ITU

Remote Desktop Connection for IT Pros/Everybody

Lowell Vanderpool
TECH SAVVY PRODUCTIONS

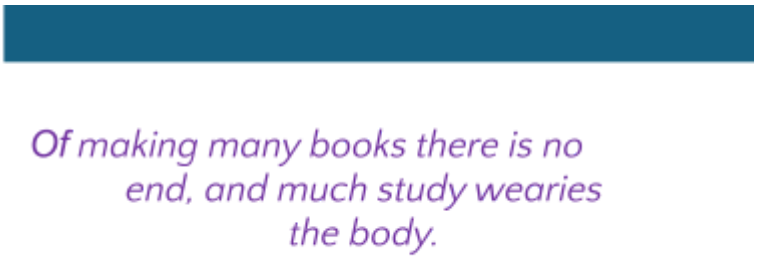
CONTACT US:

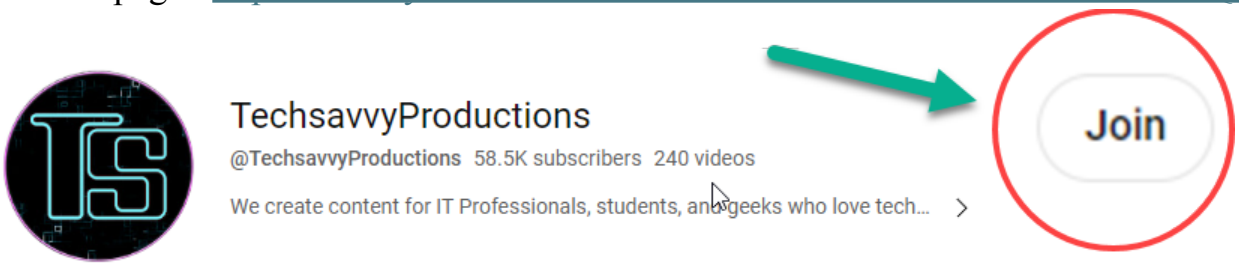
mrvanderpool@techsavvyproductions.com

SUPPORT US:

Please consider becoming a channel member:

- you get an early viewing of all our video content
- access to the complete series of videos for each subject
- links to video notes and PowerPoint slide deck both in MS-Word and PDF format
- Our eBook and resources folder
- Join our channel membership, it's $2.99/month); see the "Join" button on our channel homepage. https://www.youtube.com/channel/UCCAXBGYIJnScl0IFKXOIlsQ/join





"Everybody can be great... because anybody can serve. You don't have to have a college degree to serve. You don't have to make your subject and verb agree to serve. You only need a heart full of grace. A soul generated by love." Martin Luther King Jr.

.

SOCIAL MEDIA AND WEBSITE:

Check out our YouTube channel for more content!
YouTube: https://www.youtube.com/user/vanderl2796/featured
Check out our Website: https://www.techsavvyproductions.com
Facebook: https://www.facebook.com/TechSavvyTeamFL
Twitter: https://twitter.com/vanderl2796
Telegram: https://t.me/Lowell901
Mr.V Linkedin: https://www.linkedin.com/in/lowell-vanderpool-57970623/
Email: mrvanderpool@techsavvyproductions.com

We translate subtitles on our videos into many languages:

Tech Savvy Productions

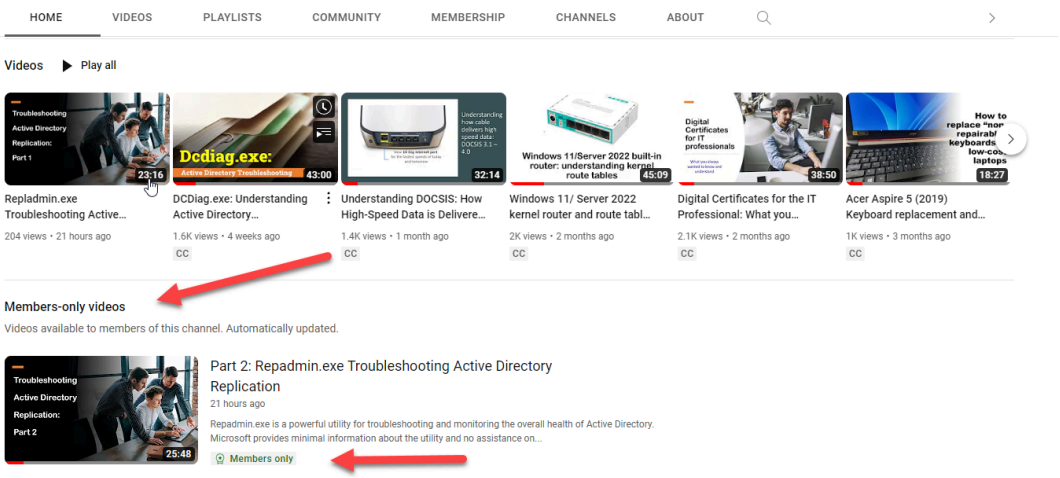
Two free ways to support our channel, like the video if it helped you better understand technology or the topic, and subscribe. Thank you for taking the time to do these helpful steps!

Like

Where is member only resources?

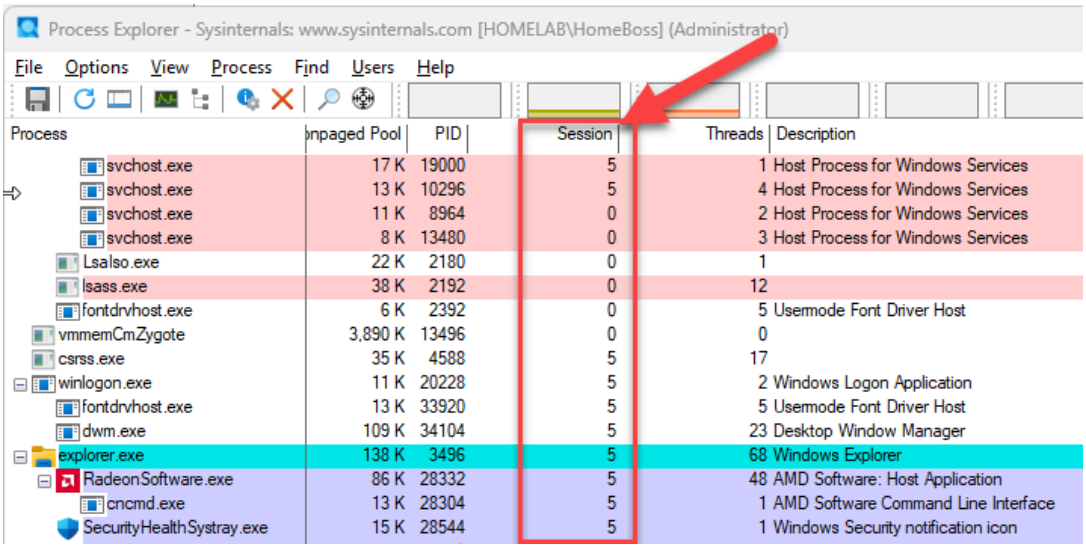
Channel homepage and click on “Playlists” you will see a Members-only playlist.

Computers are like air conditioners, they stop working properly if you open Windows

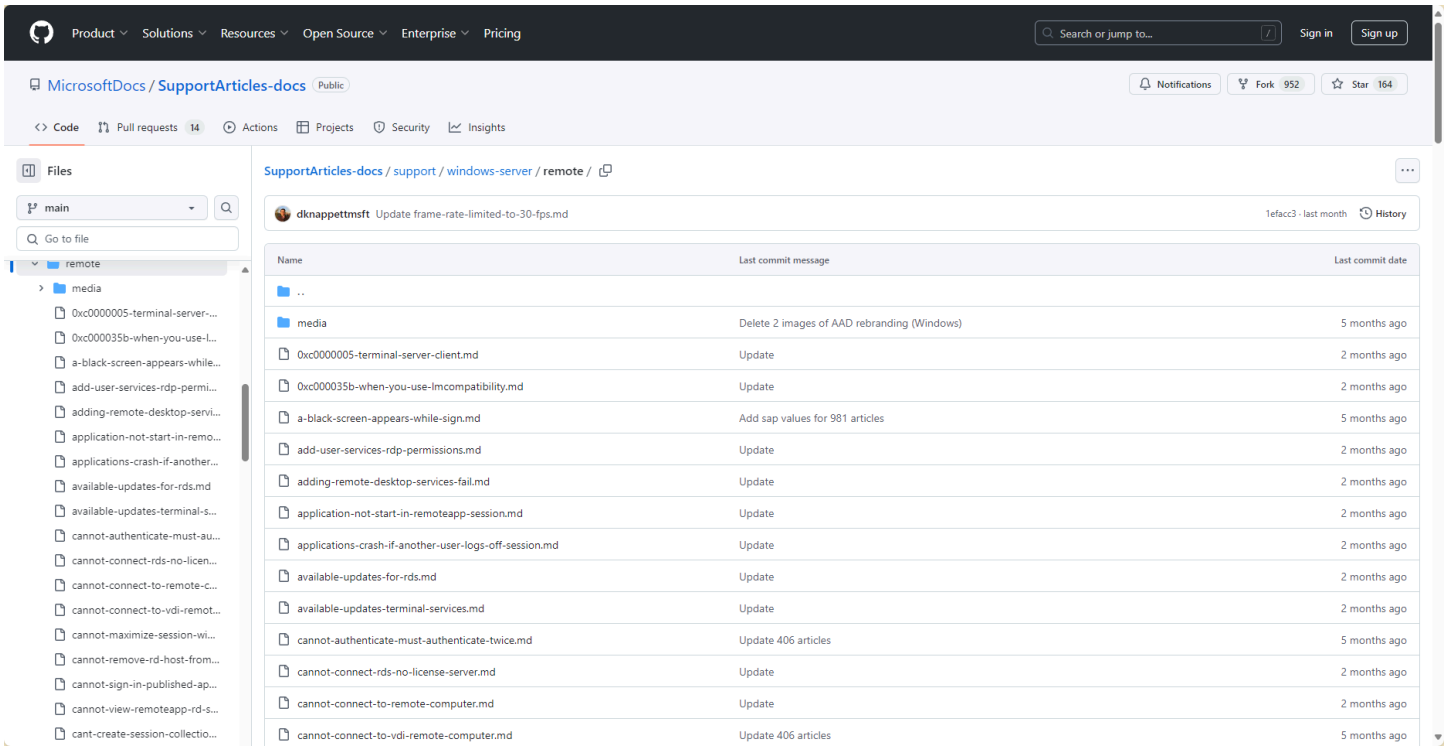


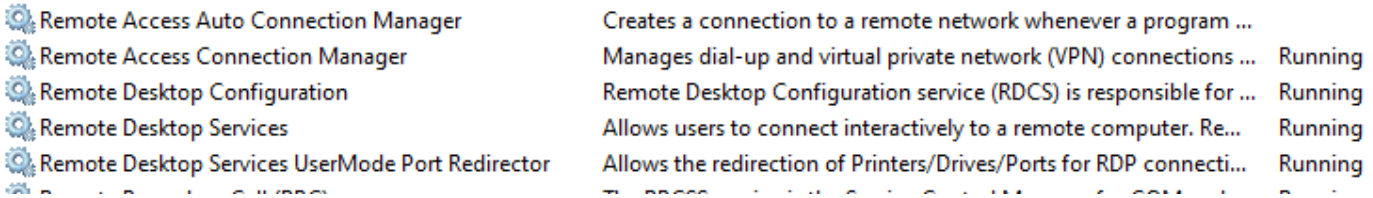
Contents

Tech Savvy Productions



Many great articles on GitHub on troubleshooting many issues with RDC: SupportArticles-docs/support/windows-server/remote at main · MicrosoftDocs/SupportArticles-docs · GitHub. 24





DISCLAIMER:

All scripts are provided "as is" without any warranties or guarantees of any kind, either express or implied, including but not limited to the implied warranties of merchantability, fitness for a particular purpose, and non-infringement. Use of this script is at your own risk.

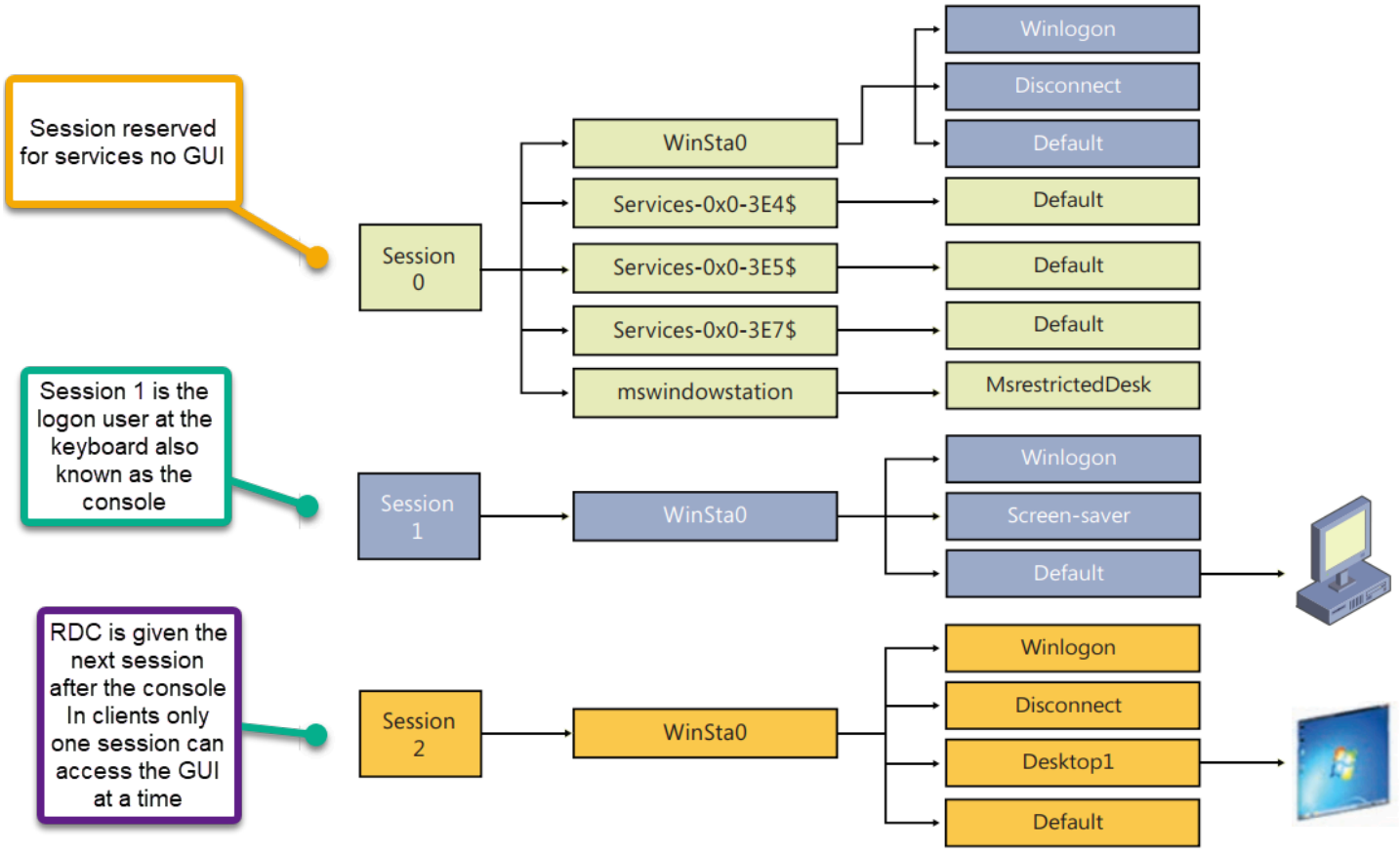
The author shall not be held responsible or liable for any damages, data loss, or other issues that arise from the use or misuse of these scripts. You are solely responsible for ensuring that the script is suitable for your specific requirements and for any consequences that result from its execution.

Before running these scripts, you should thoroughly test it in a controlled environment to ensure it functions as expected. It is recommended to backup any critical data before use.

By using these script provided, you agree to these terms and acknowledge that you have read and understood this disclaimer.

Understanding Sessions, Window Stations, Desktops

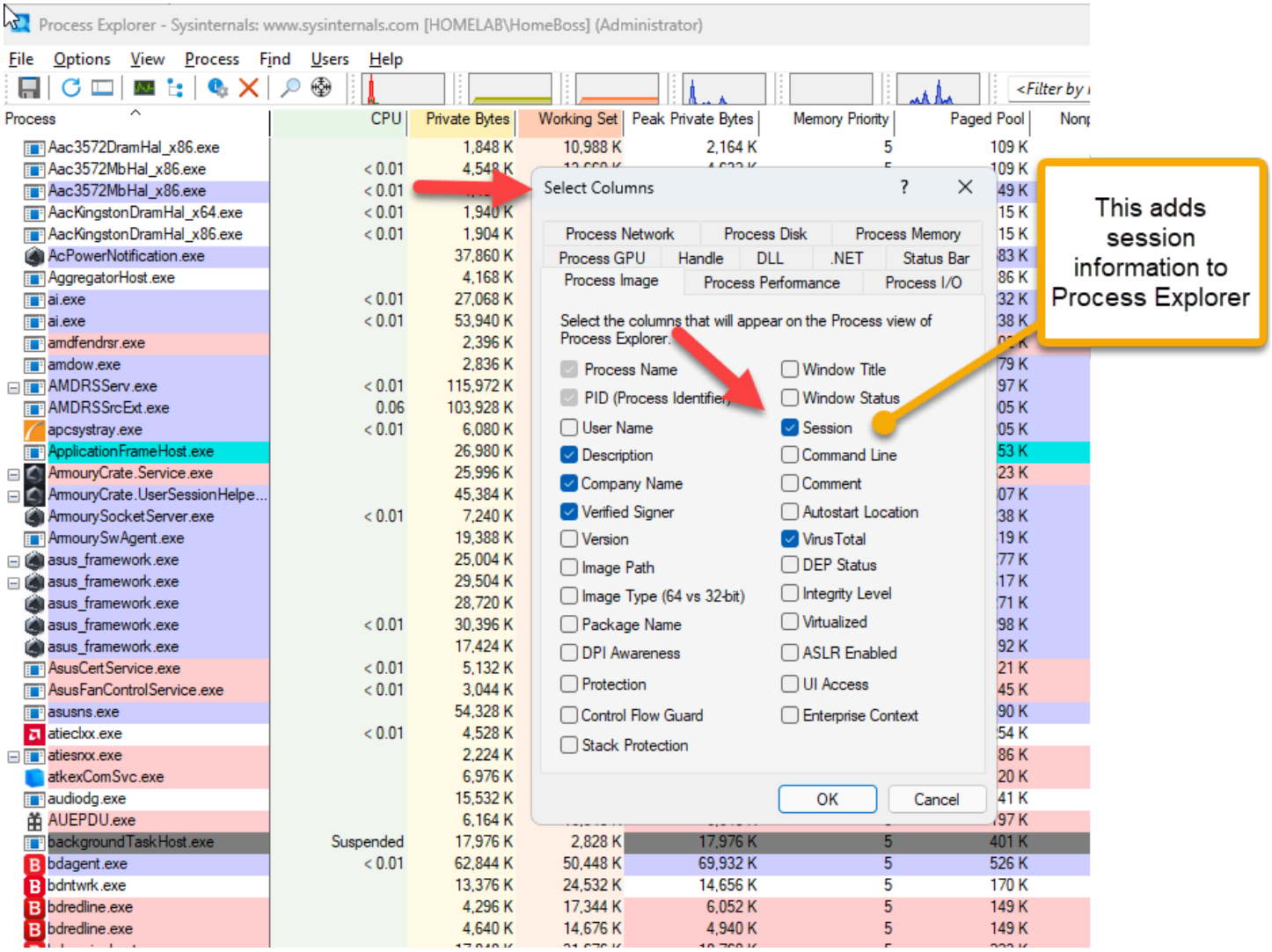
If you read the title and say " I know most of that stuff already", you most likely do not. These are concepts most do not understand at all. So let's learn about Sessions, Window Stations and Desktops



. Relationship between sessions, window stations, and desktops.

Let's start with an overview of the hierarchy, an example of which is depicted in Figure, and then define the terms. At the outermost layer are **remote desktop services (RDS) purple** sessions, formerly known as terminal services (TS) sessions. Each session contains one or more window stations, which contain desktops. Each of these securable objects has resources allocated for its sole use. There is also a loose relationship between these and logon sessions created by the LSA. Although Windows documentation doesn't always make a clear distinction between LSA logon sessions and RDS sessions, they are completely separate entities.

Note The term *console session* is sometimes mistaken as a synonym for *session 0*. The console session is the remote desktop services session associated with the locally attached keyboard, video, and mouse. If all active sessions on a computer are remote desktop sessions, the console session remains connected and displays a logon screen. It might or might not happen to be session 0 on Windows XP/Windows 2003, but it is never session 0 on Windows Vista or newer.



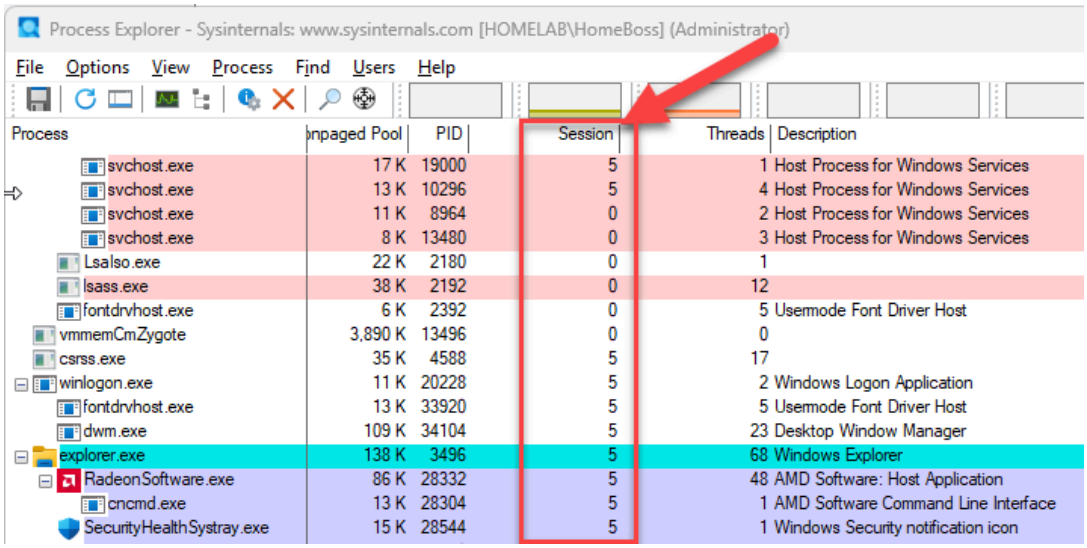


Figure 1Notice session 5 for the user, it is a RDC session

```
PS C:\Users\homeboss.HOMELAB> query user
 USERNAME              SESSIONNAME        ID  STATE   IDLE TIME  LOGON TIME
>homeboss              console             5  Active      20:12  8/29/2024 4:13 AM
```

Figure 2the query user command shows the user and what session ID they have

Remote desktop services sessions

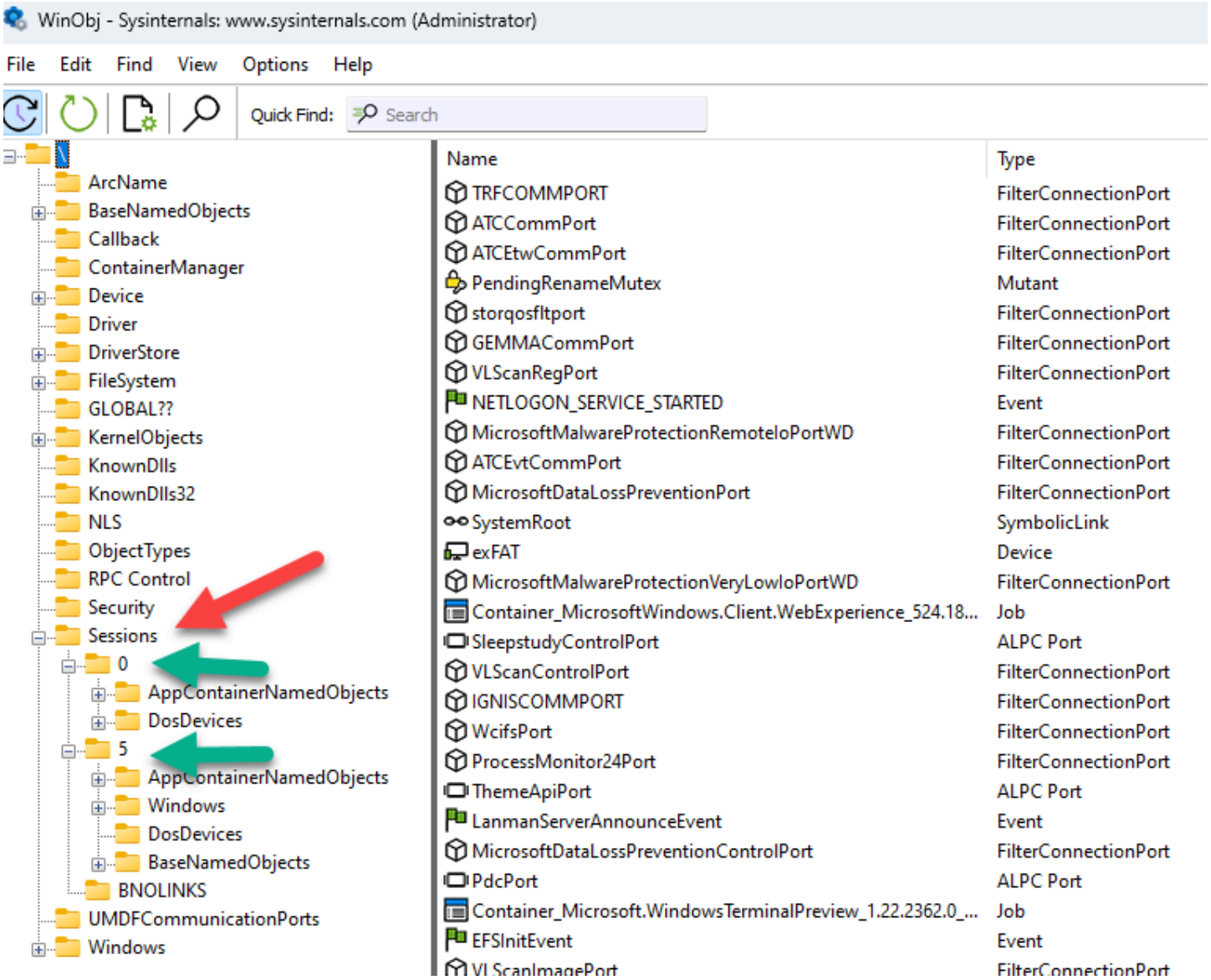
Remote desktop services support multiple interactive user sessions on a single computer. Introduced in Windows NT 4.0 Terminal Server Edition, they were not incorporated into the Windows client operating system family until Windows XP. Features they support include Fast User Switching, Remote Desktop, Remote Assistance, Remote Applications Integrated Locally (RAIL, a.k.a. RemoteApps), and virtual machine integration features.

- An important limitation of Windows clients (Windows XP, Windows Vista, Windows 7, Windows 8.x, and Windows 10) is that only one interactive session can be active at a time.

- That is, while processes can continue to run in multiple disconnected sessions simultaneously, only one session can update a display device and receive keyboard and mouse input.

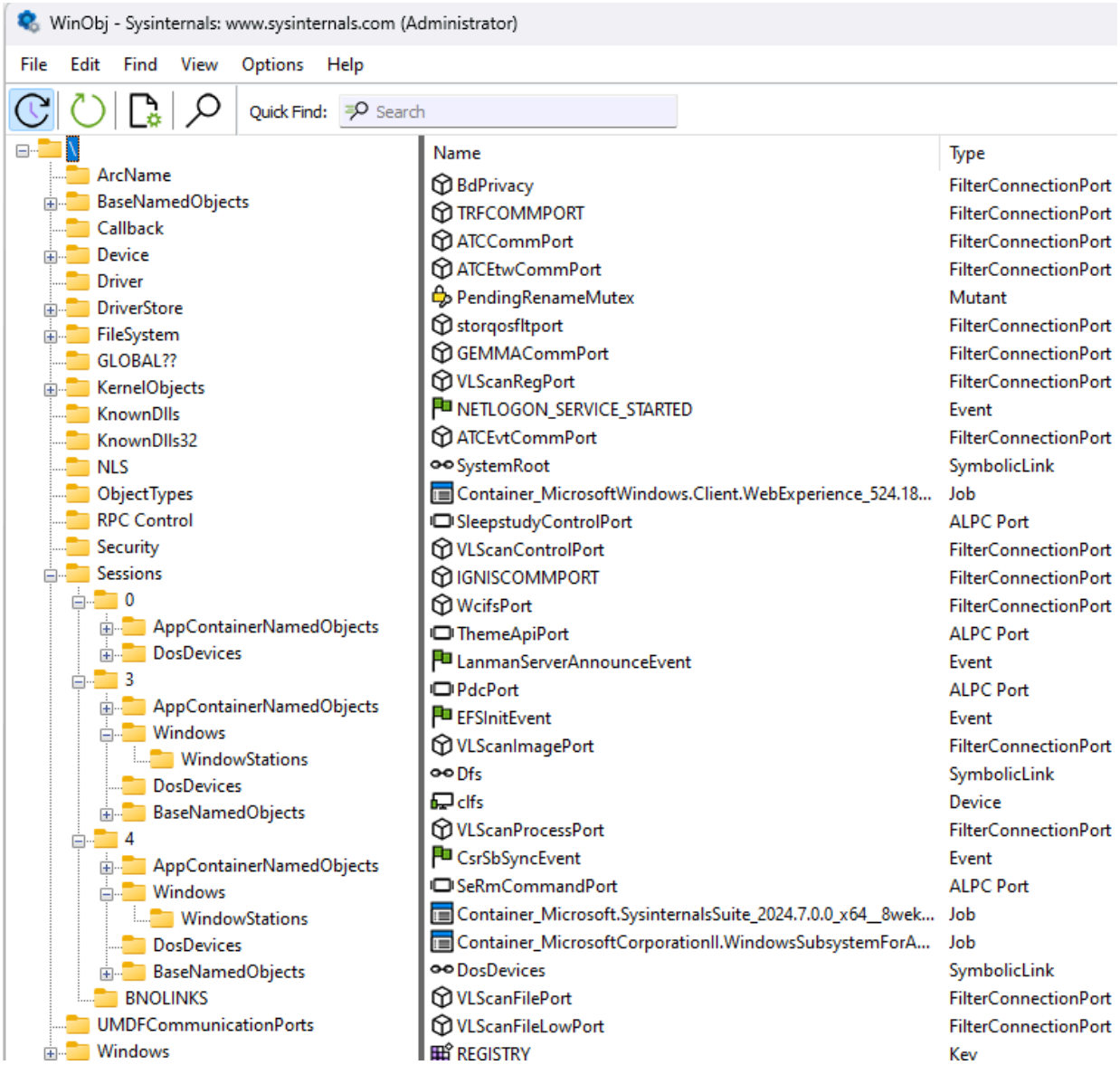
- A further limitation was that a *domain-joined* Windows XP computer supported at most only one interactive session. For example, if a user were logged on at the console, you could log on to the computer via Remote Desktop using the same account and continue that session, but you could not log on with a different user account unless the first user were logged off.

- Remote desktop services sessions are identified by an incrementing numeric session ID, starting with session 0. Windows defines a global namespace in the Object Manager and a session-private "local" namespace for each session numbered 1 and higher to provide isolation between sessions. The global namespace serves as the local namespace for processes in session 0. WinObj offers a graphical view of the Object Manager namespace



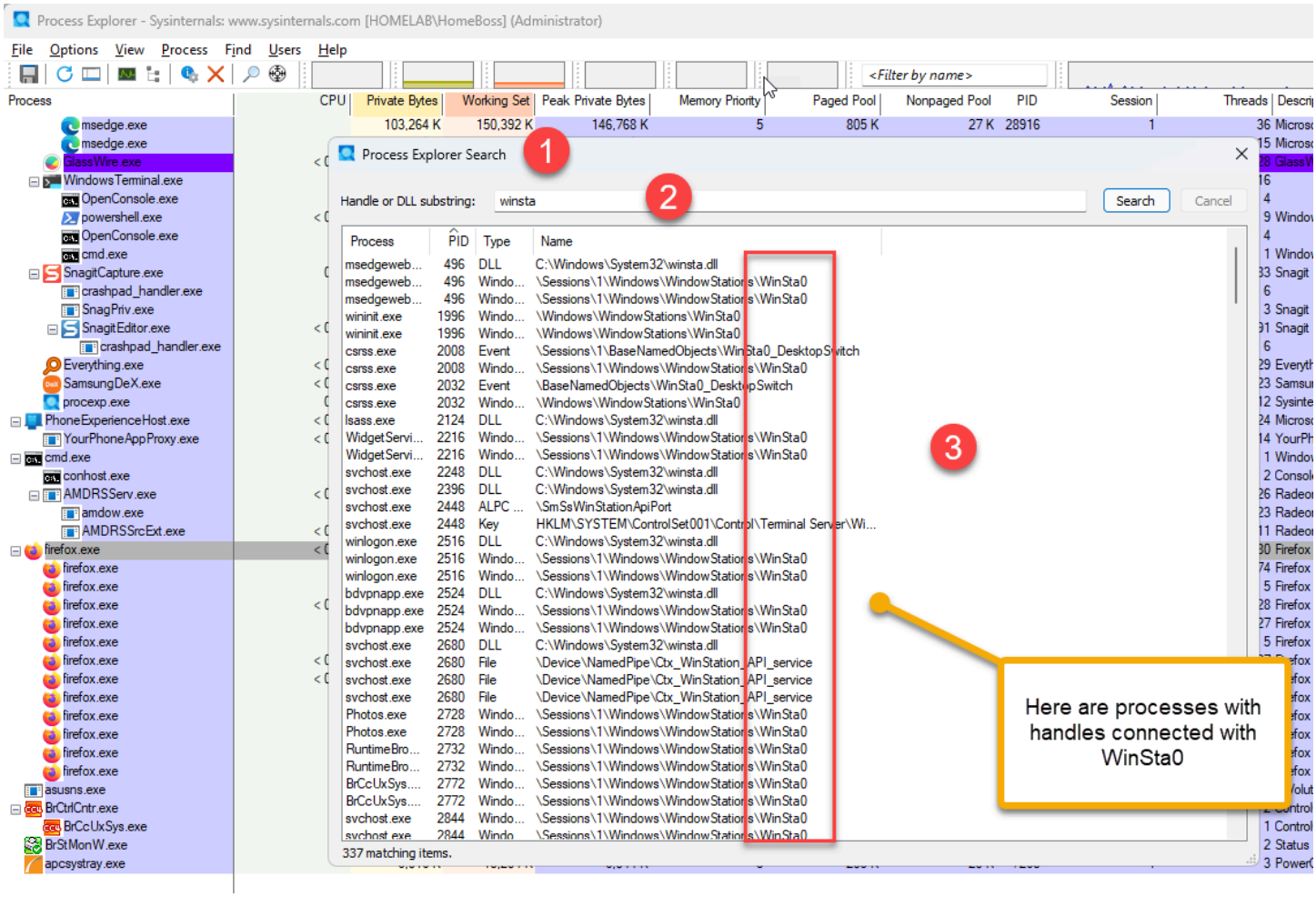
- **System processes and Windows services** always run in remote desktop services **session 0.**
- In Windows XP and Windows Server 2003, the first interactive user to log on to a computer also used session 0 and, consequently, used the same local namespace as services. Windows XP and Windows Server 2003 created sessions 1 and higher only when needed; if the first user logged off before a second one logged on, the second user used session 0 as well. Consequently, on a domain-joined Windows XP, session 0 was always the only session.
- In Windows Vista and newer, services run in session 0, but for security reasons all interactive user sessions run in sessions 1 and higher.

