BATSH Version 2.20

To run WINDOWS commands from a text file. Line by Line.
Like BATCH (.BAT) files in DOS,
but with some WINDOWS specific commands,
and not all the DOS features.

For WINDOWS 3.1 & WINDOWS95

Click one of the following help topics to get more information:
Installation
Commands
Variables
Release Notes
License
Hints

Thomas Nyffenegger / nyffenegger@fmi.ch
http://www.fmi.ch/groups/ThomasNyffenegger/Group.html

WINDOWS and DOS stand for the PC operating systems of Microsoft Corporation.

Installation

Place the files **BATSH.EXE** and **BATSH.HLP** together anywhere on the disk.

Write with NOTEPAD a text file with one command per line.
Save the text file with the extension .BSH
Windows 3.1: Associate with File-Manager the program BATSH.EXE. [*File - Associate*...]. Include the program path in the association.
To install your command-file in Program-Manager, you can drag the file with the mouse from File-Manager to a Program-Manager group.For this procedure, both File- and Program-Manager must be visible on the WINDOWS screen.
You can change the icon shown in Program-Manager with the menu [*File - Properties*...]
Windows95: Click with the right mouse button on the new file and choose [Open With] from the menu.
Select BATSH as program to use. If the BATSH program is not shown in the selection, you must search it with the [Other..] option.
Activate the box [Always use this program...]

Run your new command-file with a double click on the file-name.

The default text file editor supplied with WINDOWS.
Any other writing program can be used. The file must be **saved as** type Text-Only (without extra formatting characters).

License

I give this program away as **freeware**.
It's features will never reach the standard of a commercial product, but it may suit you, if you need a few commands only.

You are not allowed to modify the program,
sell it (except distribution-costs),
or use it as part of a commercial software package.

The program can be distributed without prior request as part of a public domain software library.

The Author has no warranty, obligations or liability for any problems that may be encountered using this program.

Thomas Nyffenegger, FMI, PBX 2543, 4002 Basel, CH
Email: nyffenegger@fmi.ch
Web: http://www.fmi.ch/groups/ThomasNyffenegger/Group.html

Hints

Run BATSH command files within a BATSH file with the command:

```
BATSH.EXE file-name parameter
```

If WAIT is ON (default), the processing waits for the branched BATSH file to return (like the ***CALL*** command in DOS).

For missing DOS commands you can use BATSH to run DOS batch-files.
To hide running DOS commands you must create a PIF file with the settings:
'display windowed' and 'close on exit'.
Use the BATSH commands RUN HIDE or RUN ICON to run the program.
See the PRINT command for an example using internal DOS commands.

Be careful with searching window titles. The command:

```
LABEL Waiting to close Clock
WAITCLOSE Clock
```

is waiting for the BATSH icon closing and not for the Clock program.

Due to variable substitution, the character % must be typed in BATSH files as %%

At runtime, each command-line is stripped from leading and trailing SPACE and TAB characters. This allows to structure the file, but it also restricts parameters not ending with a SPACE or TAB. (It does not affect MESSAGE lines)

Windows 3.1: For more control you can use the WINDOWS supplied RECORDER and record a macro. Assign a shortcut-key (single function-key) and store the macro in a file. It can be run from BATSH as RECORDER -H shortcut-key macro-file.

.

Release Notes

Version 2.20:
The ICON command can change WINDOWS95 taskbar icons.
A network connection-name (NETADD command) may contain spaces.
The hidden icon label in command-mode (/C) is uppercase.
Support for Windows95 is added in the help-file.

Version 2.14:
Bug-fix:
The program is now running stable under WINDOWS95.
This is not a 95 update, since some commands (ICON, WAITDROP, PRINTER...) can not be used yet.
INI Variables
The filename-part can be written with an extension, so any text file with the structure of a windows initialisation file can now be used.

Version 2.12:
Bug-fix:
Under some conditions, the PRINT command kept waiting much longer than it took to print the file.

