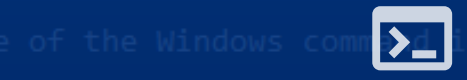
Process CMD Commands in Windows 11 for Testing

Essential Commands for Process Management and Testing



Introduction to Process Management in Windows 11

What are Processes?

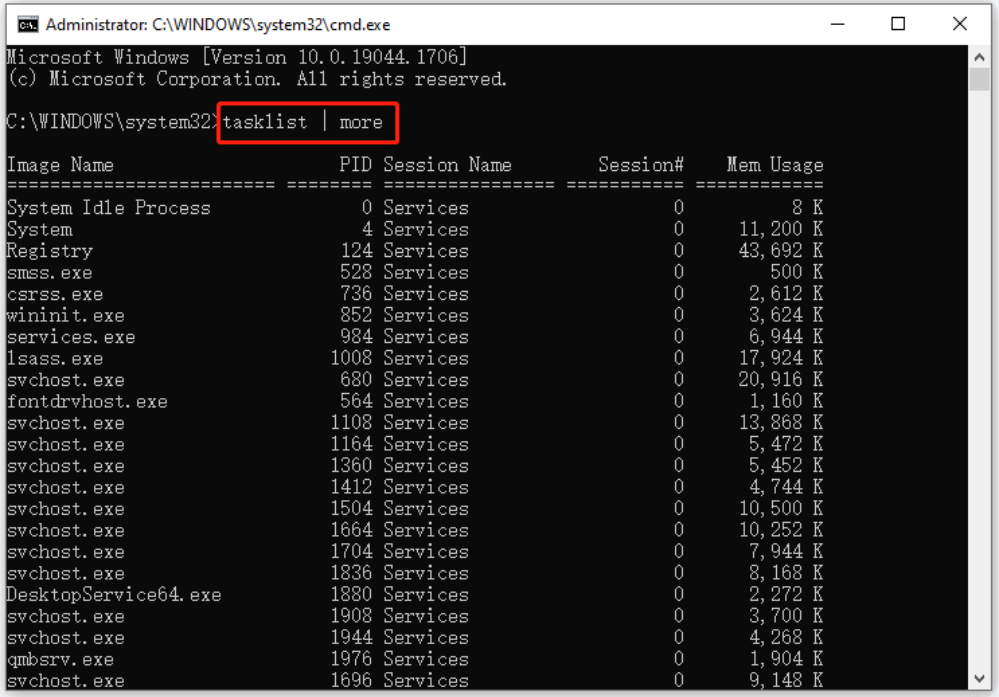
- Running instances of **applications** and **system services**
- Each process has unique **Process ID (PID)** and resource allocation

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



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
```

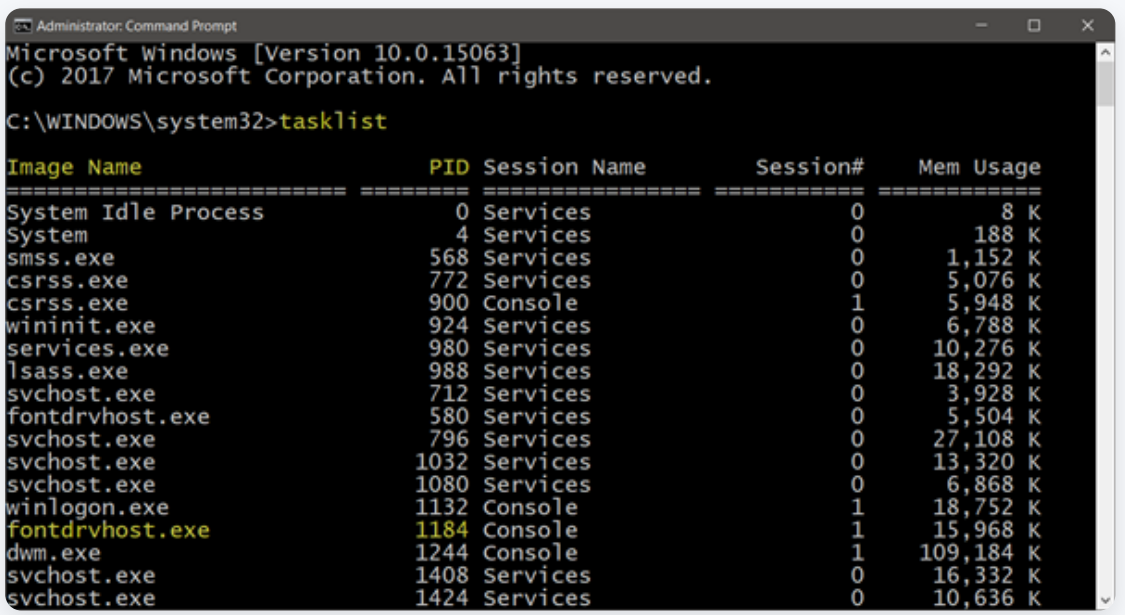
```
tasklist /svc
```

Filtering Examples

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

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

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



Taskkill Command - Syntax and Examples

<> Syntax

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

Key Parameters

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

Basic Usage

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

