Novell DOS 7, Caldera OpenDOS 7.01, DR-DOS 7.02-7.05), "7.00" (CCI Multiuser DOS 7.00), "7.07" (DR-DOS 7.07), "7.1" (IMS Multiuser DOS 7.1), "7.21" (CCI Multiuser DOS 7.21),^[23] "7.22" (CCI Multiuser DOS 7.22) etc.^{[11][25][23]}

%YESCHAR%

This variable can be used to define the character displayed by some commands in messages for "Yes" in [Y,N] queries, thereby overriding the current system default (typically "Y" in English versions of DR-DOS). If it contains a string, only the first character, uppercased, will be taken. Some commands also support a command line parameter /Y to automatically assume "Yes" on queries, thereby suppressing such prompts. If, however, the parameter /Y:y is used to specify the "Yes" character (thereby overriding any %YESCHAR% setting), queries are not suppressed. See also the related CONFIG.SYS directive YESCHAR and the environment variable %NOCHAR%.^[22]

%%CLS%

This variable defines the control sequence to be sent to the console driver to clear the screen when the CLS command is issued, thereby overriding the internal default ("←[2J" under DR-DOS, "←E" under DOS Plus 1.2 on Amstrad machines^[19] as well as under Concurrent DOS, Multiuser DOS, and REAL/32 for VT52 terminals, or "←+" under Multiuser DOS for ASCII terminals).^[23] If the variable is not defined and no ANSI.SYS console driver is detected, the DR-DOS COMMAND.COM will directly clear the screen via INT 10h/AH=00h BIOS function, like MS-DOS/PC DOS COMMAND.COM does. A special \nnn-notation for octal numbers is supported to allow the definition of special characters like ESC (ASCII-27 = "←" = 1Bh = 33o), as f.e. in SET \$CLS=\033[2J. To send the backslash ("\\") itself, it can be doubled "\\\".^{[11][20][23]}

%%DIR%

Supported by DOS Plus accepting the values "L" (long) or "W" (wide) to change the default layout of directory listings with DIR. Can be overridden using the command line options /L or /W.^{[20][19]} See also the similar environment variable %DIRCMD% and the DIR options /C and /R of the DR-DOS COMMAND.COM.^[11]

%%PAGE%

Supported by DOS Plus accepting the values "ON" or "OFF" for pagination control. Setting this to "ON" has the same affect as adding /P to commands supporting it (like DIR or TYPE).^{[20][19]}

%%LENGTH%

Used by DOS Plus to define the screen length of the console in lines. This is used to control in a portable way when the screen output should be temporarily halted until a key is pressed in conjunction with the /P option supported by various commands or with automatic pagination.^{[20][19]} See also the related environment variables %%WIDTH% and %DIRSIZE% as well as the similar pseudo-variable %_ROWS%.

%%WIDTH%

Used by DOS Plus to define the screen width of the console in columns. This is used to control in a portable way the formatting of the screen output of commands like DIR /W or TYPE filename.^{[20][19]} See also the related environment variables %%LENGTH% and %DIRSIZE% as well as the similar pseudo-variable %_COLUMNS%.

%%SLICE%

Used by DOS Plus accepting a numerical value to control the foreground/background time slicing of multitasking programs.^{[20][19]} See also the DOS Plus command SLICE.

%%\$ON%

This variable can hold an optional control sequence to switch text highlighting, reversion or colorization on. It is used to emphasize or otherwise control the display of the file names in commands like TYPE wildcard, for example `SET $ON=\033[1m` with ANSI.SYS loaded or `SET $ON=\016` for an IBM or ESC/P printer. For the special `\nnn` octal notation supported, see %%\$CLS%.^{[11][19]} While the variable is undefined by default under DOS Plus and DR-DOS, the Multiuser DOS default for an ASCII terminal equals `SET $ON=\033p`.^{[20][23]} See also the related environment variable %%\$OFF%.

%%\$OFF%

This variable can hold an optional control sequence to switch text highlighting, reversion or colorization off. It is used to return to the normal output after the display of file names in commands like TYPE wildcard, for example `SET $OFF=\033[0m` with ANSI.SYS loaded or `SET $OFF=\024` for an IBM or ESC/P printer. For the special `\nnn` octal notation supported, see %%\$CLS%.^{[11][19]} While the variable is undefined by default under DOS Plus and DR-DOS, the Multiuser DOS default for an ASCII terminal equals `SET $OFF=\033q`.^{[20][23]} See also the related environment variable %%\$ON%.

%%\$HEADER%

This variable can hold an optional control sequence issued before the output of the file contents in commands like TYPE under DR-DOS 7.02 and higher. It can be used for highlighting, pagination or formatting, f.e. when sending the output to a printer, i.e. `SET $HEADER=\017` for an IBM or ESC/P printer. For the special `\nnn` octal notation supported, see %%\$CLS%.^[20] See also the related environment variable %%\$FOOTER%.

%%\$FOOTER%

This variable can hold an optional control sequence issued after the output of the file contents in commands like TYPE under DR-DOS 7.02 and higher. It is used to return to the normal output format, i.e. `SET $FOOTER=\022\014` in the printer example above. For the special `\nnn` octal notation supported, see %%\$CLS%.^[20] See also the related environment variable %%\$HEADER%.

Datelight ROM-DOS supports a number of additional *standard environment variables* as well including:

%%DIRSIZE%

This variable is used to define non-standard screen sizes `rows[,cols]` for DIR options `/P` and `/W` (similar to %%\$LENGTH% and %%\$WIDTH% under DOS Plus).^[18]

%%NEWFILE%

This variable is automatically set to the first parameter given to the CONFIG.SYS directive NEWFILE.^[18]

%%TZ%, %%COMM%, %%SOCKETS%, %%HTTP_DIR%, %%HOSTNAME% and %%FTPDIR% are also used by ROM-DOS.^[18]

OS/2

%%BEGINLIBPATH%

Contains a semicolon-separated list of directories which are searched for DLLs *before* the

directories given by the `%LIBPATH%` variable (which is set during system startup with the special `CONFIG.SYS` directive `LIBPATH`). It is possible to specify relative directories here, including `"."` for the current working directory. See also the related environment variable `%ENDLIBPATH%`.