- This increased separation between end-user processes and system processes is called *session 0 isolation*



Window stations

- Each remote desktop services session contains one or more named *window stations*.
- A window station is a securable object that contains a clipboard, an atom table,7 and one or more desktops.
- Every process is associated with one window station. Within a session, only the window station named ***WinSta0*** can display a user interface or receive user input.



Love is patient, love is kind. It does not envy, it does not boast, it is not proud. It does not dishonor others, it is not self-seeking, it is not easily angered, it keeps no record of wrongs. Love does not delight in evil but rejoices with the truth. It always protects, always trusts, always hopes, always perseveres.

Tech Savvy Productions

```
C:\Users\homeboss.HOMELAB>handle -a | findstr WinSta
  218: ALPC Port     \SmSsWinStationApiPort
  384: Key           HKLM\SYSTEM\ControlSet001\Control\Terminal Server\WinStations
  190: WindowStation \Sessions\1\Windows\WindowStations\WinSta0
  1C0: WindowStation \Sessions\1\Windows\WindowStations\WinSta0
  7F8: File  (---)   \Device\NamedPipe\Ctx_WinStation_API_service
  804: File  (---)   \Device\NamedPipe\Ctx_WinStation_API_service
  808: File  (---)   \Device\NamedPipe\Ctx_WinStation_API_service
  1F0: Event         \BaseNamedObjects\WinSta0_DesktopSwitch
   F4: WindowStation \Sessions\1\Windows\WindowStations\WinSta0
  100: WindowStation \Sessions\1\Windows\WindowStations\WinSta0
   BC: WindowStation \Sessions\1\Windows\WindowStations\WinSta0
   D4: WindowStation \Sessions\1\Windows\WindowStations\WinSta0
   C4: WindowStation \Windows\WindowStations\WinSta0
  30C: WindowStation \Windows\WindowStations\WinSta0
   F0: WindowStation \Sessions\1\Windows\WindowStations\WinSta0
   F8: WindowStation \Sessions\1\Windows\WindowStations\WinSta0
  154: WindowStation \Sessions\1\Windows\WindowStations\WinSta0
  15C: WindowStation \Sessions\1\Windows\WindowStations\WinSta0
  1CC: WindowStation \Sessions\1\Windows\WindowStations\WinSta0
  1D4: WindowStation \Sessions\1\Windows\WindowStations\WinSta0
  154: WindowStation \Sessions\1\Windows\WindowStations\WinSta0
  15C: WindowStation \Sessions\1\Windows\WindowStations\WinSta0
  15C: WindowStation \Sessions\1\Windows\WindowStations\WinSta0
```

- In sessions 1 and higher, Windows creates only a WinSta0 window station. In session 0, in addition to WinSta0, Windows creates a separate window station for every LSA logon session associated with a service, with the locally unique identifier (LUID) of the logon session incorporated into the window station name.

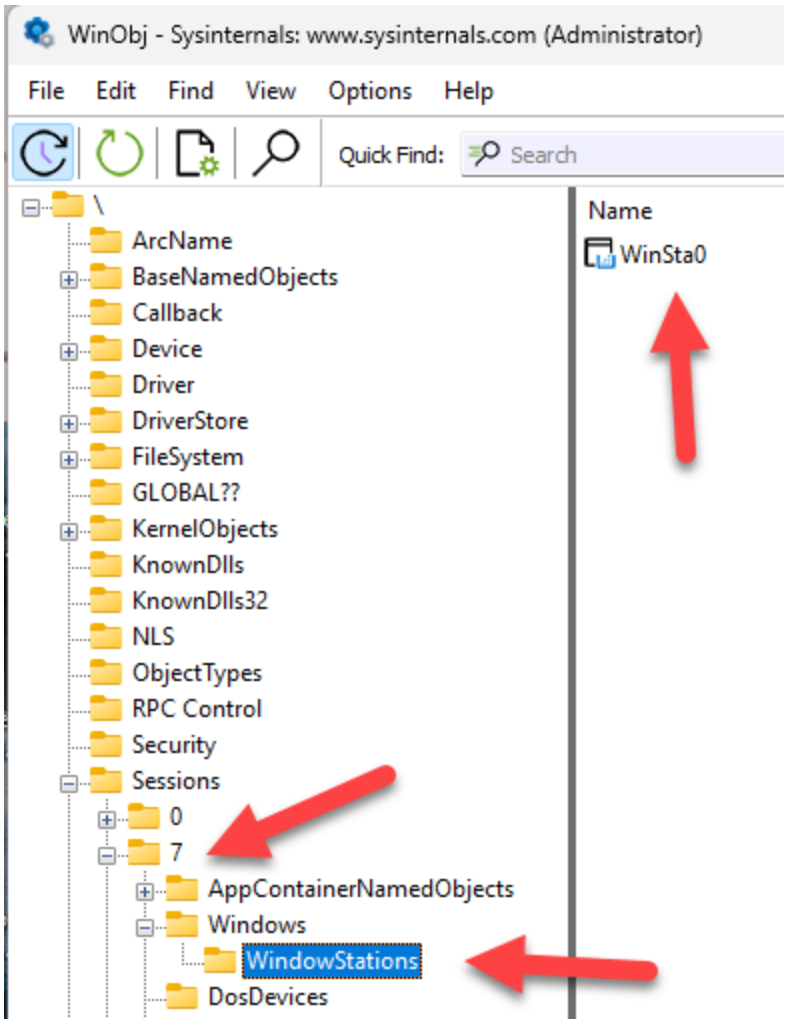
(لُوقَا 6:27-28، 32-36)

أَمَّا أَنَا فَأَقُولُ لَكُمْ: أَحِبُّوا أَعْدَاءَكُمْ، وَصَلُّوا مِنْ أَجْلِ الَّذِينَ 44 [a]›.سَمِعْتُمْ أَنَّهُ قِيلَ: ‹أَحِبَّ صَاحِبَكَ، وَأَبْغِضْ عَدُوَّكَ» 43
يَضْطَهِدُونَكُمْ، 45 فَتَكُونُوا بِذَلِكَ أَبْنَاءَ أَبِيكُمُ الَّذِي فِي السَّمَاءِ. لِأَنَّ اللهَ يَجْعَلُ الشَّمْسَ تُشْرِقُ عَلَى الخُطَاةِ وَالصَّالِحِينَ، وَيُرسِلُ
المَطَرَ إِلَى الأَبْرَارِ وَالأَشْرَارِ. 46 فَإِنْ أَحبَبْتُمْ مَنْ يُحِبُّونَكُمْ فَقَطْ، فَأَيَّةَ مُكَافَئَةٍ تَسْتَحِقُّونَ؟ أَفَلَا يَفْعَلُ جَامِعُو الضَّرَائِبِ ذَلِكَ أَيْضًا؟
47 وَإِنْ كُنْتُمْ تُحَيُّونَ إِخْوَتَكُمْ فَقَطْ، فَمَا الَّذِي يُمَيِّزُكُمْ عَنِ الآخَرِينَ؟ أَفَلَا يَفْعَلُ حَتَّى عَابِدُو الأَوْثَانِ ذَلِكَ أَيْضًا؟ 48 لِذَلِكَ كُونُوا
.كَامِلِينَ كَمَا أَنَّ أَبَاكُمُ السَّمَاوِيَّ كَامِلٌ

```
Get-Process | Select-Object Name, Id, SessionId
```

```
PS C:\Users\homeboss.HOMELAB> Get-Process | Select-Object Name, Id, SessionId

Name                           Id SessionId
----                           -- ---------
Aac3572DramHal_x86          12620         0
Aac3572MbHal_x86            16240         0
Aac3572MbHal_x86            20680         1
AacKingstonDramHal_x64      14872         0
AacKingstonDramHal_x86      14744         0
AcPowerNotification          4608         1
AggregatorHost              12276         0
ai                           9932         1
ai                          12608         1
amdfendrsr                   5760         0
amdow                       23016         1
AMDRSServ                   22280         1
AMDRSSrcExt                 23416         1
apcsystray                   7268         1
ApplicationFrameHost        11928         1
ArmouryCrate.Service         5068         0
ArmouryCrate.UserSessionHelper  4344         1
ArmourySocketServer         14736         1
ArmourySwAgent              17688         1
asus_framework               6336         1
asus_framework              16504         1
```

[43] »También han oído que se dijo: “Ama a tu prójimo y odia a tu enemigo.” [44] Pero yo les digo:
Amen a sus enemigos, y oren por quienes los persiguen. [45] Así ustedes serán hijos de su Padre
que está en el cielo; pues él hace que su sol salga sobre malos y buenos, y manda la lluvia
sobre justos e injustos. [46] Porque si ustedes aman solamente a quienes los aman, ¿qué premio
recibirán? Hasta los que cobran impuestos para Roma se portan así. [47] Y si saludan solamente a
sus hermanos, ¿qué hacen de extraordinario? Hasta los paganos se portan así. [48] Sean ustedes
perfectos, como su Padre que está en el cielo es perfecto.

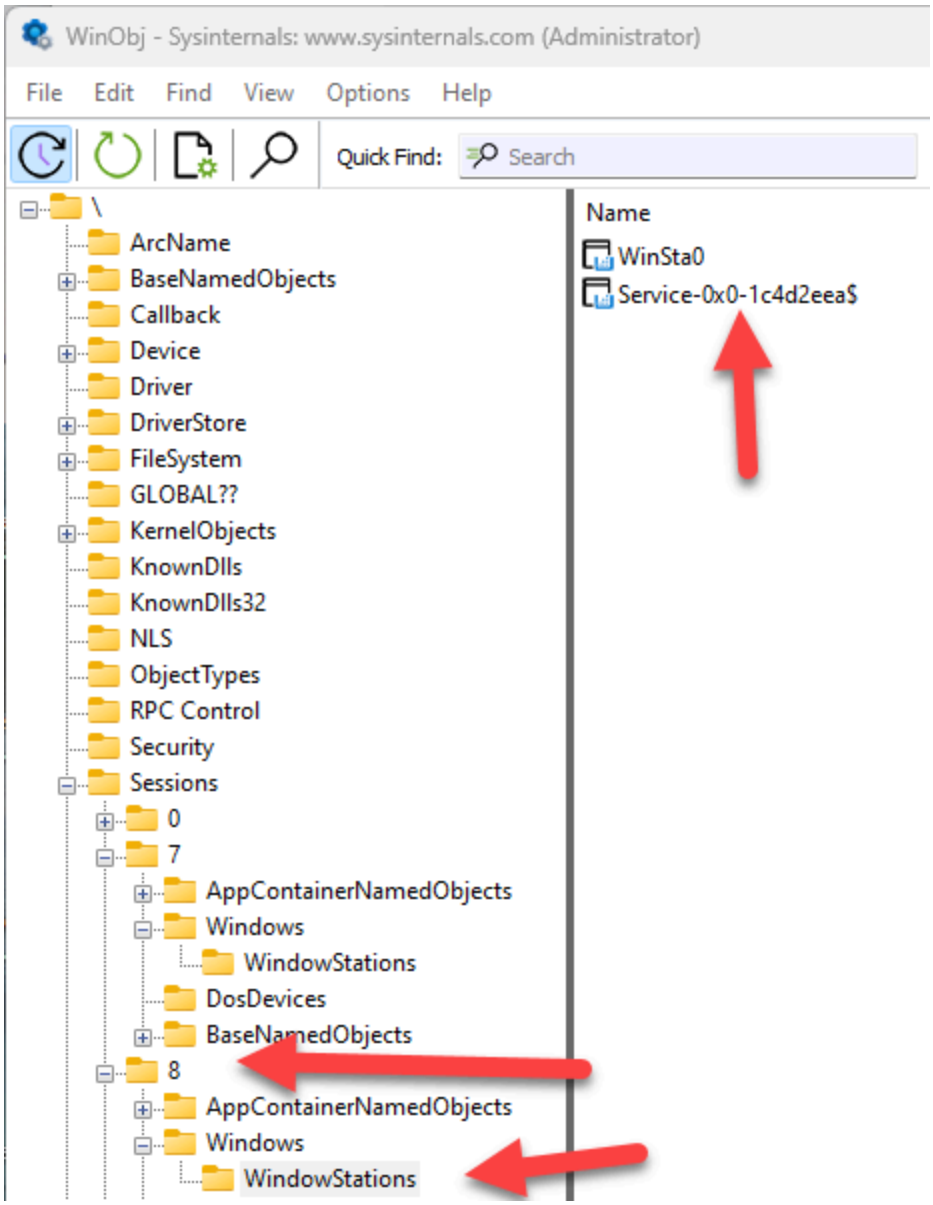
- 

  - For example, service processes that run as System run in the *Service-0x0-3e7$* window station,
  - while those that run as Network Service run in the *Service-0x0-3e4$* window station. These window stations cannot display a user interface or receive user input.

सबसे प्रेम रखो

(लूका 6:27-28, 32-36)

43 "तुमने सुना है: कहा गया है 'तू अपने पड़ौसी से प्रेम कर और शत्रु से घृणा कर।' 44 किन्तु मैं कहता हूँ अपने शत्रुओं से भी
प्यार करो। जो तुम्हें यातनाएँ देते हैं, उनके लिये भी प्रार्थना करो। 45 ताकि तुम स्वर्ग में रहने वाले अपने पिता की सिद्ध
संतान बन सको। क्योंकि वह बुरों और भलों सब पर सूर्य का प्रकाश चमकाता है। पापियों और धर्मियों, सब पर वर्षा कराता
है। 46 यह मैं इसलिये कहता हूँ कि यदि तू उन्हीं से प्रेम करेगा जो तुझसे प्रेम करते हैं तो तुझे क्या फल मिलेगा। क्या ऐसा
तो कर वसूल करने वाले भी नहीं करते? 47 यदि तू अपने भाई बंदों का ही स्वागत करेगा तो तू औरों से अधिक क्या कर रहा
है? क्या ऐसा तो विधर्मी भी नहीं करते? 48 इसलिये परिपूर्ण बनो, वैसे ही जैसे तुम्हारा स्वर्ग-पिता परिपूर्ण है।



PsExec –s cmd.exe runs a command prompt in the *Service-0x0-3e7$* window station and redirects its console I/O to PsExec. PsExec's *–i* option lets you specify the remote desktop services session and runs the target process in its WinSta0 window station.

- A service configured to run as System can also be configured to Allow Service To Interact With Desktop.
- When so configured, the service runs in session 0's *WinSta0* instead of *Service-0x0-3e7$*. When the interactive user was also in session 0, this allowed the service to interact directly with the end user through the display and user input such as the mouse and keyboard. In hindsight, this wasn't a good idea as I'll describe shortly, and Microsoft has recommended against using this technique— and with session 0 isolation, this no longer works. (The Interactive Services Detection service, UI0Detect, offers partial mitigation.)

Desktops

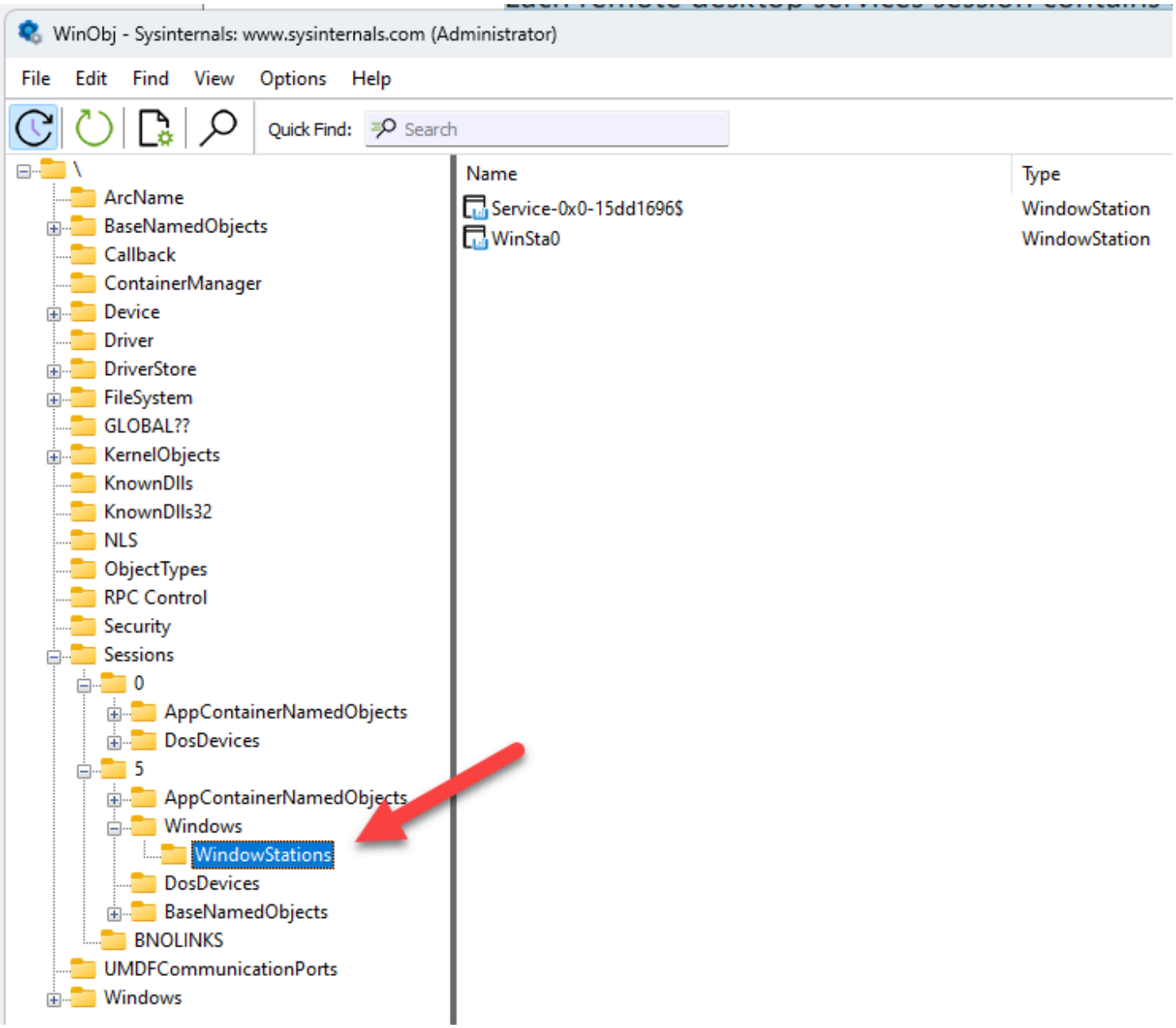
Each window station contains one or more desktops.

- A desktop is a securable object with a logical display surface on which applications can render UI in the form of windows.

> **Note** The desktops described here are unrelated to the Desktop abstraction at the top of the Windows Explorer shell namespace. Also, the Windows 10 multiple-desktops feature does not create new instances of the type of desktop described here, unlike the Sysinternals Desktops utility.

マタイの福音書 5:43-48

43 『隣人を愛し、敵を憎め』とは、よく言われることです。 44 しかし、わたしは言います。敵を愛し、迫害する人のために祈りなさい。 45 それこそ、天の父の子どもであるあなたがたに、ふさわしいことです。天の父は、悪人にも善人にも太陽の光を注ぎ、正しい人にも正しくない人にも分け隔てなく雨を降らせてくださいます。 46 自分を愛してくれる人だけを愛したからといって、取り立てて自慢できるでしょうか。悪人でも、そのくらいのことはしています。 47 気の合う友達とだけ親しくしたところで、ほかの人とどこが違うと言えるでしょう。神を信じなくても、そのくらいのことはだれでもします。 48 ですから、あなたがたは、天の父が完全であるように完全でありなさい。



- Multiple desktops can contain UI, but only one can be displayed at a time.
- **There are typically three desktops** in the interactive window station:
  - Default, The Default desktop is where user applications run by default
  - Screen-saver, The Screen-saver desktop is where Windows runs the screen saver if password protection is enabled
  - and Winlogon, The Winlogon desktop, also known as the secure desktop, is where Windows transfers control when you press Ctrl+Alt+Del and the default place to display UAC elevation dialog boxes.
- **Permissions on the Winlogon desktop restrict access only to programs running as System**, which protects secure operations involving password entry.

- As a process is associated with a window station, each of its threads is associated with a desktop within the window station.
- Although individual threads of a process can be associated with different desktops, they are usually associated with a single desktop.

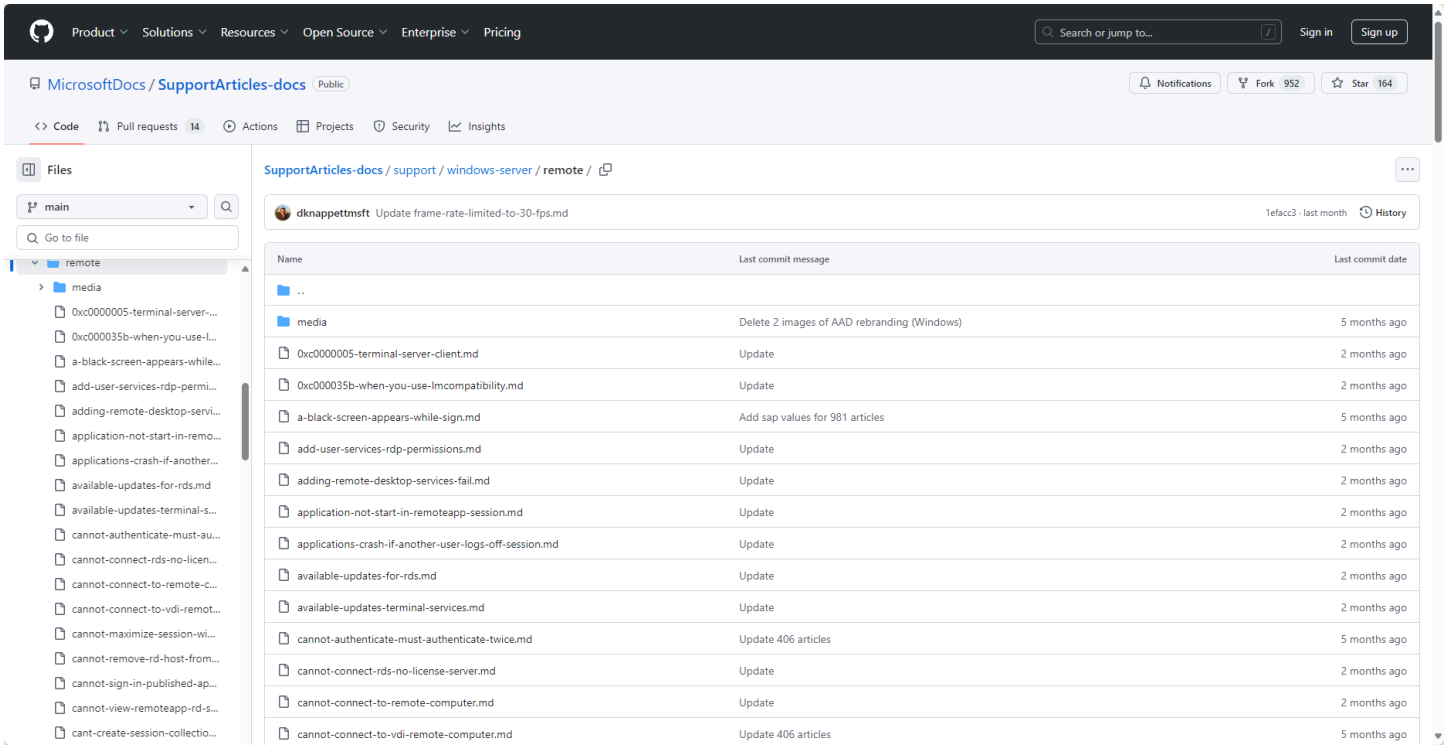
Several Sysinternals utilities, including Process Explorer and Process Monitor, identify the session ID to which a process belongs.

- Although none of the utilities directly identify the window station or desktops that a process is associated with, Process Explorer's Handle View can offer hints in the form of open handles to window stations or desktop objects.
- For example, Process Explorer shows a process running as System in session 0 with open handles to the \Default desktop and the \Windows\WindowStations\Service-0x0-3e7$ window station.

Mwagale Abalabe Bammwe

43 "Mwawulira ab'edda bwe baagambibwa nti, 'Yagalanga muliraanwa wo, naye okyawenga
mulabe wo.' 44 Naye nze mbagamba nti, Mwagalenga abalabe bammwe! Musabirenga
ababayigganya! 45 Bwe mulikola bwe mutyo muliba baana ba Kitammwe ali mu ggulu. Kubanga
omusana gwe agwakiza abalungi n'ababi, era n'enkuba agitonnyeza ab'amazima n'abatali ba
mazima. 46 Bwe mwagala ababaagala bokka, mufunamu ki? N'abawooza bwe bakola bwe batyo.
47 Era bwe munaalamusanga baganda bammwe bokka, munaaba mulina njawulo ki n'abalala?
Kubanga ne bannamawanga bwe bakola bwe batyo. 48 Naye mmwe kibagwanira okubeeranga
abatuukirivu nga Kitammwe ali mu ggulu bw'ali Omutuukirivu."

Many great articles on GitHub on troubleshooting many issues with RDC: [SupportArticles-docs/support/windows-server/remote at main · MicrosoftDocs/SupportArticles-docs · GitHub]



More articles on RDC troubleshooting: [Guidance for troubleshooting RDS session connectivity - Windows Server | Microsoft Learn]

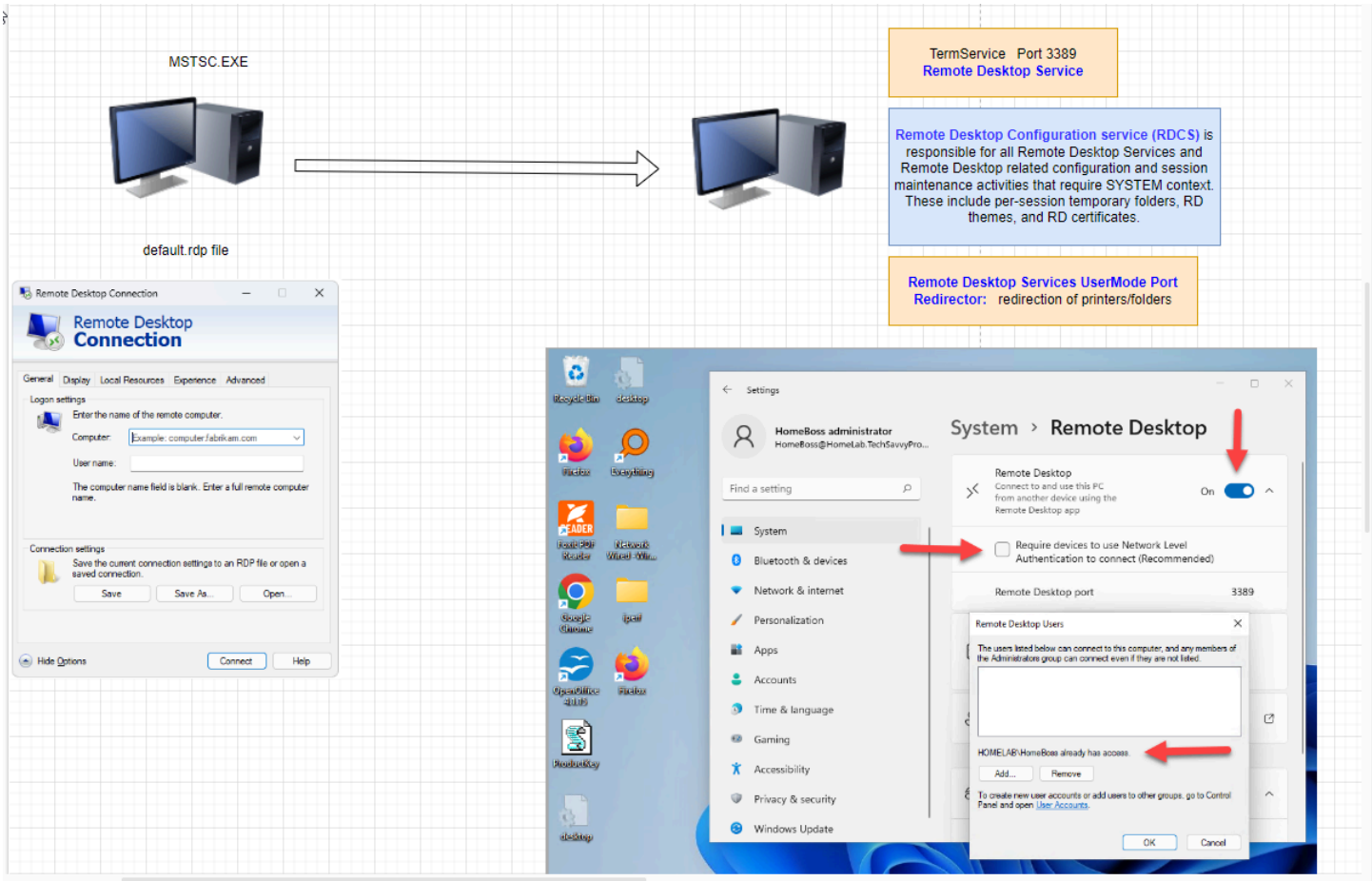
RDC (Remote Desktop Connection) Client-Side Components

RDC is an extension of the T.120 family of protocol standards:

- **T.120** is a suite of point-to-multipoint communication protocols for teleconferencing, videoconferencing, and computer-supported collaboration.[1] It provides for application sharing, online chat, file sharing, and other functions. The protocols are standardized by the ITU Telecommunication Standardization Sector (ITU-T)
- A multichannel capable protocol allows for separate virtual channels for carrying the following information:
    - presentation data
    - serial device communication
    - licensing information
    - highly encrypted data, such as keyboard, mouse activity

ਸਭ ਲੋਕਾਂ ਨੂੰ ਪਿਆਰ ਕਰੋ

43 “ਤੁਸੀਂ ਸੁਣਿਆ ਹੈ ਜੋ ਇਹ ਕਿਹਾ ਗਿਆ ਸੀ, ‘ਤੁਸੀਂ ਆਪਣੇ ਗੁਆਂਢੀ ਨਾਲ ਵੀ ਪਿਆਰ ਕਰੋ [a] ਅਤੇ ਆਪਣੇ ਵੈਰੀ ਨਾਲ
ਵੈਰ ਰੱਖੋ।’ 44 ਪਰ ਮੈਂ ਤੁਹਾਨੂੰ ਆਖਦਾ ਹਾਂ ਕਿ ਆਪਣੇ ਵੈਰੀਆਂ ਨਾਲ ਵੀ ਪਿਆਰ ਕਰੋ। ਅਤੇ ਜੋ ਤੁਹਾਨੂੰ ਸਤਾਉਣ ਉਨ੍ਹਾਂ ਲਈ
ਪ੍ਰਾਰਥਨਾ ਕਰੋ। 45 ਜੇਕਰ ਤੁਸੀਂ ਅਜਿਹਾ ਕਰੋਂਗੇ, ਫ਼ੇਰ ਤੁਸੀਂ ਆਪਣੇ ਪਿਤਾ ਦੇ ਜਿਹੜਾ ਸਵਰਗ ਵਿੱਚ ਹੈ, ਸੱਚੇ ਪੁੱਤਰ ਹੋਵੋਗੇ,
ਕਿਉਂਕਿ ਪਿਤਾ ਆਪਣਾ ਸੂਰਜ, ਬੁਰੇ ਅਤੇ ਭਲੇ ਦੋਹਾਂ ਉੱਪਰ ਹੀ ਚੜ੍ਹਾਉਂਦਾ ਹੈ। ਤੁਹਾਡਾ ਪਿਤਾ ਧਰਮੀਆਂ ਅਤੇ ਕੁਧਰਮੀਆਂ
ਉੱਪਰ ਵੀ ਮੀਂਹ ਵਰਸਾਉਂਦਾ ਹੈ। 46 ਜੇਕਰ ਤੁਸੀਂ ਉਨ੍ਹਾਂ ਨਾਲ ਹੀ ਪਿਆਰ ਕਰੋ ਜਿਹੜੇ ਤੁਹਾਡੇ ਨਾਲ ਪਿਆਰ ਕਰਦੇ ਹਨ ਤਾਂ
ਤੁਹਾਨੂੰ ਕੋਈ ਫ਼ਲ ਨਹੀਂ ਮਿਲੇਗਾ। ਕੀ ਮਸੂਲੀਏ ਵੀ ਇਹੀ ਨਹੀਂ ਕਰਦੇ? 47 ਜੇਕਰ ਤੁਸੀਂ ਸਿਰਫ਼ ਆਪਣੇ ਦੋਸਤਾਂ ਨਾਲ ਚੰਗਾ
ਵਰਤਾਓ ਕਰਦੇ ਹੋ ਤਾਂ ਤੁਸੀਂ ਦੂਸਰੇ ਲੋਕਾਂ ਨਾਲੋਂ ਚੰਗੇ ਨਹੀਂ ਹੋ। ਕੀ ਕੁਧਰਮੀ ਵੀ ਅਜਿਹਾ ਨਹੀਂ ਕਰਦੇ? 48 ਸੋ ਜਿਵੇਂ ਤੁਹਾਡਾ
ਪਿਤਾ ਜਿਹੜਾ ਸੁਰਗ ਵਿੱਚ ਹੈ, ਸੰਪੂਰਣ ਹੈ ਤਿਵੇਂ ਤੁਸੀਂ ਵੀ ਸੰਪੂਰਣ ਹੋਵੋ।



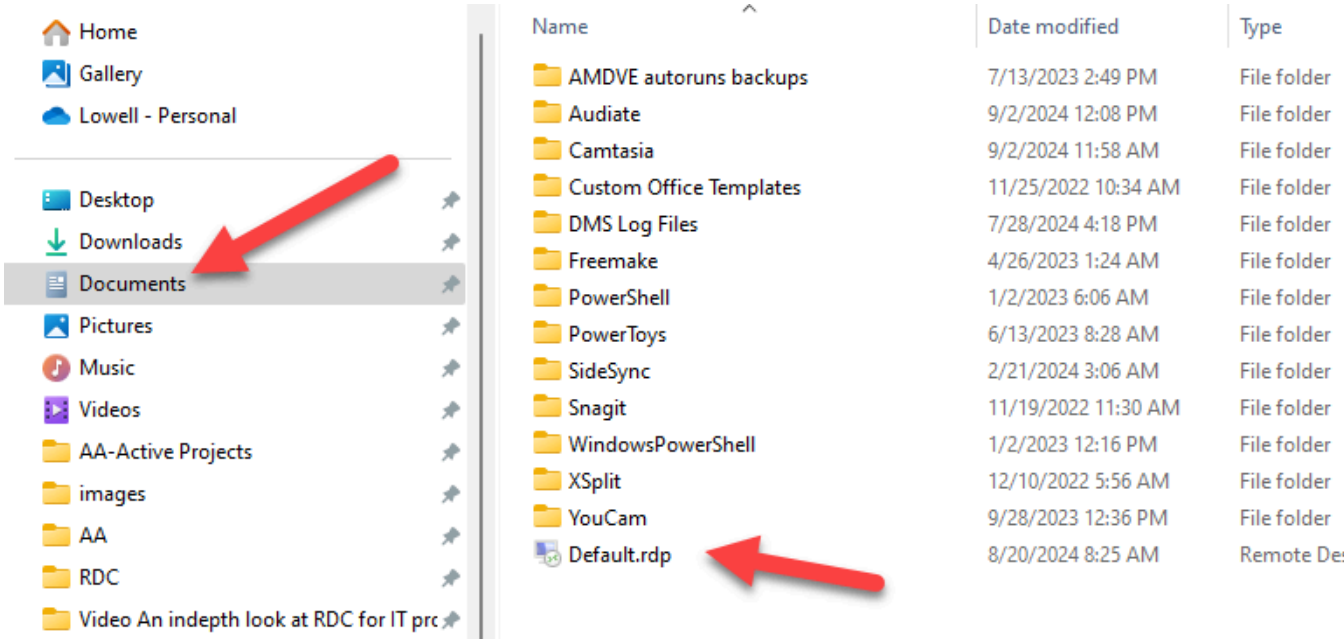
When using RDC on the client side to connect to a remote server or client, several services, processes, and components are involved. Here is an overview of the key components:

1. Remote Desktop Services (RDS) Client Services

- **TermService (Remote Desktop Services)**: This service allows users to connect to the machine via RDP. It must be running on both the client and server.
  - **PowerShell Check**: `Get-Service -Name TermService`
  - **Command-Line Check**: `sc query TermService`
  - **Troubleshooting**: If not running, start the service using `Start-Service -Name TermService` or `net start TermService`.
  - Check for errors in Event Viewer under "Applications and Services Logs > Microsoft > Windows > TerminalServices-LocalSessionManager".