Version 2.11:
Bug-fix:
A disabled Help option ($HELP) got active after using the MENU command.
The error-message for an invalid $HELP variable definition was incorrect.
CLOSE (not QUIT) a BATSH instance with an open message-box, did produce a system error.
Each SHOW command did rearrange the desktop icons.
The $NET variable was not reset for all NETADD and NETPORT/REMOTE conditions.
In some networks, the IF REMOTE command did not search all drives and ports, due to non standard return codes.

New:
$TIME/$DATE The system TIME & DATE is read or set using these variables.
SHOW New parameters allow to move and size a window, or define the location of a BATSH message-box.

Index of Commands

In command descriptions, optional parameters are marked with [ ] brackets. For using the option, you must type it without the brackets.
For commands returning an ERRORLEVEL, the value for successful operation is 0.
All commands with filename parameters use the DOS 8.3 names.

;	comment line
/C /M /E	command-line parameters
CD	change current directory
CLOSE	ask a window application to close [with save]
COPY	copy a file - single files only
DEL	delete a file - single file only
EXACT	toggle case sensitivity for text parameters
EXIT	stop processing and quit BATSH.EXE.
file	run a DOS or WINDOWS program
FONT	toggle fixed- or variable-pitch display font
GOTO	jump with processing to a label
ICON	change the icon image
IF	conditions to branch program execution
INC	increase/decrease the value of a numeric variable
LABEL	change the title of the BATSH.EXE icon
MD	make a new directory
MENU	define own BATSH menu options
MESSAGE	display a message box for user response
NETADD	add a network connection
NETSTOP	stop network connection
PLAY	play WAVE sound file
PRINT	print a print-file (Windows3.1 only)
PRINTER	set WINDOWS default printer
QUIT	ask an application to quit [w/o save]
RD	remove a directory
REM	comment line
RESTART	restart WINDOWS
RUN	run a DOS or WINDOWS program
SET	set or remove a variable
SHOW	modify the display status of a window (or BATSH icon)
SPLIT	split a variable in two parts.
TRACE	display each command (for debugging)
WAIT	wait toggle or wait for a specific event
WAITCLOSE	wait until a window is closed
WAITOPEN	wait until a window is open

REM

REM *txt*
;*txt*

3 types of comment lines, including empty lines

txt	comment

/C /M /E

Start-up parameter

BATSH.EXE /C *command*
With this parameter, BATSH can run a single command from the command-line. The BATSH icon will be hidden and the command is used as hidden icon-label (upper-case).
BATSH.EXE /M *command*
BATSH can load all the commands in one step into memory.
This allows you to run BATSH files from removable disks.
BATSH.EXE /E *command*
For compatibility with previous BATSH versions, the start-up command-line for BATSH is case-sensitive only when using the new /E switch.

The M and/or E parameter can be set as default parameter with a BATSH.INI initialisation file in the WINDOWS directory. The entry for the parameters is:

```
[default]
parameter=M E
```

The command **SET batsh|default|parameter=ME** can do this for you.

CD , MD , RD

CD *txt*
change current directory

MD *txt*
make new directory

RD *txt*
remove existing directory
(the directory must be empty)

txt directory or path name

The variable $CD holds the value of the current directory.

Return value:
errorlevel 2 for errors

CLOSE
QUIT

CLOSE *txt*
(close window with title -*txt*- [with save])

QUIT *txt*
(quit window with title -*txt*- [w/o save])

txt: substring of a window title

Return value
ERRORLEVEL 2 if window not found

COPY

COPY *file1 file2*
(copy a file - single files only)

file1 (source*)*, ***file2*** (destination) - full DOS-filename

! existing files will be overwritten !
! compressed (MS compress) files will be expanded !

Use the XCOPY command from DOS to copy groups of files.

Return value:
errorlevel 2 if file not found
errorlevel 3 for copy errors

DEL

DEL *file*
(delete a file - single file only)

! if possible, access restrictions are reset before deletion

Return value:
errorlevel 2 if file not found
errorlevel 3 for file access errors

EXACT

EXACT *sw* or
EXACT=*sw*

sw	ON	(default)
	OFF	

When referring to window-titles the *txt* parameters are treated case sensitive by default. The EXACT command can toggle this on/off

EXIT

EXIT

stop processing and close the current instance of the program.

