Batch Script - Process

In this chapter, we will discuss the various processes involved in Batch Script.

Viewing the List of Running Processes

In Batch Script, the TASKLIST command can be used to get the list of currently running processes within a system.

Syntax

```
TASKLIST [/S system [/U username [/P [password]]]] [/M [module] | /SVC | /V
[/FO format] [/NH]
```

> Following are the description of the options which can be presented to the TASKLIST command.

S.No.	Options & Description
1.	**/S system** Specifies the remote system to connect to
2.	**/U** **[domain\]user** Specifies the user context under which the command should execute.
3.	**/P [password]** Specifies the password for the given user context. Prompts for input if omitted.
4.	**/M [module]** Lists all tasks currently using the given exe/dll name. If the module name is not specified all loaded modules are displayed.
5.	**/SVC** Displays services hosted in each process.
6.	**/V** Displays verbose task information.
7.	**/FI filter** Displays a set of tasks that match a given criteria specified by the filter.
8.	**/FO format** Specifies the output format. Valid values: "TABLE", "LIST", "CSV".
9.	**/NH** Specifies that the "Column Header" should not show in the output. Valid only for "TABLE" and "CSV" formats.

Examples

```
TASKLIST
```

The above command will get the list of all the processes running on your local system. Following is a snapshot of the output which is rendered when the above command is run as it is. As you can see from the following output, not only do you get the various processes running on your system, you also get the memory usage of each process.

```
Image Name                     PID       Session Name       Session#     Mer
========================= ========    ================ =========== ========
System Idle Process              0        Services            0
System                           4        Services            0           2
smss.exe                       344        Services            0         1,0
csrss.exe                      528        Services            0         3,8
csrss.exe                      612        Console             1        41,7
wininit.exe                    620        Services            0         3,5
winlogon.exe                   648        Console             1         5,8
services.exe                   712        Services            0         6,2
lsass.exe                      720        Services            0         9,7
svchost.exe                    788        Services            0        10,0
svchost.exe                    832        Services            0         7,6
dwm.exe                        916        Console             1       117,4
nvvsvc.exe                     932        Services            0         6,6
nvxdsync.exe                   968        Console             1        16,3
nvvsvc.exe                     976        Console             1        12,7
svchost.exe                   1012        Services            0        21,6
svchost.exe                    236        Services            0        33,8
svchost.exe                    480        Services            0        11,1
svchost.exe                   1028        Services            0        11,1
svchost.exe                   1048        Services            0        16,1
wlanext.exe                   1220        Services            0        12,5
conhost.exe                   1228        Services            0         2,5
svchost.exe                   1276        Services            0        13,8
svchost.exe                   1420        Services            0        13,4
spoolsv.exe                   1556        Services            0         9,3
```

```
tasklist > process.txt
```

The above command takes the output displayed by tasklist and saves it to the process.txt file.

```
tasklist /fi "memusage gt 40000"
```

The above command will only fetch those processes whose memory is greater than 40MB. Following is a sample output that can be rendered.

```
Image Name                     PID      Session Name     Session#     Mem Us
=========================    ======== ================ =========== =========
dwm.exe                        916     Console             1        127,912
explorer.exe                  2904     Console             1        125,868
ServerManager.exe             1836     Console             1         59,796
WINWORD.EXE                   2456     Console             1        144,504
chrome.exe                    4892     Console             1        123,232
chrome.exe                    4976     Console             1         69,412
chrome.exe                    1724     Console             1         76,416
chrome.exe                    3992     Console             1         56,156
chrome.exe                    1168     Console             1        233,628
chrome.exe                     816     Console             1         66,808
```

Killing a Particular Process

Allows a user running Microsoft Windows XP professional, Windows 2003, or later to kill a task from a Windows command line by process id (PID) or image name. The command used for this purpose is the TASKILL command.

Syntax

```
TASKKILL [/S system [/U username [/P [password]]]] { [/FI filter]
[/PID processid | /IM imagename] } [/T] [/F]
```

> Following are the description of the options which can be presented to the TASKKILL command.

S.No.	Options & Description
1.	**/S system** Specifies the remote system to connect to
2.	**/U** **[domain\]user** Specifies the user context under which the command should execute.
3.	**/P [password]** Specifies the password for the given user context. Prompts for input if omitted.
4.	**/FI** **FilterName** Applies a filter to select a set of tasks. Allows "*" to be used. ex. imagename eq acme* See below filters for additional information and examples.
5.	**/PID** **processID** Specifies the PID of the process to be terminated. Use TaskList to get the PID.
6.	**/IM** **ImageName** Specifies the image name of the process to be terminated. Wildcard '*' can be used to specify all tasks or image names.
7.	**/T** Terminates the specified process and any child processes which were started by it.
8.	**/F** Specifies to forcefully terminate the process(es).

Examples

```
taskkill /f /im notepad.exe
```

The above command kills the open notepad task, if open.

```
taskill /pid 9214
```

The above command kills a process which has a process of 9214.

Starting a New Process

DOS scripting also has the availability to start a new process altogether. This is achieved by using the START command.

Syntax

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

Wherein

- **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.

> Following are the description of the options which can be presented to the START command.

S.No.	Options & Description
1.	**/MIN** Start window Minimized
2.	**/MAX** Start window maximized.
3.	**/LOW** Use IDLE priority class.
4.	**/NORMAL** Use NORMAL priority class.
5.	**/ABOVENORMAL** Use ABOVENORMAL priority class.
6.	**/BELOWNORMAL** Use BELOWNORMAL priority class.
7.	**/HIGH** Use HIGH priority class.
8.	**/REALTIME** Use REALTIME priority class.

Examples

```
START "Test Batch Script" /Min test.bat
```

The above command will run the batch script test.bat in a new window. The windows will start in the minimized mode and also have the title of “Test Batch Script”.

```
START "" "C:\Program Files\Microsoft Office\Winword.exe" "D:\test\TESTA.txt
```

The above command will actually run Microsoft word in another process and then open the file TESTA.txt in MS Word.