

# Process CMD Commands in Windows 11 for Testing

Essential Commands for Process Management and Testing

```
Select Command Prompt
Microsoft Windows [Version 10.0.19044.1620]
(c) Microsoft Corporation. All rights reserved.

C:\Users\lance>help
For more information on a specific command, type HELP command-name
ASSOC          Displays or modifies file extension associations.
ATTRIB         Displays or changes file attributes.
BREAK          Sets or clears extended CTRL+C checking.
BCDEDIT        Sets properties in boot database to control boot load
CALL           Calls one batch program from another.
CD             Displays the name of or changes the current directory.
CHCP           Displays or sets the active code page number.
CHDIR          Displays the name of or changes the current directory.
CHKDSK         Checks a disk for errors.
CHKNTFS        Displays or modifies the checking of disk at boot time.
CLS            Clears the screen.
CMD            Starts a new instance of the Windows command interpreter.
COLOR          Sets the default console foreground and background colors.
COMP           Compares the contents of two files or sets of files.
COMPACT        Displays or alters the compression of files on NTFS partitions.
CONVERT        Converts FAT volumes to NTFS. You cannot convert the
```



# Introduction to Process Management in Windows 11

## What are Processes?

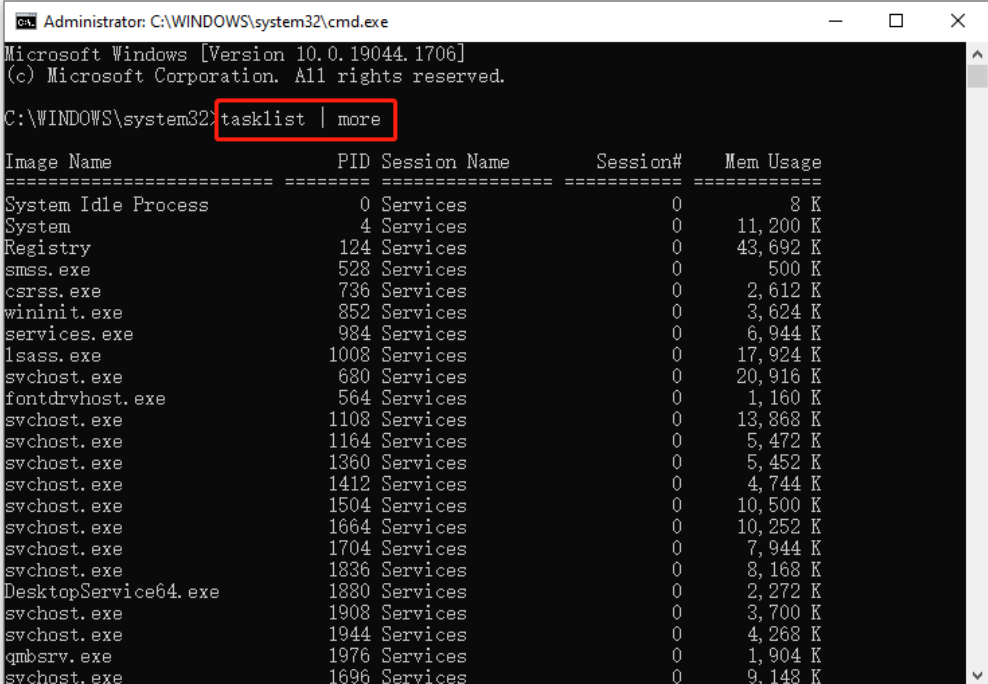
- Running instances of **applications** and **system services**
- Each process has unique **Process ID (PID)** and resource allocation

## Why Important for Testing?

- Monitor **resource usage** during application testing
- Identify and terminate **unresponsive** or **resource-intensive** processes

