

How to Check, Enable or Disable SMB Protocol Versions on Windows

 woshub.com/smb-1-0-support-in-windows-server-2012-r2

June 7, 2021

```
PS C:\Windows\system32> Get-SmbServerConfiguration | select EnableSMB1Protocol,EnableSMB2Protocol
EnableSMB1Protocol EnableSMB2Protocol
-----
False              True
```

The **Server Message Block (SMB)** is a client-server network protocol used in Windows networks to provide remote access to shared files, printers, other network resources, and for interprocess communication. TCP port 445 is used for communication between SMB clients and servers. In this article, we'll explore the versions of the SMB protocol available across different Windows versions (and how they relate to samba versions in Linux); how to check the SMB version in use on your computer; and how to enable/disable the SMBv1, SMBv2, and SMBv3 protocol dialects on a Windows computer (client or server).

SMB Protocol Versions in Windows

There are several versions of the SMB protocol (dialects) that have consistently appeared in new Windows versions (and Samba).

- **CIFS** – Windows NT 4.0
- **SMB 1.0** – Windows 2000
- **SMB 2.0** – Windows Server 2008 and Windows Vista SP1 (supported in Samba 3.6)
- **SMB 2.1** – Windows Server 2008 R2 and Windows 7 (Samba 4.0)
- **SMB 3.0** – Windows Server 2012 and Windows 8 (Samba 4.2)
- **SMB 3.0.2** – Windows Server 2012 R2 and Windows 8.1 (partially supported in Samba)
- **SMB 3.1.1** – Windows Server 2016 and Windows 10 (support for this feature was introduced in Samba 4.3, and it is now used by default)
- **SMB 3.1.1 (*)** – this version of SMB was introduced in Windows 11 and Windows Server 2022. It received support for SMB over QUIC, compression of SMB file traffic, and encryption with AES-256-GCM and AES-256-CCM support. Starting with Windows 11 24H2, SMB packet signing is mandatory.

The **Samba** package is used to implement the SMB protocol in Linux/Unix. We have indicated in brackets which versions of Samba support each SMB dialect.

SMB is a client-server protocol in which an SMB server provides access to shared resources for SMB clients.

In SMB communication, the client and server use the maximum SMB protocol version supported by both the client and the server. Use the following summary table to determine which SMB protocol version is selected when different versions of Windows interact.

Operating System	Win 10/11, Win Server 2016/ 2019/ 2022/ 2025	Windows 8.1, Win Server 2012 R2	Windows 8,Server 2012	Windows 7,Server 2008 R2	Windows Vista,Server 2008	Windows XP, Server 2003 and earlier
Windows 10 and 11, Windows Server 2016/2019/ 2022/2025	SMB 3.1.1	SMB 3.02	SMB 3.0	SMB 2.1	SMB 2.0	SMB 1.0
Windows 8.1, Server 2012 R2	SMB 3.02	SMB 3.02	SMB 3.0	SMB 2.1	SMB 2.0	SMB 1.0
Windows 8, Server 2012	SMB 3.0	SMB 3.0	SMB 3.0	SMB 2.1	SMB 2.0	SMB 1.0
Windows 7, Server 2008 R2	SMB 2.1	SMB 2.1	SMB 2.1	SMB 2.1	SMB 2.0	SMB 1.0
Windows Vista, Server 2008	SMB 2.0	SMB 2.0	SMB 2.0	SMB 2.0	SMB 2.0	SMB 1.0
Windows XP, 2003 and earlier	SMB 1.0	SMB 1.0	SMB 1.0	SMB 1.0	SMB 1.0	SMB 1.0

For example, when a Windows 8.1 client computer connects to a Windows Server 2016 file server, the SMB 3.0.2 protocol is used.

According to this table, Windows XP and Windows Server 2003 can only use SMB 1.0 to access shared files and folders on the server (this protocol is disabled by default on modern Windows versions). This means that computers running Windows XP or Windows Server 2003/R2 (no longer supported) will not be able to access shared folders on machines running newer versions of Windows.

For example, clients using Windows XP or Server 2003 will be unable to access the SYSVOL and NETLOGON folders on domain controllers or log in to Active Directory (AD) if a DC is running Windows Server 2012 R2 or a later version.

When trying to connect to a resource on a file server with SMB v1 disabled from legacy clients, an error appears:

The specified network name is no longer available

A computer running a desktop OS version (such as Windows 10 or 11) and being used as an SMB server can support a maximum of **20** concurrent SMB connections.

