Installing Windows Updates

Windows Updates deliver security patches, bug fixes, driver updates, and feature improvements from Microsoft. Keeping systems updated is a fundamental part of maintaining a secure and stable environment. Updates can be installed through the Windows Update GUI, Settings app, Windows Server Update Services (WSUS), or via command line tools including PowerShell and `wuauclt`.

Background

Windows Update has evolved significantly across OS versions. On modern systems (Windows 10/11, Server 2016+), the `PSWindowsUpdate` PowerShell module and the built-in `UsoClient.exe` / `wuauclt.exe` utilities are the primary CLI tools. In enterprise environments, update distribution is typically managed centrally through WSUS, Microsoft Endpoint Configuration Manager (MECM/SCCM), or Windows Autopatch.

Understanding which update categories apply to a system helps prioritize deployment:

Update Type	Description	Typical Release Cadence
Security Updates	Patches for CVEs and vulnerabilities	Monthly (Patch Tuesday)
Cumulative Updates	Bundled quality + security fixes	Monthly
Feature Updates	Major OS version upgrades	Annually (Windows 10/11)
Driver Updates	Hardware driver updates via WU	As needed
Definition Updates	Defender antimalware signatures	Daily / multiple times daily
Optional Updates	Non-critical quality improvements	As needed

> **Note:** Patch Tuesday falls on the second Tuesday of each month. Emergency out-of-band updates may be released at any time for critical vulnerabilities.

Usage

Checking for and Installing Updates via Settings (GUI)

Navigate to **Settings → Windows Update** and click **Check for updates**.

Windows Update settings page showing available updates *Screenshot: Windows Update page in Settings showing pending updates, their status, and the "Check for updates" button*

Installing Updates via PowerShell (PSWindowsUpdate Module)

The `PSWindowsUpdate` module is the most capable CLI option for managing updates on individual machines.

Install the module (run as Administrator):

```
Install-Module -Name PSWindowsUpdate -Force -Scope AllUsers
```

Import and list available updates:

```
Import-Module PSWindowsUpdate
Get-WindowsUpdate
```

Install all available updates:

```
Install-WindowsUpdate -AcceptAll -AutoReboot
```

Install only security updates, without auto-reboot:

```
Install-WindowsUpdate -Category "Security Updates" -AcceptAll -IgnoreReboot
```

Install a specific update by KB article number:

```
Install-WindowsUpdate -KBArticleID KB5034441 -AcceptAll
```

Hide an update to prevent it from installing:

```
Hide-WindowsUpdate -KBArticleID KB5034441
```

> **Tip:** Use `-Verbose` with any `PSWindowsUpdate` command to see detailed progress output. Useful when scripting or troubleshooting stuck installs.

Using UsoClient.exe (Windows 10/11 and Server 2016+)

`UsoClient.exe` is the Update Session Orchestrator client, replacing much of `wuauclt` functionality on modern Windows.

```
# Scan for updates
UsoClient.exe StartScan

# Download detected updates
UsoClient.exe StartDownload

# Install downloaded updates
UsoClient.exe StartInstall

# Trigger a full scan, download, and install in sequence
UsoClient.exe ScanInstallWait
```

> **Warning:** `UsoClient.exe` does not return meaningful exit codes and provides no console output. Use `PSWindowsUpdate` or check Event Viewer (`Applications and Services Logs > Microsoft > Windows > WindowsUpdateClient`) to verify results.

Using wuauclt.exe (Legacy / Windows 7 / Server 2008 R2)

```
# Force detection of updates from WSUS or WU
wuauclt.exe /detectnow

# Trigger installation of detected updates
wuauclt.exe /updatenow

# Report current update status to WSUS server
wuauclt.exe /reportnow
```

> **Note:** `wuauclt.exe` is largely deprecated on Windows 10/11 and Server 2016+. Commands may appear to run but have no effect. Use `UsoClient.exe` or `PSWindowsUpdate` on modern systems.