```
taskkill /pid 1234
```

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

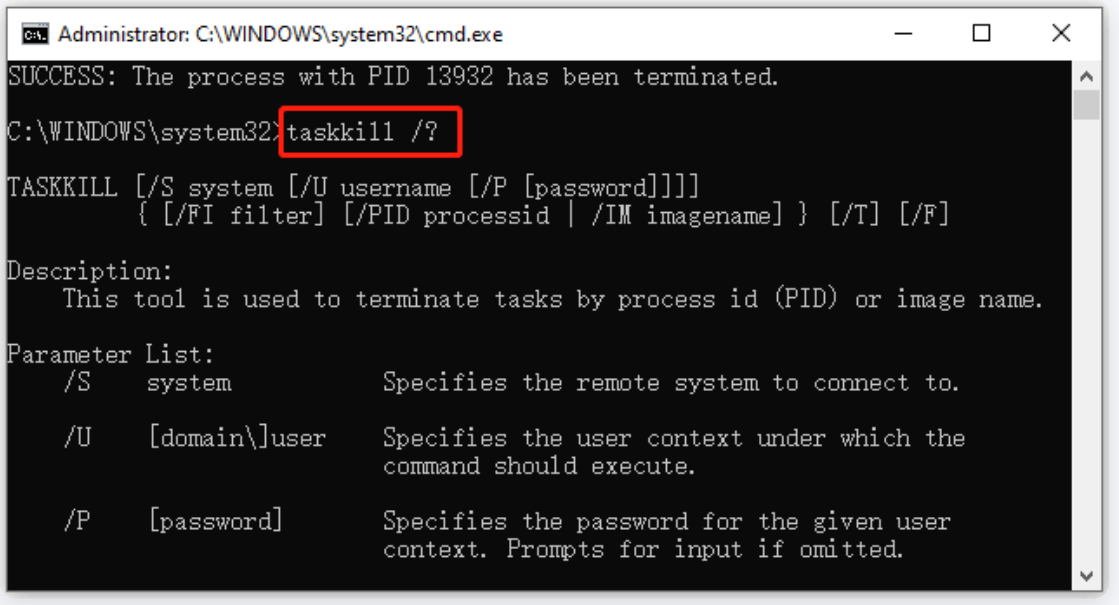
Advanced Examples

```
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*



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
```

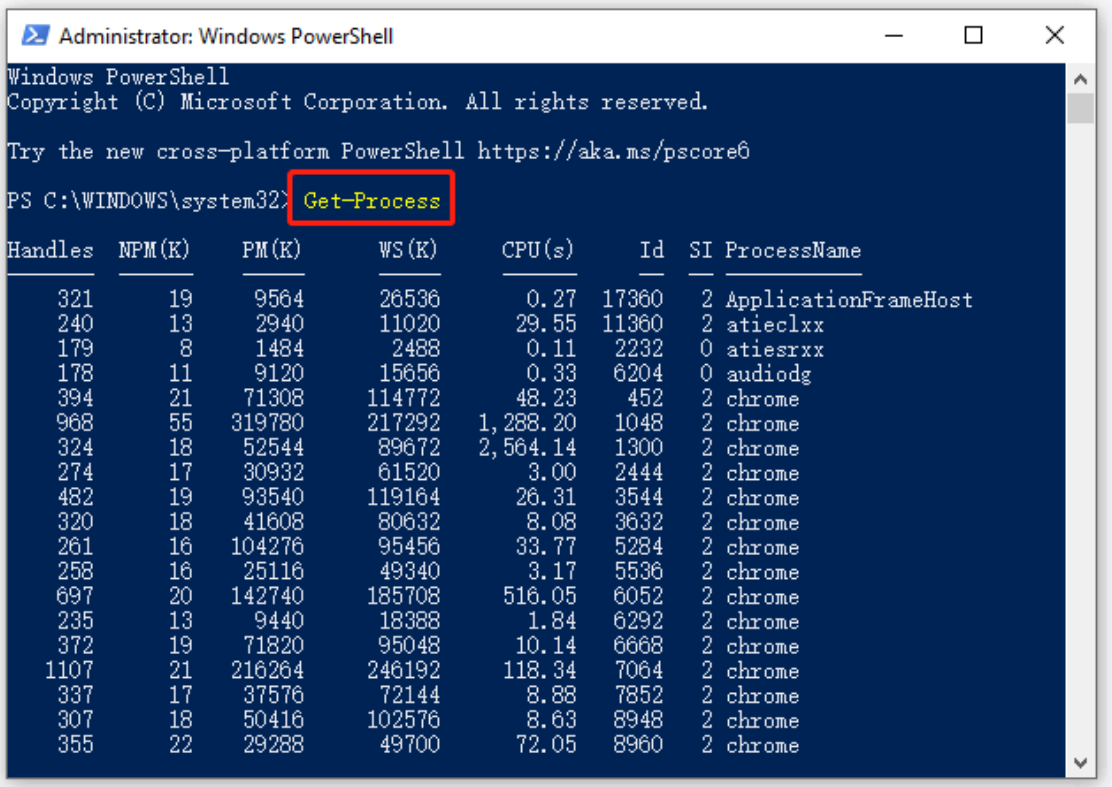
Advanced Examples

```
wmic process where "WorkingSetSize>10000000"
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