%ENDLIBPATH%

a list of directories to be searched for DLLs like `%BEGINLIBPATH%`, but searched *after* the list of directories in `%LIBPATH%`.

Windows

These environment variables refer to locations of critical operating system resources, and as such generally are not user-dependent.^[26]

%APPDATA%

Contains the full path to the *Application Data* directory of the logged-in user. Does not work on Windows NT 4.0 SP6 UK.

%LOCALAPPDATA%

This variable is the temporary files of *Applications*. Its uses include storing of desktop themes, Windows error reporting, caching and profiles of web browsers.

%ComSpec%/ %COMSPEC%

The `%ComSpec%` variable contains the full path to the command processor; on the Windows NT family of operating systems, this is `cmd.exe`, while on Windows 9x, `%COMSPEC%` is `COMMAND.COM`.

%OS%

The `%OS%` variable contains a symbolic name of the operating system family to distinguish between differing feature sets in batchjobs. It resembles an identically named environment variable `%OS%` found in all DOS-related operating systems of Digital Research-origin like Concurrent DOS, Multiuser DOS, REAL/32, DOS Plus, DR DOS, Novell DOS and OpenDOS. `%OS%` always holds the string `"Windows_NT"` on the Windows NT family.^[27]

%PATH%

This variable contains a semicolon-delimited (do not put spaces in between) list of directories in which the command interpreter will search for an executable file that matches the given command. Environment variables that represent paths may be nested within the `%PATH%` variable, but only at one level of indirection. If this sub-path environment variable itself contains an environment variable representing a path, `%PATH%` will not expand properly in the variable substitution. Equivalent to the Unix `$PATH` variable.

%ProgramFiles%, %ProgramFiles(x86)%, %ProgramW6432%

The `%ProgramFiles%` variable points to the *Program Files* directory, which stores all the installed programs of Windows and others. The default on English-language systems is `"C:\Program Files"`. In 64-bit editions of Windows (XP, 2003, Vista), there are also `%ProgramFiles(x86)%`, which defaults to `"C:\Program Files (x86)"`, and `%ProgramW6432%`, which defaults to `"C:\Program Files"`. The `%ProgramFiles%` itself depends on whether the process requesting the environment variable is itself 32-bit or 64-bit (this is caused by Windows-on-Windows 64-bit redirection).

%CommonProgramFiles%

This variable points to the *Common Files* directory. The default is `"C:\Program Files\Common Files"` in the English version of Windows.

%OneDrive%

The %OneDrive% variable is a special system-wide environment variable found on Windows NT and its derivatives. Its value is the path of where (if installed and setup) the Onedrive directory is located. The value of %OneDrive% is in most cases "C:\Users\{Username}\OneDrive\".

%SystemDrive%

The %SystemDrive% variable is a special system-wide environment variable found on Windows NT and its derivatives. Its value is the drive upon which the system directory was placed. The value of %SystemDrive% is in most cases "C:".

%SystemRoot%

The %SystemRoot% variable is a special system-wide environment variable found on the Windows NT family of operating systems. Its value is the location of the system directory, including the drive and path. The drive is the same as %SystemDrive% and the default path on a clean installation depends upon the version of the operating system. By default:

- Windows XP and newer versions use "\WINDOWS".
- Windows 2000, NT 4.0 and NT 3.1 use "\WINNT".
- Windows NT 3.5 and NT 3.51 uses "\WINNT35".
- Windows NT 4.0 Terminal Server uses "\WTSRV".

%windir%

This variable points to the *Windows* directory. (On the Windows NT family of operating systems, it is identical to the %SystemRoot% variable). Windows 95–98 and Windows ME are, by default, installed in "C:\Windows". For other versions of Windows, see the %SystemRoot% entry above.

User management variables store information related to resources and settings owned by various user profiles within the system. As a general rule, these variables do not refer to critical system resources or locations that are necessary for the OS to run.

%ALLUSERSPROFILE% (%PROGRAMDATA% since Windows Vista)

This variable expands to the full path to the *All Users* profile directory. This profile contains resources and settings that are used by all system accounts. Shortcut links copied to the *All Users\ Start menu* or *Desktop* directories will appear in every user's *Start menu* or *Desktop*, respectively.

%USERDOMAIN%

The name of the *Workgroup* or *Windows Domain* to which the current user belongs. The related variable, %LOGONSERVER%, holds the hostname of the server that authenticated the current user's login credentials (name and password). For home PCs and PCs in a workgroup, the authenticating server is usually the PC itself. For PCs in a Windows domain, the authenticating server is a domain controller (a primary domain controller, or PDC, in Windows NT 4-based domains).

%USERPROFILE%

A special system-wide environment variable found on Windows NT and its derivatives. Its value is the location of the current user's profile directory, in which is found that user's HKCU registry hive (NTUSER). Users can also use the %USERNAME% variable to determine the active users login identification.

Optional System variables are not explicitly specified by default but can be used to modify the default behavior of certain built-in console commands. These variables also do not need to be

explicitly specified as command line arguments.

Default values

The following tables shows typical default values of certain environment variables under English versions of Windows as they can be retrieved under CMD.

(Some of these variables are also defined when running `COMMAND.COM` under Windows, but differ in certain important details: Under `COMMAND.COM`, the names of environment variable are always uppercased. Some, but not all variables contain short 8.3 rather than long file names. While some variables present in the `CMD` environment are missing, there are also some variables specific to the `COMMAND` environment.)