EXIT is done automatically at the end of a BATCH command-file.

file

all commands that are not recognised as BATSH commands, are used as DOS or WINDOWS program names.

> ***file***
> program filename with optional start-up parameters (including PIF and BAT files)
> or filename with a known extension (assigned to a program).

The start-up directory for the program, can be defined with the BATSH CD command.

If wait is on (default), processing waits for launched program to close.

Return value:
errorlevel 2 if file not found
errorlevel >2 for other errors
see also: run

FONT

FONT=var (default)
FONT=fixed

set message-box font to variable-pitch spacing (default), or fixed character width.

GOTO

GOTO *txt*
:*txt*

Jump with processing to a line with the same text as label.
Label lines start with a colon **:*txt***

(never case sensitive)

ICON

ICON [*file*]
(change the icon image)

file: (optional) the first icon from this file is loaded.

without file parameter, the default BATSH icon is assigned.

WINDOWS default icons are loaded with the reserved words
STOP
INFO
EXCLAMATION
QUESTION

Return value:
errorlevel 2 if file not found.
errorlevel 3 if icon is invalid.

IF

IF [not] condition statement

Specify the conditions under which a statement will be executed

conditions:

ERRORLEVEL *number*

true for equal or higher errorlevel

EXIST *file*

for single file names w/o path

the function searches a matching file in the following directories (in this order): The current directory, the Windows directory , the Windows system directory, the BATSH.EXE directory, the DOS PATH.
The full filename with path is returned in $FILE.

for single files with full path the exact location is searched.
if the parameter *file* is valid for a group of files, the first matching name will be assigned to $FILE
the variables $FILE_N , $FILE_P will hold the file-name and file-path section of $FILE
the file-type is analysed and set to the variable $TYPE
the following file-types are recognised:

PS	postscript
PCL	HP-Printer language
BIN	binary data or program file
TXT	PC text file
UNIX	text file with LF as end-line
MAC	text file with CR as end-line
ERROR	file access is denied or its a directory-name
OTHER	file access was OK, but data did not give a result.

NETPORT *dev*

check if the device-name -dev- is a remote device.
the network name is returned in the variable $NET

dev : device name   LPT1-9 or D:-Z:

sets errorlevel 2 if network not installed or network-error

REMOTE *txt*

txt : substring of a remote service
returns the corresponding device name in $NET
sets errorlevel 2 if network not installed or network-error

WINDOW *txt*

txt: window title substring
in quotes if it includes spaces
or double-quotes if it contains quotes

string1==string2   - compare two strings

If both strings are numeric, the expression **number==>number** does a greater-equal comparison.

statement :   BATSH command

INC

INC var [+/- value]
(increase/decrease the value of a variable)

var: BATSH variable with numeric value (32-bit)

value: (optional) positive or negative number
The default value is +1

Return value:
errorlevel 2 if the variable is not numeric.

LABEL

LABEL [*txt*]

change the title of the BATSH.EXE icon
txt : new title (default is the filename of the running script file)

MESSAGE

MESSAGE [*lbl*]
[*txt*]
[@*file*]
[*txt*]
.[*endmessage parameters*]

This command is **replacing** the commands **ASK** and **CHOICE** from release 1.53

Multiple lines of text ***txt*** can be displayed.
The window is sized automatically, up to a maximum defined by the screen size.
A scroll-bar will allow more lines than the display can show.
The position of the window can be defined with the SHOW command.

The message definition start-line: **MESSAGE [*lbl*]**
lbl is an optional message box title (default MESSAGE)

The message lines
txt single or multiple text lines.
a line should not exceed the screen width
variables get translated
@file for any line starting with @ and a filename
the file is checked for type TXT and displayed.