## CMD Commands Benefits

- Lightweight** - Works when GUI tools fail
- Scriptable** - Automate process testing workflows



```
Administrator: C:\WINDOWS\system32\cmd.exe
Microsoft Windows [Version 10.0.19044.1706]
(c) Microsoft Corporation. All rights reserved.

C:\WINDOWS\system32>tasklist | more

Image Name                    PID Session Name        Session#    Mem Usage
=====
System Idle Process           0 Services             0             8 K
System                        4 Services             0          11,200 K
Registry                     124 Services             0          43,692 K
smss.exe                      528 Services             0           500 K
csrss.exe                     736 Services             0          2,612 K
wininit.exe                   852 Services             0          3,624 K
services.exe                  984 Services             0          6,944 K
lsass.exe                     1008 Services            0          17,924 K
svchost.exe                    680 Services             0          20,916 K
fontdrvhost.exe               564 Services             0           1,160 K
svchost.exe                   1108 Services            0          13,868 K
svchost.exe                    1164 Services             0           5,472 K
svchost.exe                    1360 Services            0           5,452 K
svchost.exe                    1412 Services            0           4,744 K
svchost.exe                    1504 Services            0          10,500 K
svchost.exe                    1664 Services            0          10,252 K
svchost.exe                    1704 Services            0           7,944 K
svchost.exe                    1836 Services            0           8,168 K
DesktopService64.exe          1880 Services            0           2,272 K
svchost.exe                    1908 Services            0           3,700 K
svchost.exe                    1944 Services            0           4,268 K
qmbdrv.exe                    1976 Services             0           1,904 K
svchost.exe                    1696 Services             0           9,148 K
```

# Key Process CMD Commands Overview



## tasklist

Displays **running processes** with PID, memory usage, and session information



## taskkill

Terminates processes by **name** or **PID**, with force option for unresponsive programs



## wmic process

Windows Management Instrumentation tool for **detailed process information** and management



## netstat -ano

Shows **network connections** with associated process IDs for network troubleshooting

# Tasklist Command - Syntax and Examples

## <> Syntax

```
tasklist [/s [/u \ [/p ]]] [/m | /svc | /v] [/fo  
[/nh]] [/fi ]
```

## ⚙️ Key Parameters

/m	List processes using specified module
/svc	List services for each process
/v	Display verbose information
/fo	Output format: TABLE, LIST, CSV
/fi	Filter processes by criteria