How to Check SMB Version on Windows

Run the PowerShell command to determine which SMB protocol versions (dialects) are available on a computer:

```
Get-SmbServerConfiguration | select EnableSMB1Protocol,EnableSMB2Protocol
```

```
PS C:\Windows\system32> Get-SmbServerConfiguration | select EnableSMB1Protocol,EnableSMB2Protocol

EnableSMB1Protocol EnableSMB2Protocol
-----
False              True
```

This command returned that the SMB1 protocol is disabled (`EnableSMB1Protocol = False`), and the SMB2 and SMB3 protocols are enabled (`EnableSMB2Protocol = True`).

In Windows, it is not possible to disable or enable SMBv3 or SMBv2 separately. These protocols are always enabled/disabled only together since they use the same stack. SMB client functionality is provided by the **LanmanWorkstation** service, while SMB server functionality is handled by the **LanmanServer**. Check that these services are running:

```
Get-Service LanmanServer,LanmanWorkstation
```

```
PS C:\> Get-Service LanmanServer,LanmanWorkstation

Status      Name              DisplayName
-----
Running     LanmanServer      Server
Running     LanmanWorkstation Workstation
```

For legacy Windows versions (Windows 7, Vista, and Windows Server 2008 R2/2008), you can view the enabled SMB protocols in the registry:

```
Get-Item HKLM:\SYSTEM\CurrentControlSet\Services\LanmanServer\Parameters |
ForEach-Object {Get-ItemProperty $_.pspath}
```

If this registry key doesn't contain items named `SMB1` or `SMB2`, then both of these protocols are enabled (this is the default configuration).

```
PS C:\Windows\system32> Get-Item HKLM:\SYSTEM\CurrentControlSet\Services\LanmanServer\Parameters |
Get-ItemProperty $_.pspath

PSPath           : Microsoft.PowerShell.Core\Registry::HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\LanmanServer\Parameters
PSParentPath     : Microsoft.PowerShell.Core\Registry::HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\LanmanServer
PSChildName      : Parameters
PSProvider       : Microsoft.PowerShell.Core\Registry
ServiceDll       : C:\Windows\system32\srvsvc.dll
ServiceDllUnloadOnStop : 1
EnableAuthenticateUserSharing : 0
NullSessionPipes : {}
autodisconnect   : 15
enableforcedlogoff : 1
restrictnullsessaccess : 1
Lnannounce       : 0
Size             : 3
AdjustedNullSessionPipes : 3
enablesecuritysignature : 0
requiresecuritysignature : 0
Guid             : {159, 96, 58, 73...}
```

In earlier versions of Windows, you could find out which SMB dialects are allowed to be used as clients:

```
sc.exe query mrxsmb10
```

```
SERVICE_NAME: mrxsmb10
TYPE : 2 FILE_SYSTEM_DRIVER
STATE : 4 RUNNING
(STOPPABLE, NOT_PAUSABLE, IGNORES_SHUTDOWN)
WIN32_EXIT_CODE : 0 (0x0)
SERVICE_EXIT_CODE : 0 (0x0)
CHECKPOINT : 0x0
WAIT_HINT : 0x0
```

```
sc.exe query mrxsmb20
```

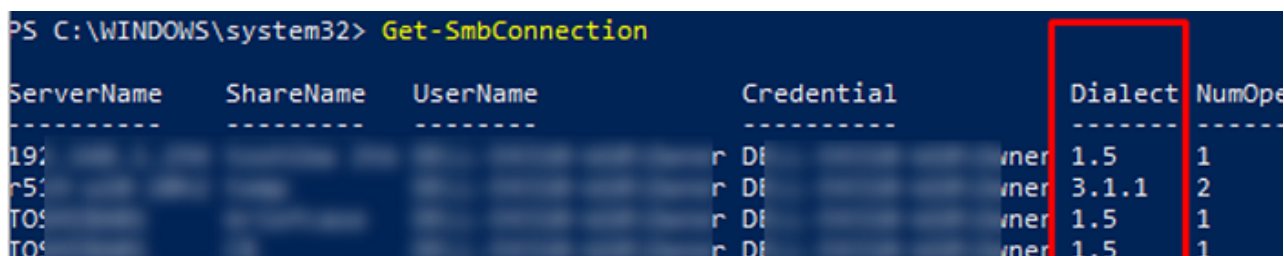
```
SERVICE_NAME: mrxsmb20
TYPE : 2 FILE_SYSTEM_DRIVER
STATE : 4 RUNNING
(STOPPABLE, NOT_PAUSABLE, IGNORES_SHUTDOWN)
WIN32_EXIT_CODE : 0 (0x0)
SERVICE_EXIT_CODE : 0 (0x0)
CHECKPOINT : 0x0
WAIT_HINT : 0x0
```

In both cases, the services are running (**STATE = 4 Running**). This means that the current Windows device can connect to both SMBv1 and SMBv2 hosts.