The message definition end-line
the end-line must start with a dot (and with optional parameters)

- the end of the message lines.
until here, all lines are treated as message.

[*endmessage parameters*]
The user response to the message is defined with the last line. It can be a BUTTON, a KEY or an INPUTFIELD with BUTTON.

The message box with an OK button:

.[*lbl*],[*sec*].
bl The default button label is OK.
You may type your own button label (except ask and choice).
sec optional time-out value in seconds. (default 20 seconds)
A value of 0 will disable the time-out function.

The message box with an INPUT field and OK button:

.ASK *var*
let the user assign a value to a variable.
The old value of an existing variable is given as default.
var variable name to assign response.(For the variable $PASSWORD, typed characters are shown as *).

Return value:
errorlevel 2 if empty

The message box with a KEY press response:

.CHOICE k1,k2,k3
select by different keys -with errorlevel returned-
not exactly the DOS 6.x equivalent
k1 is the default key on time-out -errorlevel 1-

you must mention the options in the text message, the key options are not shown)

k1-k9 a key letter (k1 is the default key)
(never case sensitive)

Return value

errorlevel according to the position on the command line: k1 ->1 k2 ->2

Special keys:

$S Space
$E Enter
$A any other key that is not in the key list -except Alt
# and number - time-out value in seconds

MENU

menu [*mnu*] [*txt*]

Up to two menu options can be defined for a BATSH icon.

mnu name that is shown
in quotes if it includes spaces
or double-quotes if it contains quotes

txt menu-action (a file or a program).
see $HELP and $CLOSE description.

Additional menu definitions will replace the second one.

Examples:

option to start Desktop Settings

```
menu 'Desktop Settings'   control.exe desktop
```

option to play the solitaire game from windows

```
menu "It's game time"   sol.exe
```

erase all user options

```
menu
```

erase the 'Desktop Settings' menu

```
menu 'Desktop Settings'
```

NETADD

NETADD *dev txt* [*pass*]

new network connection or reconnect

dev device name LPT1-LPT9 or D:-Z: ?:
LPT4: - **LPT9:** are not supported in all the networks.
?: takes the first free disk name for connection
and returns the connected drive name in $NET

txt connection string
\\server-name\service-name
or \\server\service%%user-name with user-name
in quotes if it includes spaces
or double-quotes if it contains quotes

pass optional password

Return value:
errorlevel >0 for various network errors

Note:
Microsoft Network, has a limit of 12 characters for the service-name (including optional user-name).

NETSTOP

NETSTOP [QUIT] *dev*

stop network connection

dev device name LPT1-9 or D:-Z:

QUIT use this keyword to ignore open files and print-jobs (forced disconnection)

Return value:
errorlevel >0 for various network errors

PLAY

PLAY *file*

***file*: **WAVE sound file

Return value:
errorlevel 2 if file not found

PRINT (For Windows 3.1 only / See note below for WINDOWS95)

PRINT *file*

!! file gets deleted !!

print a print-file to the WINDOWS 3.1 defined default printer

file name of a file with raw-data.

Return value:
errorlevel 2 if file not found or spooler initialisation error.

!!! The file is deleted automatically after printing. If you want to keep it, you must copy it first **!!!**

Restrictions:
- This function is using Print-Manager. Some printer drivers do not print with the spooler active, and can therefore not be used with this command.

- It can print to Local Ports only (LPT,COM). Not to another file.
- Printing to ports listed (in printer-control) as *not present* , will not return an error
- BATSH processing stops until the print-manager is closed or the print-job is processed. PRINTMANAGER errors will occur, if the printing BATSH instance is closed while waiting (via visible BATSH-icon or by remote command).