Variable	Locale specific	Windows XP (CMD)	Windows Vista and later
%ALLUSERSPROFILE% ^[28]	Yes	C:\Documents and Settings\All Users	C:\ProgramData
%APPDATA% ^[28]	Yes	C:\Documents and Settings\{username}\Application Data	C:\Users\{username}\AppData\Local
%CommonProgramFiles% ^[28]	Yes	C:\Program Files\Common Files	C:\Program Files\Common Files
%CommonProgramFiles(x86)% ^[28]	Yes	C:\Program Files (x86)\Common Files <i>(only in 64-bit version)</i>	C:\Program Files (x86)\Common Files <i>(only in 64-bit version)</i> ^[28]
%CommonProgramW6432% ^[28]	Yes	%CommonProgramW6432% <i>(not supported, not replaced by any value)</i>	C:\Program Files <i>(only in 64-bit version)</i> ^[28]
%COMPUTERNAME%	No	{computername}	{computername}
%ComSpec%	No	C:\Windows\System32\cmd.exe	C:\Windows\System32\cmd.exe
%HOMEDRIVE% ^[28]	No	C:	C: ^[28]
%HOMEPATH% ^[28]	Yes	\Documents and Settings\{username}	\Users\{username}
%LOCALAPPDATA% ^[28]	Yes	%LOCALAPPDATA% <i>(not supported, not replaced by any value)</i>	C:\Users\{username}\AppData\Local
%LOGONSERVER%	No	\\{domain_logon_server}	\\{domain_logon_server}
%PATH%	Yes	C:\Windows\system32;C:\Windows;C:\Windows\System32\Wbem;{plus program paths}	C:\Windows\System32\cmd.exe;C:\Windows\System32\WindowsPowerShell\v1.0\
%PATHEXT%	No	.COM;.EXE;.BAT;.CMD;.VBS;.VBE;.JS;.WSF;.WSH	.com;.exe;.bat;.cmd;.vbs;.vbe;.js;.wsf;.wsh
%ProgramData% ^[28]	Yes	%ProgramData% <i>(not supported, not replaced by any value)</i>	%SystemDrive%\ProgramData
%ProgramFiles% ^[28]	Yes	%SystemDrive%\Program Files	%SystemDrive%\Program Files
%ProgramFiles(x86)% ^[28]	Yes	%SystemDrive%\Program Files (x86) <i>(only in 64-bit version)</i>	%SystemDrive%\Program Files (x86) <i>(only in 64-bit version)</i> ^[28]
%ProgramW6432% ^[28]	Yes	%ProgramW6432% <i>(not supported, not replaced by any value)</i>	%SystemDrive%\Program Files <i>(only in 64-bit version)</i> ^[28]
%PROMPT%	No	Code for current command prompt format, usually \$P\$G	Code for current command prompt format, usually \$P\$G
%PSModulePath%		%PSModulePath% <i>(not supported, not replaced by any value)</i>	%SystemRoot%\system32\WindowsPowerShell\v1.0\
%PUBLIC% ^[28]	Yes	%PUBLIC% <i>(not supported, not replaced by any value)</i>	%SystemDrive%\Users\Public
%SystemDrive% ^[28]	No	C:	C: ^[28]
%SystemRoot% ^[28]	No	The Windows directory, usually C:\Windows, formerly C:\WINNT	%SystemDrive%\Windows
%TEMP% ^[28] and %TMP% ^[28]	Yes	%SystemDrive%\Documents and Settings\{username}\Local Settings\Temp	%SystemRoot%\temp; %SystemRoot%\AppData\Local\Temp; %SystemRoot%\AppData\Local\Microsoft\Windows\Temporary Internet Files\Content.IE5\
%USERDOMAIN%	No	{userdomain}	{userdomain}
%USERNAME%	No	{username}	{username}
%USERPROFILE% ^[28]	Yes	%SystemDrive%\Documents and Settings\{username}	%SystemDrive%\Users\{username}

<code>%windir%</code> ^[28]	No	<code>%SystemDrive%\WINDOWS</code>	<code>%SystemDri</code>
---------------------------------------	----	------------------------------------	-------------------------

In this list, there is no environment variable that refers to the location of the user's *My Documents* directory, so there is no standard method for setting a program's home directory to be the *My Documents* directory.

Pseudo-environment variables

The command processors in DOS and Windows also support pseudo-environment variables. These are values that are fetched like environment variables, but are not truly stored in the environment but computed when requested.

DOS

Besides true environment variables, which are statically stored in the environment until changed or deleted, a number of pseudo-environment variables exist for batch processing.

The so-called *replacement parameters* or *replaceable parameters* (Microsoft / IBM terminology) aka *replacement variables* (Digital Research / Novell / Caldera terminology)^[21] or *batch file parameters* (JP Software terminology)^[10] `%1..%9` and `%0` can be used to retrieve the calling parameters of a batchjob, see SHIFT. In batchjobs, they can be retrieved just like environment variables, but are not actually stored in the environment.

Some command-line processors (like DR-DOS COMMAND.COM,^[21] Multiuser DOS MDOS.COM/TMP.EXE (Terminal Message Process), JP Software 4DOS, 4OS2, 4NT, Take Command and Windows cmd.exe) support a type of pseudo-environment variables named *system information variables* (Novell / Caldera terminology)^[21] or *internal variables* (JP Software terminology),^[10] which can be used to retrieve various possibly dynamic, but read-only information about the running system in batch jobs. The returned values represent the status of the system in the moment these variables are queried; that is, reading them multiple times in a row may return different values even within the same command; querying them has no direct effect on the system. Since they are not stored in the environment, they are not listed by SET and do not exist for external programs to retrieve. If a true environment variable of the same name is defined, it takes precedence over the corresponding variable until the environment variable is deleted again. They are not case-sensitive. While almost all such variables are prefixed with an underscore ("_") by 4DOS etc. by convention (f.e. %_SECOND%),^[10] they are not under DR-DOS COMMAND.COM (f.e. %OS_VERSION%).

In addition, 4DOS, 4OS2, 4NT, and Take Command also support so called *variable functions*,^[10] including user-definable ones. They work just like *internal variables*, but can take optional parameters (f.e. %@EVAL[]%) and may even change the system status depending on their function.

System information variables supported by DR-DOS COMMAND.COM:

%AM_PM%

This pseudo-variable returns the ante- or post-midday status of the current time. The returned string depends on the locale-specific version of DR-DOS, f.e. "am" or "pm" in the

English version. It resembles an identically named *identifier variable* in Novell NetWare login scripts.

