START

Start a program, command or batch script, opens in a new/separate Command Prompt window.

```
Syntax
      START "title" [/D path] [options] "command" [parameters]

Key:
   title       Text for the CMD window title bar (required.)
   path        Starting directory.
   command     The command, batch file or executable program to run.
   parameters  The parameters passed to the command.

Options:
   /MIN         Start window Minimized.
   /MAX         Start window Maximized.
   /W or /WAIT  Start application and wait for it to terminate.
                (see below)

   /LOW         Use IDLE priority class.
   /NORMAL      Use NORMAL priority class.
   /ABOVENORMAL Use ABOVENORMAL priority class.
   /BELOWNORMAL Use BELOWNORMAL priority class.
   /HIGH        Use HIGH priority class.
   /REALTIME    Use REALTIME priority class.

   /B         Start application without creating a new window. In this case
              Ctrl-C will be ignored - leaving Ctrl-Break as the only way to
              interrupt the application.

   /I         Ignore any changes to the current environment, typically made with SET.
              Use the original environment passed to cmd.exe

   /NODE      The preferred Non-Uniform Memory Architecture (NUMA)
              node as a decimal integer.

   /AFFINITY  The processor affinity mask as a hexadecimal number.
              The process will be restricted to running on these processors.

   Options for 16-bit WINDOWS programs only

   /SEPARATE  Start in separate memory space. (more robust) 32 bit only.
   /SHARED    Start in shared memory space. (default) 32 bit only.
```

Always include a **TITLE** this can be a simple string like "My Script" or just a pair of empty quotes ""
According to the Microsoft documentation, the *title* is optional, but depending on the other options chosen you can have problems if it is omitted.

If *command* is an internal cmd command or a batch file then the command processor is run with the /K switch to cmd.exe. This means that the window will remain after the command has been run.

In a batch script, a `START` command without `/wait` will run the program and just continue, so a script containing nothing but a `START` command will close the CMD console and leave the new program running.

Document files can be invoked through their file association just by typing the name of the file as a command.
e.g. `START "" MarchReport.DOC` will launch the application associated with the .DOC file extension and load the document.

To minimise any chance of the wrong exectuable being run, specify the full path to *command* or at a minimum include the file extension:
`START "" notepad.exe`

If you START an application without a file extension (for example `WinWord` instead of `WinWord.exe`)then the PATHEXT environment variable will be read to determine which file extensions to search for and in what order.
The default value for the PATHEXT variable is: `.COM;.EXE;.BAT;.CMD`

Start - run in parallel

> The default behaviour of START is to instantiate a new process that runs in parallel with the main process. For arcane technical reasons, this does not work for some types of executable, in those cases the process will act as a blocker, pausing the main script until it's complete.
>
> In practice you just need to test it and see how it behaves.

Often you can work around this issue by creating a one line batch script (`runme.cmd`) to launch the executable, and then call that script with `START runme.cmd`

Start /Wait

The `/WAIT` option should reverse the default 'run in parallel' behaviour of `START` but again your results will vary depending on the item being started, for example:

```
Echo Starting
START /wait "demo" calc.exe
Echo Done
```

The above will start the calculator and wait before continuing. However if you replace `calc.exe` with `Winword.exe`, to run Word instead, then the `/wait` will stop working, this is because `Winword.exe` is a stub which launches the main Word application and then exits.

A similar problem will occur when starting a batch file, by default START will run the equivalent of CMD /K which opens a second command window and leaves it open. In most cases you will want the batch script to complete and then just close it's CMD console to resume the initial batch script. This can be done by explicitly running CMD /C ...

```
Echo Starting
START /wait "demo" CMD /c demoscript.cmd
Echo Done
```

Add `/B` to have everything run in a single window.

In a batch file, an alternative is to use TIMEOUT to delay processing of individual commands.

START vs CALL

Starting a new process with CALL, is very similar to running `START /wait`, in both cases the calling script will (usually) pause until the second script has completed.

Starting a new process with CALL, will run in the same shell environment as the calling script. For a GUI application this makes no difference, but a second 'called' batch file will be able to change variables and pass those changes back to the caller.

In comparison `START` will instantiate a new `CMD.exe` shell for the called batch. This will inherit variables from the calling shell, but any variable changes will be discarded when the second script ends.

Run a program

To start a new program (not a batch script), you don’t have to use `CALL` or `START`, just enter the path/file to be executed, either on the command line or within a batch script. This will behave as follows:

- On the command line, CMD.EXE does not wait for the application to terminate and control immediately returns to the command prompt.
- Running a program from within a batch script, CMD.EXE will pause the initial script and wait for the application to terminate before continuing.
- If you run one batch script from another *without* using either `CALL` or `START`, then the first script is terminated and the second one takes over.