## Basic Usage

```
tasklist
```

```
tasklist /v
```

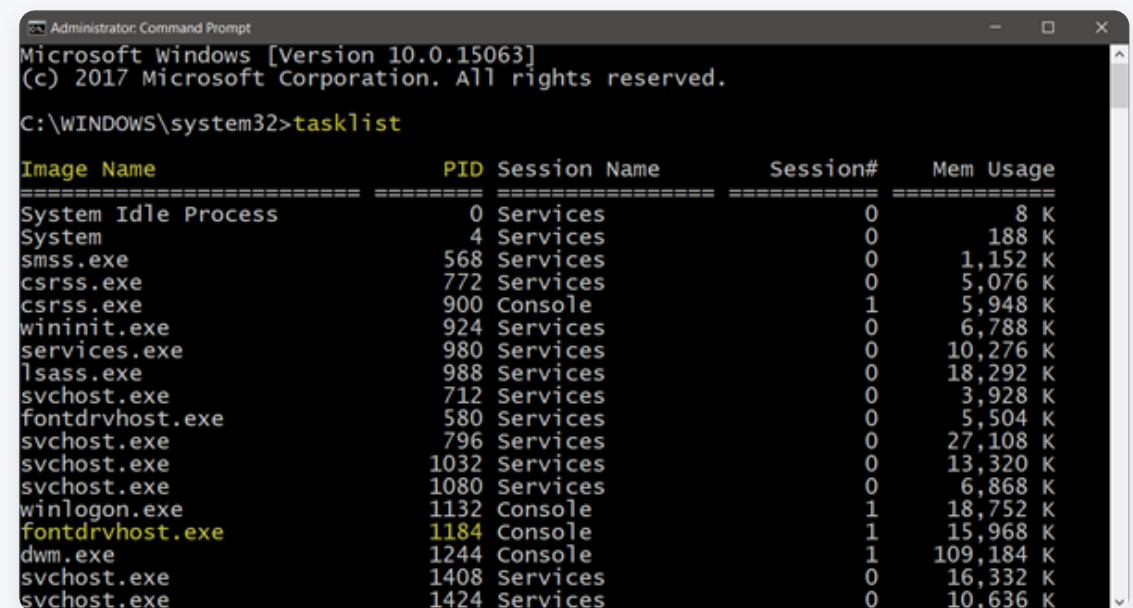
```
tasklist /svc
```

## Filtering Examples

```
tasklist /fi "imagename eq notepad.exe"
```

```
tasklist /fi "memusage gt 100000"
```

```
tasklist /fi "status eq running"
```



```
Administrator: Command Prompt
Microsoft Windows [Version 10.0.15063]
(c) 2017 Microsoft Corporation. All rights reserved.

C:\WINDOWS\system32>tasklist

Image Name                      PID Session Name        Session#    Mem Usage
=====
System Idle Process             0 Services             0             8 K
System                          4 Services             0            188 K
smss.exe                        568 Services             0           1,152 K
csrss.exe                       772 Services             0           5,076 K
csrss.exe                       900 Console                1           5,948 K
wininit.exe                     924 Services             0           6,788 K
services.exe                   980 Services             0          10,276 K
lsass.exe                      988 Services             0          18,292 K
svchost.exe                     712 Services             0           3,928 K
fontdrvhost.exe                 580 Services             0           5,504 K
svchost.exe                     796 Services             0          27,108 K
svchost.exe                    1032 Services             0          13,320 K
svchost.exe                    1080 Services             0           6,868 K
winlogon.exe                   1132 Console                1          18,752 K
fontdrvhost.exe                1184 Console                1          15,968 K
dwm.exe                        1244 Console                1         109,184 K
svchost.exe                    1408 Services             0          16,332 K
svchost.exe                    1424 Services             0          10,636 K
```

# Taskkill Command - Syntax and Examples

## <> Syntax

```
taskkill [/s [/u \ [/p ]]] {[/fi ] [/pid | /im  
]} [/f] [/t]
```

## ⚙️ Key Parameters

<code>/f</code>	Forcefully terminate process
<code>/im</code>	Kill process by image name
<code>/pid</code>	Kill process by PID
<code>/t</code>	Terminate child processes
<code>/fi</code>	Filter processes by criteria

## Basic Usage

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

```
taskkill /pid 1234
```

```
taskkill /f /im chrome.exe
```

## Advanced Examples

```
taskkill /f /im "my app.exe" /t
```

```
taskkill /fi "memusage gt 100000"
```

```
taskkill /s remote-pc /u admin /p password /im  
app.exe
```

*Use /f parameter for unresponsive processes*

A screenshot of a Windows Command Prompt window titled "Administrator: C:\WINDOWS\system32\cmd.exe". The prompt shows the successful termination of a process with PID 13932. Below this, the command `C:\WINDOWS\system32>taskkill /?` is entered, with `taskkill` highlighted by a red rectangle. The output displays the syntax for the TASKKILL command: `TASKKILL [/S system [/U username [/P [password]]] { [/FI filter] [/PID processid | /IM imagename] } [/T] [/F]`. It also includes a "Description:" section stating that the tool is used to terminate tasks by PID or image name, and a "Parameter List:" section detailing the options: `/S` (specifies remote system), `/U` (specifies user context), and `/P` (specifies password).

Administrator: C:\WINDOWS\system32\cmd.exe

SUCCESS: The process with PID 13932 has been terminated.

C:\WINDOWS\system32>**taskkill** /?

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

Description:  
This tool is used to terminate tasks by process id (PID) or image name.

Parameter List:

/S	system	Specifies the remote system to connect to.
/U	[domain\]user	Specifies the user context under which the command should execute.
/P	[password]	Specifies the password for the given user context. Prompts for input if omitted.

# WMIC Command for Process Management

# ⚙️ What is WMIC?


**Windows Management Instrumentation** Command-line tool  
Provides **detailed system information** and management capabilities  
More **powerful** than tasklist for process analysis

## <> Syntax

```
wmic process [where condition] get properties
```

## ☰ Key Properties

- Name** Process name
- ProcessId** Process ID (PID)
- CommandLine** Command line arguments
- WorkingSetSize** Memory usage in bytes



WMIC is deprecated in Windows 11 and requires manual installation as a "Feature on Demand"

## Basic Examples

```
wmic process get Name,ProcessId

wmic process where "Name='notepad.exe'" get ProcessId

wmic process get Name,CommandLine,PageFileUsage
```