%DAY%

This pseudo-variable returns the days of the current date in a 2-digit format with leading zeros, f.e. "01".."31". See also the similar pseudo-variable %_DAY%. It resembles an identically named *identifier variable* in Novell NetWare login scripts.

%DAY_OF_WEEK%

This pseudo-variable returns the day name of the week in a 3-character format. The returned string depends on the locale-specific version of DR-DOS, f.e. "Sun", "Mon", "Tue", "Wed", "Thu", "Fri", or "Sat" in the English version. It resembles an identically named *identifier variable* in Novell NetWare login scripts.

%ERRORLEVEL%

In `COMMAND.COM` of DR-DOS 7.02 and higher, this pseudo-variable returns the last error level returned by an external program or the `RETURN` command, f.e. "0".."255".^{[29][30]} See also the identically named pseudo-variable %ERRORLEVEL% under Windows and the `IF ERRORLEVEL` conditional command.

%ERRORLVL%

In DR-DOS 7.02 and higher, this pseudo-variable returns the last error level in a 3-digit format with leading zeros, f.e. "000".."255".^{[29][30]} Under Multiuser DOS, this is a true environment variable automatically updated by the shell to the return code of exiting programs.^[23] See also the related pseudo-variable %ERRORLEVEL% under DR-DOS and the `IF ERRORLEVEL` command.

%GREETING_TIME%

This pseudo-variable returns the 3-level day greeting time. The returned string depends on the locale-specific version of DR-DOS, f.e. "morning", "afternoon", or "evening" in the English version. It resembles an identically named *identifier variable* in Novell NetWare login scripts.

%HOUR%

This pseudo-variable returns the hours of the current time in 12-hour format without leading zeros, f.e. "1".."12". It resembles an identically named *identifier variable* in Novell NetWare login scripts.

%HOUR24%

This pseudo-variable returns the hours of the current time in 24-hour format in a 2-digit format with leading zeros, f.e. "00".."23". It resembles an identically named *identifier variable* in Novell NetWare login scripts. See also the similar pseudo-variable %_HOUR%.

%MINUTE%

This pseudo-variable returns the minutes of the current time in a 2-digit format with leading zeros, f.e. "00".."59". It resembles an identically named *identifier variable* in Novell NetWare login scripts. See also the similar pseudo-variable %_MINUTE%.

%MONTH%

This pseudo-variable returns the months of the current date in a 2-digit format with leading zeros, f.e. "01".."12". It resembles an identically named *identifier variable* in Novell NetWare login scripts. See also the similar pseudo-variable %_MONTH%.

%MONTH_NAME%

This pseudo-variable returns the month name of the current date. The returned string

depends on the locale-specific version of DR-DOS, f.e. "January", "February", "March", "April", "May", "June", "July", "August", "September", "October", or "December" in the English version. It resembles an identically named *identifier variable* in Novell NetWare login scripts.

%NDAY_OF_WEEK%

This pseudo-variable returns the number of day of the current week, f.e. "1".."7" (with "1" for Sunday). It resembles an identically named *identifier variable* in Novell NetWare login scripts.

%OS_VERSION%

This pseudo-variable returns the version of the operating system depending on the current setting of the environment variable %VER%. If %VER% is not defined, %OS_VERSION% returns "off". It resembles an identically named *identifier variable* in Novell NetWare login scripts, which may return versions also for non-DR-DOS versions of DOS.

%SECOND%

This pseudo-variable returns the seconds of the current time in a 2-digit format with leading zeros, f.e. "00".."59". It resembles an identically named *identifier variable* in Novell NetWare login scripts. See also the similar pseudo-variable %_SECOND%.

%SHORT_YEAR%

This pseudo-variable returns the year of the current date in a 2-digit format with leading zeros, f.e. "93".."99", "00".."92". It resembles an identically named *identifier variable* in Novell NetWare login scripts.

%YEAR% and %_YEAR%

Supported since Novell DOS 7, the %YEAR% pseudo-variable returns the year of the current date in a 4-digit format, f.e. "1980".."2099". It resembles an identically named *identifier variable* in Novell NetWare login scripts. DR-DOS 7.02 and higher added %_YEAR% for compatibility with 4DOS, returning the same value.^[10]

%/%

In COMMAND.COM of DR-DOS 7.02 and higher, this pseudo-variable returns the current SwitChar setting of the system, either "/" (DOS style) or "-" (Unix style).^{[31][32]} See also the related CONFIG.SYS directive SWITCHAR and the environment variable %SWITCHAR%.

%_CODEPAGE%

This pseudo-variable returns the systems' current code page ("1".."65533"), f.e. "437", "850", "858". This variable was originally introduced by 4DOS,^[10] but also became available with COMMAND.COM since DR-DOS 7.02. See also the CHCP command.

%_COLUMNS%

This pseudo-variable returns the current number of screen columns depending on the display mode, f.e. "40", "80", "132", etc. This variable was originally introduced by 4DOS,^[10] but also became available with COMMAND.COM since DR-DOS 7.02. See also a similar environment variable \$_WIDTH% under DOS Plus.

%_COUNTRY%

This pseudo-variable returns the systems' current country code ("1".."65534"), f.e. "1" for USA, "44" for UK, "49" for Germany, "20049" with ISO 8601, "21049" with ISO 8601 and Euro support.^{[32][33]} This variable was originally introduced by 4DOS,^[10] but also became available with COMMAND.COM since DR-DOS 7.02. See also the CONFIG.SYS directive COUNTRY.

%_DAY%

This pseudo-variable returns the days of the current date without leading zeros, f.e. "1".. "31". This variable was originally introduced by 4DOS,^[10] but also became available with `COMMAND.COM` since DR-DOS 7.02. See also the similar pseudo-variable %DAY%.

%_HOUR%

This pseudo-variable returns the hours of the current time in 24-hour format without leading zeros, f.e. "0".. "23". This variable was originally introduced by 4DOS,^[10] but also became available with `COMMAND.COM` since DR-DOS 7.02. See also the similar pseudo-variable %HOUR24%.