Scheduling a Reboot After Updates

```
# Schedule restart for 11:00 PM tonight
shutdown /r /t 0 /f
# or schedule for a specific time using Task Scheduler
$trigger = New-ScheduledTaskTrigger -Once -At "23:00"
$action = New-ScheduledTaskAction -Execute "shutdown.exe" -Argument "/r /f /t 60"
Register-ScheduledTask -TaskName "PostUpdateReboot" -Trigger $trigger -Action $action -
RunLevel Highest
```

Update Process Flow

```
flowchart TD
    A([Start]) --> B[Scan for Updates\nUsoClient / PSWindowsUpdate / GUI]
    B --> C{Updates\nAvailable?}
    C -- No --> D([System Up to Date])
    C -- Yes --> E[Review Update List\nCategories, KBs, Size]
    E --> F{Approve /\nProceed?}
    F -- No / Defer --> G[Hide or Defer Update]
    G --> D
    F -- Yes --> H[Download Updates]
    H --> I[Install Updates]
    I --> J{Reboot\nRequired?}
    J -- No --> K[Verify Installation\nGet-WUHistory / WinVer]
    J -- Yes --> L[Schedule or Perform Reboot]
    L --> K
    K --> M([Done])
```

Configuration

Configuring Windows Update via Group Policy

Group Policy is the standard method for controlling update behavior in domain environments.

Key GPO paths:

```
Computer Configuration
└── Administrative Templates
    └── Windows Components
        └── Windows Update
            └── Manage end user experience
            └── Manage updates offered from Windows Server Update Services
```

GPO Setting	Description
Configure Automatic Updates	Sets update behavior (notify, download, install)
Specify intranet Microsoft update service location	Points clients to a WSUS server
No auto-restart with logged-on users	Prevents forced reboots when users are active
Configure auto-restart required notification alerts	Manages restart notification behavior
Turn off access to all Windows Update features	Disables the WU client entirely (use with WSUS)

Configuring Windows Update via Registry

For workgroup machines or scripted deployments without GPO:

```
# Point a machine to a WSUS server
$WSUSServer = "http://wsus.domain.local:8530"
$WUPath = "HKLM:\SOFTWARE\Policies\Microsoft\Windows\WindowsUpdate"
$AUPath  = "HKLM:\SOFTWARE\Policies\Microsoft\Windows\WindowsUpdate\AU"

New-Item -Path $WUPath -Force | Out-Null
New-Item -Path $AUPath -Force | Out-Null

Set-ItemProperty -Path $WUPath -Name "WUServer"       -Value $WSUSServer
Set-ItemProperty -Path $WUPath -Name "WUStatusServer" -Value $WSUSServer
Set-ItemProperty -Path $AUPath -Name "UseWUServer"    -Value 1 -Type DWord
Set-ItemProperty -Path $AUPath -Name "AUOptions"      -Value 4 -Type DWord  # 4 = Auto
download and schedule install
```

> **Warning:** Incorrect WSUS registry settings can prevent a machine from receiving any updates. Always verify connectivity to the WSUS URL before applying at scale.

Deferring Feature and Quality Updates (Windows 10/11 Pro+)

```
# Defer quality updates by 14 days
Set-ItemProperty -Path "HKLM:\SOFTWARE\Policies\Microsoft\Windows\WindowsUpdate\AU" `
    -Name "DeferQualityUpdates" -Value 1 -Type DWord
Set-ItemProperty -Path "HKLM:\SOFTWARE\Policies\Microsoft\Windows\WindowsUpdate\AU" `
    -Name "DeferQualityUpdatesPeriodInDays" -Value 14 -Type DWord

# Defer feature updates by 60 days
Set-ItemProperty -Path "HKLM:\SOFTWARE\Policies\Microsoft\Windows\WindowsUpdate\AU" `
    -Name "DeferFeatureUpdates" -Value 1 -Type DWord
Set-ItemProperty -Path "HKLM:\SOFTWARE\Policies\Microsoft\Windows\WindowsUpdate\AU" `
    -Name "DeferFeatureUpdatesPeriodInDays" -Value 60 -Type DWord
```

Common Use Cases

Patching Multiple Remote Machines via PSWindowsUpdate

```
# Install all updates on a list of remote servers (WinRM must be enabled)
$servers = @("SRV01", "SRV02", "SRV03")

Invoke-WUJob -ComputerName $servers `
    -Script { Import-Module PSWindowsUpdate; Install-WindowsUpdate -AcceptAll -AutoReboot |
Out-File "C:\WU_$(hostname)_$(Get-Date -f yyyyMMdd).log" } `
    -Confirm:$false `
    -RunNow
```

> **Note:** `Invoke-WUJob` uses Task Scheduler on the remote machine to run the update job under the SYSTEM account, bypassing double-hop credential issues common with `Invoke-Command`.