WINDOWS95:
Print-Manager is not used in WINDOWS95 printing.
But DOS printing is now spooled by WINDOWS and you can use the DOS copy command to print a file:

RUN ICON COMMAND /C COPY /B *file* LPT1:

PRINTER

PRINTER # *txt*

set WINDOWS default printer

#	number 1..9 for output port LPT1: - LPT9:
txt	substring of an installed printer driver

Return value:
errorlevel 2 if printer driver not found or printer not defined

Note:
The default printer can not be set with an active Printer-Settings window.
You can save and restore the current Default-Printer with the INI-variable win|windows|device

WINDOWS95:
The default printer setting from BATSH may not work for a WINDOWS95 specific application, when it reads the setting from the new DAT file and not from WIN.INI.

RESTART

RESTART [*file*]

restart WINDOWS

The optional parameter ***file*** is the path and filename of a DOS program-file to run, after Windows has been terminated and before WINDOWS restarts.

Return value:
errorlevel 2 if an application refuses to close

Note:
DOS batch (.BAT) files must be run as *COMMAND /C or /K file*.
See the DOS help-file for further help.

RUN

RUN [*sw*] *txt*

To define the display status at start-up of a program, and to activate programs with names, identical to one of the BATSH commands.

sw (optional)

HIDE	hide program -*txt*
ICON	minimise program -*txt*
NORMAL	default window-size
FULL	run program full screen

(some programs e.g. BATSH use their on start-up display status)

txt program filename with optional start-up parameters
or filename with a known extension (assigned to a program file).

The start-up directory for the program, can be defined with the BATSH CD command.

If WAIT is ON (default), processing waits for launched program to close

Return value:
errorlevel 2 if file not found, >2 for other errors
see also: file

SET

set variable to value *txt*

SET *var=txt*
If the variable already exists, the new ***txt*** value replaces the old one.

remove variable

SET *var=*

The DOS environment variables defined at WINDOWS start-up can be read, but not set.
You can define BATSH variables with DOS-variable names, but they will only be valid for the current BATSH file (not for launched programs).

Note:
The ***txt*** parameter can not have leading and trailing SPACES or trailing TABS. To store such a value, you must add an extra character and remove it later with the SPLIT command.

SHOW

SHOW *sw* [*txt*]

modify the display status of an existing window, or the BATSH icon.

sw	HIDE	hide window -*txt*
	ICON	minimise window -*txt*
	NORMAL	restore window
	FULL	set window -*txt*- full screen
	x:y[:w:h]	move upper-left corner of window -*txt*- to position x:y (in percent of the screen-size). Optional zoom window to size w(width):h(height).

txt (optional) substring of a window title

Note:
Without ***txt*** parameter

the BATSH icon is hidden or visible

the x:y parameter defines the location of subsequent MESSAGE windows. A value of 0:0 will reset to the default centre position.

The SHOW command without parameter rearranges the desktop icons in WINDOWS 3.1.

Return value:
errorlevel 2 if window not found

SPLIT

SPLIT var [txt]

Search the first occurrence of a word (txt), in the variable var. The result is stored (without the matching word) in the special variables **$P1** - part1 and **$P2** - part2.

var: BATSH variable
txt: search word/character or number

If **txt** is numeric, SPLIT will divide the variable at the given position.

For using numbers as text-search, or words with spaces, you must put the **txt** parameter in quotes or double-quotes.

Depending on the EXACT state, the SPLIT command will treat upper- and lower-case letters as different characters (default).

TRACE

TRACE *sw* or
TRACE=*sw*

each command is shown in a message-box - before execution

(for debugging purpose)

sw	ON
	OFF (default)

WAIT

WAIT *sw*

as on/off toggle

sw ON (default) wait for launched programs to be closed
OFF

as wait for a specific event

sw DROP (WINDOWS 3.1 only)
Wait until a file is dropped on BATSH-icon
The filename is returned in the variable $DROP.
A hidden BATSH icon is made visible before the wait.
For multiple files, the name of the first file is kept.
sw time in seconds - wait processing

WINDOWS95:
The DROP feature in Windows95 is no longer supported for ICONS (Taskbar).

WAITCLOSE

WAITCLOSE [*sec*] *txt*

wait until window with title -***txt***- is closed

(if more than one title matches, the first is monitored)

sec	time-out in seconds (optional)
txt	substring of a window title

Return value:
errorlevel 2 if window not found, errorlevel 3 if time-out reached.