2. Processes

- **mstsc.exe (Microsoft Terminal Services Client)**: The main executable that handles Remote Desktop connections.
    - **Verification**: Open Task Manager (Ctrl+Shift+Esc) and look for mstsc.exe.
    - **Command-Line Check**: `tasklist | findstr mstsc.exe`
    - **Mstsc.**exe requires a configuration file to run. Default.rdp is a hidden file in Documents folder

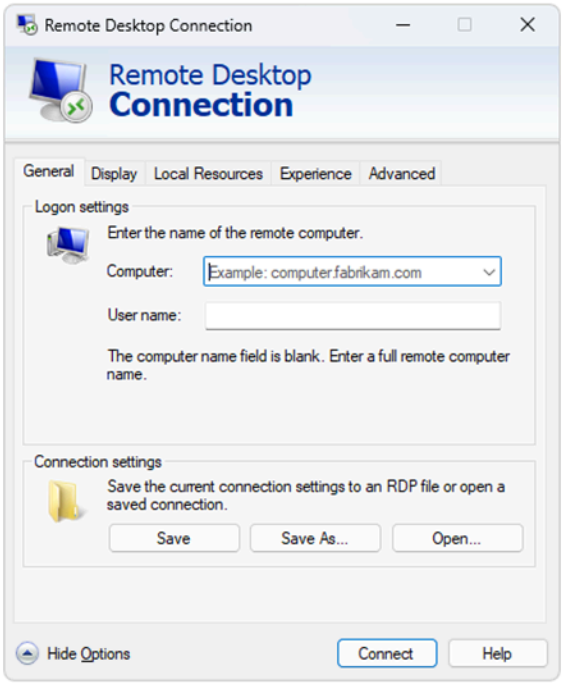



```
C:\Users\homeboss.HOMELAB>tasklist | findstr mstsc.exe
mstsc.exe                     7964 Console                    1    243,720 K
```

Любовь к врагам

43 Вы слышали, что было сказано: «Люби ближнего твоего»[a] и «Ненавидь врага твоего»[b]. 44 Я же говорю вам: любите ваших врагов и молитесь о тех, кто преследует вас, 45 чтобы вам быть истинными сынами вашего Небесного Отца. Ведь Он повелевает солнцу светить и злым, и добрым и посылает дождь как на праведных, так и на неправедных. 46 Если вы любите тех, кто любит вас, то какая вам за это награда? Не делают ли то же самое даже сборщики налогов?[c] 47 Если вы приветствуете только ваших братьев, то что в этом особенного? Разве язычники не делают того же? 48 Поэтому будьте совершенны, как совершенен ваш Небесный Отец.

3. Critical Components



- **RDP Client DLLs**:
  - **mstscax.dll**: ActiveX control that drives the RDC client interface. If corrupted or missing, RDC will not function correctly.
    - **Verification**: Use SFC.exe to replace the DLL if it is missing or corrupted.
- **Rdpclip.exe (Remote Desktop Clipboard)**: Manages clipboard redirection between the client and the remote session.
  - **Verification**: Check the process in Task Manager or use `tasklist | findstr rdpclip.exe`.
  - **Troubleshooting**: If clipboard redirection fails, restart rdpclip.exe by killing the process and relaunching it from the command line: `taskkill /im rdpclip.exe /f & start rdpclip.exe`.

4. Network Components

- **Port 3389**: The default port for RDP communication.
  - **Verification**: Use `Test-NetConnection -ComputerName <remote_server> -Port 3389` in PowerShell or
- **Firewall Rules**: Ensure that both inbound and outbound rules for RDP (TCP 3389) are allowed.
  - **Verification**:
    - **PowerShell**: `Get-NetFirewallRule | Where-Object { $_.DisplayName -like "*Remote*Desktop*" -and $_.Enabled -eq "True" }`
    - **GUI**: Use "Windows Defender Firewall with Advanced Security" to manually check or modify rules.

5. Group Policy Settings

- **Group Policy Object (GPO) Settings**: Specific policies like "Allow users to connect remotely using Remote Desktop Services" can be configured to enable or disable RDC.
  - **Verification**:
    - **PowerShell**: `Get-GPResultantSetOfPolicy -Computer <ComputerName> -ReportType HTML -Path C:\gporeport.html` (Open the HTML file in a browser to view applied policies).

- **Command Line**: `gpresult /h C:\gporeport.html`

6. Certificates and Encryption

- **TLS/SSL Certificates**: Used to secure RDP sessions. Issues with certificates (like expired or self-signed certificates) can prevent connections.
  - **Verification**: Check certificates in "Certificate Manager" (certmgr.msc) under "Remote Desktop" folder.
- **PowerShell Check**: `Use Get-ChildItem -Path Cert:\LocalMachine\Remote Desktop` to view and manage certificates.

7. DNS and Name Resolution

- **DNS Settings**: The client must resolve the remote server's name correctly.
  - **Verification**: Use `nslookup <remote_server>` or `ping <remote_server>` to check name resolution.

8. Third-Party Tools for Troubleshooting

- **Other Tools**:
  - **Wireshark**: For deep packet inspection and analysis of RDP traffic. You can filter for tcp.port == 3389 to view only RDP traffic.

Verifying and Troubleshooting Steps on Remote computer

1. **Verify Services**: Ensure TermService is running.
2. **Check Processes**: rdpclip.exe is running.
3. **Verify Network Configuration**: Ensure RDP port 3389 is open and allowed through the firewall.
4. **Check Group Policies**: Review GPO settings related to Remote Desktop.
5. **Validate Certificates**: Ensure certificates are valid and trusted.

Steps to Verify RDP Status Using PowerShell on remote host

1. Start a Remote PowerShell Session

To begin, open a remote PowerShell session to the target computer. Use the Enter-PSSession or Invoke-Command cmdlet to run commands remotely.

```
Enter-PSSession -ComputerName RemoteComputerName -Credential
(Get-Credential)
```

Or, use Invoke-Command to run commands without entering an interactive session:

```
Invoke-Command -ComputerName RemoteComputerName -ScriptBlock {
<Your Commands> } -Credential (Get-Credential)
```

2. Check the Status of RDP Services

```
Get-Service -Name TermService
```

This will return the status of the Remote Desktop Services. The status should be Running if RDP is enabled.

3. Check RDP Port Configuration in the Registry

Ensure that RDP is enabled and check the RDP port (default is 3389). Use the following command:

```
Invoke-Command -ComputerName RemoteComputerName -ScriptBlock {

    Get-ItemProperty -Path
'HKLM:\System\CurrentControlSet\Control\Terminal Server\' -Name
fDenyTSConnections

}
```

If fDenyTSConnections is set to 0, it means that RDP connections are allowed. If it is set to 1, RDP connections are disabled.

4. Verify the Windows Firewall Configuration for RDP

Check if the Windows Firewall is allowing RDP traffic:

```
Invoke-Command -ComputerName RemoteComputerName -ScriptBlock {
    Get-NetFirewallRule -DisplayName "Remote Desktop - User
Mode (TCP-In)"
}
```

Make sure the firewall rule is enabled (Enabled should be True), and the action is set to Allow.

5. Check the Listening State of RDP Port

To verify that the RDP port (TCP 3389) is listening, use the following command:

```
Action                              : 2
```

This command checks if the RDP port is listening on the remote computer.

Conclusion

By combining these checks, you can determine whether RDP is enabled and properly configured on a remote PC. This approach helps to diagnose connectivity issues, verify service status, check firewall configurations, and ensure that the necessary ports are open and listening.

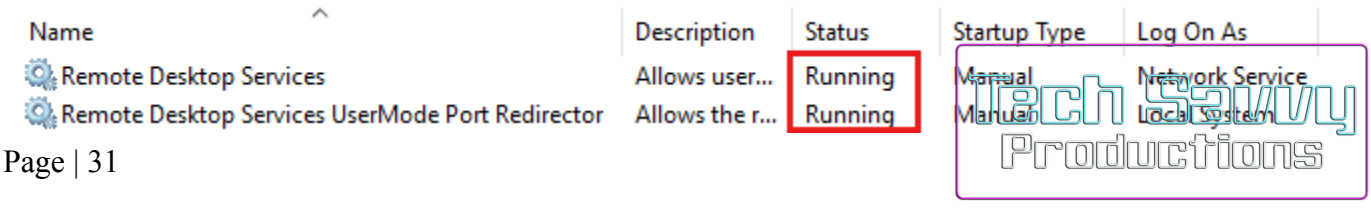
For more information on using PowerShell for remote management, you can refer to the Microsoft PowerShell Documentation.

Check the status of the RDP services

On both the local (client) computer and the remote (target) computer, the following services should be running:

- Remote Desktop Services (TermService)
- Remote Desktop Services UserMode Port Redirector (UmRdpService)

You can use the Services MMC snap-in to manage the services locally or remotely. You can also use PowerShell to manage the services locally or remotely (if the remote computer is configured to accept remote PowerShell cmdlets).



Check the status of the RDP listener

For this procedure, use a PowerShell instance that has administrative permissions. For a local computer, you can also use a command prompt that has administrative permissions. However, this procedure uses PowerShell because the same cmdlets work both locally and remotely.

1. To connect to a remote computer, run the following cmdlet:

PowerShell

Enter-PSSession -ComputerName <computer name>

Enter qwinsta.

```
[rdwagw01]: PS C:\> qwinsta
 SESSIONNAME       USERNAME                 ID  STATE   TYPE        DEVICE
>services                                    0  Disc
 console                                     1  Conn
 31c5ce94259d4...                        65536  Listen
 rdp-tcp                                 65537  Listen
```

If the list includes rdp-tcp with a status of Listen, the RDP listener is working. Proceed to Check the RDP listener port.

Explanation of the Output

1. **ID (Session ID)**:
   Each session has a unique identifier (ID) assigned by the system. This ID helps differentiate between different sessions. The IDs are numeric values that indicate the session number.
2. **State**:
   This represents the current state of the session. The common states you might see are:
   - **Disc (Disconnected)**: The session is disconnected from the client, but still exists on the server. The user's session remains active in the background and can be reconnected.
   - **Conn (Connected)**: The session is actively connected to a client. The user is currently logged in and interacting with the session.
   - **Listen**: The server is in a "listening" state, waiting for a new connection to be established.
3. **Console**:
   The console session (ID 1) represents the physical keyboard and monitor directly attached to the computer. This is usually the session that is initiated when someone logs into the machine locally, rather than remotely.

4. **rdp-tcp**:
   This indicates the protocol and transport mechanism being used for Remote Desktop Protocol (RDP) sessions. The rdp-tcp entry with ID 65536 and ID 65537 in a "Listen" state shows that the RDSH or computer is listening for incoming RDP connections on the TCP protocol. Multiple listening IDs (e.g., 65536, 65537) represent different listeners or configurations for handling RDP traffic.
5. **Session Names**:
   This column (in your example, the entries are blank or filled with GUID-like strings such as 31c5ce94259d4...) can display the name of the session or the unique identifier associated with it. Sometimes, it may show a unique identifier representing the session's context.

Otherwise, continue at step 4.

Export the RDP listener configuration from a working computer.

1. Sign in to a computer that has the same operating system version as the affected computer has, and access that computer's registry (for example, by using Registry Editor).
2. Navigate to the following registry entry:

HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Terminal Server\WinStations\RDP-Tcp

3. Export the entry to a .reg file. For example, in Registry Editor, right-click the entry, select **Export**, and then enter a filename for the exported settings.
4. Copy the exported .reg file to the affected computer.

To import the RDP listener configuration, open a PowerShell window that has administrative permissions on the affected computer (or open the PowerShell window and connect to the affected computer remotely).

1. To back up the existing registry entry, enter the following cmdlet:

cmd /c 'reg export "HKLM\SYSTEM\CurrentControlSet\Control\Terminal Server\WinStations\RDP-tcp" C:\Rdp-tcp-backup.reg'

To remove the existing registry entry, enter the following cmdlets:

Remove-Item -path 'HKLM:\SYSTEM\CurrentControlSet\Control\Terminal Server\WinStations\RDP-tcp' -Recurse -Force

To import the new registry entry and then restart the service, enter the following cmdlets:

3. cmd /c 'regedit /s c:\<filename>.reg'
4. Restart-Service TermService -Force

Replace <filename> with the name of the exported .reg file.

Test the configuration by trying the remote desktop connection again. If you still can't connect, restart the affected computer.

If you still can't connect, check the status of the RDP self-signed certificate.

Check the status of the RDP self-signed certificate

1. Has to be run with admin credentials on a domain.

```
# Prompt for target computer name

$TargetComputer = Read-Host -Prompt "Enter the name of the remote computer (e.g., PC1234 or IP address)"



# Prompt for credentials with a description

$Credential = Get-Credential -Message "Enter credentials for the remote computer ($TargetComputer). Use a domain account with administrative privileges."



# Display the initial operation

Write-Output "Attempting to connect to $TargetComputer with provided credentials..."



# Define a script block for operations on the remote machine

$ScriptBlock = {

    param ($ComputerName)
```

```
Write-Output "Stopping Remote Desktop Services (TermService) on $ComputerName..."

try {
    # Stop the Remote Desktop Services
    Stop-Service -Name "TermService" -Force -ErrorAction Stop
    Write-Output "Remote Desktop Services stopped successfully."
}
catch {
    Write-Output "Failed to stop Remote Desktop Services: $_"
    exit 1
}

Write-Output "Deleting the RDP self-signed certificate..."

try {
    # Define the certificate thumbprint location for RDP
    $RdpCertificatePath = "HKLM:\SOFTWARE\Microsoft\SystemCertificates\Remote Desktop\Certificates"

    # Remove the RDP self-signed certificate
    Remove-Item -Path "$RdpCertificatePath\*" -Recurse -Force -ErrorAction Stop
    Write-Output "RDP self-signed certificate deleted successfully."
```

```
    }
    catch {
        Write-Output "Failed to delete the RDP self-signed certificate: $_"
        exit 1
    }

    Write-Output "Restarting Remote Desktop Services (TermService)..."

    try {
        # Start the Remote Desktop Services
        Start-Service -Name "TermService" -ErrorAction Stop
        Write-Output "Remote Desktop Services restarted successfully."
    }
    catch {
        Write-Output "Failed to restart Remote Desktop Services: $_"
        exit 1
    }

    Write-Output "Operation completed successfully on $ComputerName."
}

# Execute the script block on the remote computer
try {
```

```
    Invoke-Command -ComputerName $TargetComputer -Credential $Credential -ArgumentList
$TargetComputer -ScriptBlock $ScriptBlock -ErrorAction Stop

    Write-Output "Script executed successfully on $TargetComputer."

}

catch {

    Write-Output "Script execution failed on $TargetComputer: $_"

}
```

2. has not been recreated, check the permissions of the MachineKeys folder.

Check the permissions of the MachineKeys folder

1. On the affected computer, open Explorer, and then navigate to *C:\ProgramData\Microsoft\Crypto\RSA*.
2. Right-click **MachineKeys**, select **Properties** > **Security** > **Advanced**.
3. Make sure that the following permissions are configured:
    - **Builtin\Administrators**: **Full control**
    - **Everyone**: **Read**, **Write**

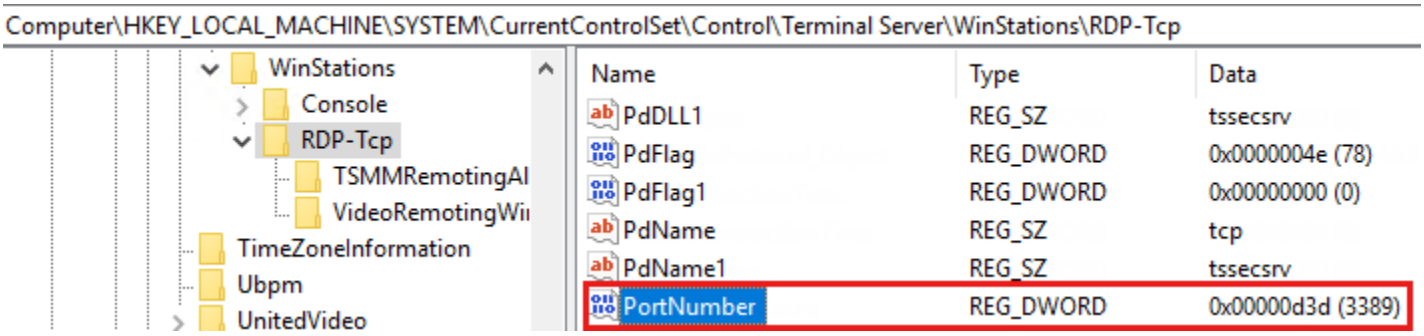
Check the RDP listener port

On both the local (client) computer and the remote (target) computer, the RDP listener should be listening on port 3389. No other applications should be using this port.

To check or change the RDP port, use the Registry Editor:

1. Go to the Start menu, select **Run**, then enter *regedt32* into the text box that appears.
    - To connect to a remote computer, select **File**, and then select **Connect Network Registry**.
    - In the **Select Computer** dialog box, enter the name of the remote computer, select **Check Names**, and then select **OK**.

2. Open the registry and navigate to HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Terminal Server\WinStations\<listener>.



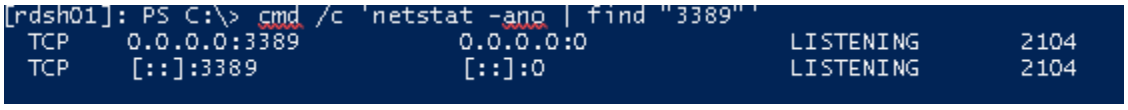
3. If **PortNumber** has a value other than **3389**, change it to **3389**.

Check that another application isn't trying to use the same port

For this procedure, use a PowerShell instance that has administrative permissions. For a local computer, you can also use a command prompt that has administrative permissions. However, this procedure uses PowerShell because the same cmdlets work locally and remotely.

1. Open a PowerShell window. To connect to a remote computer, enter Enter-PSSession -ComputerName <computer name>.
2. Enter the following command:

```
cmd /c 'netstat -ano | find "3389"'
```



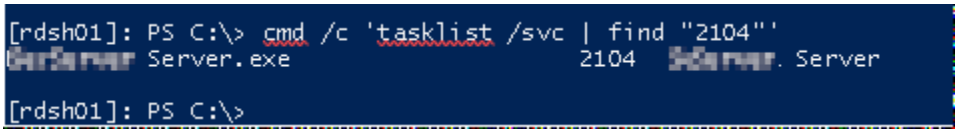
Look for an entry for TCP port 3389 (or the assigned RDP port) with a status of **Listening**.

Note

The process identifier (PID) for the process or service using that port appears under the PID column.

To determine which application is using port 3389 (or the assigned RDP port), enter the following command:

```
cmd /c 'tasklist /svc | find "<pid listening on 3389>"'
```



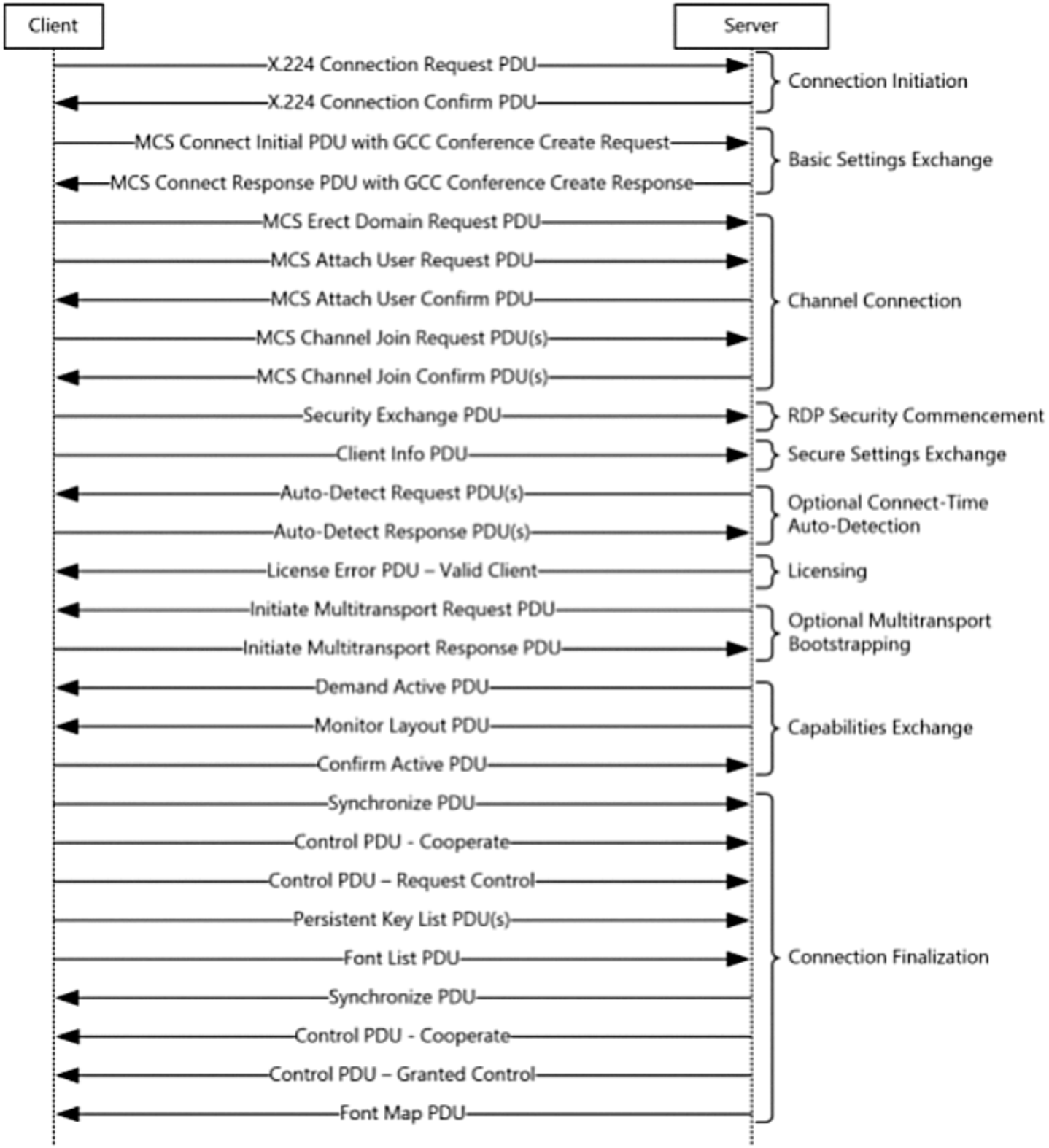
Look for an entry for the PID number that is associated with the port (from the netstat output). The services or processes that are associated with that PID appear on the right column.

If an application or service other than Remote Desktop Services (*TermServ.exe*) is using the port, you can resolve the conflict by using one of the following methods:

- Configure the other application or service to use a different port (recommended).
- Uninstall the other application or service.
- Configure RDP to use a different port, and then restart the Remote Desktop Services service (not recommended).

Remote Desktop Handshake sequence

[43] ท่านได้ยินคำที่กล่าวไว้ว่า ‘จงรักเพื่อนบ้านของเจ้า[a] และเกลียดชังศัตรูของเจ้า’ [44] แต่เราขอบอกท่านว่า จงรักศัตรูของท่าน และอธิษฐานให้บรรดาคนที่กดขี่ข่มเหงท่าน [45] เพื่อว่าท่านจะได้เป็นบรรดาบุตรของพระบิดาในสวรรค์ของท่าน เพราะว่าพระองค์เป็นผู้ที่ให้ดวงอาทิตย์ของพระองค์ขึ้นส่องยังคนชั่วและคนดี และโปรดให้ฝนโปรยลงบนคนที่มีความชอบธรรมและคนที่ไม่มีความชอบธรรม [46] ถ้าหากว่าท่านรักบรรดาผู้ที่รักท่าน แล้วท่านจะได้รางวัลอะไรเล่า พวกคนเก็บภาษีมิได้ทำเช่นนั้นด้วยหรือ [47] ถ้าท่านพูดทักทายกับพี่น้องของท่านเท่านั้น ท่านทำอะไรเกินกว่าคนอื่นๆ เล่า แม้แต่บรรดาคนนอกก็ยังทำอย่างนั้นมิใช่หรือ [48] ฉะนั้นท่านจงเป็นคนดีเพียบพร้อมทุกประการ เช่นเดียวกับพระบิดาในสวรรค์ของท่านที่ดีเพียบพร้อมทุกประการ



RDC Components in Windows 11

Windows 11 RDC Components for the RDC server:

1. **mstsc.exe (Microsoft Terminal Services Client)**
    - **Function**: This is the executable for the Remote Desktop Connection client. It provides the user interface to connect to a Remote Desktop server.
    - **Type**: Process
    - **Details**: Located in `C:\Windows\System32\mstsc.exe`.
2. **termsrv.dll (Terminal Server Remote Protocol Service DLL)**
    - **Function**: Implements the core functionality of the RDC server protocol stack, allowing multiple remote desktop connections.
    - **Type**: DLL
    - **Details**: Located in `C:\Windows\System32\termsrv.dll`.
3. **rdpclip.exe (Remote Desktop Clipboard Process)**
    - **Function**: Provides clipboard redirection between the client and the server in an RDC session.
    - **Type**: Process
    - **Details**: Runs in the background to support the sharing of clipboard data.
4. **rdpshell.exe (Remote Desktop Shell)**
    - **Function**: Manages the remote desktop shell environment, providing a remote user interface and session management.
    - **Type**: Process
    - **Details**: Supports seamless windowing and session management features.
5. **Remote Desktop Services (TermService)**
    - **Function**: Allows multiple users to connect interactively to a computer and manage remote desktops.
    - **Type**: Service
    - **Service Name**: TermService
    - **Details**: Manages all remote desktop connections to the computer.
6. **rdpinit.exe (RDP Initialization Process)**
    - **Function**: Initializes a remote desktop session, setting up the connection parameters and required environment.
    - **Type**: Process
    - **Details**: Responsible for session startup and initial configuration.

Services, Processes, and Critical Components of Remote Desktop Connection (RDC) on the Server Side for Windows Server 2016 -2025

1. Remote Desktop Services (RDS) and its Components:

- **Remote Desktop Services (RDS) / TermService:**
  The core service responsible for handling remote desktop sessions. It manages remote desktop connections, licensing, and terminal services.
  Troubleshooting:
  - Check if the service is running:
    - **PowerShell:** `Get-Service -Name TermService`
    - **Command Line:** `sc query TermService`
    - **GUI Tool:**
      - Open Services.msc and locate "Remote Desktop Services."
  - Restart the service if needed:
    - **PowerShell:** `Restart-Service -Name TermService`
    - **Command Line:** `net stop TermService && net start TermService`

2. Remote Desktop Protocol (RDP):

- **Rdpinit.exe / Rdpclip.exe:**
  These processes are involved in initializing the RDP session and clipboard redirection. They ensure that the session starts correctly and that the clipboard can be shared between the client and the server.
  Troubleshooting:
  - Use Task Manager or **PowerShell**: `Get-Process -Name rdpinit, rdpclip`
  - *Restart if needed:* `Stop-Process -Name rdpinit, rdpclip -Force`

    `Start-Process -FilePath "C:\Windows\System32\rdpclip.exe"`

3. Remote Desktop Configuration Service:

- **SessionEnv (Remote Desktop Configuration Service):**
  Configures RDP settings for the server, ensuring sessions are correctly established based on policies.
  Troubleshooting:
  - Verify the service is running:
    - **PowerShell:** `Get-Service -Name SessionEnv`
  - Restart if needed:
    - **PowerShell:** `Restart-Service -Name SessionEnv`

4. Remote Desktop Licensing Service:

- **TermServLicensing:**
  Manages Remote Desktop Client Access Licenses (CALs) that are required for client connections.
  Troubleshooting:
  - Check service status:
    - **PowerShell:** `Get-Service -Name TermServLicensing`
  - Review license status and errors in Event Viewer (eventvwr.msc):
    - Look under Applications and Services Logs > Microsoft > Windows > RemoteDesktopServices-Licensing.

5. Windows Audio Endpoint Builder:

- **AudioSrv (Windows Audio Endpoint Builder):**
  Ensures audio is redirected correctly from the server to the client for applications requiring sound.
  Troubleshooting:
  - Check service status:
    - **PowerShell:** `Get-Service -Name AudioSrv`
  - Restart if needed:
    - **PowerShell:** `Restart-Service -Name AudioSrv`

6. Remote Desktop Gateway Service:

- **Tsgateway (Remote Desktop Gateway):**
  Allows secure access to remote desktops through HTTPS, often used in corporate environments.
  Troubleshooting:

- Check the service:
  - **PowerShell:** `Get-Service -Name TSGateway`
- Restart service if needed:
  - **PowerShell:** `Restart-Service -Name TSGateway`

7. RDS Endpoint and RDP-TCP Listener:

- **Listener Configuration (RDP-TCP):**
  Manages the network connections that the server listens for RDP connections.
  Troubleshooting:
  - Check for active listeners:
    - **PowerShell:**
      ```
      Get-WmiObject -Namespace root\cimv2\terminalservices -Class Win32_TSGeneralSetting | Select-Object TerminalName, SecurityLayer, EncryptionLevel
      ```
  - Verify listening ports:
    - **Command Line:** `netstat -a | findstr :3389`

8. RemoteFX and other RDP-related DLLs (e.g., rdpcore.dll, rdpbase.dll)

- **Function**: Enhances remote desktop sessions with GPU acceleration and USB redirection.
- **Type**: DLL
- **Details**: Supports advanced multimedia and peripheral functionality in RDS.

9. Remote Desktop Web Access (rdweb.exe)

- **Function**: Provides a web-based gateway for remote desktop sessions and RemoteApps.
- **Type**: Service
- **Service Name**: Rdweb (Remote Desktop Web Access)

10. Remote Desktop Connection Broker (Tssdis.exe)

- **Function**: Manages connections to available RD Session Host servers in the farm. Supports load balancing and reconnecting a user to an existing session.
- **Type**: Service
- **Service Name**: Tssdis (Remote Desktop Connection Broker)
- **Details**: Critical for managing multiple session hosts and directing traffic properly.

Tools and Utilities for Troubleshooting

1. PowerShell Commands:

- **`Get-RDSessionHost`:** Retrieves information about session hosts in a Remote Desktop deployment.
- **`Test-NetConnection`:** Checks if a remote server is reachable over RDP port (default 3389).

```
Test-NetConnection -ComputerName <ServerName> -Port 3389
```

2. Command-Line Utilities:

- **qwinsta / query session:**
  Displays the status of all sessions on a terminal server.

```
qwinsta /server:<ServerName>
```

- **quser / query user:**
  Displays information about user sessions on the RD session host.

```
quser /server:<ServerName>
```

- **mstsc /admin:**
  Use this to connect to a remote server in admin mode, bypassing the RDP user limit.

3. GUI Tools:

- **Remote Desktop Services Manager (tsadmin.msc):**
  Allows managing remote desktop sessions, disconnecting users, and monitoring session status.
- **Event Viewer (eventvwr.msc):**
  Monitor and analyze the Event Logs for Remote Desktop Services.

4. Third-Party Tools:

- **NirSoft Utilities:**
  - **CurrPorts:** Displays the list of all currently opened TCP/IP and UDP ports on the local computer, helping to identify whether the RDP port is in use.
  - **NetworkLatencyView:** Monitors network latency and helps identify network issues impacting RDP performance.