Multiprocessor systems

Processor affinity is assigned as a hex number but calculated from the binary positions (similar to NODRIVES)

Hex	Binary	Processors
1	00000001	Proc 1
3	00000011	Proc 1+2
7	00000111	Proc 1+2+3
C	00001100	Proc 3+4 etc

Specifying /NODE allows processes to be created in a way that leverages memory locality on NUMA systems. For example, two processes that communicate with each other heavily through shared memory can be created to share the same preferred NUMA node in order to minimize memory latencies. They allocate memory from the same NUMA node when possible, and they are free to run on processors outside the specified node.

```
start /NODE 1 app1.exe
start /NODE 1 app2.exe
```

These two processes can be further constrained to run on specific processors within the same NUMA node.

In the following example, app1 runs on the low-order two processors of the node, while app2 runs on the next two processors of the node. This example assumes the specified node has at least four logical processors. Note that the node number can be changed to any valid node number for that computer without having to change the affinity mask.

```
    start /NODE 1 /AFFINITY 0x3 app1.exe
    start /NODE 1 /AFFINITY 0xc app2.exe
```

Running executable (.EXE) files

When a file that contains a .exe header, is invoked from a CMD prompt or batch file (with or without START), it will be opened as an executable file. The filename extension does not have to be .EXE. The file header of executable files start with the 'magic sequence' of ASCII characters 'MZ' (0x4D, 0x5A) The 'MZ' being the initials of Mark Zibowski, a Microsoft employee at the time the file format was designed.

Command Extensions

If Command Extensions are enabled, external command invocation through the command line or the START command changes as follows:

Non-executable files can be invoked through their file association just by typing the name of the file as a command. (e.g. WORD.DOC would launch the application associated with the .DOC file extension). This is based on the setting in `HKCU\Software\Microsoft\Windows\CurrentVersion\Explorer\FileExts\.ext\OpenWithList`, or if that is not specified, then the file associations - see ASSOC and FTYPE.

When executing a command line whose first token is the string `CMD` without an extension or path qualifier, then `CMD` is replaced with the value of the COMSPEC variable. This prevents picking up CMD.EXE from the current directory.

When executing a command line whose first token does NOT contain an extension, then CMD.EXE uses the value of the COMSPEC environment variable. This prevents picking up CMD.EXE from the current directory.

When executing a command line whose first token does NOT contain an extension, then CMD.EXE uses the value of the PATHEXT environment variable to determine which extensions to look for and in what order. The default value for the PATHEXT variable is: `.COM;.EXE;.BAT;.CMD` Notice the syntax is the same as the PATH variable, with semicolons separating the different elements.

When searching for an executable, if there is no match on any extension, then looks to see if the name matches a directory name. If it does, the START command launches the Explorer on that path. If done from the command line, it is the equivalent to doing a CD /D to that path.

Errorlevels

If the command is successfully started `ERRORLEVEL =`*`unchanged`*, typically this will be 0 but if a previous command set an errorlevel, that will be preserved (this is a bug).
If the command fails to start then `ERRORLEVEL = 9059`
`START /WAIT` *`batch_file`* - will return the ERRORLEVEL specified by EXIT

START is an internal command.

Examples

Run a minimised Login script:
```
START "My Login Script" /Min Login.cmd
```

Start a program and wait for it to complete before continuing:
```
START "" /wait autocad.exe
```

Open a file with a particular program:
```
START "" "C:\Program Files\Microsoft Office\Winword.exe" "D:\Docs\demo.txt"
```

Open Windows Explorer and list the files in the current folder (.) :
```
C:\any\old\directory> START .
```

Open a webpage in the default browser, note the protocol is required (https://)
```
START https://ss64.com
```

Open a webpage in Microsoft Edge:
```
%windir%\explorer.exe microsoft-edge:https://ss64.com
```
or with a hard-coded path:
```
"C:\Program Files (x86)\Microsoft Edge\Application\msedge.exe" https://ss64.com
```

Connect to a new printer: (this will setup the print connection/driver)
```
START \\print_server\printer_name
```

Start an application and specify where files will be saved (Working Directory):
```
START /D C:\Documents\ /MAX "Maximised Notes" notepad.exe
```

“Do not run; scorn running with thy heels” ~ Shakespeare, The Merchant of Venice

Related commands

WMIC process call create "c:\some.exe","c:\exec_dir" - This method returns the PID of the started process.
CALL - Call one batch program from another.
CMD - can be used to call a subsequent batch and ALWAYS return even if errors occur.
TIMEOUT - Delay processing of a batch file/command.
TITLE - Change the title displayed above the CMD window.
RUN commands Start → Run commands.
ScriptRunner - Run one or more scripts in sequence.
Run a script - How to create and run a batch file.
Q162059 - Opening Office documents.
Equivalent PowerShell: Start-Process - Start one or more processes.
Equivalent bash command (Linux) : open - Open a file in it’s default application.
Equivalent macOS command: open - Open a file in a chosen application.