WAITOPEN

WAITOPEN [*sec*] *txt*

wait until window with title ***txt*** is open

sec	time-out in seconds (optional)
txt	substring of a window title

Return value:
errorlevel 2 if window is already open
errorlevel 3 if time-out reached.

VARIABLES

BATSH variables can be at any place on a line. (except see note below)
Before executing a command, all text-parts with variable reference are replaced with the variable value.
The variable reference for command line (start-up) parameters is:

%0 - %9 (%**0** is the command-file with path)

The value of other variables is referred to with the expression:

%var%
var variable name

All variable names are stored in uppercase, but the values may have lower- and upper-case text.

Variable Types:
Commandline variables

See above.

DOS variables

You can read the DOS environment variables as normal variable reference.

INI file variables

file|section|keyname

BATSH global variables

$$*var*

Special Variables

Local variables

All variable names that are not of one of the above type, are used as local variables. They are only valid for the running BATSH file.

Note:
The statement part from an IF command must start with a valid BATSH command.
Example:

If errorlevel 2 **goto** %var%
and not: If errorlevel 2 %var%

Special Environment Variables

Special variables are variables connected to a program function. Their value can be accessed and set by internal functions.
The names are reserved variable names.

$DATE
Read and set the system date: month/day/year   without leading zeros.
The year must be in the range 1980-2099
$TIME
Read and set the system time: hour:minute:second
In 24h format and without leading zeros.

$FILE , $FILE_N , $FILE_P
$TYPE
These variables are set from a successful IF EXIST command.

$DROP (Windows 3.1 only)
Has the first file-name from a WAIT DROP command stored.

$PASSWORD
When used with the   MESSAGE ASK   command, typed characters are shown as * hidden characters in the input field.

$P1 , $P2
The two parts of a variable as result from a SPLIT command

$HELP   default: BATSH.HLP
$CLOSE   no default
The variable $HELP and $CLOSE define a file or a program that activates, when the user selects the corresponding menu option. Programs (not files) may have parameters.
You can disable the help option by setting $HELP to empty.

$INSTANCE
Number of BATSH programs running.

$NET
- device name from an IF REMOTE or NETADD command.
- network name from an IF NETPORT command.

$CD
The current disk and directory (set with the CD command)

$VER
The version number (3 digits)

Global Variables

$$var

Variables starting with **$$** are kept in WINDOWS memory (global).
They can be accessed from each instance of the BATSH.EXE program, and get deleted only when WINDOWS is stopped, or when they are set to empty with the command:

SET $$var=

Note:
The global space is 256 characters only (for names and assigned values).

For programmers:
The global variables are stored as GlobalAtom String
The format is   var=value[tab]var=value[tab]
The actual handle for the GlobalAtom is stored in the WIN.INI file as

```
        [BATSH]
        Global.20=
```

You must verify the handle before using it. In case of an invalid string, all global-variables are empty.

INI Variables

file|section|keyname (the character **|** is ASCII # 124)

With this type of variables, you can access initialisation files from WINDOWS and other applications.

You read information by referring the variable in a command

%*file|section|keyname*%

and you write or delete entries with the SET command

SET *file|section|keyname*=value
SET *file|section|keyname*= (to delete *keyname*)
SET *file|section*|= (to delete whole *section*)

As **filename** you can use the name without the extension. The default file-extension is **.INI**.
When you set a new INI variable, all parts that are not yet defined will be created. The default directory for the files is the WINDOWS directory.

Example: Reading the current language setting:

%system|boot.description|language.dll%

Windows and Windows applications use initialisation files to configure themselves according to setting in these files.

Format of initialisation files:

The files are structured into groups called *sections*. Each section has the format:

```
[section]
keyname=value
```

The files are in ANSI text format and are mainly using an **.INI** or **.INF** name extension.

Always have a backup of the initialisation files. Incorrect settings can lead to unexpected results when you run Windows.