- **Sysinternals Tools:**
  - **TCPView:** Shows detailed listings of all TCP and UDP endpoints, useful for verifying that the RDP listener is active.
  - **PsPing:** Network utility for testing latency and packet loss for RDP connections.

Components of Remote Desktop Connection (RDC) in Microsoft Products

1. RDC Components in Microsoft VDI Products:

Virtual Desktop Infrastructure (VDI):
Yes, RDC components are integral to Microsoft's Virtual Desktop Infrastructure (VDI) products. VDI allows users to create a virtualized environment where desktops are hosted on a

centralized server. Microsoft's implementation of VDI is heavily dependent on Remote Desktop Services (RDS) and its underlying RDC components.

- **RDS and VDI Relationship:**
  - **Remote Desktop Services (RDS):** Acts as the foundation for Microsoft's VDI solutions. It provides the infrastructure to deliver virtualized applications, session-based desktops, and virtual desktops.
  - **Remote Desktop Protocol (RDP):** The protocol used to connect to these VDI environments. RDP components, such as TermService and rdpinit.exe, enable secure connections to virtual desktops.
  - **Remote Desktop Gateway (RD Gateway):** Helps manage secure connections from outside the corporate network to the internal VDI infrastructure.

URL for more details: Microsoft Remote Desktop Services and VDI

2. RDC Components Used by Remote Assistance:

Remote Assistance: Yes, Remote Assistance uses several RDC components. Remote Assistance is a Windows feature that allows a user to temporarily share control of their computer with someone they trust (e.g., an IT support technician).

- **Common Components with RDC:**
  - **Remote Desktop Protocol (RDP):** Both RDC and Remote Assistance use RDP for establishing connections. However, while RDC is used for full desktop control, Remote Assistance allows shared control with the option to request and release control back to the user.
  - **Session Manager Components:** The same components that manage RDC sessions (rdpclip.exe, rdpinit.exe, etc.) also manage Remote Assistance sessions.
  - **Port 3389:** Remote Assistance uses the same port (3389) for RDP communication, which can lead to conflicts if multiple RDP-based services are running.

URL for more details: Remote Assistance Overview

3. Other Microsoft Products and Features Utilizing RDC Components:

- Windows Admin Center:
  Utilizes RDP for remote server management, allowing administrators to manage servers from a central web-based console.
  - **URL:** Windows Admin Center Overview
- Microsoft Endpoint Manager:
  Integrates with RDP to manage remote devices, enabling IT administrators to access remote desktops directly.

    - **URL:** Microsoft Endpoint Manager
- Azure Virtual Desktop (AVD):
  Uses RDC components to provide a full desktop experience in the Azure cloud. AVD relies on RDP for secure connections between the client and Azure-hosted virtual desktops.
    - **URL:** Azure Virtual Desktop
- Microsoft Intune:
  Uses RDC for remote management and support features, allowing administrators to connect to user desktops to assist with configuration or troubleshooting.
    - **URL:** Microsoft Intune Overview

4. Older Microsoft Products Utilizing RDC Components:

- Windows Home Server:
  Allowed remote access to files and applications via RDP, enabling users to connect remotely to their home network.
    - **URL:** Windows Home Server Documentation
- Small Business Server (SBS):
  Used RDC to provide remote access to desktops and applications within a small business network.
    - **URL:** Windows Small Business Server Overview

Remote Desktop Protocol (RDP)

1. Microsoft Learn - Remote Desktop Protocol (RDP)

- An official overview of RDP from Microsoft, detailing the protocol's design, security features, and capabilities.
- **URL:** Microsoft Learn - Remote Desktop Protocol (RDP)

2. MS-RDP: Remote Desktop Protocol Technical Specifications

- A comprehensive technical document by Microsoft detailing the RDP protocol's specifications, including transport protocols, security, and session management.
- **URL:** MS-RDP: Remote Desktop Protocol Technical Specifications

3. Microsoft Security Guidance for Remote Desktop Protocol

- Provides security best practices and configuration guidelines for securing RDP connections to prevent unauthorized access and mitigate security threats.
- **URL:** Microsoft Security Guidance for Remote Desktop Protocol

4. RDP Security in Windows Server - Microsoft Documentation

- Focuses on security features within RDP, including encryption, authentication, and mitigation against common RDP-based attacks.
- **URL:** RDP Security in Windows Server

5. Remote Desktop Services - Technical Reference (Microsoft Docs)

- Offers a deep dive into the services and components involved in RDP, including its dependencies, service configuration, and detailed descriptions of various related components.
- **URL:** Remote Desktop Services - Technical Reference

6. Sysinternals - RDP-related Tools and Insights

- The Sysinternals website, part of Microsoft, provides tools like TCPView and Process Explorer that help diagnose and troubleshoot RDP-related issues. The site also has blog posts and articles on RDP components.
- **URL:** Sysinternals by Mark Russinovich

7. Rapid7 - Deep Dive into RDP: Security Analysis

- A comprehensive security analysis and deep dive into the RDP protocol by Rapid7, a well-known cybersecurity company. The article discusses vulnerabilities, security configurations, and how attackers exploit RDP.
- **URL:** Rapid7 - Deep Dive into RDP

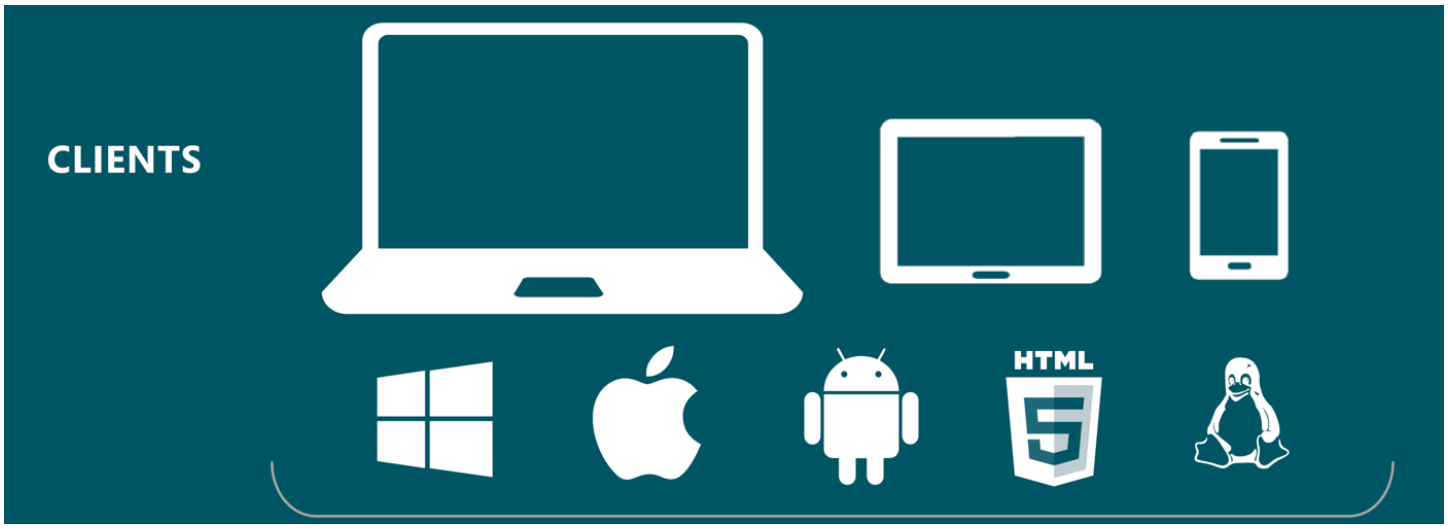
Additional Resources:

- **PCAP Analysis of RDP Traffic (Wireshark Wiki)**
  - An excellent resource on analyzing RDP traffic using Wireshark, ideal for network admins looking to understand how RDP traffic flows and identifying potential issues.
  - **URL:** Wireshark Wiki - RDP
- **Reddit Sysadmin Community - RDP Security Discussions**
  - While not a single document, the Sysadmin subreddit often has deep technical discussions and user experiences related to RDP configurations, issues, and solutions.
  - **URL:** Reddit Sysadmin Community

These resources will provide a thorough understanding of the RDP protocol, its security features, technical specifications, and best practices for configuration and troubleshooting.

Remote Desktop Connection (RDC) is an essential tool for Windows IT professionals

allowing remote access and management of systems. Understanding its inner workings and troubleshooting common issues is crucial for maintaining seamless connectivity and productivity.



How RDC Works

Remote Desktop operates through the Remote Desktop Protocol (RDP), a network communications protocol developed by Microsoft that enables remote access to another

computer over a network. When you initiate an RDC session, the client (local machine) establishes a connection with the remote computer (server) using RDP, allowing the user to interact with the remote system as if they were physically present. The server transmits the display output to the client, while input from the keyboard and mouse is sent back to the server (MS Learn) (MS Learn).

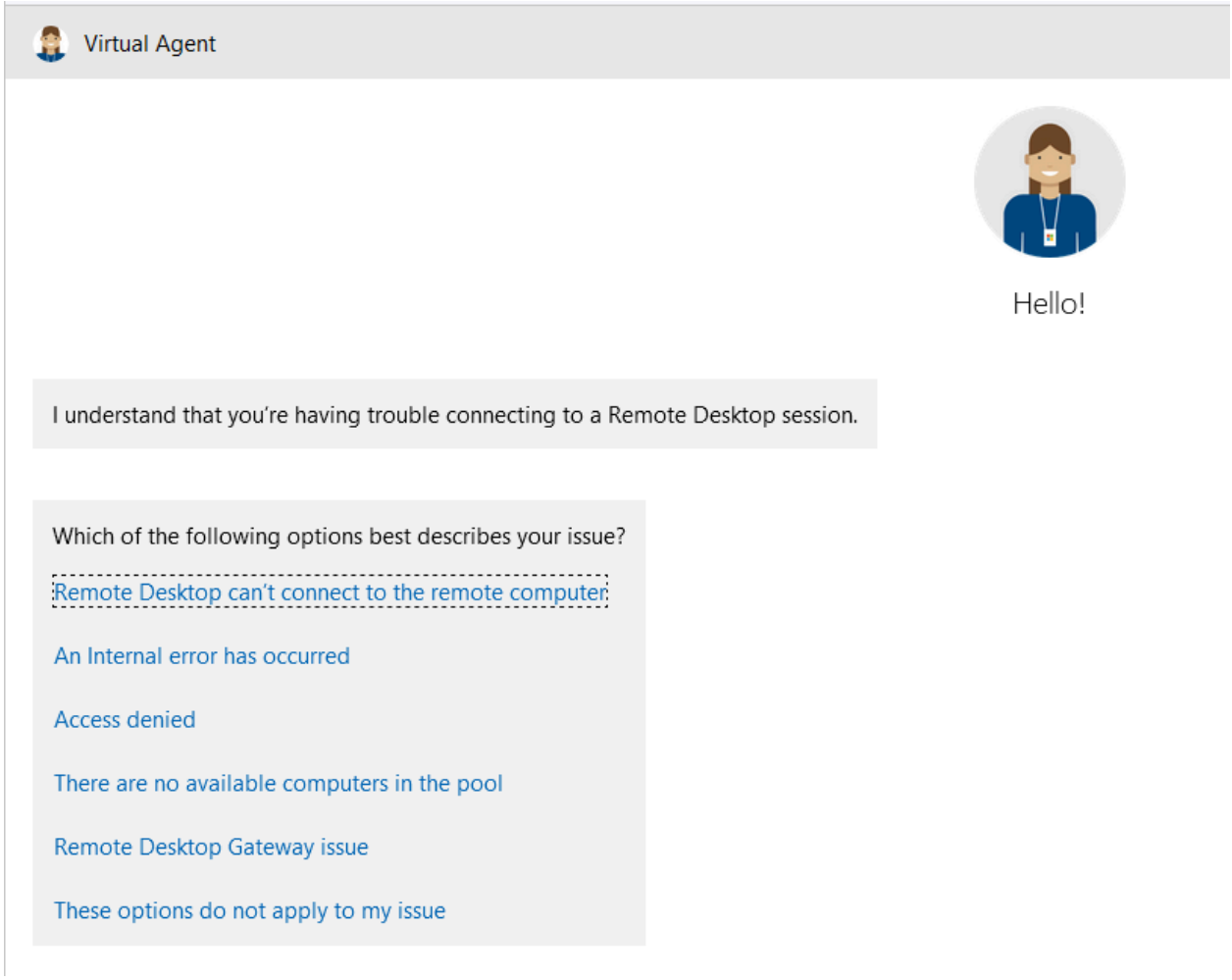
RDC also involves listener components on the server, which handle incoming connections and ensure that they are authorized based on user credentials and network policies (MS Learn).

Key Processes and Functions

- **Authentication**: The Network Level Authentication (NLA) ensures that the user is authenticated before establishing a session, adding a security layer to prevent unauthorized access (Techanta).
- **Session Redirection**: RDC can redirect devices such as printers and audio from the client to the server, enabling local resources to be used in the remote session (MS Learn).
- **Session Management**: IT admins can manage multiple simultaneous RDC sessions on servers using tools like Remote Desktop Services Configuration (MS Learn).

- https://vsa.services.microsoft.com/en-us/v1.0/?partnerId=7d74cf73-5217-4008-833f-87a1a278f2cb&flowId=DMC&initialQuery=rdssc

Try our Virtual Agent - It can help you quickly identify and fix common RD Sessions connectivity issues



10 Common Problems faced by admins using RDC to remote Access a client or server

10 common problems experienced by admins using Remote Desktop Connection (RDC) to access remote computers, along with detailed step-by-step procedures to remedy each problem. Each solution includes command-line utilities, PowerShell commands, and GUI tools where applicable.

1. RDC Connection Fails: Network Issues

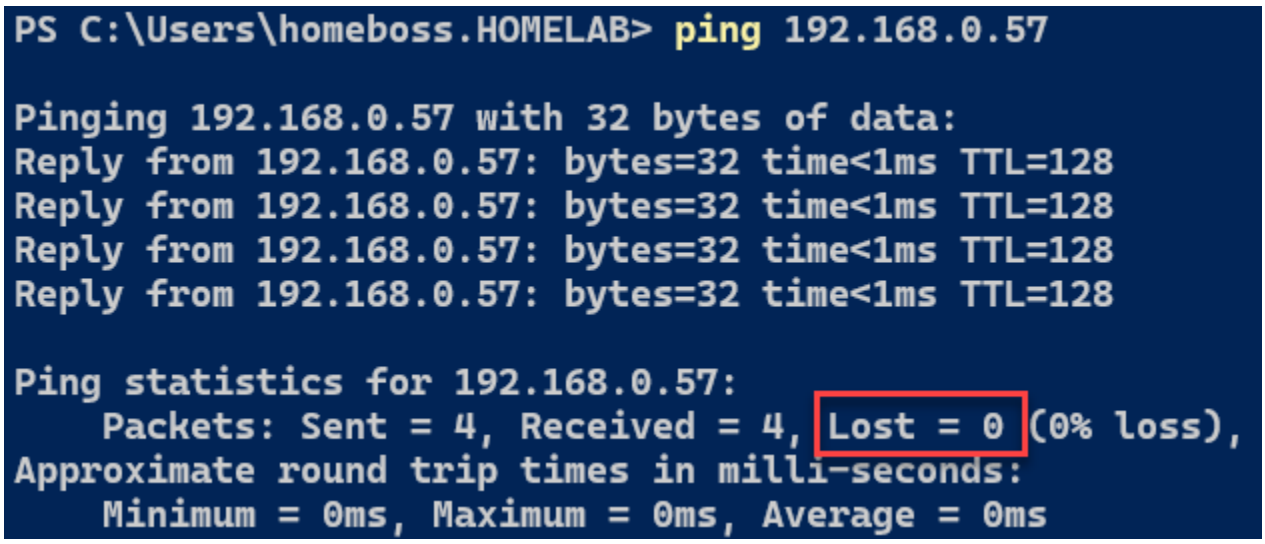
Problem: The connection to the remote desktop fails due to network issues such as incorrect IP, DNS resolution problems, or network connectivity issues.

Solution:

Command-Line Utilities:

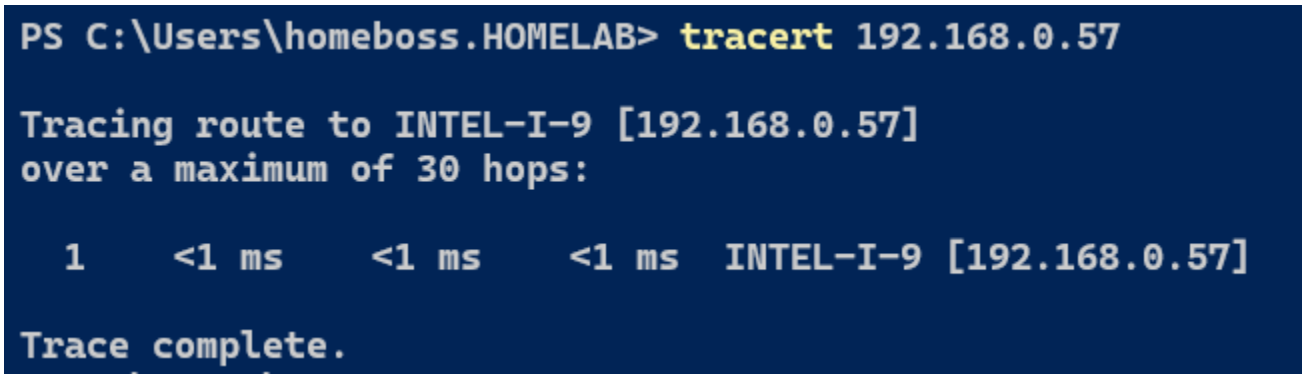
- Use ping to check connectivity:

```
ping <remote_ip_or_hostname>
```



- Use tracert to trace the route to the remote machine:

```
tracert <remote_ip_or_hostname>
```

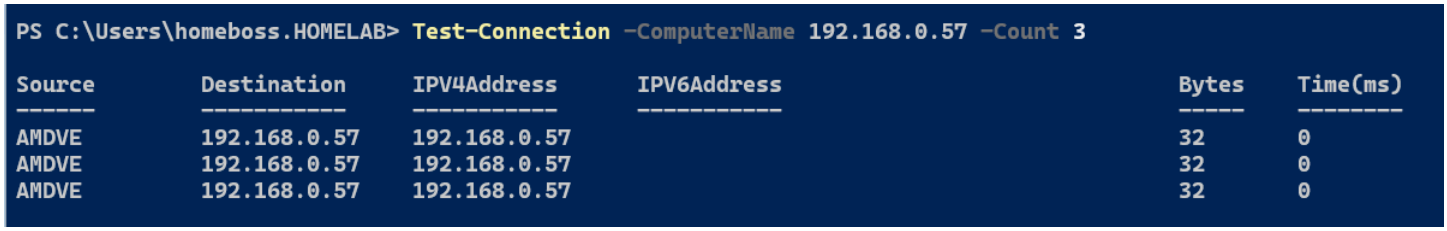


PowerShell:

Check network connectivity using Test-Connection:

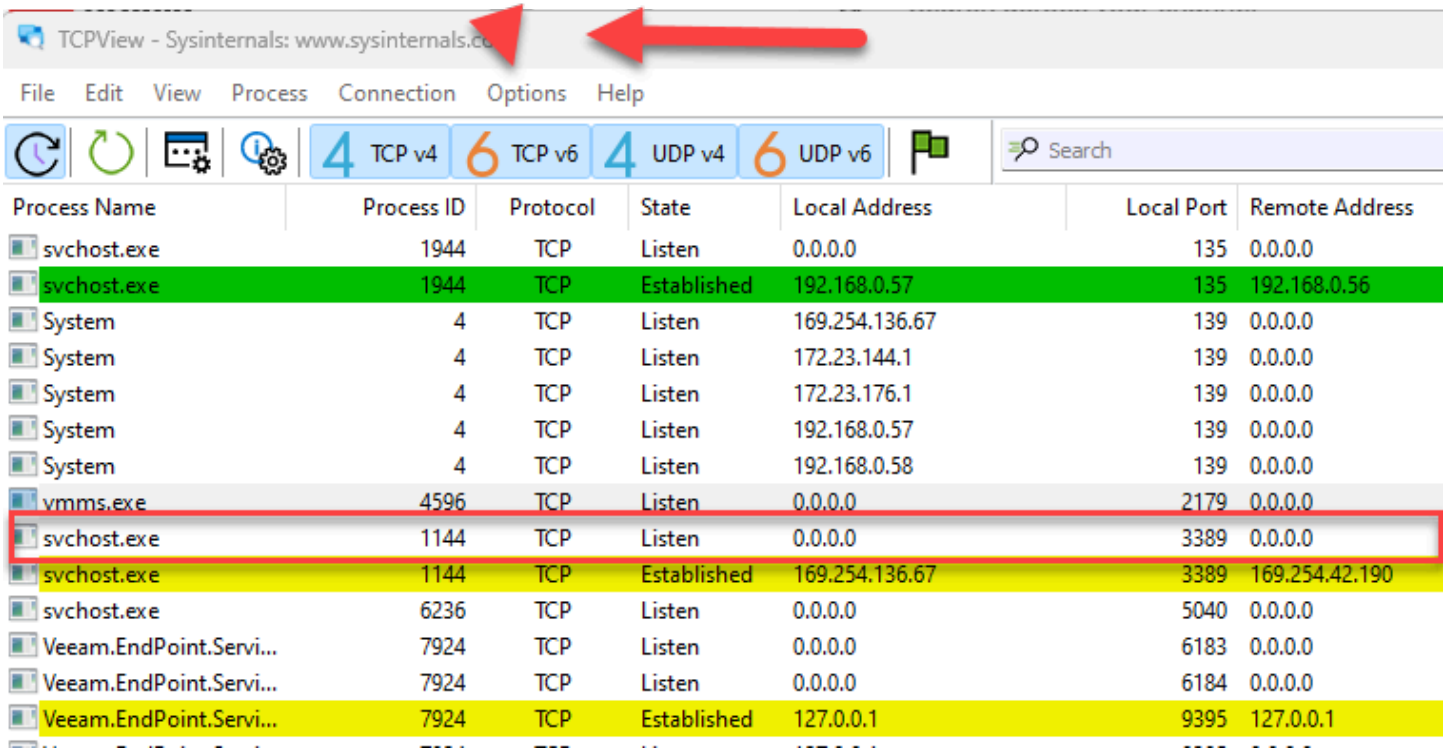
powershell

```
Test-Connection -ComputerName <remote_ip_or_hostname> -Count 3
```



GUI Tools:

Use Sysinternals' TCPView to check for open connections and troubleshoot network ports.



2. Incorrect Remote Desktop Configuration

Problem: Remote Desktop is not enabled on the remote computer or the computer is not configured to accept connections.

Solution:

Command-Line Utilities:

Enable Remote Desktop using reg command:

```
reg add "HKLM\SYSTEM\CurrentControlSet\Control\Terminal Server" /v fDenyTSConnections /t REG_DWORD /d 0 /f
```

PowerShell:

Enable Remote Desktop:

powershell

```
Set-ItemProperty -Path
"HKLM:\System\CurrentControlSet\Control\Terminal Server" -Name
"fDenyTSConnections" -Value 0

Enable-NetFirewallRule -DisplayGroup "Remote Desktop"
```

Script Tool:

Description to remotely enable RDC on client or computer:

How often do you, as an admin, need to use Remote Desktop Connection (RDC) to access a computer or server, only to find that RDC isn't enabled? In my experience, this happens all too frequently. To help with this, I've provided a tested PowerShell script below that guides you through enabling RDC remotely, modifying the firewall settings, and ensuring the domain admin is added to the appropriate group for RDC access.

The best part? You don't need to modify the script! It will prompt you for the necessary information and display a success or failure message for each step. Give it a try!

Description: Remotely Enabling RDC on a Client or Server

Script Overview:
This PowerShell script allows you to remotely enable RDC on a client or server, configure the necessary firewall settings, and add the domain admin to the Remote Desktop Users group. The script is interactive—it prompts you for all required details and provides feedback at each stage of the process.

Instructions:

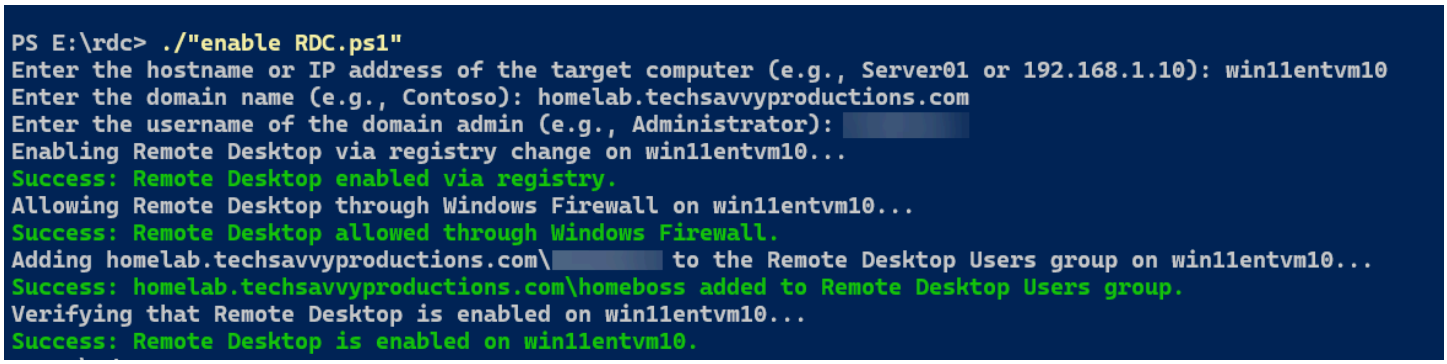
1. Copy the script below and paste it into a text editor.
2. Save it with a .ps1 extension (e.g., Enable-RDC.ps1).
3. Run it in a domain environment.

```
# Prompt for input values with descriptive messages

$TargetComputer = Read-Host "Enter the hostname or IP address
of the target computer (e.g., Server01 or 192.168.1.10)"

$Domain = Read-Host "Enter the domain name (e.g., Contoso)"

$AdminUser = Read-Host "Enter the username of the domain admin
(e.g., Administrator)"
```

```
# Step 1: Enable Remote Desktop by modifying the registry value
Write-Host "Enabling Remote Desktop via registry change on $TargetComputer..."
try {
    Invoke-Command -ComputerName $TargetComputer -ScriptBlock {
        Set-ItemProperty -Path "HKLM:\System\CurrentControlSet\Control\Terminal Server" -Name "fDenyTSConnections" -Value 0
    }
    Write-Host "Success: Remote Desktop enabled via registry." -ForegroundColor Green
} catch {
    Write-Host "Failure: Unable to enable Remote Desktop via registry." -ForegroundColor Red
}

# Step 2: Allow Remote Desktop through the Windows Firewall
Write-Host "Allowing Remote Desktop through Windows Firewall on $TargetComputer..."
try {
    Invoke-Command -ComputerName $TargetComputer -ScriptBlock {
        Enable-NetFirewallRule -DisplayGroup "Remote Desktop"
    }
    Write-Host "Success: Remote Desktop allowed through Windows Firewall." -ForegroundColor Green
} catch {
    Write-Host "Failure: Unable to allow Remote Desktop through Windows Firewall." -ForegroundColor Red
```

```
}


# Step 3: Add the default domain admin to the Remote Desktop
Users group

Write-Host "Adding $Domain\$AdminUser to the Remote Desktop
Users group on $TargetComputer..."

try {

    Invoke-Command -ComputerName $TargetComputer -ScriptBlock {

        Add-LocalGroupMember -Group "Remote Desktop Users"
-Member "$using:Domain\$using:AdminUser"

    }

    Write-Host "Success: $Domain\$AdminUser added to Remote
Desktop Users group." -ForegroundColor Green

} catch {

    Write-Host "Failure: Unable to add $Domain\$AdminUser to
the Remote Desktop Users group." -ForegroundColor Red

}


# Step 4: Verify Remote Desktop is enabled

Write-Host "Verifying that Remote Desktop is enabled on
$TargetComputer..."

try {

    $RDCStatus = Invoke-Command -ComputerName $TargetComputer
-ScriptBlock {

        Get-ItemProperty -Path
"HKLM:\System\CurrentControlSet\Control\Terminal Server" -Name
"fDenyTSConnections"

    }
```

```
    if ($RDCStatus.fDenyTSConnections -eq 0) {
        Write-Host "Success: Remote Desktop is enabled on $TargetComputer." -ForegroundColor Green
    } else {
        Write-Host "Failure: Remote Desktop is not enabled on $TargetComputer." -ForegroundColor Red
    }
} catch {
    Write-Host "Failure: Unable to verify the Remote Desktop status on $TargetComputer." -ForegroundColor Red
}
```



3. Access Denied Errors

Problem: The admin receives an "Access Denied" error when trying to connect or execute tasks remotely.

Solution:

Command-Line Utilities:

Ensure the user is part of the "Remote Desktop Users" group:

```
net localgroup "Remote Desktop Users" <username> /add
```

PowerShell:

Add the user to the "Remote Desktop Users" group:

```
Add-LocalGroupMember -Group "Remote Desktop Users" -Member
"<username>"
```

GUI Tools:

Use Sysinternals' AccessChk to check the user's group membership and permissions.

4. RDC Port Blocked by Firewall

Problem: The firewall on the remote computer is blocking the Remote Desktop port (default: TCP 3389).

Solution:

Command-Line Utilities:

Open port 3389 in Windows Firewall:

```
netsh advfirewall firewall add rule name="Remote Desktop"
protocol=TCP dir=in localport=3389 action=allow
```

PowerShell:

Open the Remote Desktop port using PowerShell:

```
New-NetFirewallRule -DisplayName "Remote Desktop" -Direction
Inbound -Protocol TCP -LocalPort 3389 -Action Allow
```

GUI Tools:

Use Windows Firewall with Advanced Security GUI to manually allow the Remote Desktop connection.

5. Slow or Lagging Remote Desktop Connection

Problem: Remote Desktop performance is slow or lagging due to network bandwidth or display settings.

Solution:

Command-Line Utilities:

Test network performance using iperf: https://iperf.fr/iperf-download.php

```
iperf -c <remote_ip>
```

PowerShell:

Check network adapter settings and speed:

```
Get-NetAdapter | Select-Object Name, Status, LinkSpeed
```

GUI Tools:

Use Sysinternals' ProcMon to monitor the performance of the Remote Desktop client and identify any potential issues.

To use ProcMon (Process Monitor) to monitor the performance of the Remote Desktop Connection (RDC) client and identify potential issues,

you'll need to follow specific steps and set appropriate filters. ProcMon is a powerful tool from Microsoft's Sysinternals suite that can capture real-time file system, registry, and process/thread activity, which can help pinpoint what might be causing performance issues with RDC.

Steps to Use ProcMon for Monitoring RDC Client

1. **Download and Launch ProcMon:**
   - Download **Process Monitor (ProcMon)** from the Microsoft Sysinternals website.
   - Run ProcMon.exe as an administrator to capture the system activities.

2. **Clear Existing Filters:**
    - When ProcMon starts, it may have default filters applied. Click on Filter in the menu and select Reset Filter to clear any existing filters and ensure you capture all events initially.
3. **Set Filters for RDC Monitoring:**
    - To focus on the RDC client (usually mstsc.exe), set a filter:
        - Go to Filter > Filter... or press Ctrl + L.
        - Add a new filter condition:
          Process Name | **is** | **mstsc.exe** | **Include**
        - Click Add and then OK.
    - This filter will limit ProcMon to only show events related to the RDC client process, reducing noise and making it easier to identify relevant activities.
4. **Capture File and Registry Operations:**
    - Start a Remote Desktop session using the RDC client.
    - ProcMon will begin capturing events related to file system access, registry queries, and modifications made by mstsc.exe.
    - Look for events that may indicate slowdowns or errors, such as:
        - **NAME NOT FOUND**: The RDC client is trying to access a file or registry entry that does not exist.
        - **ACCESS DENIED**: The RDC client lacks the necessary permissions to perform an operation.
        - **BUFFER OVERFLOW**: There is an attempt to read or write more data than the buffer size can handle, which could be a sign of performance problems or configuration issues.
5. **Filter for Network Activity:**
    - To monitor network-related performance, include filters for network-related operations. Unfortunately, ProcMon does not natively capture network traffic, but you can still check for operations involving WinSock or TCP.
    - Add filters like:
      Operation | **begins with** | **TCP** | **Include**
6. **Analyze CPU or Disk-Related Issues:**
    - Look for events that suggest high CPU usage or disk I/O issues, such as:
        - **IRP_MJ_CREATE** or **IRP_MJ_WRITE** operations taking a long time to complete.
        - High frequency of file access or registry operations.
7. **Save and Analyze the Log:**
    - Save the captured events by going to File > Save.
    - Use the saved log to analyze recurring patterns or specific errors contributing to the performance degradation.

Additional Tips:

- **Filter by Time:** If you know when the performance issue occurs, use the Timestamp filter to narrow down the events to a specific period.
- **Use Highlighting:** Use the Highlight feature (Ctrl + H) to make certain events stand out, such as errors or specific registry operations that might indicate problems.
- **Correlate with Other Tools:** Use ProcMon in conjunction with other tools like **Performance Monitor** (PerfMon) or **Resource Monitor** to get a comprehensive view of system performance.

Summary:

By setting up appropriate filters and using ProcMon to monitor file, registry, and network operations related to the RDC client, you can identify performance bottlenecks, missing dependencies, or permission issues. This approach will help you pinpoint the root cause of RDC performance issues and take the necessary steps to resolve them.

6. Remote Desktop Service Crashes or Stops

Problem: The Remote Desktop service (TermService) on the remote machine crashes or stops.

Solution:

ommand-Line Utilities:

Restart the Remote Desktop service:

```
net stop TermService && net start TermService
```

PowerShell:

Restart the service using PowerShell:

```
Restart-Service -Name TermService
```

GUI Tools:

Use Sysinternals' Process Explorer to monitor and restart services.

7. Remote Desktop Licensing or Certificate Issues

Problem: Remote Desktop licensing or certificates are causing connection problems.

Solution:

Command-Line Utilities:

Reset Remote Desktop licensing or certificates:

```
del /f /s /q %windir%\System32\lserver\*.*
```

PowerShell:

Remove and reconfigure the RDP certificate:

```
Remove-Item "HKLM:\SOFTWARE\Microsoft\MSLicensing" -Recurse
```

GUI Tools:

Use CertMgr.msc to manage certificates manually.

8. Clipboard Redirection Not Working

Problem: The clipboard functionality (copy-paste) does not work over the RDC session.