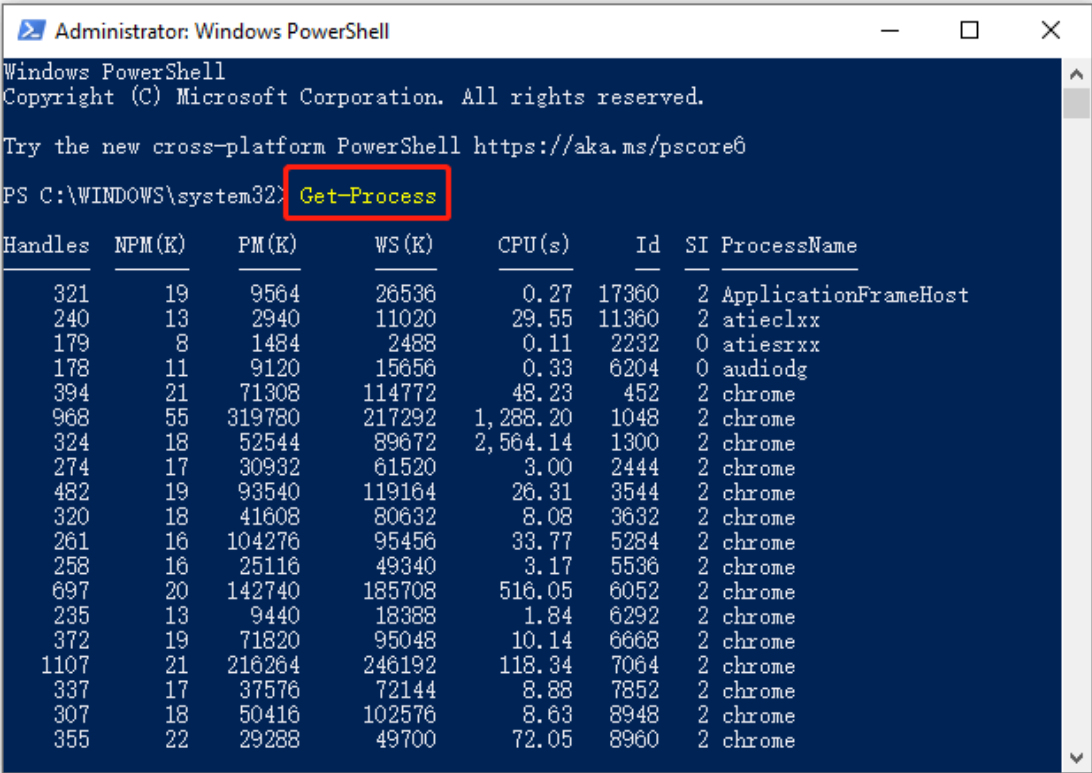
## Advanced Examples

```
wmic process where "WorkingSetSize>100000000" get Name,ProcessId

wmic process call terminate "notepad.exe"

wmic /node:remote-pc process get Name,ProcessId
```

Use */output:filename.txt* to save results to file



# Advanced Process Testing Techniques



## Combining Commands

- **Pipeline monitoring:** `tasklist | find "process"`
- **Resource tracking:** `for /l %i in (1,1,10) do tasklist /fi "imagename eq app.exe" & timeout /t 5`
- **Batch termination:** `tasklist /fi "memusage gt 100000" > temp.txt & for /f "tokens=2" %i in (temp.txt) do taskkill /pid %i`



## Performance Monitoring

- **Memory tracking:** `tasklist /fi "imagename eq app.exe" /v | find "Working Set"`
- **Resource logging:** `tasklist /v > C:\logs\processes_%date:~-4,4%%date:~-10,2%%date:~-7,2%.txt`
- **Network connections:** `netstat -ano | find "PID"`



## Troubleshooting

- **Zombie processes:** `tasklist /fi "status eq not responding"`
- **Resource hogs:** `tasklist /fi "memusage gt 200000" /v`



## Automated Testing

- **Process lifecycle:** `start app.exe & timeout /t 10 & taskkill /im app.exe`
- **Stress testing:** `for /l %i in (1,1,50) do start app.exe`



▸ **Process ancestry:** `wmic process where "ProcessId=1234" get ParentProcessId`

▸ **Batch automation:** `@echo off & tasklist /fi "imagename eq app.exe" & if %errorlevel% equ 1 (start app.exe)`

# Best Practices for Process Testing in Windows 11

## Effective Testing

- ✓ **Create baseline** before testing
- ✓ **Log results** with timestamps
- ✓ **Test in isolation** to avoid interference
- ✓ **Monitor resources** during tests

## Common Pitfalls

- ✗ **Forgetting /f** with unresponsive processes
- ✗ **Not checking privileges** before terminating
- ✗ **Ignoring child processes** when terminating
- ✗ **Using WMIC** without checking availability

## Security Considerations

- 🛡️ **Run as admin** only when necessary
- 🛡️ **Verify process identity** before termination
- 🛡️ **Avoid exposing sensitive data** in logs
- 🛡️ **Use proper authentication** for remote systems

## Performance Tips

- ↗️ **Filter output** to reduce processing time
- ↗️ **Use specific criteria** instead of listing all
- ↗️ **Cache results** for repeated queries
- ↗️ **Batch operations** to reduce overhead

## Resources for Further Learning

[Microsoft Docs](#)[PowerShell 7 Guide](#)[Windows Sysinternals](#)[Tech Community](#)[Microsoft Learn](#)