%_MINUTE%

This pseudo-variable returns the minutes of the current time without leading zeros, f.e. "0".. "59". This variable was originally introduced by 4DOS,^[10] but also became available with `COMMAND.COM` since DR-DOS 7.02. See also the similar pseudo-variable %MINUTE%.

%_MONTH%

This pseudo-variable returns the months of the current date without leading zeros, f.e. "1".. "12". This variable was originally introduced by 4DOS,^[10] but also became available with `COMMAND.COM` since DR-DOS 7.02. See also the similar pseudo-variable %MONTH%.

%_ROWS%

This pseudo-variable returns the current number of screen rows depending on the display mode, f.e. "25", "43", "50", etc. This variable was originally introduced by 4DOS,^[10] but also became available with `COMMAND.COM` since DR-DOS 7.02. See a similar environment variable \$\$LENGTH% under DOS Plus.

%_SECOND%

This pseudo-variable returns the seconds of the current time without leading zeros, f.e. "0".. "59". This variable was originally introduced by 4DOS,^[10] but also became available with `COMMAND.COM` since DR-DOS 7.02. See also the similar pseudo-variable %SECOND%.

System information variables supported by DR-DOS `COMMAND.COM` with networking loaded:

%LOGIN_NAME%

This pseudo-variable returns the user name. This always worked with NETX, but it will also work with Personal NetWare's ODI/VLM if the current drive is a PNW-mapped drive (otherwise an empty string is returned). See also the similarly named environment variable %LOGINNAME%.

%P_STATION%

This pseudo-variable returns the physical station number in a format "?????????????". The value depends on the MAC address of the network adapter, but can be overridden. It resembles an identically named *identifier variable* in Novell NetWare login scripts.

%STATION%

This pseudo-variable returns the logical station number starting with "1" for the first client. The numbers are assigned by the file server and remain static for as long as the IPX connection remains established. It resembles an identically named *identifier variable* in Novell NetWare login scripts.

%FULL_NAME%

This pseudo-variable returns the full name of the logged in user, if available. It resembles an identically named *identifier variable* in Novell NetWare login scripts. See also the related pseudo-variable %LOGIN_NAME%.

Windows

Dynamic environment variables (also named *internal variables* or *system information variables* under DOS) are pseudo-environment variables supported by `CMD.EXE` when command-line extensions are enabled, and they expand to various discrete values whenever queried, that is, their values can change when queried multiple times even within the same command. While they can be used in batch jobs and at the prompt, they are not stored in the environment. Consequently, they are neither listed by `SET` nor do they exist for external programs to read. They are not case-sensitive.

Indirectly, they are also supported under Windows' `COMMAND.COM`, which has been modified to internally call `CMD.EXE` to execute the commands.

%CD%

This pseudo-variable expands to the current directory equivalent to the output of the command `CD` when called without arguments. While a long filename can be returned under `CMD.EXE` depending on the current directory, the fact that the current directory will always be in `8.3` format under `COMMAND.COM` will cause it to return a short filename under `COMMAND.COM`, even when `COMMAND` internally calls `CMD`.

%CMDCMDLINE%

This pseudo-variable expands to the original startup parameters of `CMD.EXE`, f.e. "`C:\Windows\system32\cmd.exe`". Under Windows' `COMMAND.COM`, this may return something like "`C:\Windows\system32\cmd.exe /c ...`" due to the fact that `COMMAND.COM` calls `CMD.EXE` internally.

%CMDEXTVERSION%

This pseudo-variable expands to the version of the command-line extensions of `CMD.EXE`, if enabled (e.g. "`1`" under Windows NT, "`2`" under Windows 2000 and Windows XP).

%DATE%

This pseudo-variable expands to the current date. The date is displayed according to the current user's date format preferences.

%ERRORLEVEL%

This pseudo-variable expands to the last set error level, a value between "`0`" and "`255`" (without leading zeros).^{[29][34][30]} External commands and some internal commands set error levels upon execution. See also the identically named pseudo-variable `%ERRORLEVEL%` under DR-DOS and the `IF ERRORLEVEL` command.

%HIGHESTNUMANODENUMBER%

This pseudo-variable returns the number of the highest NUMA node.

%RANDOM%

This pseudo-variable returns a random number between "`0`" and "`32767`".

%TIME%

This pseudo-variable returns the current time. The time is displayed according to the current user's time format preferences. If the `%TIME%` and `%DATE%` variables are both used, it is important to read them both in this particular order in rapid succession in order to avoid midnight-rollover problems.

Other shells

Unix-like shells have similar dynamically generated variables, bash's \$RANDOM being a well-known example. However, since these shells have a concept of local variables, they are described as special local variables instead.^[35]

See also

- [Variable \(computer science\)](#)
- [List of Unix commands](#)
- [List of DOS commands](#)
- [Special folder](#)
- [Modules Environment](#)
- [PWB shell](#)
- [Windows Registry](#)

Notes

1. Applications not coping properly with environment variables longer than 128 characters include [CadStar PCB](#), [Checkit](#), [Computer Select CD-ROM](#), [RenderMan](#), [VINES Network](#), [Windows 3.0](#), and [Word for Windows 6.0](#).
2. [4DOS](#) and [MS-DOS 7.0](#) set the length to 127 and insert ASCII-13 at this position, but [Borland](#) libraries set the length to the invalid value of 128 and do not insert an ASCII-13. For shorter command lines, 4DOS/NDOS and DR-DOS also insert an ASCII-0 after the ASCII-13, but not when invoked via [INSTALL](#) directive; MS-DOS does not.