Solution:

Command-Line Utilities:

Restart rdpclip process:

```
taskkill /f /im rdpclip.exe && start rdpclip.exe
```

PowerShell:

Restart the process using PowerShell:

```
Stop-Process -Name rdpclip -Force; Start-Process -Name rdpclip
```

GUI Tools:

Use Task Manager to manually restart the rdpclip.exe process.

9. Black Screen After Login

Problem: The user experiences a black screen after logging in via Remote Desktop.

Solution:

Command-Line Utilities:

Restart Windows Explorer to refresh the user interface:

```
taskkill /f /im explorer.exe && start explorer.exe
```

PowerShell:

Restart the process using PowerShell:

```
 Stop-Process -Name explorer -Force; Start-Process explorer
```

GUI Tools:

Use Sysinternals' Process Explorer to locate and restart the Explorer process.

10. Remote Desktop Not Working After System Update

Problem: After a system update, Remote Desktop stops working due to incompatible settings or services.

Solution:

Command-Line Utilities:

Rollback the recent updates if needed:

```
wusa /uninstall /kb:<KB_Number>
```

PowerShell:

Check for and remove specific updates:

```
Get-HotFix | Where-Object { $_.Description -like "*Update*" }
        Uninstall-WindowsFeature -Name <FeatureName>
```

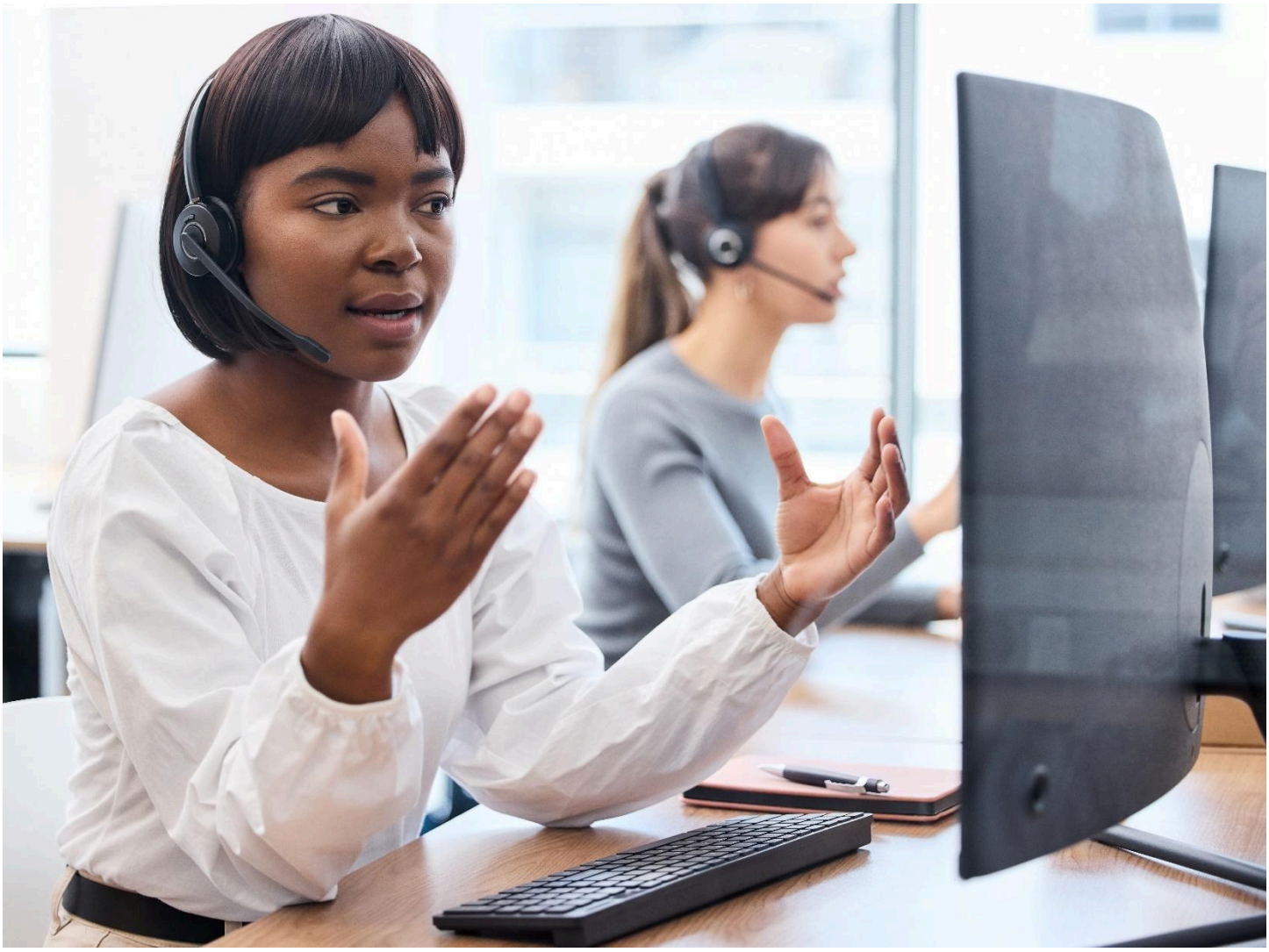
GUI Tools:

Use Windows Update History to view and uninstall problematic updates.

These steps should help you troubleshoot and resolve common Remote Desktop issues effectively. Let me know if you need further details on any specific problem!

Common Issues and Troubleshooting

1. **Connection Problems**:
    - **Network Configuration**: Connection failures often stem from network issues or firewall restrictions. Checking both local and remote firewall settings to ensure port 3389 (or the configured RDP port) is open can resolve connectivity issues (Microsoft Answers).
    - **Service Configuration**: Ensure that Remote Desktop Services are properly configured and running on the remote machine. Misconfiguration of the service can prevent successful connections (MS Learn) (Techanta).
2. **Slow Performance**:
    - **Bandwidth and Network Settings**: Adjusting settings such as disabling visual effects and reducing the display resolution can help improve performance, especially over limited bandwidth (Techanta).
    - **Resource Management**: Closing unnecessary applications on the remote machine can free up resources and improve the responsiveness of the remote session (Techanta).
3. **Authentication and Login Issues**:
    - **Credential Verification**: Incorrect credentials or expired passwords can block access. Ensuring credentials are correct and resetting passwords as needed can resolve login problems (Microsoft Answers) (Techanta).
    - **Active Directory Issues**: Misconfigurations in Active Directory or group policies can prevent login. IT professionals should verify that user permissions are correctly set (Techanta).



4. **Port Conflicts**:
    - **Port Assignment Conflicts**: If another application is using the same port as RDP, it can block the connection. Using tools like netstat and tasklist to identify and resolve port conflicts can restore functionality (MS Learn).
5. **Screen Resolution and Display Issues**:
    - **Display Settings**: Mismatched screen resolutions between the client and server can cause scaling problems. Adjusting the RDC client's display options to match the remote machine's resolution can resolve these issues (Techanta).
6. **Audio and Printing Redirection Failures**:
    - **Driver Updates**: Ensuring that both the local and remote machines have up-to-date drivers for audio and printers can prevent redirection issues (Techanta).

Resources for In-Depth Troubleshooting

- Microsoft Learn - Remote Desktop Services Overview (MS Learn)
- Microsoft Learn - Troubleshooting RDP Client Connection Problems (MS Learn)
- Techanta - How to Troubleshoot Common Remote Desktop Issues (Techanta)

These resources provide detailed solutions and further reading for IT professionals seeking to master Remote Desktop troubleshooting and ensure smooth, secure remote conections.

Did you know? One of the languages we create subtitles for is ***Catalan***.

Catalan is spoken as a primary language in many regions of Europe, including Spain, Andorra, and France:

- 

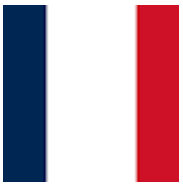
  Spain

  Catalan is spoken in the autonomous communities of Aragon, Balearic Islands, Catalonia, Murcia, and Valencia, and roughly 22% of the Spanish population speak it as a first language.

- 

  Andorra

  Catalan is the official language of Andorra.

- 

  France

  Catalan is spoken in the Roussillon region of France.

Catalan is also spoken in the city of Alghero in Sardinia, Italy.

IT info about Spain

- The average salary for Technical Support is $23000 per year in Catalonia, Spain -Glassdoor.com
- The estimated total pay for a Systems Administrator is €32,000 per year in the Spain area, with an average salary of €30,000 per year.

- There are 143 Spanish data centers, of which the majority are located in and around Madrid and Barcelona. Large carrier neutral Spanish data centers feature rich ecosystems and state-of-the-art equipment, ensuring maximum uptime and connectivity to over 892 service providers

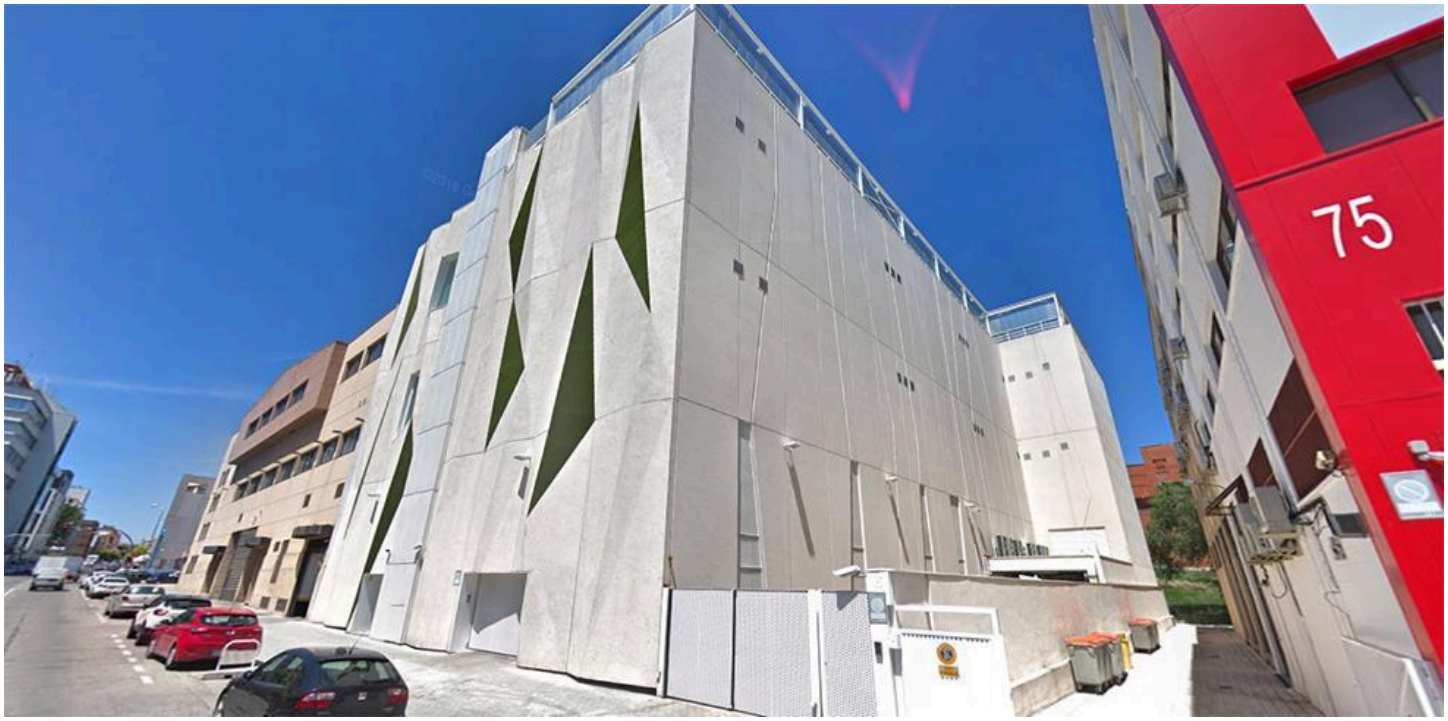


Figure 3Madrid Spain Data Center NetActuate

General Remote Desktop connection troubleshooting

In this article

Try our Virtual Agent - It can help you quickly identify and fix common RD Sessions connectivity issues

Use these steps when a Remote Desktop client can't connect to a remote desktop but doesn't provide messages or other symptoms that would help identify the cause.

Check the status of the RDP protocol

Check the status of the RDP protocol on a local computer

To check and change the status of the RDP protocol on a local computer, see How to enable Remote Desktop.

Note

If the remote desktop options are not available, see **Check whether a Group Policy Object is blocking RDP**.

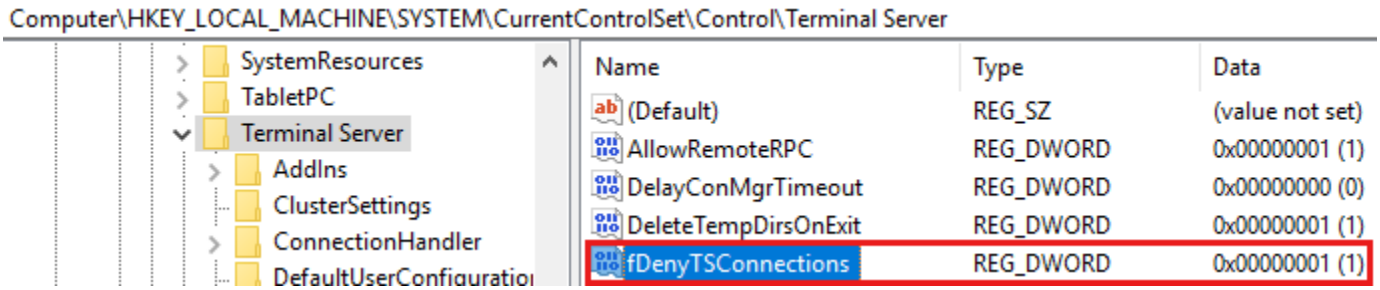
Check the status of the RDP protocol on a remote computer

Important

This section, method, or task contains steps that tell you how to modify the registry. However, serious problems might occur if you modify the registry incorrectly. Therefore, make sure that you follow these steps carefully. For protection, back up the registry before you modify it so that you can restore it if a problem occurs. For more information about how to back up and restore the registry, see **How to back up and restore the registry in Windows**.

To check and change the status of the RDP protocol on a remote computer, use a network registry connection:

1. First, go to the **Start** menu, then select **Run**. In the text box that appears, enter *regedt32*.
2. In the Registry Editor, select **File**, then select **Connect Network Registry**.
3. In the **Select Computer** dialog box, enter the name of the remote computer, select **Check Names**, and then select **OK**.
4. Navigate to HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Terminal Server and to HKEY_LOCAL_MACHINE\SOFTWARE\Policies\Microsoft\Windows NT\Terminal Services.



- If the value of the **fDenyTSConnections** key is **0**, then RDP is enabled.
- If the value of the **fDenyTSConnections** key is **1**, then RDP is disabled.

2. To enable RDP, change the value of **fDenyTSConnections** from **1** to **0**.

Check whether a Group Policy Object (GPO) is blocking RDP on a local computer

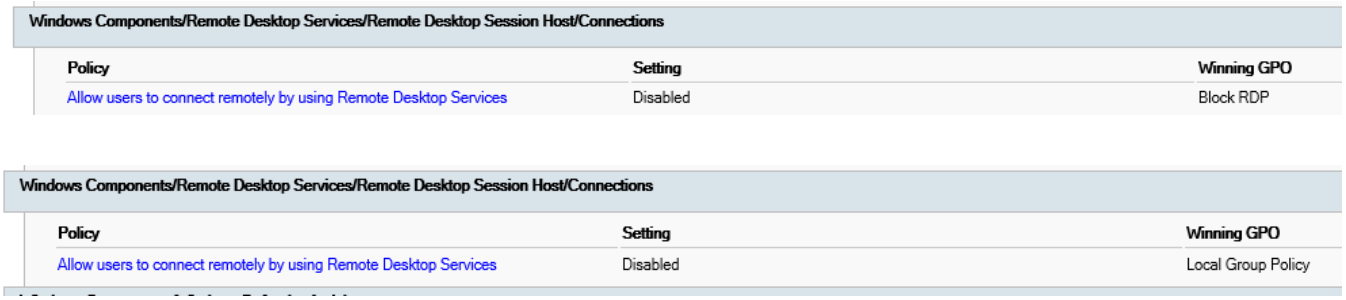
If you can't turn on RDP in the user interface or the value of **fDenyTSConnections** reverts to **1** after you've changed it, a GPO may be overriding the computer-level settings.

To check the group policy configuration on a local computer, open a command prompt window as an administrator, and enter the following command:

```
gpresult /h c:\gpo\gpresult.html
```

After this command finishes, open *gpresult.html*. In **Computer Configuration\Administrative Templates\Windows Components\Remote Desktop Services\Remote Desktop Session Host\Connections**, find the **Allow users to connect remotely by using Remote Desktop Services** policy.

- If the setting for this policy is **Enabled**, Group Policy is not blocking RDP connections.
- If the setting for this policy is **Disabled**, check **Winning GPO**. This is the GPO that is blocking RDP connections.



Windows Components/Remote Desktop Services/Remote Desktop Session Host/Connections		
Policy	**Setting**	**Winning GPO**
Allow users to connect remotely by using Remote Desktop Services	Disabled	Block RDP

Windows Components/Remote Desktop Services/Remote Desktop Session Host/Connections		
Policy	**Setting**	**Winning GPO**
Allow users to connect remotely by using Remote Desktop Services	Disabled	Local Group Policy

Check whether a GPO is blocking RDP on a remote computer

To check the Group Policy configuration on a remote computer, the command is almost the same as for a local computer:

```
gpresult /S <computer name> /h c:\gpo\gpresult-<computer name>.html
```

The file that this command produces (*gpresult-<computer name>.html*) uses the same information format as the local computer version (*gpresult.html*) uses.

Modifying a blocking GPO

You can modify these settings in the Group Policy Object Editor (GPE) and Group Policy Management Console (GPM). For more information about how to use Group Policy, see Advanced Group Policy Management.

To modify the blocking policy, use one of the following methods:

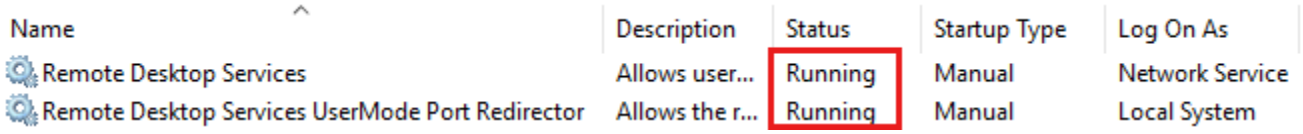
- In GPE, access the appropriate level of GPO (such as local or domain), and navigate to **Computer Configuration** > **Administrative Templates** > **Windows Components** > **Remote Desktop Services** > **Remote Desktop Session Host** > **Connections** > **Allow users to connect remotely by using Remote Desktop Services**.
    1. Set the policy to either **Enabled** or **Not configured**.
    2. On the affected computers, open a command prompt window as an administrator, and run the gpupdate /force command.
- In GPM, navigate to the organizational unit (OU) in which the blocking policy is applied to the affected computers and delete the policy from the OU.

Check the status of the RDP services

On both the local (client) computer and the remote (target) computer, the following services should be running:

- Remote Desktop Services (TermService)
- Remote Desktop Services UserMode Port Redirector (UmRdpService)

You can use the Services MMC snap-in to manage the services locally or remotely. You can also use PowerShell to manage the services locally or remotely (if the remote computer is configured to accept remote PowerShell cmdlets).



On either computer, if one or both services are not running, start them.

Note

If you start the Remote Desktop Services service, select **Yes** to automatically restart the Remote Desktop Services UserMode Port Redirector service.

Check that the RDP listener is functioning

Important

This section, method, or task contains steps that tell you how to modify the registry. However, serious problems might occur if you modify the registry incorrectly. Therefore, make sure that you follow these steps carefully. For protection, back up the registry before you modify it so that you can restore it if a problem occurs. For more information about how to back up and restore the registry, see **How to back up and restore the registry in Windows**.

Check the status of the RDP listener

For this procedure, use a PowerShell instance that has administrative permissions. For a local computer, you can also use a command prompt that has administrative permissions. However, this procedure uses PowerShell because the same cmdlets work both locally and remotely.

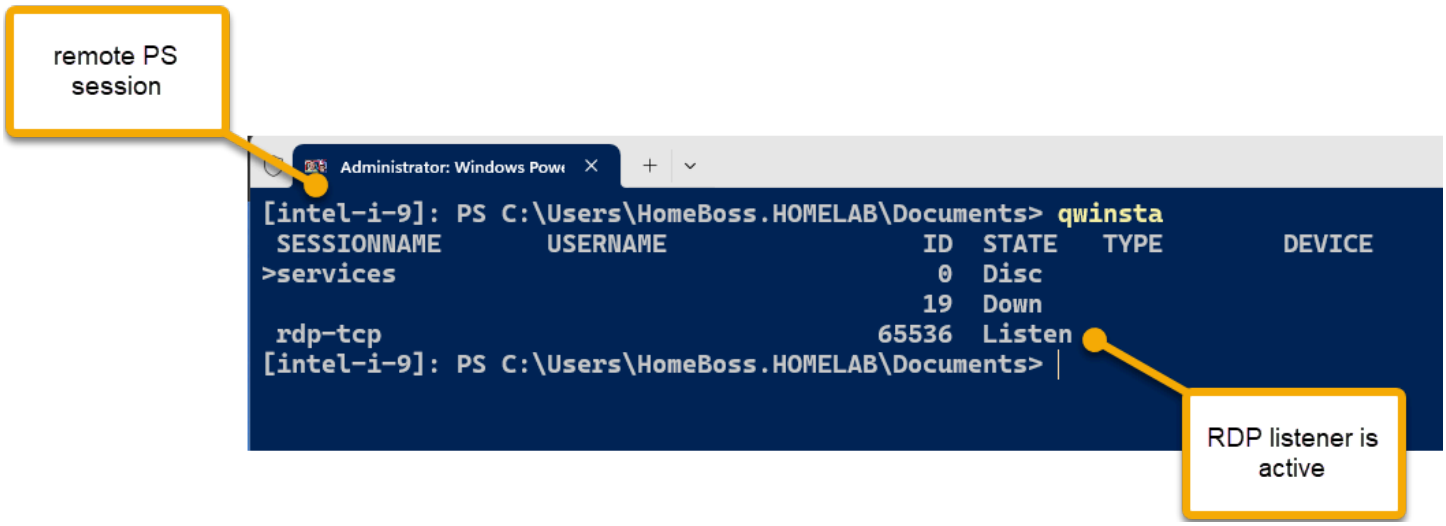
1. To connect to a remote computer, run the following cmdlet:

   Enter-PSSession -ComputerName <computer name>

2. Enter qwinsta.

```
[rdwagw01]: PS C:\> qwinsta
 SESSIONNAME       USERNAME                 ID  STATE   TYPE        DEVICE
>services                                   0  Disc
 console                                    1  Conn
 31c5ce94259d4...                       65536  Listen
 rdp-tcp                                65537  Listen
```

3. If the list includes rdp-tcp with a status of Listen, the RDP listener is working.



4. Proceed to Check the RDP listener port. Otherwise, continue at step 4.

```
cmd /c 'netstat -ano | find "3389"'
```

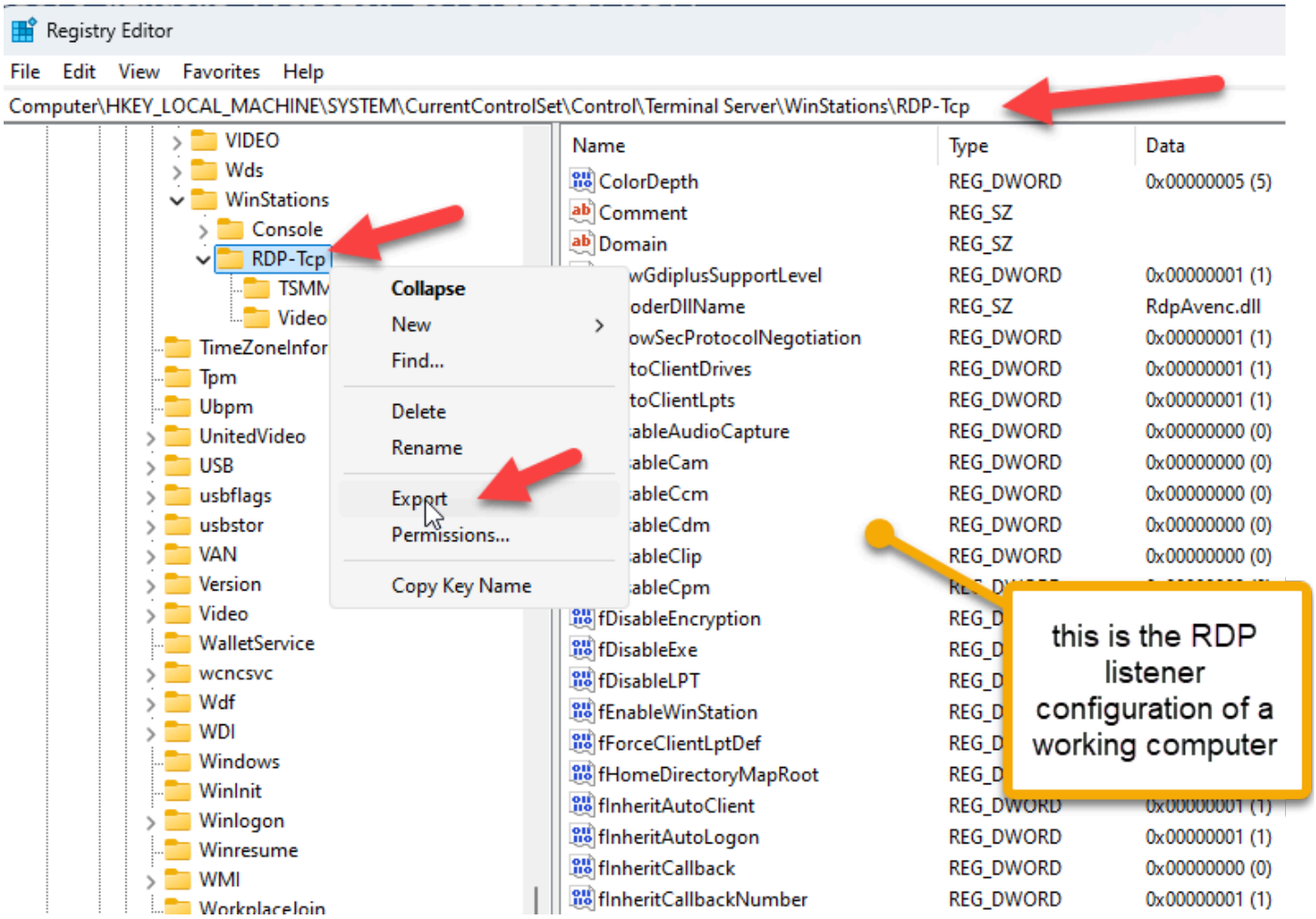
```
PS C:\Users\homeboss.HOMELAB> cmd /c 'netstat -ano | find "3389"'
  TCP    0.0.0.0:3389           0.0.0.0:0              LISTENING       2660
  TCP    192.168.0.56:21003     192.168.0.57:3389      ESTABLISHED     23644
  TCP    [::]:3389              [::]:0                 LISTENING       2660
  UDP    0.0.0.0:3389           *:*                                    2660
  UDP    0.0.0.0:55111          192.168.0.57:3389                      23644
  UDP    [::]:3389              *:*                                    2660
```

5. Export the RDP listener configuration from a working computer.

    a. Sign in to a computer that has the same operating system version as the affected computer has, and access that computer's registry (for example, by using Registry Editor).

    a. Navigate to the following registry entry:

    HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Terminal Server\WinStations\RDP-Tcp



a. Export the entry to a .reg file. For example, in Registry Editor, right-click the entry, select **Export**, and then enter a filename for the exported settings.

a. Copy the exported .reg file to the affected computer.

2. To import the RDP listener configuration, open a PowerShell window that has administrative permissions on the affected computer (or open the PowerShell window and connect to the affected computer remotely).

   a. To back up the existing registry entry, enter the following cmdlet:

   PowerShell
   ```
   cmd /c 'reg export
   "HKLM\SYSTEM\CurrentControlSet\Control\Terminal
   Server\WinStations\RDP-tcp"
   C:\Rdp-tcp-backup.reg'
   ```

   a. To remove the existing registry entry, enter the following cmdlets:

PowerShell

```
Remove-Item -path
'HKLM:\SYSTEM\CurrentControlSet\Control\Terminal
Server\WinStations\RDP-tcp' -Recurse -Force
```

a. To import the new registry entry and then restart the service, enter the following cmdlets:

PowerShell

```
cmd /c 'regedit /s c:\<filename>.reg'
Restart-Service TermService -Force
```

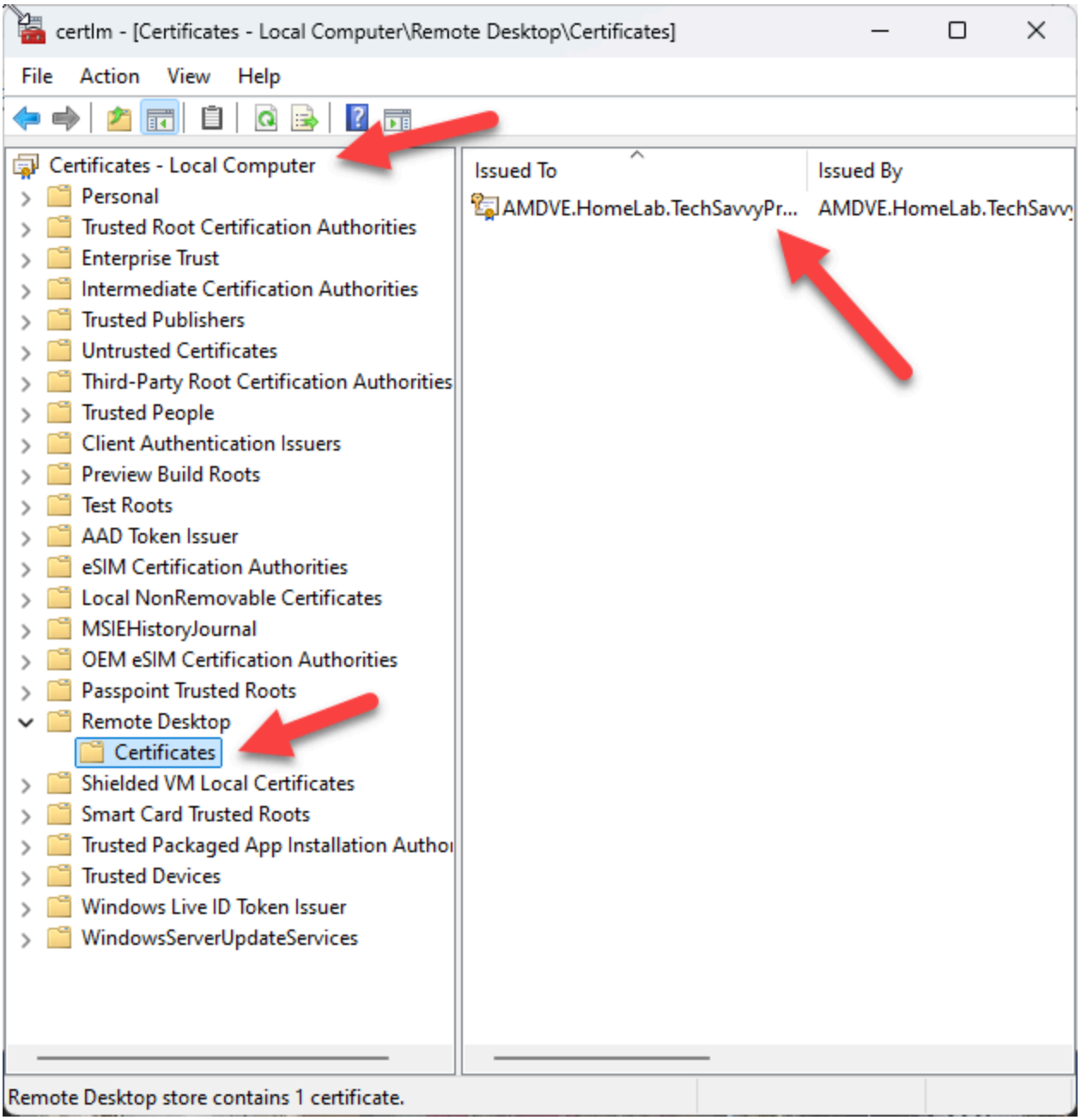
Replace <filename> with the name of the exported .reg file.

2. Test the configuration by trying the remote desktop connection again. If you still can't connect, restart the affected computer.
3. If you still can't connect, check the status of the RDP self-signed certificate.

Check the status of the RDP self-signed certificate

1. If you still can't connect, open the Certificates MMC snap-in. When you are prompted to select the certificate store to manage, select **Computer account**, and then select the affected computer.

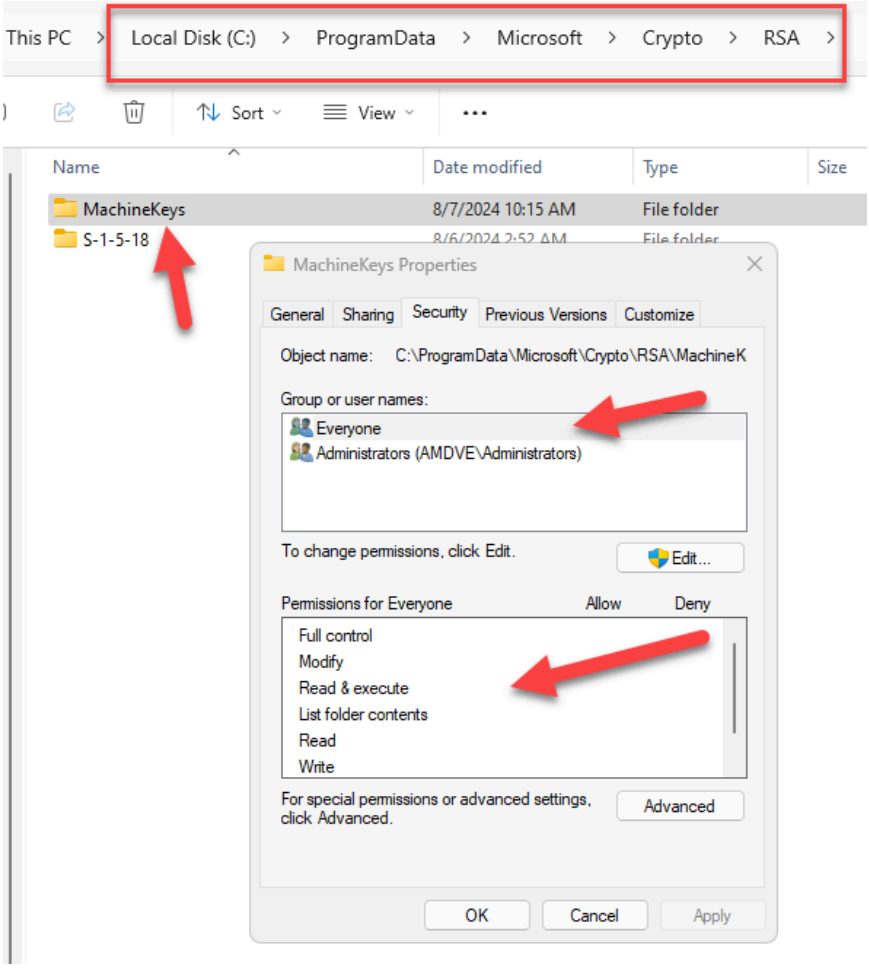
NOW ALL HAS BEEN HEARD;
HERE IS THE CONCLUSION OF THE MATTER:
FEAR GOD AND KEEP HIS COMMANDMENTS,
FOR THIS IS THE DUTY OF ALL MANKIND.
FOR GOD WILL BRING EVERY DEED INTO JUDGMENT,
INCLUDING EVERY HIDDEN THING,
WHETHER IT IS GOOD OR EVIL.