Find Used SMB Protocol Versions with Get-SMBConnection

When communicating over the SMB protocol, the client and server negotiate and use the highest dialect version supported by both sides. Use the PowerShell cmdlet to see which SMB protocol versions clients are using to connect to an SMB server:

```
Get-SMBConnection
```



```
PS C:\WINDOWS\system32> Get-SmbConnection
```

ServerName	ShareName	UserName	Credential	Dialect	NumOpen
192.168.1.1	C:\	Administrator	Administrator	1.5	1
192.168.1.1	C:\	Administrator	Administrator	3.1.1	2
192.168.1.1	C:\	Administrator	Administrator	1.5	1
192.168.1.1	C:\	Administrator	Administrator	1.5	1

The SMB version that is used to connect to the remote server (*ServerName*) is listed in the **Dialect** column.

To find out if SMB encryption (introduced in SMB 3.0) is being used, run the following command:

```
Get-SmbConnection | ft ServerName,ShareName,Dialect,Encrypted,UserName
```

In Linux, you can view a list of the SMB connections and protocols used by Samba:

```
$ sudo smbstatus
```

List the SMB dialects that active clients use to connect to the file server, along with the number of connections:

```
Get-SmbSession | Select-Object -ExpandProperty Dialect | Sort-Object -Unique
```

```
PS C:\Windows\system32> Get-SmbSession | Select-Object Dialect | Group-Object Dialect | Select Name, Count
Name                                     Count
----                                     -
2.10                                     898
3.02                                     8
```

In this example, 898 clients are connected to the file server via SMB 2.1 (Windows 7/Windows Server 2008 R2), and an additional 12 clients are connected via SMB 3.02.

You can use [PowerShell to enable auditing of the SMB versions](#) used for the connections:

```
Set-SmbServerConfiguration -AuditSmb1Access $true
```

SMB connection events can then be [queried from the Event Viewer logs using PowerShell](#).

```
Get-WinEvent -LogName Microsoft-Windows-SMBServer/Audit
```

Why Is SMBv1 Disabled in Newer Windows Versions?

Due to serious security issues and critical vulnerabilities, SMB protocol version 1.0 is deprecated and disabled by default in modern versions of Windows (the WannaCrypt and Petya ransomware attacks are good examples of how vulnerabilities in the SMBv1 protocol can be exploited). The legacy SMB1 protocol has been replaced by the newer and more secure SMB2 and SMB3 protocols.

Starting with Windows 10 version 809, the SMB1 client and server services are disabled by default during clean operating system installations.

Check that SMB 1.0 is disabled using the command:

```
Get-WindowsOptionalFeature -Online -FeatureName SMB1Protocol
```

```
PS C:\> Get-WindowsOptionalFeature -Online -FeatureName SMB1Protocol
FeatureName       : SMB1Protocol
DisplayName       : SMB 1.0/CIFS File Sharing Support
Description       : Support for the SMB 1.0/CIFS file sharing protocol
RestartRequired  : Possible
State             : Disabled
CustomProperties  : ServerComponent\Description : Support for the SMB
                    protocol.
                    ServerComponent\DisplayName : SMB 1.0/CIFS File S
                    ServerComponent\Id : 487
                    ServerComponent\Type : Feature
                    ServerComponent\UniqueName : FS-SMB1
                    ServerComponent\Display\Update\Name : SMB1Protoco
```

Although the legacy SMB 1.0 protocol can still be enabled on Windows devices, it is not recommended for security reasons. Disabling SMB 1 may prevent Windows from accessing shared folders on legacy devices (old NAS versions, shared network printers, Windows XP/Server 2003 devices, old Linux versions, etc.) that don't support the newer versions of the protocol.

Any clients running Windows XP or Windows Server 2003, as well as any other legacy devices that only support SMBv1, should be updated or isolated.