References

1. Robbins, Arnold; Beebe, Nelson H. F. (May 2005). Apandi, Tatiana; Randal, Allison; Witwer, Adam (eds.). *Classic Shell Scripting* (1 ed.). O'Reilly. ISBN 978-0-596-00595-5.
2. "unset - unset values and attributes of variables and functions – " (http://pubs.opengroup.org/onlinepubs/9699919799/utilities/V3_chap02.html#unset). *Commands & Utilities Reference, The Single UNIX Specification - The Open Group Base Specifications, IEEE Std 1003.1-2008* (Issue 7 ed.). The IEEE and The Open Group. 2016 [2001]. Archived (https://web.archive.org/web/20171218175244/http://pubs.opengroup.org/onlinepubs/9699919799/utilities/V3_chap02.html) from the original on 2017-12-18. Retrieved 2017-12-18.
3. fgrose (2015-07-31) [2011]. "The unset builtin command" (<http://wiki.bash-hackers.org/commands/builtin/unset>). *Bash Hackers Wiki*. Archived (<https://web.archive.org/web/20171218175214/http://wiki.bash-hackers.org/commands/builtin/unset>) from the original on 2017-12-18. Retrieved 2017-12-18.
4. Jacobs, Mike; Koren, Alexander; Satran, Michael (2017-11-14). "Store and retrieve settings and other app data" (<https://docs.microsoft.com/en-us/windows/uwp/app-settings/store-and-retrieve-app-data>). Microsoft. Archived (<https://web.archive.org/web/20171218175200/http://docs.microsoft.com/de-de/windows/uwp/design/app-settings/store-and-retrieve-app-data>) from the original on 2017-12-18. Retrieved 2017-12-18.
5. Raymond, Eric (2002-08-02) [Originally published 2001]. "The BROWSER project" (<http://www.catb.org/~esr/BROWSER/index.html>). Archived (<https://web.archive.org/web/2019092123921/http://www.catb.org/~esr/BROWSER/index.html>) from the original on 2019-09-21. Retrieved 2020-10-21.

6. "LWN - Development" (<https://lwn.net/2001/0201/devel.php3>). *LWN.net*. 2001-02-01. Archived (<https://web.archive.org/web/20190208125048/https://lwn.net/2001/0201/devel.php3>) from the original on 2019-02-08. Retrieved 2020-10-21. "Standardizing a BROWSER environment variable. Eric Raymond is promoting the use of a new environment variable, BROWSER, to complement the existing PAGER, MAILER, and EDITOR variables. This effort is being tested as an experiment in hacking social systems."
7. Špaček, Michal (2020-01-19) [First version published 2013]. "Env::Browser-0.06" (<https://metacpan.org/pod/Env::Browser>). *metacpan.org*. Archived (<https://web.archive.org/web/20201021202950/https://metacpan.org/pod/Env::Browser>) from the original on 2020-10-21. Retrieved 2020-10-21. "Env::Browser - Process environment variable \$BROWSER and run web browser"
8. "webbrowser — Convenient Web-browser controller" (<https://docs.python.org/3/library/webbrowser.html>). *docs.python.org*. Archived (<https://web.archive.org/web/20201013205951/http://docs.python.org/3/library/webbrowser.html>) from the original on 2020-10-13. Retrieved 2020-10-21.
9. "DefaultWebBrowser - Debian Wiki" (<https://wiki.debian.org/DefaultWebBrowser>). *wiki.debian.org*. 2020-06-10. Archived (<https://web.archive.org/web/20201021204658/http://wiki.debian.org/DefaultWebBrowser>) from the original on 2020-10-21. Retrieved 2020-10-21. "The environment variable BROWSER can be set to[...] always choose your desired browser."
10. Brothers, Hardin; Rawson, Tom; Conn, Rex C.; Paul, Matthias R.; Dye, Charles E.; Georgiev, Luchezar I. (2002-02-27). *4DOS 8.00 online help*.
11. Paul, Matthias R. (1997-07-30) [1994-05-01]. *NWDOS-TIPs — Tips & Tricks rund um Novell DOS 7, mit Blick auf undokumentierte Details, Bugs und Workarounds* (<http://www.antonis.de/dos/dos-tuts/mpdostip/html/nwdostip.htm>). Release 157 (in German) (3 ed.). Archived (<https://web.archive.org/web/20161104235829/http://www.antonis.de/dos/dos-tuts/mpdostip/html/nwdostip.htm>) from the original on 2016-11-04. Retrieved 2014-08-06. {{cite book}}: |work= ignored (help) (NB. NWDOSTIP.TXT is a comprehensive work on Novell DOS 7 and OpenDOS 7.01, including the description of many undocumented features and internals. The provided link points to a HTML-converted version of the file, which is part of the MPDOSTIP.ZIP collection.) [1] (<https://web.archive.org/web/20190601152204/https://www.sac.sk/download/text/mpdostip.zip>)
12. Paul, Matthias R. (1997-05-01) [1995-03-01]. "Hinweise zu JPSofts 4DOS 5.5b/c, 5.51, 5.52a und NDOS" (<http://www.antonis.de/dos/dos-tuts/mpdostip/html/4dostip.htm>). *MPDOSTIP* (in German). Archived (<https://web.archive.org/web/20161104211143/http://www.antonis.de/dos/dos-tuts/mpdostip/html/4dostip.htm>) from the original on 2016-11-04. Retrieved 2015-05-08. (NB. The provided link points to a HTML-converted version of the 4DOS5TIP.TXT file, which is part of the MPDOSTIP.ZIP collection.) [2] (<https://web.archive.org/web/20190601152204/https://www.sac.sk/download/text/mpdostip.zip>)
13. Paul, Matthias R. (1997-10-02). "Caldera OpenDOS 7.01/7.02 Update Alpha 3 IBMBIO.COM README.TXT" (<https://web.archive.org/web/20031004074600/http://www-student.informatik.uni-bonn.de/~frinke/ibmbioa3.zip>). Archived from the original (<http://www.uni-bonn.de/~uvs180/download/ibmbioa3.zip>) on 2003-10-04. Retrieved 2009-03-29. [3] (<https://web.archive.org/web/20181225154705/http://mirror.macintosharchive.org/max1zzz.co.uk/+Windows%20%20DOS/DOS/System/Novell/Support/Bins/Op702src.zip>)
14. Paul, Matthias R. (1997-05-27) [1996]. "SETENV v1.11" (<https://web.archive.org/web/20090215001349/http://www.uni-bonn.de/~uvs180/mpdostip/html#M.SETENV>). Archived from the original (<http://www.uni-bonn.de/~uvs180/mpdostip/html#M.SETENV>) on 2009-02-15. Retrieved 2019-08-09. "[...] SETENV [...] to hide and later restore the [...] pre-environment [...] By using SETENV.COM you can save some KiloBytes of rare DOS memory [...] depending on the number of drivers loaded by `INSTALL=/INSTALLHIGH=/HIINSTALL=` and the current size of the pre-environment. [...] this original [...] feature cannot be found in any known memory manager/optimizer. [...]"