2. In the **Certificates** folder under **Remote Desktop**, delete the RDP self-signed certificate.
3. On the affected computer, restart the Remote Desktop Services service.
4. Refresh the Certificates snap-in.
5. If the RDP self-signed certificate has not been recreated, [check the permissions of the MachineKeys folder](#).

Check the permissions of the MachineKeys folder

1. On the affected computer, open Explorer, and then navigate to *C:\ProgramData\Microsoft\Crypto\RSA*.
2. Right-click **MachineKeys**, select **Properties** > **Security** > **Advanced**.
3. Make sure that the following permissions are configured:
    - **Builtin\Administrators**: **Full control**
    - **Everyone**: **Read**, **Write**



Check the RDP listener port

On both the local (client) computer and the remote (target) computer, the RDP listener should be listening on port 3389. No other applications should be using this port.

Important

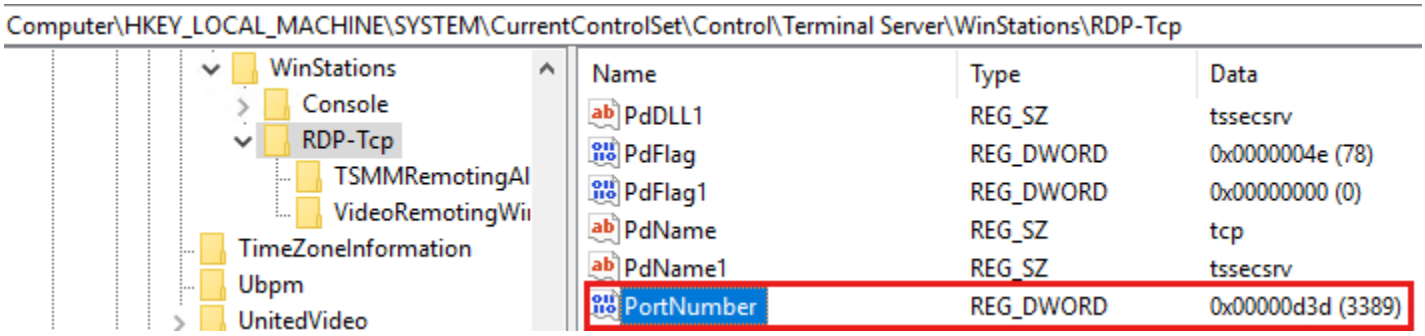
This section, method, or task contains steps that tell you how to modify the registry. However, serious problems might occur if you modify the registry incorrectly. Therefore, make sure that you follow these steps carefully. For protection, back up the registry before you modify it so that you can restore it if a problem occurs. For more information about how to back up and restore the registry, see **How to back up and restore the registry in Windows**.

To check or change the RDP port, use the Registry Editor:

1. Go to the Start menu, select **Run**, then enter *regedt32* into the text box that appears.

    - To connect to a remote computer, select **File**, and then select **Connect Network Registry**.
    - In the **Select Computer** dialog box, enter the name of the remote computer, select **Check Names**, and then select **OK**.

2. Open the registry and navigate to HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Terminal Server\WinStations\<listener>.



3. If **PortNumber** has a value other than **3389**, change it to **3389**.

    Important

    You can operate Remote Desktop services using another port. However, we don't recommend you do this. This article doesn't cover how to troubleshoot that type of configuration.

4. After you change the port number, restart the Remote Desktop Services service.

Check that another application isn't trying to use the same port

For this procedure, use a PowerShell instance that has administrative permissions. For a local computer, you can also use a command prompt that has administrative permissions. However, this procedure uses PowerShell because the same cmdlets work locally and remotely.

1. Open a PowerShell window. To connect to a remote computer, enter Enter-PSSession -ComputerName <computer name>.
2. Enter the following command:

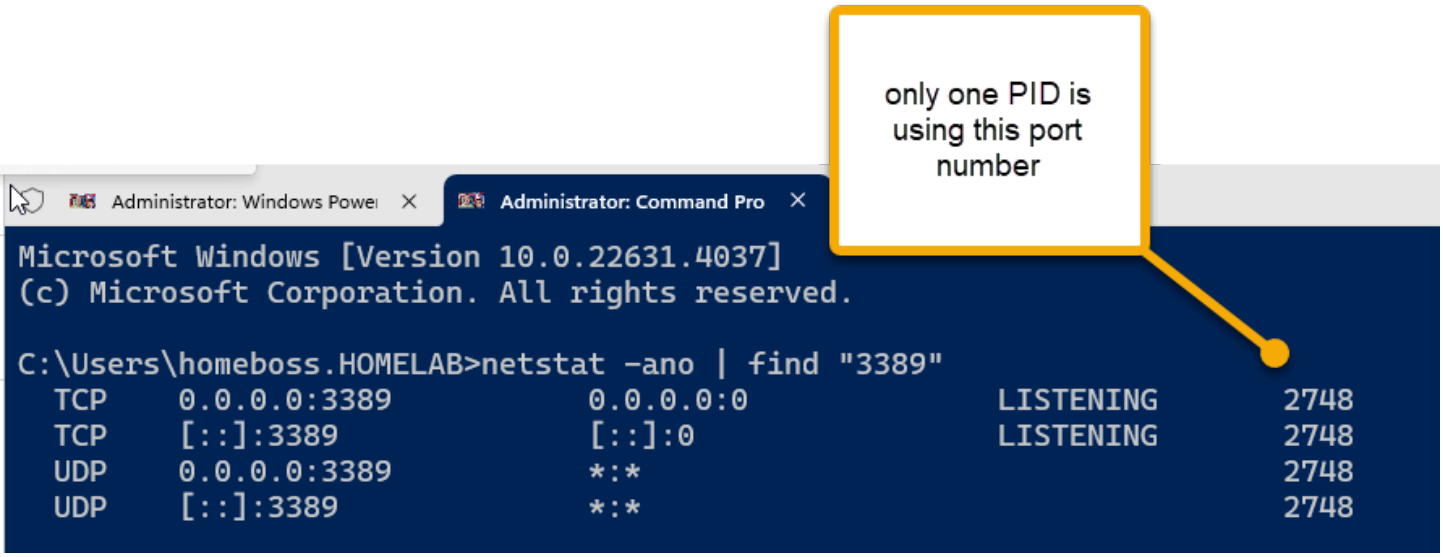
PowerShell
cmd /c 'netstat -ano | find "3389"'

```
[rdsh01]: PS C:\> cmd /c 'netstat -ano | find "3389"'
  TCP    0.0.0.0:3389           0.0.0.0:0              LISTENING       2104
  TCP    [::]:3389              [::]:0                 LISTENING       2104
```

3. Look for an entry for TCP port 3389 (or the assigned RDP port) with a status of **Listening**.

Note

The process identifier (PID) for the process or service using that port appears under the PID column.



4. To determine which application is using port 3389 (or the assigned RDP port), enter the following command:

PowerShell
cmd /c 'tasklist /svc | find "<pid listening on 3389>"'

```
[rdsh01]: PS C:\> cmd /c 'tasklist /svc | find "2104"'
[redacted] Server.exe                  2104 [redacted]. Server

[rdsh01]: PS C:\>
```

5. Look for an entry for the PID number that is associated with the port (from the netstat output). The services or processes that are associated with that PID appear on the right column.
6. If an application or service other than Remote Desktop Services (*TermServ.exe*) is using the port, you can resolve the conflict by using one of the following methods:

- Configure the other application or service to use a different port (recommended).
- Uninstall the other application or service.
- Configure RDP to use a different port, and then restart the Remote Desktop Services service (not recommended).

Check whether a firewall is blocking the RDP port

Use the psping tool to test whether you can reach the affected computer by using port 3389.

1. Go to a different computer that isn't affected and download psping.
2. Open a command prompt window as an administrator, change to the directory in which you installed psping, and then enter the following command:

   ```
   psping -accepteula <computer IP>:3389
   ```

```
PS C:\Users\homeboss.HOMELAB> psping -accepteula 192.168.0.57:3389

PsPing v2.12 - PsPing - ping, latency, bandwidth measurement utility
Copyright (C) 2012-2023 Mark Russinovich
Sysinternals - www.sysinternals.com

TCP connect to 192.168.0.57:3389:
5 iterations (warmup 1) ping test:
Connecting to 192.168.0.57:3389 (warmup): from 192.168.0.56:25838: 1.00ms
Connecting to 192.168.0.57:3389: from 192.168.0.56:25841: 0.67ms
Connecting to 192.168.0.57:3389: from 192.168.0.56:25842: 0.54ms
Connecting to 192.168.0.57:3389: from 192.168.0.56:25843: 1.08ms
Connecting to 192.168.0.57:3389: from 192.168.0.56:25847: 0.75ms

TCP connect statistics for 192.168.0.57:3389:
  Sent = 4, Received = 4, Lost = 0 (0% loss),
  Minimum = 0.54ms, Maximum = 1.08ms, Average = 0.76ms
```

3. Check the output of the psping command for results such as the following:

   - Connecting to \<computer IP\>: The remote computer is reachable.
   - **(0% loss): All attempts to connect succeeded**.
   - The remote computer refused the network connection: The remote computer is not reachable.
   - **(100% loss): All attempts to connect failed.**

2. Run psping on multiple computers to test their ability to connect to the affected computer.

3. Note whether the affected computer blocks connections from all other computers, some other computers, or only one other computer.
4. Recommended next steps:

    - Engage your network administrators to verify that the network allows RDP traffic to the affected computer.
    - Investigate the configurations of any firewalls between the source computers and the affected computer (including Windows Firewall on the affected computer) to determine whether a firewall is blocking the RDP port.

winsta

1. Syntax
2. Related links

Applies to: Windows Server 2022, Windows Server 2019, Windows Server 2016, Windows Server 2012 R2, Windows Server 2012

Displays information about sessions on a Remote Desktop Session Host server. The list includes information not only about active sessions but also about other sessions that the server runs.

Note

This command is the same as the **query session command**. To find out what's new in the latest version, see **What's New in Remote Desktop Services in Windows Server**.

Syntax

```
qwinsta [<sessionname> | <username> | <sessionID>]
[/server:<servername>] [/mode] [/flow] [/connect] [/counter]
```

Parameters

Expand table

Parameter	Description
<sessionname>	Specifies the name of the session that you want to query.
<username>	Specifies the name of the user whose sessions you want to query.

Parameter	Description
<sessionID>	Specifies the ID of the session that you want to query.
/server:<servername>	Identifies the rd Session Host server to query. The default is the current server.
/mode	Displays current line settings.
/flow	Displays current flow-control settings.
/connect	Displays current connect settings.
/counter	Displays current counters information, including the total number of sessions created, disconnected, and reconnected.
/?	Displays help at the command prompt.

Remarks

- A user can always query the session to which the user is currently logged on. To query other sessions, the user must have special access permission.
- If you don't specify a session using the <*username*>, <*sessionname*>, or *sessionID* parameters, this query will display information about all active sessions in the system.
- When **qwinsta** returns information, a greater than (>) symbol is displayed before the current session. For example:

```
C:\>qwinsta
  SESSIONNAME    USERNAME       ID STATE    TYPE    DEVICE
  console        Administrator1  0 active    wdcon
  >rdp-tcp#1     User1           1 active    wdtshare
  rdp-tcp                        2 listen    wdtshare
                                 4 idle
                                 5 idle
```

Where:

  - **SESSIONNAME** specifies the name assigned to the session.
  - **USERNAME** indicates the user name of the user connected to the session.
  - **STATE** provides information about the current state of the session.
  - **TYPE** indicates the session type.
  - **DEVICE**, which isn't present for the console or network-connected sessions, is the device name assigned to the session.

- Any sessions in which the initial state is configured as DISABLED won't show up in the **qwinsta** list until they're enabled.

Examples

To display information about all active sessions on server *Server2*, type:

```
qwinsta /server:Server2
```

To display information about active session *modeM02*, type:

```
qwinsta modeM02
```

Related links

- Command-Line Syntax Key
- query session command
- Remote Desktop Services (Terminal Services) Command Reference

Remote Desktop Services (Terminal Services) command-line tools reference

Applies to: Windows Server 2022, Windows Server 2019, Windows Server 2016, Windows Server 2012 R2, Windows Server 2012

Learn about the available Remote Desktop Services (Terminal Services) command-line tools, with descriptions and links for more detailed information.

To find out what's new in the latest version, see What's New in Remote Desktop Services in Windows Server.

Command	Description
change	Changes the Remote Desktop Session Host server settings for sign in, COM port mappings, and install mode.
change logon	Enables or disables logons from client sessions on an Remote Desktop Session Host server, or displays current logon status.
change port	Lists or changes the COM port mappings to be compatible with MS-DOS applications.
change user	Changes the install mode for the Remote Desktop Session Host server.
chglogon	Enables or disables logons from client sessions on an Remote Desktop Session Host server, or displays current logon status.
chgport	Lists or changes the COM port mappings to be compatible with MS-DOS applications.
chgusr	Changes the install mode for the Remote Desktop Session Host server.
flattemp	Enables or disables flat temporary folders.
logoff	Signs out a user from a session on an Remote Desktop Session Host server and deletes the session from the server.
msg	Sends a message to a user on an Remote Desktop Session Host server.
mstsc	Creates connections to Remote Desktop Session Host servers or other remote computers.
qappsrv	Displays a list of all Remote Desktop Session Host servers on the network.
qprocess	Displays information about processes that are running on an Remote Desktop Session Host server.
query	Displays information about processes, sessions, and Remote Desktop Session Host servers.
query process	Displays information about processes that are running on an Remote Desktop Session Host server.
query session	Displays information about sessions on an Remote Desktop Session Host server.
query termserver	Displays a list of all Remote Desktop Session Host servers on the network.
query user	Displays information about user sessions on an Remote Desktop Session Host server.
quser	Displays information about user sessions on an Remote Desktop Session Host server.
qwinsta	Displays information about sessions on an Remote Desktop Session Host server.
rdpsign	Enables you to digitally sign a Remote Desktop Protocol (.rdp) file.
reset session	Enables you to reset (delete) a session on an Remote Desktop Session Host server.
rwinsta	Enables you to reset (delete) a session on an Remote Desktop Session Host server.
shadow	Enables you to remotely control an active session of another user on an Remote Desktop Session Host server.
tscon	Connects to another session on an Remote Desktop Session Host server.
tsdiscon	Disconnects a session from an Remote Desktop Session Host server.
tskill	Ends a process running in a session on an Remote Desktop Session Host server.
tsprof	Copies the Remote Desktop Services user configuration information from one user to another.

mstsc.exe (the Microsoft Remote Desktop client)

Here are some detailed resources for **mstsc.exe** (the Microsoft Remote Desktop client) to help you understand its usage, parameters, and troubleshooting:

1. **Microsoft Learn - MSTSC Documentation**: This resource provides in-depth documentation on the usage of MSTSC, including command-line options and how to configure remote desktop sessions (MS Learn).
2. **Winaero - MSTSC Command Line Arguments**: This article lists and explains the various command-line options for MSTSC.exe, including connection parameters, display settings, and admin mode (Winaero).
3. **InterWorks - Custom Remote Desktop Shortcuts**: This guide covers how to create custom RDP files and use MSTSC commands to manage remote desktop sessions with custom parameters (InterWorks).
4. **AirDroid - MSTSC Overview and Common Issues**: This resource provides a general overview of MSTSC usage, including steps for launching remote desktop sessions and troubleshooting common connection issues (AirDroid).
5. **Microsoft Q&A - Installing and Repairing MSTSC**: A discussion on how to reinstall or repair MSTSC.exe if it's missing or not functioning, including using system file checker and DISM commands (MS Learn).

What Is mstsc.exe and How to Use it

https://www.airdroid.com/remote-support/what-is-mstsc-exe/

This article aims to provide a comprehensive understanding of mstsc.exe and its significance, and provide a guide on how to use mstsc to enable remote connection and connect remote computers.

What is mstsc.exe

mstsc.exe, short for Microsoft Terminal Services Client, is a command-line utility in Windows that opens the Remote Desktop Connection toll. This tool is integral to Windows operating

systems, allowing users to connect to and control remote computers, making it essential for IT support, remote work, and accessing resources from different locations, making it a critical component for remote work and network management.

The Parameters of mstsc

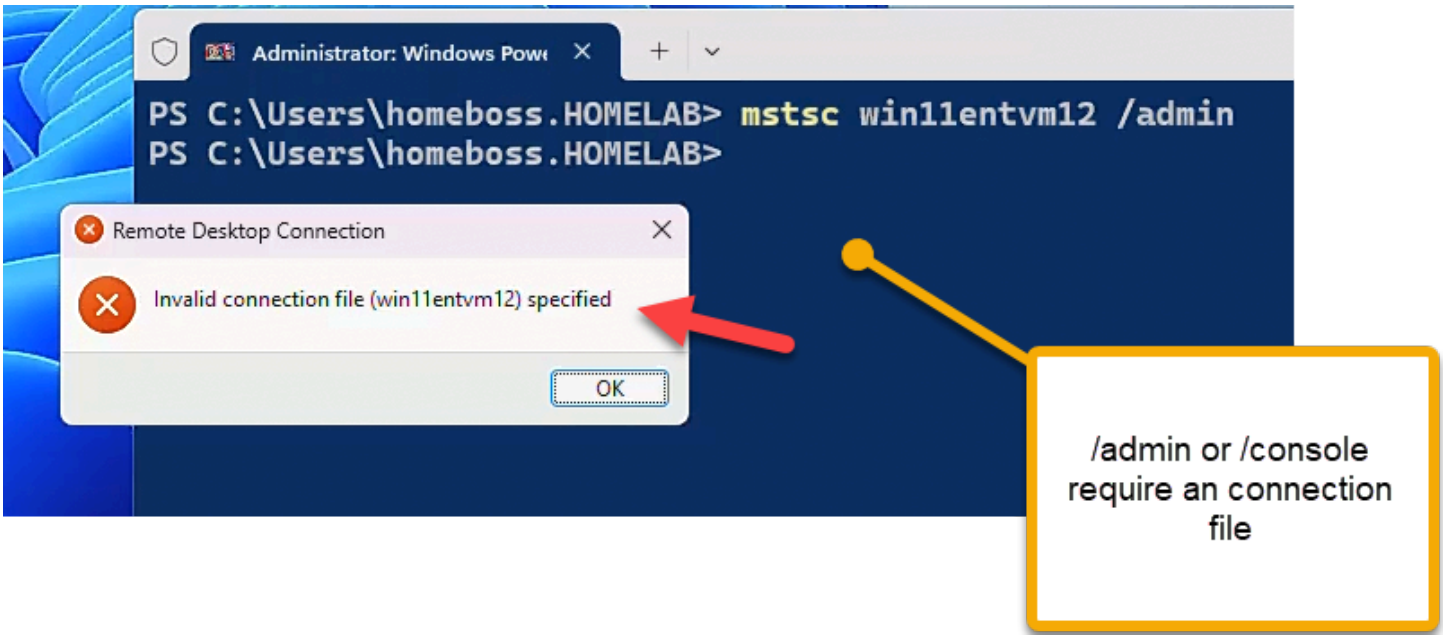
The general syntax for using matsc.exe with command-line parameters is:

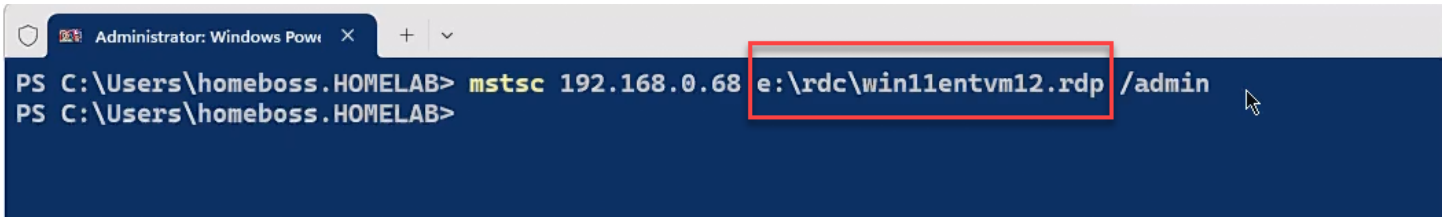
```
mstsc [<connectionfile>]
[/v:<server>[:<port>]] [/admin] [/f] [/w:<width> /h:<height>]
[/public] [/span] mstsc /edit <connectionfile> mstsc /migrate
```

Common Parameters:

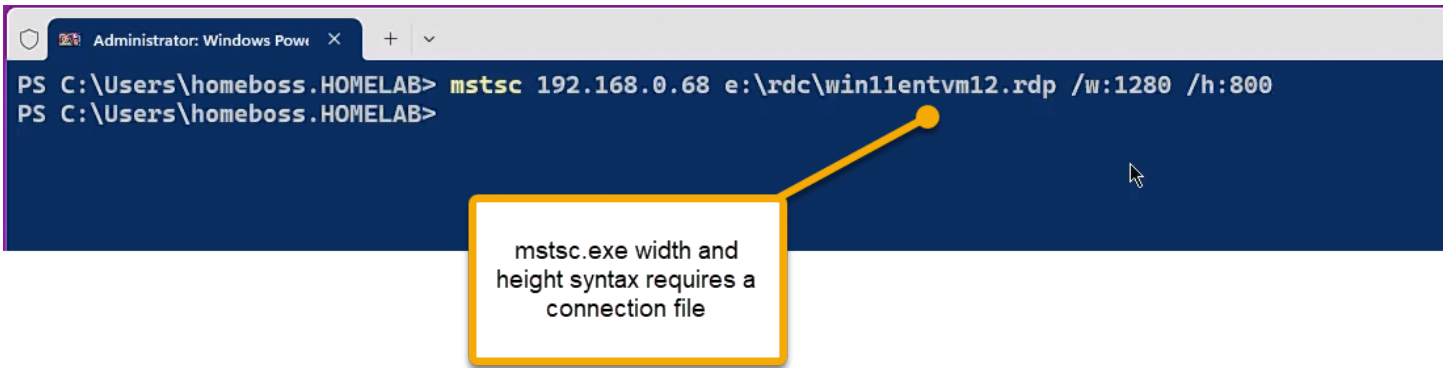
- **`/v:<computer_name>`:** Specifies the remote computer name or IP address to connect to.
- **`<connectionfile>`:** The name of an.rdp file containing the connection setting.
- **`/v: <server>[:<port>]`:** Connects to the specified remote computer and, optionally, the port number.

- **`/admin` or `/console`:** Connects to the console session of the remote computer, often used for server management.





- **`/f:`** Launched the Remote Desktop session in full-screen mode.
- **`/w:<width> and /h:<height>:`** Sets the dimensions of the Remote Desktop screen.



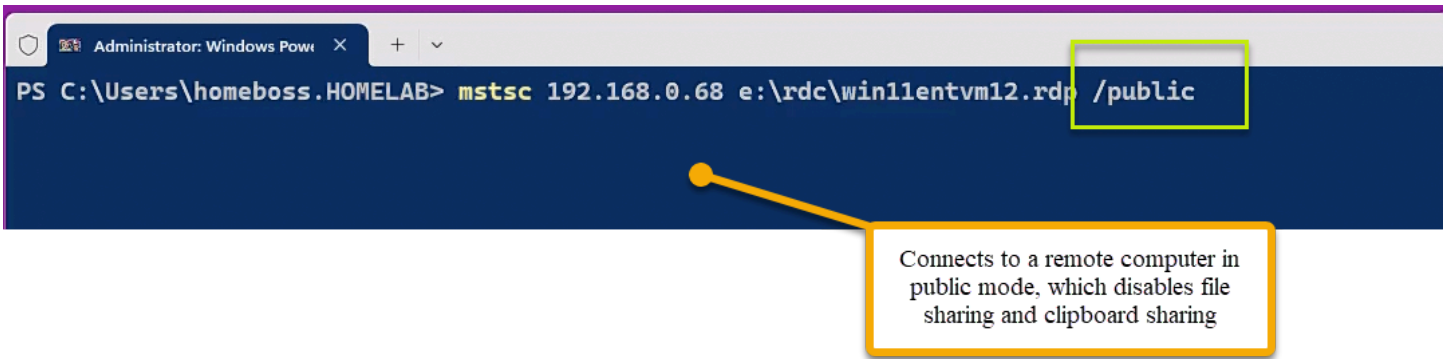
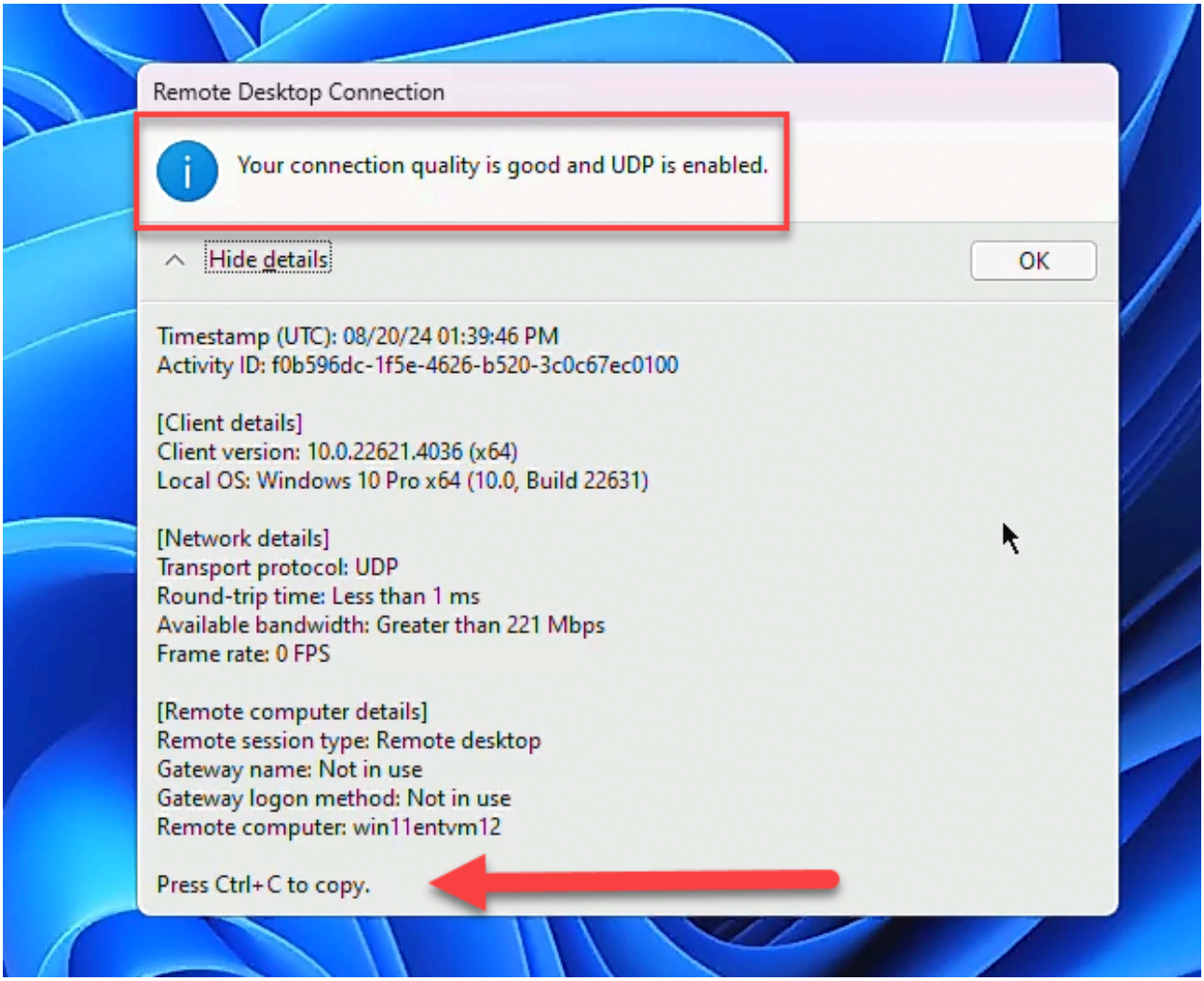
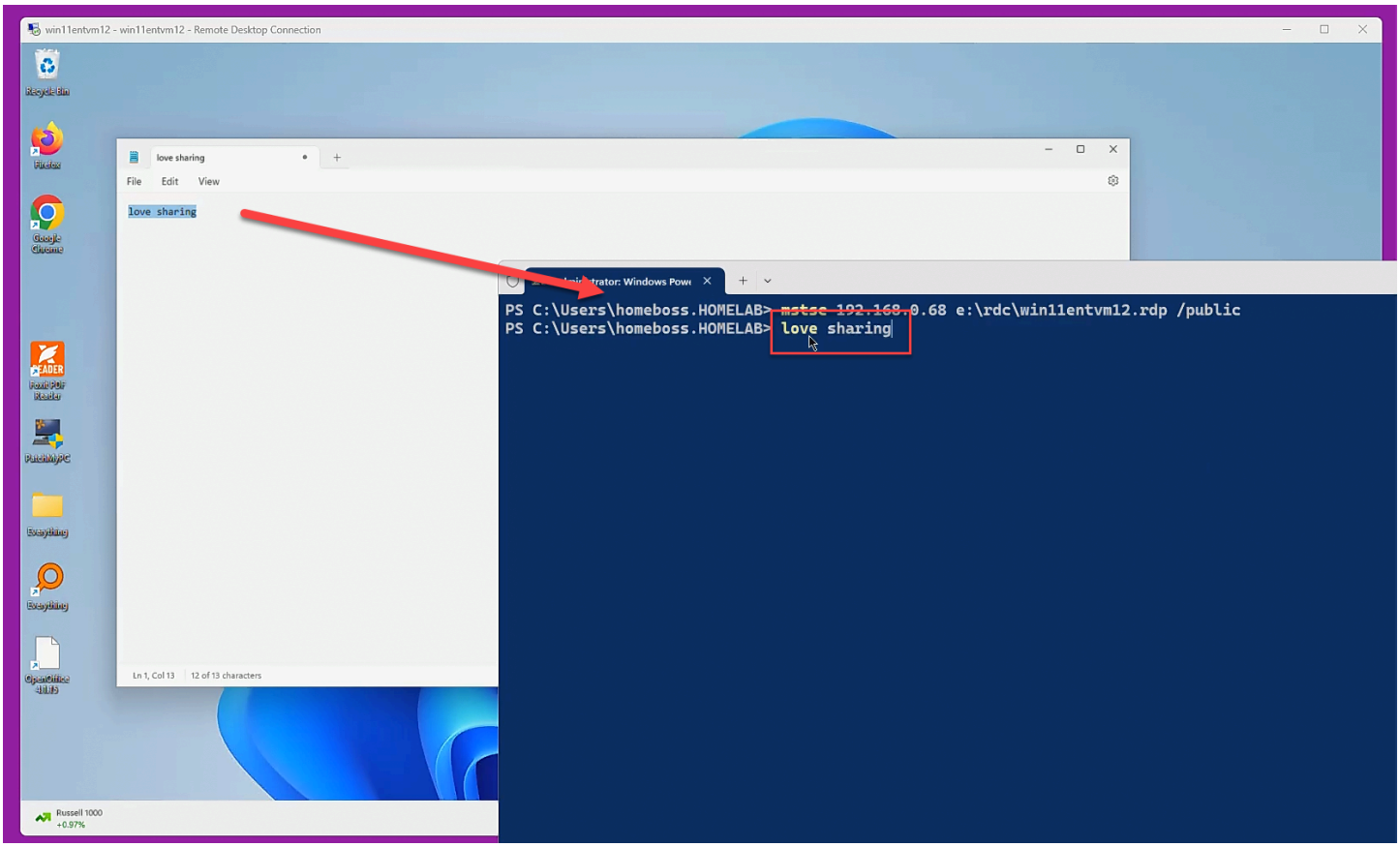


Figure 4this did not work





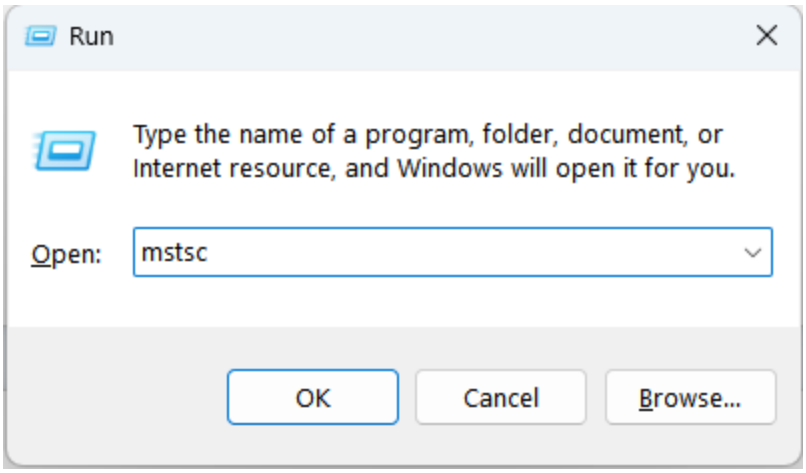
- **`/public:`** Connects to a remote computer in public mode, which disables file sharing and clipboard sharing.
- **`/span:`** Matched the Remote Desktop window to the local desktop spanning multiple monitors.