If there are no legacy devices left on your network that only support SMBv1, disable this protocol on all Windows computers.

How to Enable and Disable SMBv1, SMBv2, and SMBv3 on Windows

Let's look at ways to enable and disable different SMB versions on Windows. We'll cover SMB client and server management separately since they are different Windows components.

Windows 11, 10, 8.1, Windows Server 2025/2022/2019/2016/2012R2:

Disable SMBv1 client and server:

```
Disable-WindowsOptionalFeature -Online -FeatureName smb1protocol
```

Disable SMBv1 server only:

```
Set-SmbServerConfiguration -EnableSMB1Protocol $false
```

Enable SMBv1 client and server:

```
Enable-WindowsOptionalFeature -Online -FeatureName smb1protocol
```

If SMB1 isn't used for over 15 days, the built-in **SMB1Protocol-Deprecation** feature automatically disables it in Windows.

Enable only SMBv1 server:

```
Set-SmbServerConfiguration -EnableSMB1Protocol $true
```

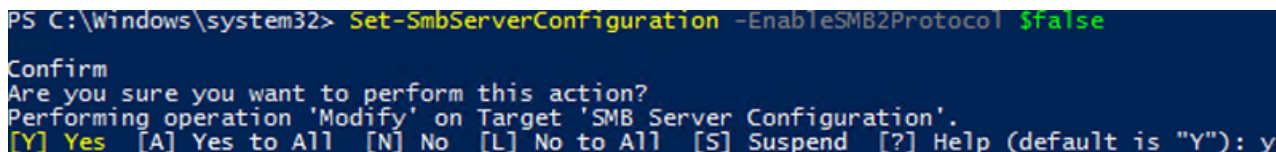
Learn more about how to enable or disable [the SMBv1 protocol on Windows 10/11 and Windows Server](#).

Disable SMBv2 and SMBv3 server:

```
Set-SmbServerConfiguration -EnableSMB2Protocol $false
```

Enable SMBv2 and SMBv3 server:

```
Set-SmbServerConfiguration -EnableSMB2Protocol $true
```



```
PS C:\Windows\system32> Set-SmbServerConfiguration -EnableSMB2Protocol $false
Confirm
Are you sure you want to perform this action?
Performing operation 'Modify' on Target 'SMB Server Configuration'.
[Y] Yes [A] Yes to All [N] No [L] No to All [S] Suspend [?] Help (default is "Y"): y
```

Windows 7, Vista, and Windows Server 2008 R2/2008:

Disable SMBv1 server:

```
Set-ItemProperty -Path
"HKLM:\SYSTEM\CurrentControlSet\Services\LanmanServer\Parameters" SMB1 -Type
DWORD -Value 0 -Force
```



```

PS C:\Windows\system32> Set-ItemProperty -Path "HKLM:\SYSTEM\
ype DWORD -Value 0 -Force
PS C:\Windows\system32> Get-Item HKLM:\SYSTEM\CurrentControlS
temProperty $_.pspath}

PSPath : Microsoft.PowerShell.Core\Reg
PSParentPath : Microsoft.PowerShell.Core\Reg
PSChildName : Parameters
PSProvider : Microsoft.PowerShell.Core\Reg
ServiceDll : C:\Windows\system32\svcsvc.dl
ServiceDllUnloadOnStop : 1
EnableAuthenticateUserSharing : 0
NullSessionPipes : <>
autodisconnect : 15
enableforcedlogoff : 1
restrictnullsessaccess : 1
Lmannounce : 0
Size : 3
AdjustedNullSessionPipes : 3
enablesecuritysignature : 0
requiresecuritysignature : 0
Guid : {159, 96, 58, 73...}
SMB2 : 1
SMB1 : 0

```

Enable SMBv1 server:

```

Set-ItemProperty -Path
"HKLM:\SYSTEM\CurrentControlSet\Services\LanmanServer\Parameters" SMB1 -Type
DWORD -Value 1 -Force

```

Disable SMBv1 client:

```

sc.exe config lanmanworkstation depend= bowser/mrxsmb20/nt
sc.exe config mrxsmb10 start= disabled

```

Enable SMBv1 client:

```

sc.exe config lanmanworkstation depend= bowser/mrxsmb10/mrxsmb20/nt
sc.exe config mrxsmb10 start= auto