Viewing Update History

```
# View last 20 installed updates
Get-WUHistory -Last 20 | Select-Object Date, Title, Result | Format-Table -AutoSize
```

PowerShell output showing Get-WUHistory results *Screenshot: Terminal output of `Get-WUHistory` listing recently installed KB articles, dates, and result status*

Checking for a Specific KB Installation

```
# Check if KB5034441 is installed
Get-HotFix -Id KB5034441

# Alternatively using PSWindowsUpdate history
Get-WUHistory | Where-Object { $_.KB -eq "KB5034441" }
```

Uninstalling a Problematic Update

```
# Uninstall via wusa.exe
wusa.exe /uninstall /kb:5034441 /quiet /norestart

# Or via DISM for cumulative updates on Server Core
dism.exe /Online /Remove-Package
/PackageName:Package_for_RollupFix~31bf3856ad364e35~amd64~~19041.1234.1.7
```

> **Warning:** Uninstalling cumulative updates on modern Windows can leave the system in an inconsistent patch state. Only do this to remediate a known bad update, and re-apply the latest good cumulative update as soon as possible.

Clearing the Windows Update Cache

Useful when updates are stuck downloading or installing:

```
# Stop update services
Stop-Service -Name wuauserv, bits, cryptsvc, msiserver -Force

# Clear the SoftwareDistribution cache
Remove-Item -Path "C:\Windows\SoftwareDistribution\Download\*" -Recurse -Force
Remove-Item -Path "C:\Windows\System32\catroot2\*" -Recurse -Force -ErrorAction
SilentlyContinue

# Restart services
Start-Service -Name cryptsvc, bits, wuauserv, msiserver
```

Automating Monthly Patching with a Script

```
# PatchServer.ps1 — basic monthly patching script
param (
    [switch]$Reboot
)

Import-Module PSWindowsUpdate -ErrorAction Stop

$LogFile = "C:\Logs\WindowsUpdate_$(Get-Date -Format 'yyyyMMdd_HHmmss').log"
New-Item -ItemType Directory -Path "C:\Logs" -Force | Out-Null

Write-Output "Starting Windows Update scan: $(Get-Date)" | Tee-Object -FilePath $LogFile

$updates = Get-WindowsUpdate -AcceptAll
if ($updates.Count -eq 0) {
    Write-Output "No updates available." | Tee-Object -FilePath $LogFile -Append
    exit 0
}

Install-WindowsUpdate -AcceptAll -IgnoreReboot | Tee-Object -FilePath $LogFile -Append

if ($Reboot) {
    Write-Output "Rebooting in 60 seconds..." | Tee-Object -FilePath $LogFile -Append
    shutdown /r /t 60 /c "Scheduled post-patching reboot"
}
```

Run it:

```
.\PatchServer.ps1 -Reboot
```

Troubleshooting

Symptom	Likely Cause	Resolution
Updates stuck at 0% download	Corrupt SoftwareDistribution cache	Clear cache (see above), restart services
`0x80070422` error	Windows Update service disabled	`Set-Service wuauserv -StartupType Automatic; Start-Service wuauserv`
`0x8024402c` error	DNS / proxy preventing WU connectivity	Check proxy settings, DNS, and firewall egress to `*.update.microsoft.com`
WSUS clients not checking in	Incorrect registry keys or WUServer unreachable	Verify WSUS URL, run `wuauclt /detectnow`, check `WindowsUpdateClient` event log
PSWindowsUpdate not found	Module not installed	`Install-Module PSWindowsUpdate -Force`
Update fails post-install	Conflicting or incomplete cumulative update	Run `sfc /scannow` and `DISM /Online /Cleanup-Image /RestoreHealth` then retry

```
# Run System File Checker
sfc /scannow

# Run DISM component store repair
DISM.exe /Online /Cleanup-Image /CheckHealth
DISM.exe /Online /Cleanup-Image /ScanHealth
DISM.exe /Online /Cleanup-Image /RestoreHealth
```

References

- Microsoft Docs — Windows Update overview
- PSWindowsUpdate module — PowerShell Gallery
- WSUS deployment guide — Microsoft Learn
- Windows Update error codes — Microsoft Learn
- UsoClient.exe documentation — Microsoft Learn
- Patch Tuesday schedule — MSRC

Revision #5
Created 2026-04-21 08:34:10 UTC by Johan Odendaal

Updated 2026-04-21 09:29:00 UTC by Johan Odendaal