How to Use mstsc.exe

Using mstsc.exe for remote desktop connections is straightforward. Here's a step-by-step guide to help you launch the application, input the necessary connection details, and configure basic settings.

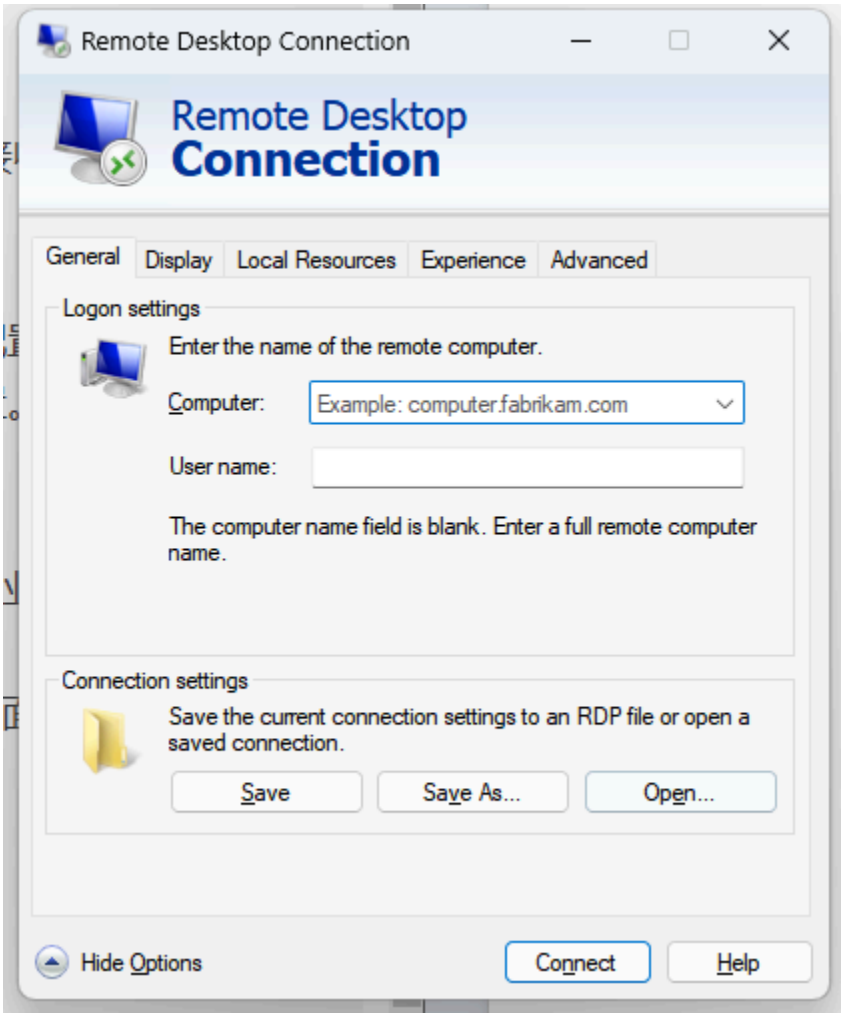
Step 1. Press `**Windows +R**` on your keyboard to open the **Run** dialog box.

Step 2. Type `mstsc` in the Run dialog box and press `Enter`. This will open the Remote Desktop Connection window.



Step 3. In the Remote Desktop Connection window, the `Computer`field, enter the IP address or hostname of the computer you want to connect to.

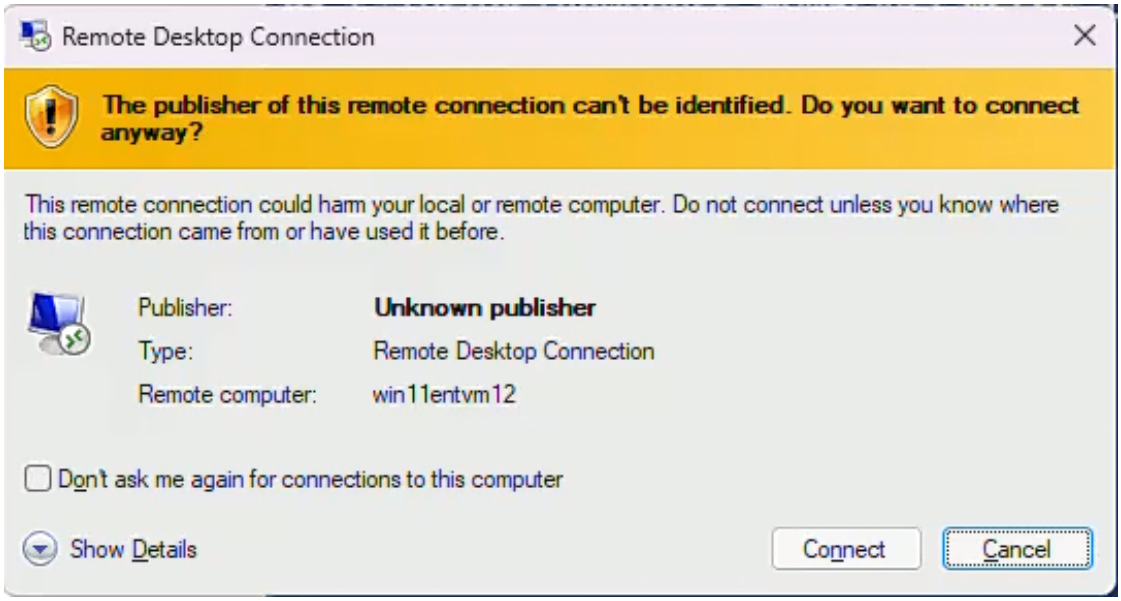
Example: `192.168.1.100` or **`remotePC.domain.com`**.

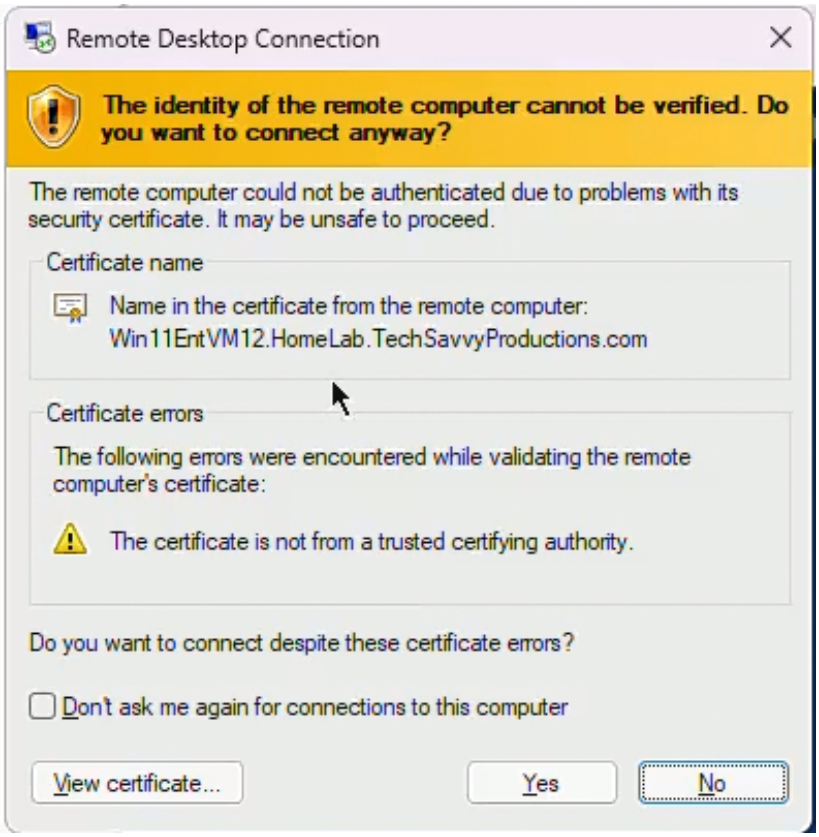


Step 4. Click on the **Show Options** button to expand the settings, and enter the username you will use to log into the remote computer. If necessary, you can also specify the domain.

Step 5. You can choose to enter your password now, or you can wait to be prompted for it after establishing the connection.

Step 6. Once all the information is entered, click the **Connect** button. If prompted, enter your password and click **OK**. You may also see a warning about the remote computer's certificate; click **Yes** to process.





Step 7. After the connection is established, you'll see the desktop of the remote computer. You can now interact with it as if you were physically present on that computer.

Troubleshooting Common Issues

When using mstsc.exe, users may encounter several common issues that can disrupt remote desktop sessions. Below are troubleshooting tips for resolving these problems.

Firewall and network setting:

- The connection fails to establish due to blocked ports or network restrictions. Ensure that the Remote Desktop Protocol(RDP) port (default is 3389) is open on both the remote and local firewalls.

```
PS C:\Users\homeboss.HOMELAB> Test-NetConnection -ComputerName win11entvm14 -Port 3389


ComputerName     : win11entvm14
RemoteAddress    : 192.168.0.141
RemotePort       : 3389
InterfaceAlias   : vEthernet (External10Gbedomain)
SourceAddress    : 192.168.0.56
TcpTestSucceeded : True
```

testing and verifying that a remote PC has an open firewall port 3389 and the terminal service responded

- Authentication issues: You receive an error stating that the credentials are incorrect or the remote computer's security settings do not allow your credentials. Double-check the username and password entered. Remember to include the domain if required(e.g., *domain\username*)
- Display and resolution issues: The remote desktop screen appears distorted, or the resolution doesn't match your local settings. Before connecting, go to **Show Option** in the **Remote Desktop Connection** windows, navigate to the **Display** tab, and adjust the resolution and color depth settings.
- **Session disconnection and how to resolve it:** The remote session frequently disconnects to times out. Check the network connection on both the local and remote computers to ensure stability. Ensure that power-saving settings on the remote computer are not causing it to go to sleep or disconnect from the network.
- Performance Problems: The remote desktop session is slow to unresponsive. Reduce the display quality by adjusting the Experience settings in the Remote Desktop

Connection window before connecting. Disable unnecessary visual effects like font smoothing or window dragging to improve performance.

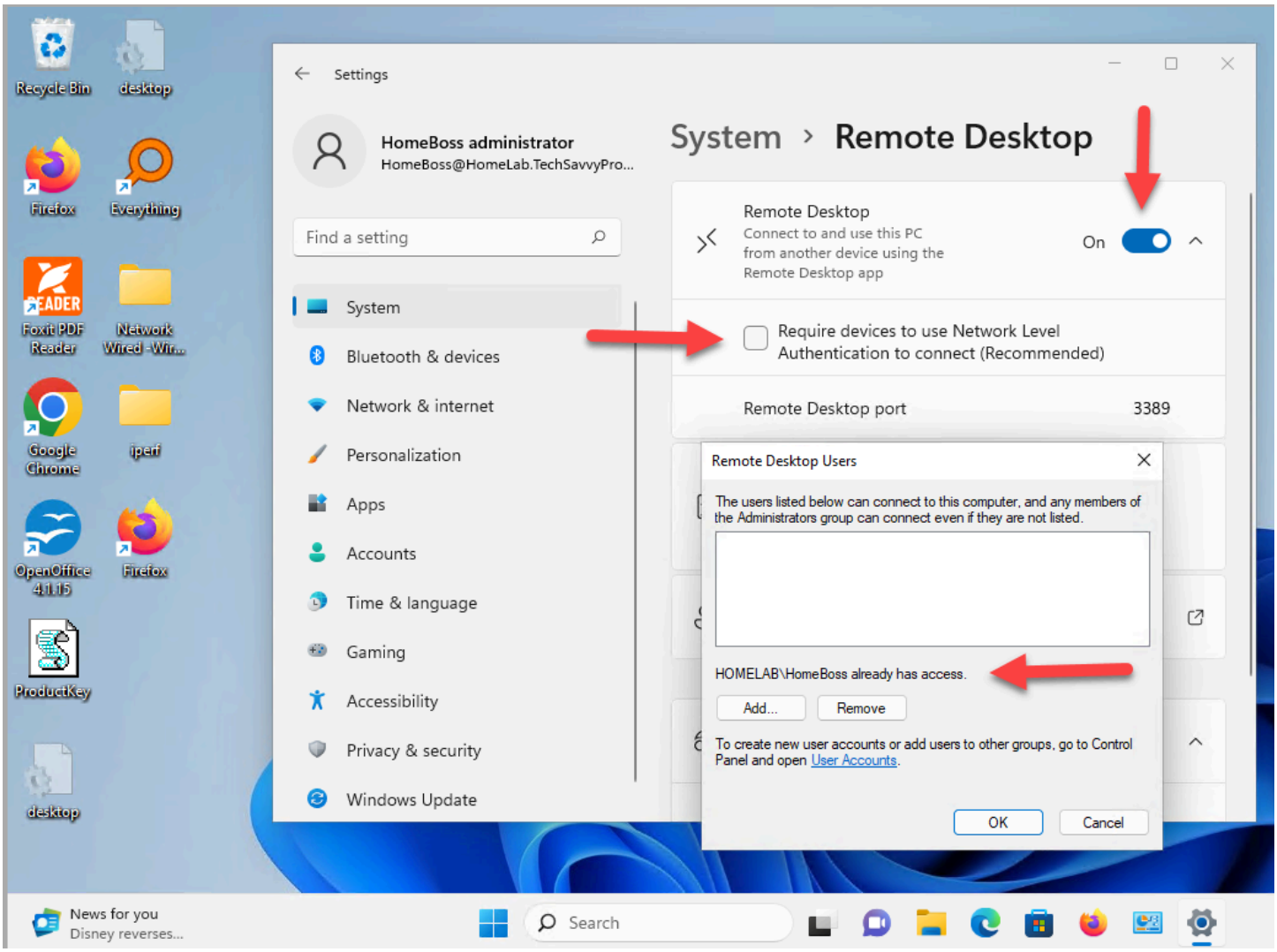


Figure 5 Homeboss and local admin are able to use RDC

Remote Access Auto Connection Manager	Creates a connection to a remote network whenever a program ...	
Remote Access Connection Manager	Manages dial-up and virtual private network (VPN) connections ...	Running
Remote Desktop Configuration	Remote Desktop Configuration service (RDCS) is responsible for ...	Running
Remote Desktop Services	Allows users to connect interactively to a remote computer. Re...	Running
Remote Desktop Services UserMode Port Redirector	Allows the redirection of Printers/Drives/Ports for RDP connecti...	Running

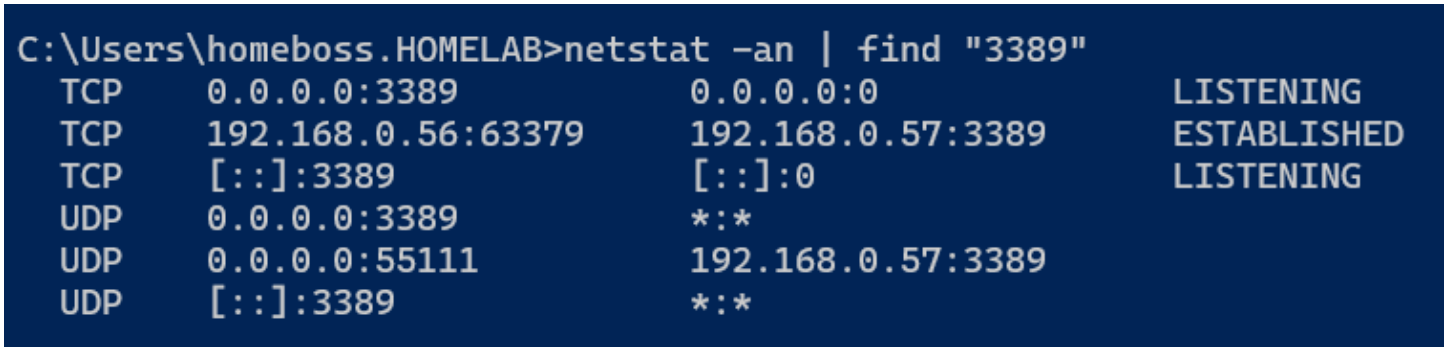
How to verify RDC client is active and that port 3389 is open

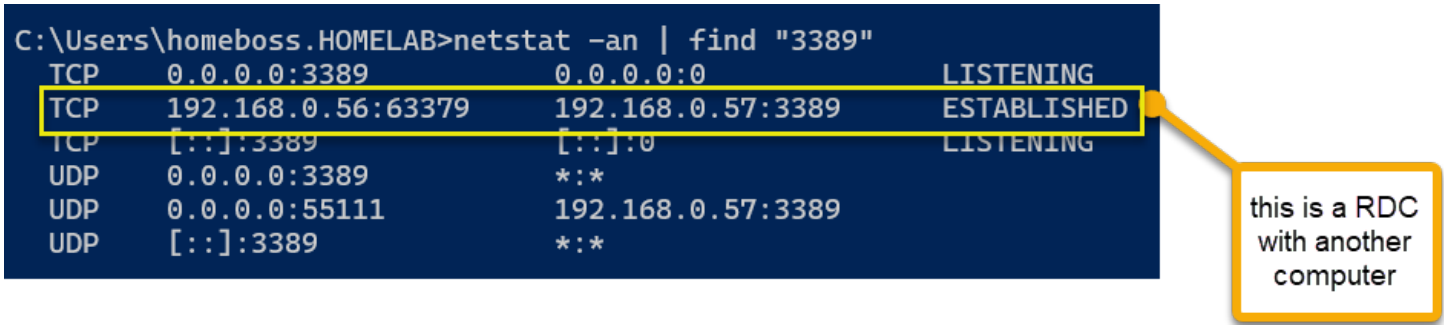
To verify that Remote Desktop Connection (RDC) client is active and that port 3389 is open and not blocked by Windows or a third-party anti-virus program, you can use a variety of built-in Windows tools and Sysinternals utilities. Below are some methods to test and verify RDC functionality.

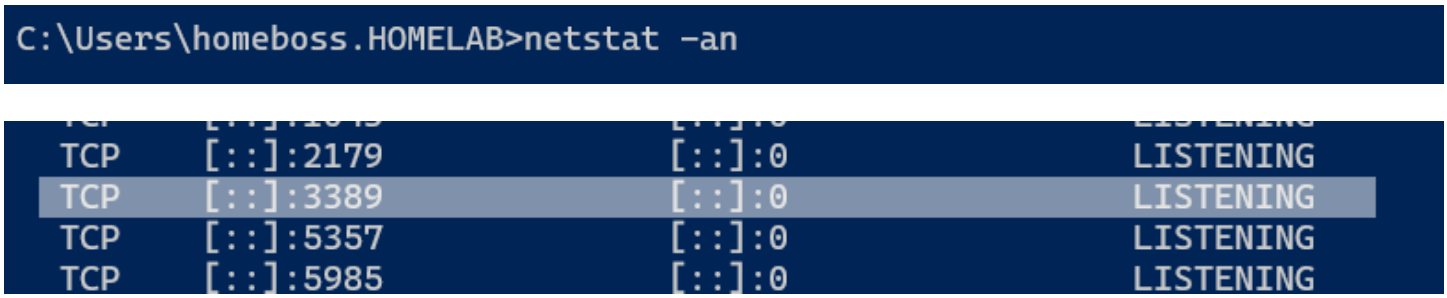
1. Command Line Tools

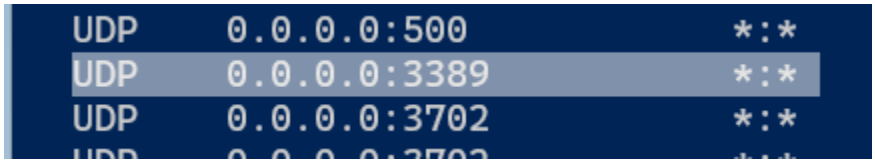
Netstat: Verify if port 3389 is listening. This verify RDC client and firewall open on local PC

```
netstat -an | find "3389"
```



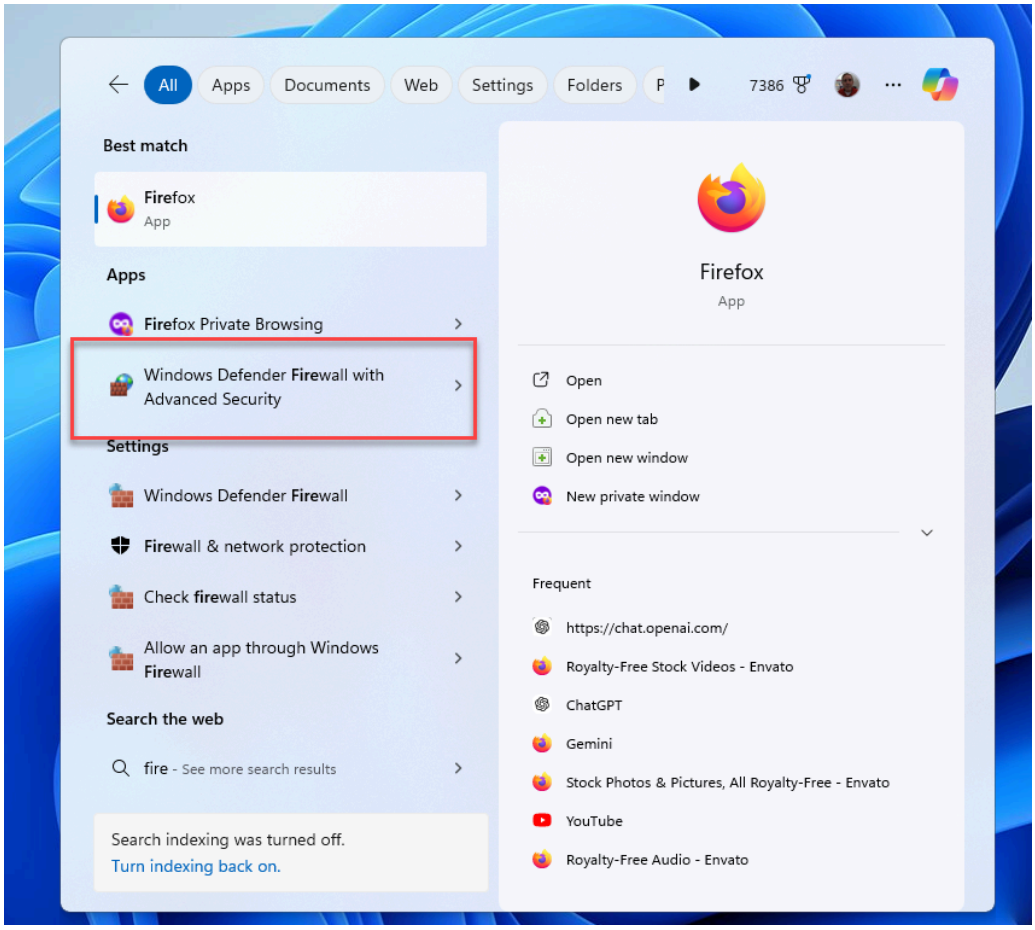






If the command shows an entry with `0.0.0.0:3389` or your local IP address listening on `3389`, it indicates that the RDC service is listening on the standard port.

2. Windows Built-in Tools



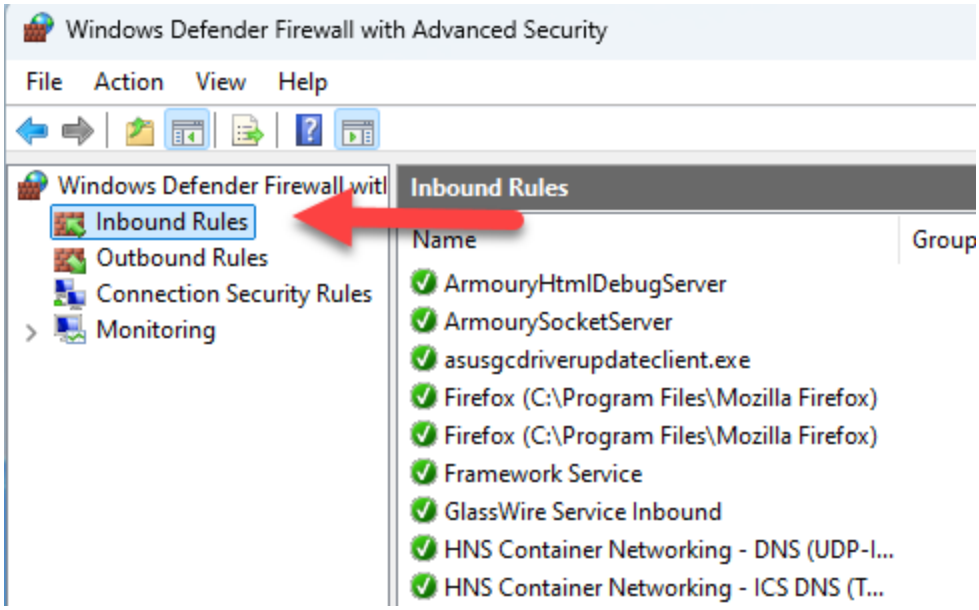
Windows Firewall Configuration Ensure port 3389 is allowed through Windows Firewall.

To check the current status of the firewall rules:

- Open Windows Firewall with Advanced Security

- Navigate to Inbound Rules.

- Look for the rule labeled **Remote Desktop (TCP-In)** and ensure it is enabled.



- **Remote Desktop Service**: Check if the Remote Desktop service is running.

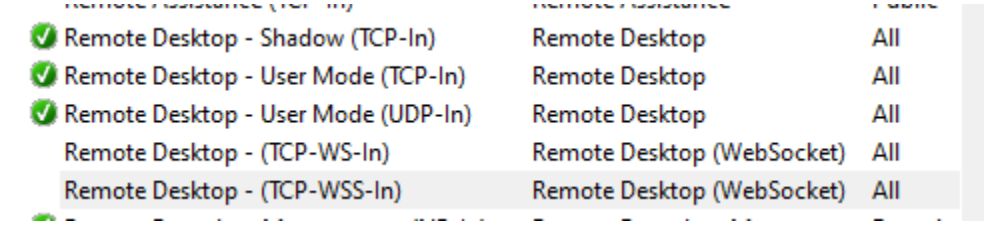


Figure 6inbound firewall rules for RDC

In the Windows Advanced Firewall GUI under Inbound Rules, the different Remote Desktop rules serve to control various network traffic types and scenarios associated with Remote Desktop Services (RDP). Let's break down the five common Remote Desktop rules typically found:

1. Remote Desktop - TCP-In

- **Purpose**: This rule allows inbound traffic over **TCP port 3389**, which is the default port used by the Remote Desktop Protocol (RDP).
- **Function**: The primary purpose is to enable Remote Desktop sessions over a TCP connection, which is the default mode for RDP traffic. TCP ensures reliable delivery of packets, making it suitable for RDP's requirements for secure and ordered communication.
- **User Mode Application**: mstsc.exe (Microsoft Terminal Services Client) and svchost.exe hosting the Remote Desktop Services (TermService) use this rule.

2. Remote Desktop - UDP-In

- **Purpose**: This rule allows inbound traffic over **UDP port 3389**.
- **Function**: Remote Desktop introduced support for **UDP** in later versions (e.g., **RDP 8.0 and above**). UDP allows for faster, low-latency transmission and is beneficial for maintaining performance on

high-latency networks, such as wireless or satellite connections. The UDP stream works in tandem with the TCP stream to optimize performance.

- **User Mode Application**: Similar to the TCP rule, mstsc.exe and svchost.exe hosting the Remote Desktop Services (TermService) use this rule, particularly when leveraging the RDP protocol's UDP capabilities.

3. Remote Desktop - Shadow (TCP-In)

- **Purpose**: This rule allows for **session shadowing** over **TCP port 3389**.
- **Function**: Session shadowing is a feature that allows an admin to view or take control of another user's Remote Desktop session. This is primarily used for remote support, training, or monitoring purposes. The rule enables this feature by ensuring that the shadowing session can establish a connection over TCP.
- **User Mode Application**: Applications like mstsc.exe and administrative tools supporting shadowing rely on this rule. Typically, this is used by administrators who need to shadow or share a user session.

4. Remote Desktop - WS-In (TCP-In)

- **Purpose**: This rule allows inbound traffic over **TCP ports for Windows Store (WS) Remote Desktop applications**.
- **Function**: This is related to Windows Store (Modern UI) versions of the Remote Desktop app that use a different traffic handling mechanism than the traditional Remote Desktop Connection. These apps may use different port ranges or protocols, and this rule facilitates their operation.
- **User Mode Application**: **rdpws.exe** (or similar) associated with Modern UI apps uses this rule to connect via Remote Desktop.

5. Remote Desktop - RemoteFX (TCP-In)

- **Purpose**: This rule is used for inbound connections via **TCP** that utilize **RemoteFX**.
- **Function**: RemoteFX is a set of protocols and technologies that enhance the visual and graphical performance of Remote Desktop sessions, particularly for virtual machines. This rule ensures that connections utilizing RemoteFX over TCP are allowed.
- **User Mode Application**: mstsc.exe and svchost.exe with RemoteFX-enabled sessions use this rule, typically seen in environments where RemoteFX is leveraged to provide richer remote desktop experiences, often with better multimedia and GPU acceleration support.

Summary of Key Points:

- **Remote Desktop - TCP-In**: Allows basic RDP traffic over TCP for traditional Remote Desktop sessions.
- **Remote Desktop - UDP-In**: Enables faster, lower-latency RDP traffic over UDP.
- **Remote Desktop - Shadow (TCP-In)**: Facilitates session shadowing for monitoring or remote assistance.

- **Remote Desktop - WS-In**: Supports connections via the Windows Store version of the Remote Desktop app.
- **Remote Desktop - RemoteFX (TCP-In)**: Allows traffic associated with enhanced graphical performance via RemoteFX.

Each of these rules has a specific role in ensuring that different types of Remote Desktop connections and features can operate smoothly within the Windows environment.

```
cmd  sc query termservice
```

You should see the service status as `RUNNING`. If it is stopped, you can start it with:

```
C:\Users\homeboss.HOMELAB>sc query termservice

SERVICE_NAME: termservice
        TYPE               : 30  WIN32
        STATE              : 4  RUNNING
                                (STOPPABLE, NOT_PAUSABLE, ACCEPTS_SHUTDOWN)
        WIN32_EXIT_CODE    : 0  (0x0)
        SERVICE_EXIT_CODE  : 0  (0x0)
        CHECKPOINT         : 0x0
        WAIT_HINT          : 0x0
```

```
cmd   sc start termservice
```

3. PowerShell Commands

Test Network Port Availability: Use `Test-NetConnection` to check if the RDC port (3389) is open on the target machine.

```
Test-NetConnection -ComputerName <remote-machine-ip> -Port 3389
```

```
PS C:\Users\homeboss.HOMELAB> Test-NetConnection -ComputerName 192.168.0.231 -Port 3389
WARNING: TCP connect to (192.168.0.231 : 3389) failed
WARNING: Ping to 192.168.0.231 failed with status: DestinationHostUnreachable


ComputerName            : 192.168.0.231
RemoteAddress           : 192.168.0.231
RemotePort              : 3389
InterfaceAlias          : vEthernet (External10Gbedomain)
SourceAddress           : 192.168.0.56
PingSucceeded           : False
PingReplyDetails (RTT)  : 0 ms
TcpTestSucceeded        : False
```

this DC was off and the error "time out" was really long

```
Test-NetConnection - 192.168.0.231:3389
   Attempting TCP connect

   Waiting for response
```

This command will show you whether the port is open and reachable.

```
PS C:\Users\homeboss.HOMELAB> Test-NetConnection -ComputerName 192.168.0.232 -Port 3389


ComputerName     : 192.168.0.232
RemoteAddress    : 192.168.0.232
RemotePort       : 3389
InterfaceAlias   : vEthernet (External10Gbedomain)
SourceAddress    : 192.168.0.56
TcpTestSucceeded : True
```

This DC was up but the time delay was really long

Verify Remote Desktop Services: Check if the Remote Desktop Service is running. Local PC

```
Get-Service -Name TermService
```

If the status is `Running`, the service is active.

```
PS C:\Users\homeboss.HOMELAB> Get-Service -Name TermService

Status   Name               DisplayName
------   ----               -----------
Running  TermService        Remote Desktop Services
```

-

Firewall Rules Check: Check if the Windows Firewall is allowing Remote Desktop traffic.

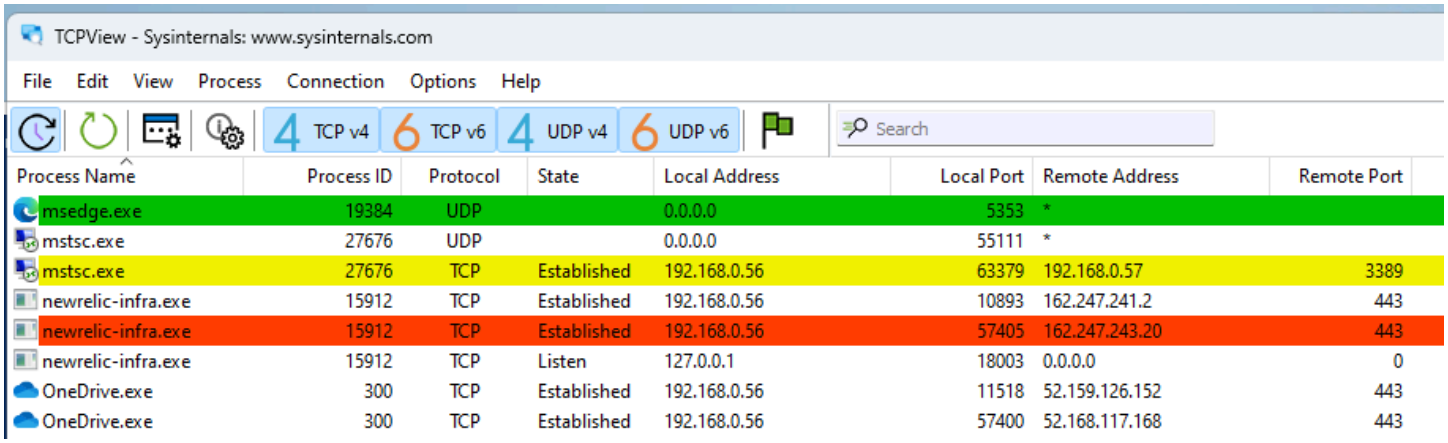
Get-NetFirewallRule -DisplayName "Remote Desktop - User Mode (TCP-In)"

```
PS C:\Users\homeboss.HOMELAB> Get-NetFirewallRule -DisplayName "Remote Desktop - User Mode (TCP-In)"


Name                          : RemoteDesktop-UserMode-In-TCP
DisplayName                   : Remote Desktop - User Mode (TCP-In)
Description                   : Inbound rule for the Remote Desktop service to allow RDP traffic. [TCP 3389]
DisplayGroup                  : Remote Desktop
Group                         : @FirewallAPI.dll,-28752
Enabled                       : True
Profile                       : Any
Platform                      : {}
Direction                     : Inbound
Action                        : Allow
EdgeTraversalPolicy           : Block
LooseSourceMapping            : False
LocalOnlyMapping              : False
Owner                         :
PrimaryStatus                 : OK
Status                        : The rule was parsed successfully from the store. (65536)
EnforcementStatus             : NotApplicable
PolicyStoreSource             : PersistentStore
PolicyStoreSourceType         : Local
RemoteDynamicKeywordAddresses : {}
PolicyAppId                   :
```

This will return the status of the firewall rule. It should be `Enabled`.