```

Disable SMBv2 server:

```

Set-ItemProperty -Path
"HKLM:\SYSTEM\CurrentControlSet\Services\LanmanServer\Parameters" SMB2 -Type
DWORD -Value 0 -Force

```

Enable SMBv2 server:

```

Set-ItemProperty -Path
"HKLM:\SYSTEM\CurrentControlSet\Services\LanmanServer\Parameters" SMB2 -Type
DWORD -Value 1 -Force

```

To enable SMB1 in Windows Server 2012, you must also change the **DependOnService** registry value from SamSS Srv2 to SamSS Srv in HKLM\SYSTEM\CurrentControlSet\Services\LanmanServer. In order to disable SMB1, you must revert the SamSS Srv2 value.

Disable SMBv2 client:

```

sc.exe config lanmanworkstation depend= bowser/mrxsmb10/nt
sc.exe config mrxsmb20 start= disabled

```

Enable SMBv2 client:

```
sc.exe config lanmanworkstation depend= bowser/mrxsmb10/mrxsmb20/lsi
sc.exe config mrxsmb20 start= auto
```

Regardless of the Windows version, you can disable the SMBv1 server on all domain-joined computers by deploying the following registry parameter through the GPO:

- Key: HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\LanmanServer\Parameters
- Name: SMB1
- Type: REG_DWORD
- Value: 0

Set the registry parameter **SMB2=0** in order to disable the SMBv2 server.

To disable the SMBv1 client, deploy the following registry item:

- Key: HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\services\mrxsmb10
- Name: Start
- Type: REG_DWORD
- Value: 4

When disabling SMB 1.0/CIFS File Sharing support on Windows, you may encounter the error 0x80070035 network path not found, errors when accessing shared SMB folders, and issues with network discovery. In this case, you must use the Discovery Service instead of the Computer Browser service ([link](#)).

How to Detect Which SMB Versions Are Enabled on Network Computers

Nmap can be used to scan the network and identify devices that are using the insecure SMBv1 protocol version. The following command will scan the specified IP subnet and display the SMB versions on the devices.

```
nmap -p445 --script smb-protocols 10.1.2.0/24 -Pn
```

```
sysops@appsrvub1:~$ nmap -p445 --script smb-protocols 192.168.31.102 -Pn
Starting Nmap 7.80 ( https://nmap.org ) at 2025-07-03 09:50 CEST
Nmap scan report for 192.168.31.102
Host is up (0.0018s latency).

PORT      STATE SERVICE
445/tcp   open  microsoft-ds

Host script results:
| smb-protocols:
|   dialects:
|     NT LM 0.12 (SMBv1) [dangerous, but default]
|     2.02
|     2.10
|     3.00
|     3.02
|_    3.11

Nmap done: 1 IP address (1 host up) scanned in 0.29 seconds
```


The computers must have the network file and printer sharing service enabled, and the Windows Defender Firewall must allow traffic on TCP port 445.

In an Active Directory domain environment, this PowerShell script can be used to get information about the enabled SMB versions on computers:

```
$serverList = Get-ADComputer -Filter {(OperatingSystem -like "Windows Server*")
-and (Enabled -eq $true)} | Select-Object -ExpandProperty Name
$results = @()
foreach ($server in $serverList) {
$serverStatus = Test-Connection -ComputerName $server -Count 1 -Quiet
if ($serverStatus) {
$SmbStatus = Invoke-Command -ComputerName $server -ScriptBlock {
Get-SmbServerConfiguration | Select-Object EnableSMB1Protocol,
EnableSMB2Protocol
}
$results += [PSCustomObject]@{
ServerName = $server
IsOnline = $true
SMBv1Enabled = $SmbStatus.EnableSMB1Protocol
SMBv2Enabled = $SmbStatus.EnableSMB2Protocol
}
}
else {
# If the server is unavailable:
$results += [PSCustomObject]@{
ServerName = $server
IsOnline = $false
SMBv1Enabled = $null
SMBv2Enabled = $null
}
}
}
$results | Format-Table -AutoSize
# Saving results to a CSV file
$results | Export-Csv -Path "C:\GetAD-SMB-Status.csv" -NoTypeInformation
```

```
$results | Export-Csv -Path "C:\SMB_Status.csv"
ServerName IsOnline SMBv1Enabled SMBv2Enabled
-----
True False True
```

In this example, we used the Get-ADComputer filter to scan only Windows Server hosts.