15. Paul, Matthias R. (2002-04-06). "Re: [fd-dev] ANNOUNCE: CuteMouse 2.0 alpha 1" (<https://marc.info/?l=freedos-dev&m=101807226917577>). *freedos-dev*. Archived (<https://web.archive.org/web/20200207130948/https://marc.info/?l=freedos-dev&m=101807226917577&w=2>) from the original on 2020-02-07. Retrieved 2020-02-07. "[...] In CTMOUSE.ASM prepareTSR routine I found a comment in regard to the zero-environment. [...] DESQview or DRDOS zero-env? [...] release environment block [...] skip if any problem [...] zero-pad for MEM style utilities [...]"
16. "PTS-DOS 2000 Pro User Manual" (http://download.paragon-software.com/doc/manual_dos_eng.pdf) (PDF). Archived (https://web.archive.org/web/20180512094512/http://download.paragon-software.com/doc/manual_dos_eng.pdf) (PDF) from the original on 2018-05-12. Retrieved 2018-05-12.
17. Paul, Matthias R. (2002-10-07). "Re: Run a COM file" (<https://groups.google.com/d/msg/alt.msdos.programmer/d7bIJY0H5M/Qu3VeTOIGVcJ>). Newsgroup: alt.msdos.programmer (news:alt.msdos.programmer). Archived (<https://archive.today/20170903230312/https://groups.google.com/forum/%23!msg/alt.msdos.programmer/d7bIJY0H5M/Qu3VeTOIGVcJ>) from the original on 2017-09-03. Retrieved 2017-09-03. [4] (<https://groups.google.com/d/msg/alt.lang.asm/PNOd9zfYow0/vXbab16j4XwJ>)
18. Datalight, Inc.; GPvNO (April 2005) [1999]. "Datalight ROM-DOS User's Guide" (https://www.datalight.com/assets/files/ROM-DOS_Users_Guide.pdf) (PDF). Datalight, Inc. 3010-0200-0716. Archived (https://web.archive.org/web/20190320151321/https://www.datalight.com/assets/files/ROM-DOS_Users_Guide.pdf) (PDF) from the original on 2019-03-20. Retrieved 2018-09-16.
19. Kotulla, Martin (November 1987). "Von CP/M zu MS-DOS, Teil 11" (<http://www.homecomputerworld.at/magazine/cpc/CPCAI-87-11.pdf>) (PDF). *Professional Computing (PC) - Schneider International* (in German). **3** (11): 100–103. Archived (<https://web.archive.org/web/20190424201007/http://www.homecomputerworld.at/magazine/cpc/CPCAI-87-11.pdf>) (PDF) from the original on 2019-04-24. Retrieved 2018-05-20.
20. Paul, Matthias R. (2002-03-26). "Updated CLS posted" (<https://marc.info/?l=freedos-dev&m=101717593306186&w=2>). freedos-dev mailing list. Archived (<https://web.archive.org/web/20181004145539/https://marc.info/?l=freedos-dev&m=101717593306186&w=2>) from the original on 2018-10-04. Retrieved 2014-08-06.
21. *Caldera DR-DOS 7.02 User Guide* (<https://web.archive.org/web/20161104235434/http://www.drdsos.net/documentation/usergeng/uglontoc.htm>). Caldera, Inc. 1998 [1993, 1997]. Archived from the original (<http://www.drdsos.net/documentation/usergeng/uglontoc.htm>) on 2016-11-04. Retrieved 2013-08-10.
22. Paul, Matthias R. (2017-08-14) [2017-08-07]. "The continuing saga of Windows 3.1 in enhanced mode on OmniBook 300" (<http://hpmuseum.org/forum/thread-8774-post-77196.html#pid77196>). *MoHPC - the Museum of HP Calculators*. Archived (<https://web.archive.org/web/20180501185933/http://hpmuseum.org/forum/thread-8774-post-77196.html>) from the original on 2018-05-01. Retrieved 2018-05-01. "[...] set DRSYS=ON (optional to tell SYS you are aware of the fact that you're running it in a foreign environment and want to proceed anyway without having to individually ACK some warnings and extra info screens displayed in this scenario otherwise) [...]"
23. *CCI Multiuser DOS 7.22 GOLD Online Documentation*. Concurrent Controls, Inc. (CCI). 1997-02-10. HELP.HLP.
24. *CCI Multiuser DOS 7.22 GOLD Installation Guide*. Concurrent Controls, Inc. (CCI). 1997-02-10. PRINTDOC.HLP.
25. Paul, Matthias R. (2002-02-20). "How to detect FreeCOM/FreeDOS in-batch?" (<https://marc.info/?l=freedos-dev&m=101423986407077&w=2>). freedos-dev mailing list. Archived (<https://web.archive.org/web/20181106220459/https://marc.info/?l=freedos-dev&m=101423986407077&w=2>) from the original on 2018-11-06. Retrieved 2014-08-06.

