How to Monitor and Manage Open Files in Windows

To monitor and manage files currently open on a Windows system, especially in a **file-sharing** environment (like with SMB/network shares), you can use **Computer Management** or the **Get-SmbOpenFile** PowerShell cmdlet. Each approach has its own use case scenarios depending on whether you prefer a graphical user interface (GUI) or command-line approach when dealing with **local access** or **network file shares**.

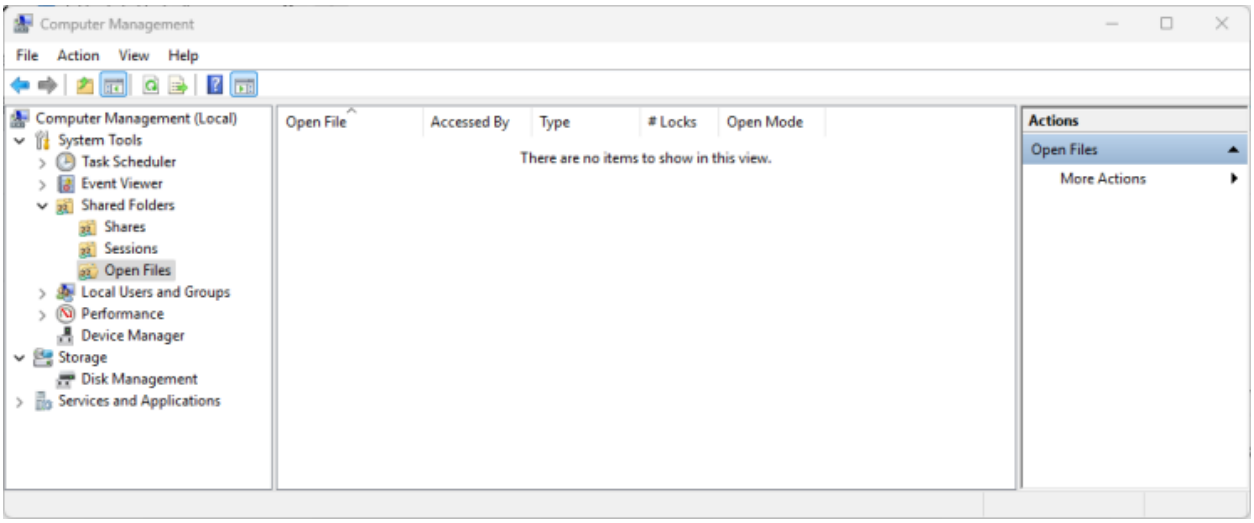
Computer Management

The built-in Computer Management console allows you to view which files are open and who is accessing them. This tool supports monitoring both your local machine and remote servers which makes it particularly useful for system administrators, who are overseeing shared resources.

To access **Computer** Management console:

- Press **Win + R**, type c**ompmgmt.msc**, and click OK.
- Right-click on Windows key and select **Computer Management**.

In the left pane, expand: **System Tools > Shared Folder > Open Files**



You will see a list of currently open files over the network (via SMB), including:

- File name/path
- Accessed by (username or computer)
- Open mode (read/write)

Actions You Can Take

- **Close Open Files:** Right-click an open file > choose Close Open File.
- **Refresh** the list as needed to see real-time activity.

The limitations with the Computer Management console lies in that it only shows **network access (SMB).** It does not show local file access. Additionally, it is not scriptable. **Use Computer Management for quick visual inspection and manual management.**

Get-SmbOpenFile cmdlet in PowerShell

The **Get-SmbOpenFile** cmdlet in PowerShell is a useful tool for system administrators to monitor and manage files currently open over the network using the SMB (Server Message Block) protocol. This command is especially handy for diagnosing file lock issues, auditing access, and identifying who has a file open on a shared network drive.

How to start using Get-SmbOpenFile

```
# List all open SMB files
Get-SmbOpenFile
```

Example with filtering for a specific file

```
Get-SmbOpenFile | Where-Object {$_.Path -like "C:\path\to\your\file.txt"}
```

Search by Partial Filename and Select Key Info

```
Get-SmbOpenFile | Where-Object {$_.Path -like "*report.docx*"} | Select-Object ClientUserName, ClientComputerName, Path
```

To Force Close a locked file

```
#Force close a locked file
Get-SmbOpenFile | Where-Object {$_.Path -like "*report.docx*"} | Close-SmbOpenFile -Force
```

⚠️ **Use the Force Close** option with caution, as it forcibly terminates the file handle and may result in data loss if users have unsaved changes. Whenever possible, notify users and request that they save and close the file or folder before proceeding.

The **Get-SmbOpenFile** cmdlet in PowerShell Requires admin privileges and Only works for **SMB-shared files** (like from a Windows file server). Use Get-SmbOpenFile when you need **automation**, **remote management**, or **detailed logging**.

Microsoft Learn has additional information about **Get-SmbOpenFile (SmbShare)**.

Disclaimer:

Please note that any PowerShell commands provided are to be used at your own risk and discretion. Running scripts comes with inherent risks. While every effort has been made to ensure accuracy, improper use of these commands can result in unintended consequences, including system instability or data loss. It is recommended to back up important data and create a system restore point before executing any PowerShell commands. Prior to deploying scripts in a live environment, consider testing with virtual machines. If you are unsure about any command or its potential impact, please seek assistance from a qualified IT professional.