26. "Recognized environment variables - Windows Deployment" (<https://learn.microsoft.com/en-us/windows/deployment/usmt/usmt-recognized-environment-variables>). 2024-01-09.
27. Brink, Shawn (2018-11-17). "Environment Variables in Windows 10 - Complete List of Environment Variables in Windows 10" (<https://www.tenforums.com/tutorials/3234-environment-variables-windows-10-a.html>). *Windows Ten Forums*. Archived (<https://web.archive.org/web/20190327135346/https://www.tenforums.com/tutorials/3234-environment-variables-windows-10-a.html>) from the original on 2019-03-27. Retrieved 2019-04-24.
28. Schulz, Hajo (2014-10-02). "Ordner für spezielle Fälle — Die "benannten Ordner" in Windows aus User- und Entwicklersicht" (<https://www.heise.de/ct/ausgabe/2014-22-Die-benannten-Ordner-in-Windows-aus-User-und-Entwicklersicht-2406251.html>) [Special purpose folders — Windows' "named folders" from a user's and developer's perspective]. *c't - magazin für computertechnik* (in German). **2014** (22). Heise Verlag: 180–184. Archived (<https://web.archive.org/web/20160709114401/https://www.heise.de/ct/ausgabe/2014-22-Die-benannten-Ordner-in-Windows-aus-User-und-Entwicklersicht-2406251.html>) from the original on 2016-07-09. Retrieved 2019-04-08.
29. Paul, Matthias R. (1997-05-01) [1993-10-01]. *BATTIPs — Tips & Tricks zur Programmierung von Batchjobs* (<http://www.antonis.de/dos/batchtut/battips/index.htm#7>) (in German). 7: ERRORLEVEL abfragen. Archived (<https://web.archive.org/web/20170823191411/http://www.antonis.de/dos/batchtut/battips/index.htm>) from the original on 2017-08-23. Retrieved 2017-08-23. {{cite book}}: |work= ignored (help) [5] (<https://www.auersoft.eu/soft/by-others/dos-exitcodes-de.html>) [6] (<https://www.auersoft.eu/soft/by-others/dos-exitcodes-en.html>) Archived (<https://archive.today/20170911103337/https://www.auersoft.eu/soft/by-others/dos-exitcodes-en.html>) 2017-09-11 at archive.today (NB. BATTIPS.TXT is part of MPDOSTIP.ZIP. The provided link points to a HTML-converted older version of the BATTIPS.TXT file.) [7] (<https://web.archive.org/web/20190601152204/https://www.sac.sk/download/text/mpdostip.zip>)
30. Auer, Eric; Paul, Matthias R.; Hall, Jim (2015-12-24) [2003-12-31]. "MS-DOS errorlevels" (<https://web.archive.org/web/20151224202118/http://www.freedos.org/technotes/technote/207.html>). Archived from the original (<http://www.freedos.org/technotes/technote/207.html>) on 2015-12-24.
31. Paul, Matthias R. (1998-01-09). *DELTREE.BAT R1.01 Extended file and directory delete* (<https://web.archive.org/web/20190408145354/http://www.lookas.net/ftp/incoming/darbui/Justas/DRDOS/DELTREE.BAT>). Caldera, Inc. Archived from the original (<http://www.lookas.net/ftp/incoming/darbui/Justas/DRDOS/DELTREE.BAT>) on 2019-04-08. Retrieved 2019-04-08.
32. *DR-DOS 7.03 WHATSNEW.TXT — Changes from DR-DOS 7.02 to DR-DOS 7.03* (<https://web.archive.org/web/20190408142232/http://www.lookas.net/ftp/incoming/darbui/Justas/DRDOS/WHATSNEW.TXT>). Caldera, Inc. 1998-12-24. Archived from the original (<http://www.lookas.net/ftp/incoming/darbui/Justas/DRDOS/WHATSNEW.TXT>) on 2019-04-08. Retrieved 2019-04-08.
33. Paul, Matthias R. (2001-06-10) [1995]. "DOS COUNTRY.SYS file format" (<http://www.columbia.edu/~em36/wpdos/MatthiasPaulCPI.zip>) (COUNTRY.LST file) (1.44 ed.). Archived (<https://web.archive.org/web/20160420065252/http://www.columbia.edu/~em36/wpdos/MatthiasPaulCPI.zip>) from the original on 2016-04-20. Retrieved 2016-08-20.
34. Allen, William; Allen, Linda. "Windows 95/98/ME ERRORLEVELs" (<https://web.archive.org/web/20110707113138/http://www.allenware.com/mcsw/errorlevels.zip>). Archived from the original (<http://www.allenware.com/mcsw/errorlevels.zip>) on 2011-07-07.
35. "Shell Variables" (<https://www.gnu.org/software/bash/manual/bash.html#Shell-Variables>). *Bash Reference Manual*.

Further reading

- Chen, Raymond (2010-05-06). "What are these strange =C: environment variables?" (<https://blogs.msdn.microsoft.com/oldnewthing/20100506-00/?p=14133/>). *The New Old Thing*. Microsoft. Archived (<https://web.archive.org/web/20180527163745/https://blogs.msdn.microsoft.com/oldnewthing/20100506-00/?p=14133%2F>) from the original on 2018-05-27. Retrieved 2017-05-27.

External links

- [environ\(7\)](https://manned.org/envron.7) (<https://manned.org/envron.7>): user environment – [Linux Programmer's Manual](#) – Overview, Conventions and Miscellanea
- [environ\(7\)](https://www.freebsd.org/cgi/man.cgi?query=environ&sektion=7) (<https://www.freebsd.org/cgi/man.cgi?query=environ&sektion=7>) – [FreeBSD Miscellaneous Information Manual](#)
- [environ\(7\)](https://keith.github.io/xcode-man-pages/envron.7.html) (<https://keith.github.io/xcode-man-pages/envron.7.html>) – [Darwin and macOS Miscellaneous Information Manual](#)
- [environ\(7\)](https://docs.oracle.com/cd/E88353_01/html/E37853/envron-7.html) (https://docs.oracle.com/cd/E88353_01/html/E37853/envron-7.html) – Solaris 11.4 Standards, Environments, Macros, Character Sets, and Miscellany Reference Manual
- "Environment Variables Wiki" (<https://web.archive.org/web/20190427235415/http://environmentvariables.org/>). Archived from the original (<http://environmentvariables.org/>) on 2019-04-27.
- [User Environment Variables](http://msdn.microsoft.com/en-us/library/windows/desktop/bb776899%28v=vs.85%29.aspx) (<http://msdn.microsoft.com/en-us/library/windows/desktop/bb776899%28v=vs.85%29.aspx>)
- [fix setx.exe not found bug](https://web.archive.org/web/20180511150138/https://www.setx.cc/) (<https://web.archive.org/web/20180511150138/https://www.setx.cc/>)

Retrieved from "https://en.wikipedia.org/w/index.php?title=Environment_variable&oldid=1241150222"

▪