4. Sysinternals Tools

TCPView: Monitor the open TCP/UDP ports on your machine, including port 3389 for Remote Desktop.



Process Name	Process ID	Protocol	State	Local Address	Local Port	Remote Address	Remote Port
msedge.exe	19384	UDP		0.0.0.0	5353	*	
mstsc.exe	27676	UDP		0.0.0.0	55111	*	
mstsc.exe	27676	TCP	Established	192.168.0.56	63379	192.168.0.57	3389
newrelic-infra.exe	15912	TCP	Established	192.168.0.56	10893	162.247.241.2	443
newrelic-infra.exe	15912	TCP	Established	192.168.0.56	57405	162.247.243.20	443
newrelic-infra.exe	15912	TCP	Listen	127.0.0.1	18003	0.0.0.0	0
OneDrive.exe	300	TCP	Established	192.168.0.56	11518	52.159.126.152	443
OneDrive.exe	300	TCP	Established	192.168.0.56	57400	52.168.117.168	443

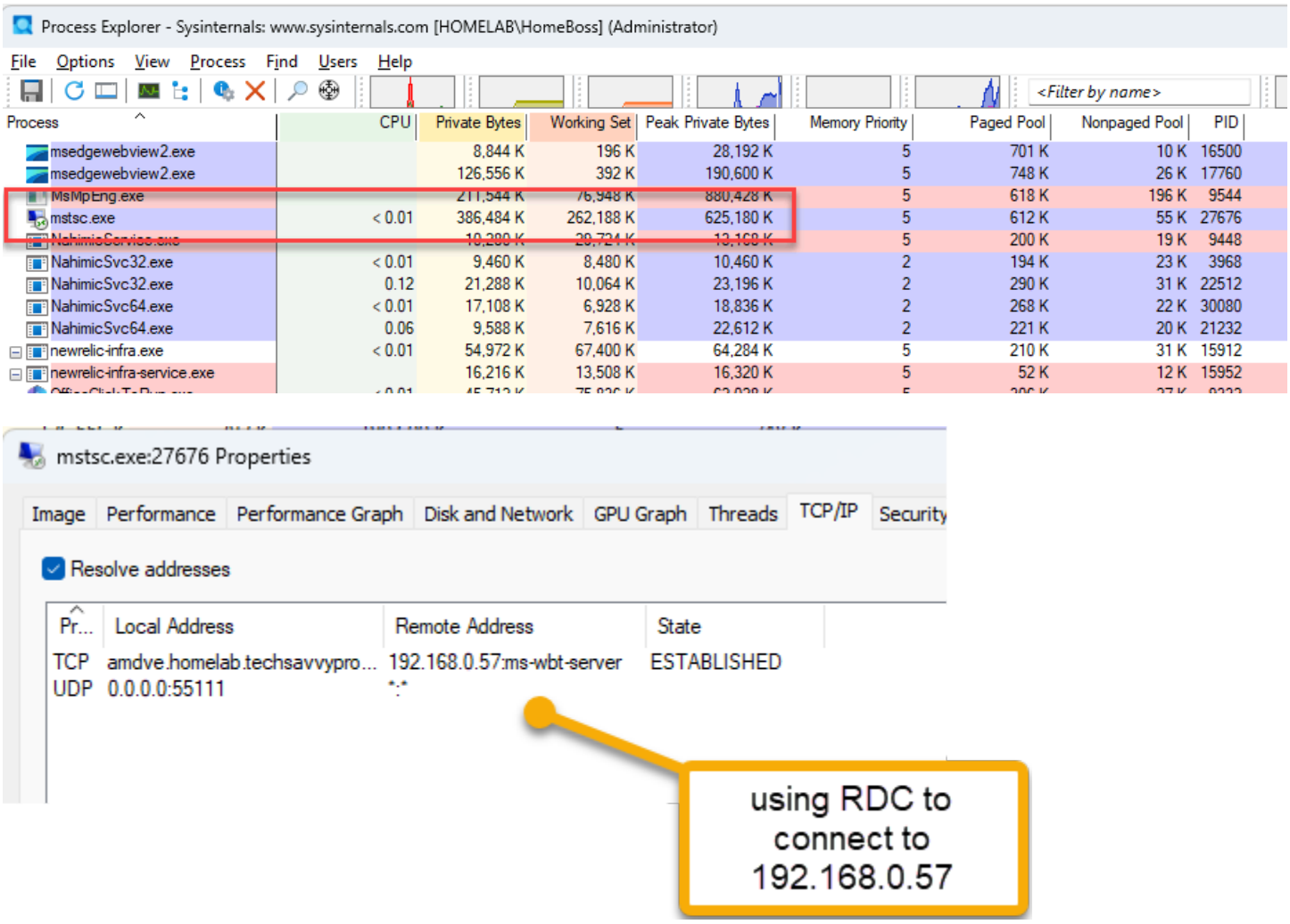
Download and run **[TCPView](https://learn.microsoft.com/en-us/sysinternals/downloads/tcpview)**.

Look for `mstsc.exe` (the Remote Desktop client) or `svchost.exe` using port 3389.

Process Explorer: Use Process Explorer to check if the `mstsc.exe` process (the RDC client) is running.

Download and run [Process Explorer](https://learn.microsoft.com/en-us/sysinternals/downloads/process-explorer).

- Look for `mstsc.exe` in the list of running processes.

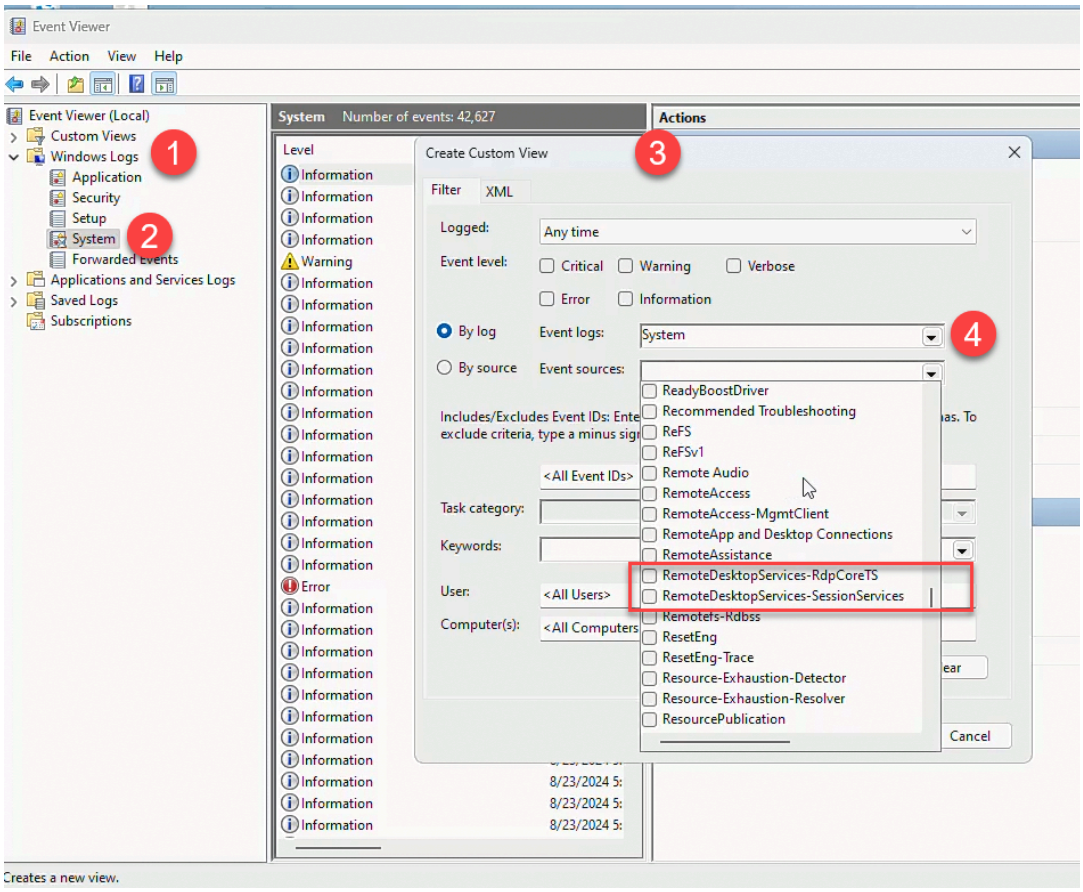


5. Event Viewer

You can also verify any errors related to Remote Desktop in Event Viewer:

- Go to `Event Viewer > Windows Logs > System` and look for any errors or warnings related to Remote Desktop Services.

By combining these tools, you can effectively verify that RDC is running correctly, that port 3389 is open, and that neither Windows Firewall nor third-party antivirus is blocking it.



Differences Between Hyper-V VM Access and RDC

When accessing virtual machines (VMs) via Hyper-V and comparing that to using Remote Desktop Connection (RDC) to access a remote client, there are some key differences and similarities regarding how these technologies work, the services they use, and troubleshooting approaches.

Differences Between Hyper-V VM Access and RDC

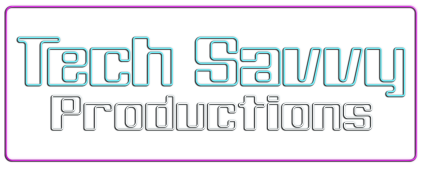
1. **Access Method:**
    - **Hyper-V VM Access (Thumbnail or VM Shell):**
        - The connection to a VM in Hyper-V is typically done using the **Hyper-V Manager** or **Virtual Machine Connection (VMC)**. When you click on the thumbnail of your VM, you open a console-based shell, which provides direct access to the virtual machine's interface, including its boot process, BIOS, and operating system shell. It is like physically accessing the machine.
        - **This connection operates directly through the Hyper-V host, bypassing network protocols like RDP (Remote Desktop Protocol)** and instead relying on virtualization technologies.
    - **Remote Desktop Connection (RDC):**
        - RDC uses **RDP**, a network protocol, to access the desktop of a remote machine that has the RDP service enabled. RDC connects to a fully booted and operational OS, allowing remote control of the graphical desktop.
        - It does not provide access to the machine's BIOS, boot sequences, or hardware settings.
2. **Services Involved:**
    - **Hyper-V VM Access**:
        - Uses **Hyper-V services** such as the **VMMS (Virtual Machine Management Service)** and **VMRS (Virtual Machine Runtime Service)** to manage the virtualization environment and the VM's state.
        - **Integration Services**: Hyper-V VMs use **Guest Services** for interactions between the host and the guest OS (e.g., for file copy, shutdown commands, etc.).
    - **RDC**:
        - Uses the **Remote Desktop Services (RDS)**, which includes the **RDP protocol**, **Remote Desktop Gateway**, and **Remote Desktop Services Session Host** (RDSSH).

- RDP service needs to be enabled on the client, and network communication happens over port **3389**.

3. **Capabilities:**
    - **Hyper-V VM Access**:
        - Provides **full control** over the VM, including booting, shutting down, and even modifying hardware resources like virtual CPUs, RAM, or network adapters.
        - Can access the **BIOS/UEFI**, observe the **boot process**, and troubleshoot issues even if the VM OS is unresponsive.
        - Does not require a functioning network connection or RDP service on the guest OS to interact with the VM.
    - **RDC**:
        - Limited to accessing a **functioning operating system** over a network. RDC requires that the remote machine is fully operational with RDP services enabled.
        - Troubleshooting is limited to post-boot issues, as you cannot access the BIOS or boot process through RDC.

Similarities Between Hyper-V VM Access and RDC

1. **Remote Access to Desktops:**
    - Both methods allow you to **remotely access** a graphical interface, controlling a desktop as though you were physically present.
2. **Keyboard, Mouse, and Display:**
    - Both technologies offer **full keyboard and mouse control**, with display forwarding to the local machine.
3. **User Experience:**
    - Once the VM is fully booted and the OS is running, the experience inside the VM's graphical desktop (via Hyper-V shell) is similar to RDC. Both allow you to run applications and interact with the OS.
4. **Session Management:**
    - Both Hyper-V VM access and RDC offer the ability to disconnect from the session while leaving the VM or remote machine running, allowing you to resume where you left off.



Unique Services to Hyper-V VMs vs. RDC

- **Unique to Hyper-V VM Access:**
    - **VMMS (Virtual Machine Management Service)**: Manages the state of the VMs (starting, stopping, saving).
    - **Hyper-V Integration Services**: Facilitates communication between the host and the guest OS for functions like time synchronization, mouse integration, and data exchange.
    - **VMC (Virtual Machine Connection)**: The direct console access used for VM interaction.
    - **Host Resource Management**: Direct control over the VM's hardware and resources.
- **Unique to RDC:**
    - **Remote Desktop Services (RDS)**: Includes **Remote Desktop Protocol (RDP)**, **RD Gateway**, and other services enabling remote desktop access over a network.
    - **Network-based Communication**: Requires a working network stack on both the client and server.

Troubleshooting Differences

1. **Troubleshooting in Hyper-V VM Access:**
    - **Boot Issues**: You can troubleshoot boot failures, access the VM's BIOS, adjust hardware resources, and view the entire boot sequence directly.
    - **Guest OS Issues**: Integration services provide insights into guest health, and you can force restarts or reset the VM without needing network access.
    - **Resource Allocation**: You can manage and allocate hardware resources (e.g., memory, CPU) directly to the VM, offering a higher level of control over the virtual environment.
2. **Troubleshooting with RDC:**
    - **Network Issues**: RDC relies on network connectivity, so if there is a network issue or RDP service is down on the remote client, you cannot connect. You'll need alternative means to access the client (like Hyper-V or console access if it's a VM).
    - **Post-Boot OS Issues**: Troubleshooting is limited to the remote OS once it's fully booted. You cannot interact with the machine during boot or modify hardware settings remotely.
    - **Service Dependent**: RDC troubleshooting often involves ensuring the **RDP service** is running and verifying firewall settings, network connectivity, and RDP configurations.

Conclusion

Accessing VMs via Hyper-V is fundamentally different from using RDC in that Hyper-V gives full control over the VM, including pre-boot operations and hardware configurations. RDC is a network-based protocol limited

to accessing a functioning OS. Troubleshooting with Hyper-V involves managing the VM at a deeper level, while RDC troubleshooting focuses more on network connectivity and service issues.

Now all has been heard;
here is the conclusion of the matter:
Fear God and keep his commandments,
for this is the duty of all mankind.
For God will bring every deed into judgment,
including every hidden thing,
whether it is good or evil.

Remote Desktop (mstsc.exe) Command Line Arguments

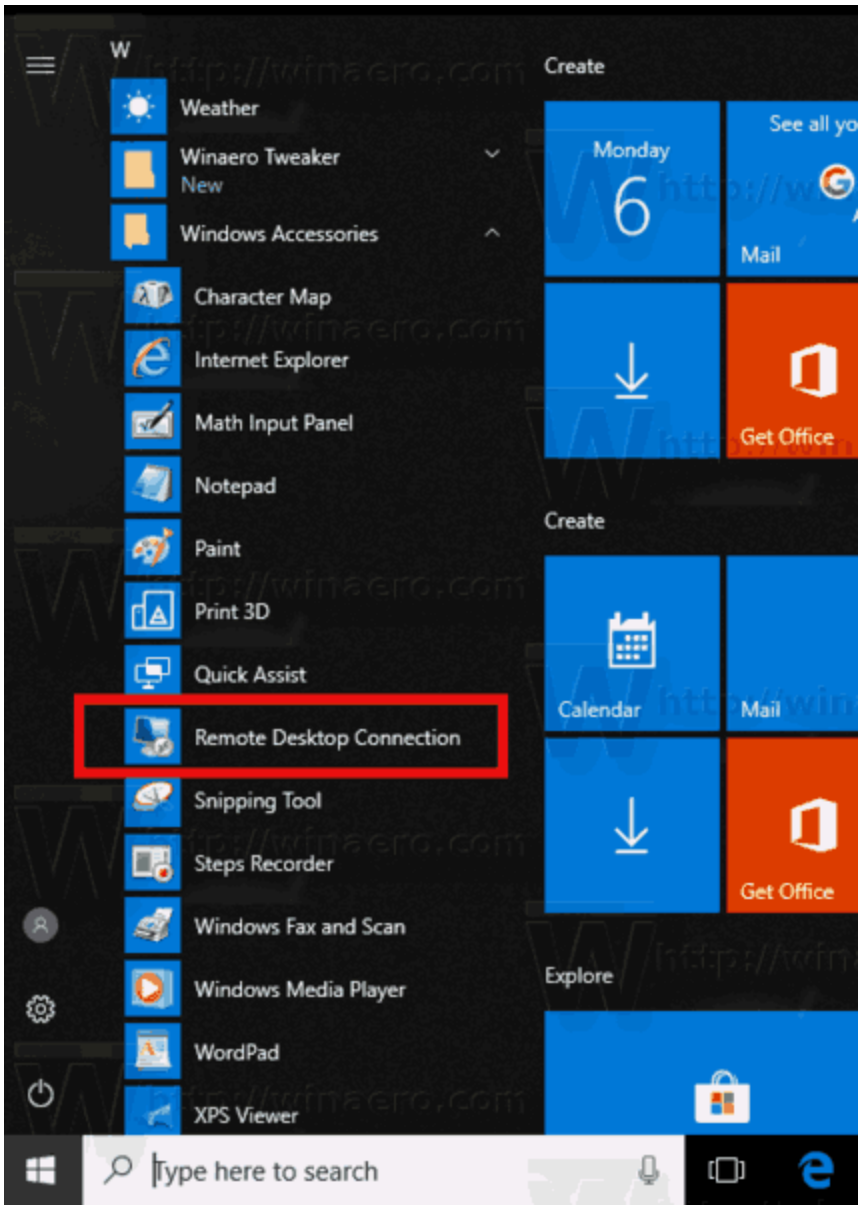
mstsc.exe is the built-in client software which allows connecting to a computer via

Remote Desktop Protocol (RDP). It is a special network protocol which allows a user to establish a connection between two computers and access the Desktop of a remote host. The local computer is often referred to as the "client". If you are running Windows, in most cases you use mstsc.exe to connect to another computer with RDP.

https://winaero.com/mstsc-exe-command-line-arguments/

Note: Any edition of Windows 10 can act as Remote Desktop Client. To host a remote session, you need to be running Windows 10 Pro or Enterprise. You can connect to a Windows 10 Remote Desktop host from another PC running Windows 10, or from an earlier Windows version like Windows 7 or Windows 8, or Linux. Windows 10 comes with both client and server software out-of-the-box, so you don't need any extra software installed.

You can find the shortcut to open a new Remote Desktop session in the Start menu. It is under Windows Accessories\Remote Desktop Connection. See the following screenshot:



Alternatively, you can launch the Remote Desktop connection app from the Run dialog (press Win + R keys together) by typing *mstsc.exe* in the Run box.

The *mstsc.exe* app supports a number of useful command line options you can apply in the Run dialog or by running the app from the command prompt or PowerShell. Let's review them.

Remote Desktop (mstsc.exe) Command Line Arguments

Tip: You can see a brief description for available options by running the following command:

mstsc.exe /?

The syntax is as follows:

```
MSTSC [<connection file>] [/v:<server[:port]>] [/g:<gateway>]
[/admin] [/f[ullscreen]] [/w:<width> /h:<height>] [/public] |
[/span] [/multimon] [/edit "connection file"]
[/restrictedAdmin] [/remoteGuard] [/prompt]
[/shadow:<sessionID> [/control] [/noConsentPrompt]]
```

"connection file" - Specifies the name of an .RDP file for the connection.

/v:<server[:port]> - Specifies the remote PC to which you want to connect.

/g:<gateway> - Specifies the RD Gateway server to use for the connection. This parameter is only read if the endpoint remote PC is specified with /v.

/admin - Connects you to the session for administering a remote PC.

/f - Starts Remote Desktop in full-screen mode.

/w:<width> - Specifies the width of the Remote Desktop window.

/h:<height> - Specifies the height of the Remote Desktop window.

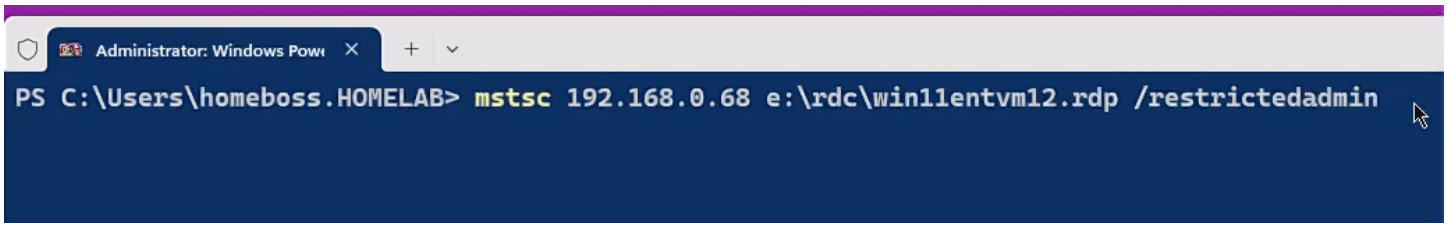
/public - Runs Remote Desktop in public mode.

/span - Matches the remote desktop width and height with the local virtual desktop, spanning across multiple monitors, if necessary. To span across monitors, the monitors must be arranged to form a rectangle.

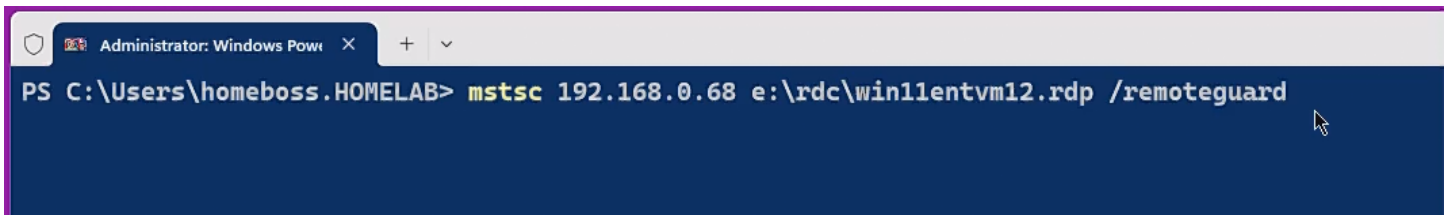
/multimon - Configures the Remote Desktop Services session monitor layout to be identical to the current client-side configuration.

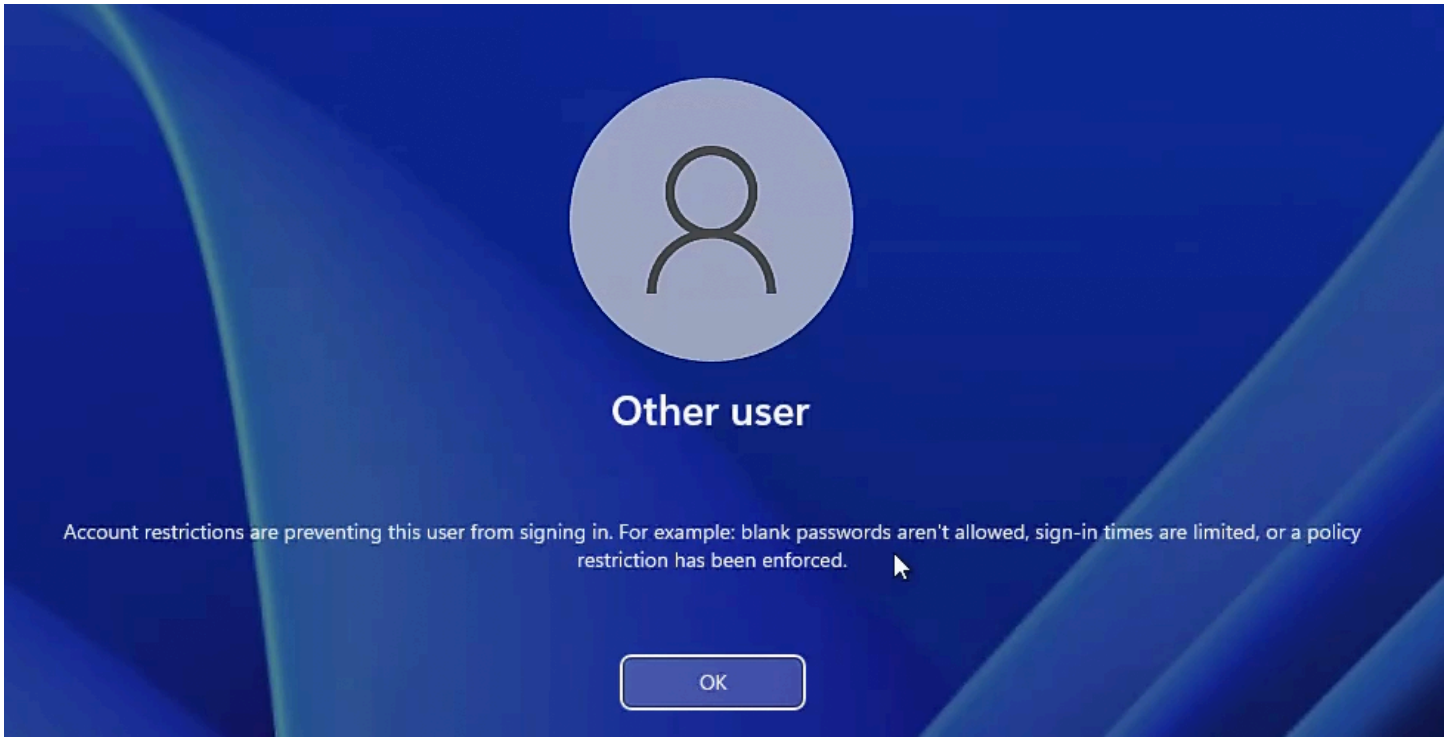
/edit - Opens the specified .RDP connection file for editing.

/restrictedAdmin - Connects you to the remote PC in Restricted Administration mode. In this mode, credentials won't be sent to the remote PC, which can protect you if you connect to a PC that has been compromised. However, connections made from the remote PC might not be authenticated by other PCs, which might impact application functionality and compatibility. This parameter implies /admin.

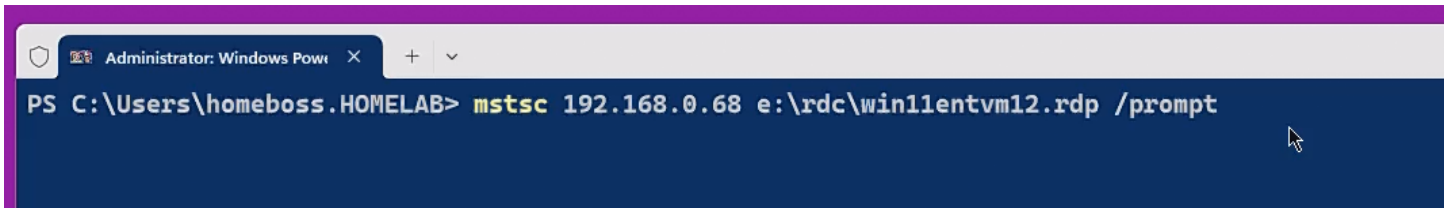


/remoteGuard - Connects your device to a remote device using Remote Guard. Remote Guard prevents credentials from being sent to the remote PC, which can help protect your credentials if you connect to a remote PC that has been compromised. Unlike Restricted Administration mode, Remote Guard also supports connections made from the remote PC by redirecting all requests back to your device.





/prompt - Prompts you for your credentials when you connect to the remote PC.



Application Used for These Options:

/shadow:<sessionID> - Specifies the ID of the session to shadow.

/control - Allows control of the session when shadowing.

/noConsentPrompt - Allows shadowing without user consent.

These options are specifically used with the **Remote Desktop Services Manager** or **Remote Desktop Session Hosts**, typically on **Windows Server** editions. They are designed for administrators to monitor, assist, or take control of user sessions remotely. Here's how you use them:

1. **Remote Desktop Services Manager**: This tool is available on Windows Servers. Administrators can view and manage user sessions connected to the server, including shadowing those sessions using the parameters mentioned.
2. **Command Prompt**: You can also use these parameters directly in the **Command Prompt** by invoking the mstsc.exe (Remote Desktop Client) with these options. For example, you can run the following command to shadow a user session:

```
mstsc /shadow:<sessionID> /control /noConsentPrompt
```

3. **Remote Desktop Connection (RDC)**: While these options aren't available through the regular GUI of Remote Desktop Connection (mstsc.exe), you can still invoke them via the command line, specifically when working with server-based environments.

Use Case Scenario:

These features are particularly useful in **corporate environments** or **data centers** where a server is hosting multiple remote desktop sessions. For instance, an IT admin might use these options to:

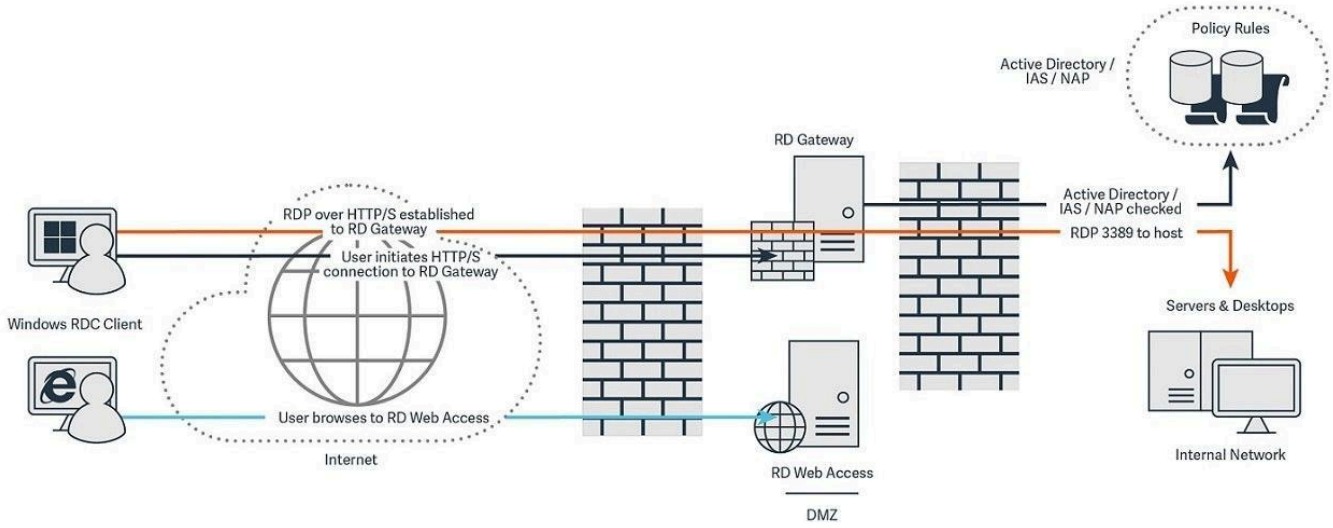
- Monitor a user's session without their input (with `/noConsentPrompt`).
- Take control of a user's session to help with troubleshooting or training (with `/control`).

Summary:

The `/shadow`, `/control`, and `/noConsentPrompt` options are typically used with **Remote Desktop Services** on Windows Server environments. Administrators use them in **Remote Desktop Services Manager** or via **Command Prompt** to shadow, view, and control user sessions running on servers.

```
PS C:\Users\homeboss.HOMELAB> query session /server:win11entvm12
 SESSIONNAME       USERNAME                 ID  STATE   TYPE        DEVICE
 services                                    0  Disc
 console                                     1  Conn
 rdp-tcp#0         John                     10  Active
 31c5ce94259d4...                        65536  Listen
 rdp-tcp                                 65537  Listen
```

mstsc



In this article

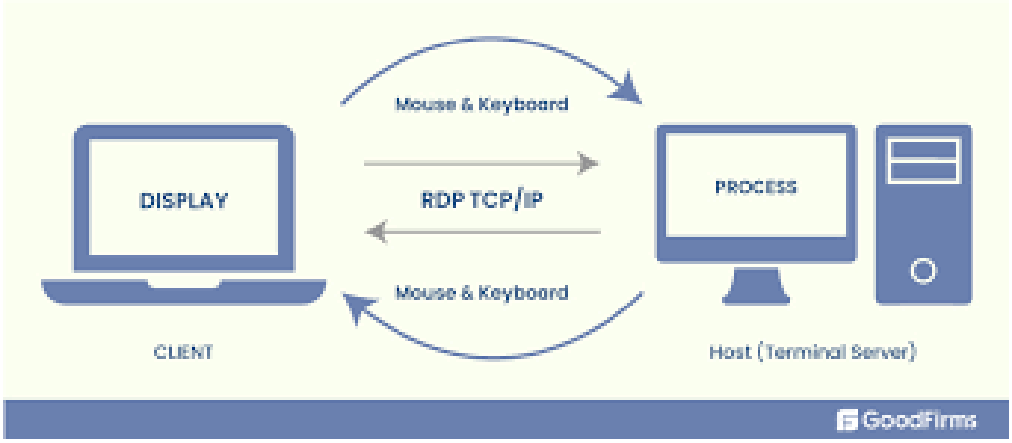
Applies to: Windows Server 2022, Windows Server 2019, Windows Server 2016, Windows Server 2012 R2, Windows Server 2012

Creates connections to Remote Desktop Session Host servers or other remote computers and edits an existing Remote Desktop Connection (.rdp) configuration file.

Syntax

mstsc.exe [<connectionfile>] [/v:<server>[:<port>]] [/g:<gateway>] [/admin] [/f <fullscreen>] [/w:<width> /h:<height>] [/public] [/multimon] [/l] [/restrictedadmin] [/remoteguard] [/prompt] [/shadow:<sessionid>] [/control] [/noconsentprompt]

mstsc.exe /edit <connectionfile>



Parameters

Parameter	Description
<connectionfile>	Specifies the name of an **.rdp** file for the connection.
/v:<server>[:<port>]	Specifies the remote computer and, optionally, the port number to which you want to connect.
/g:<gateway>	Specifies the RD Gateway server to use for the connection. This parameter is only read if the endpoint PC is specified with **/v**.
/admin	Connects you to a session for administering the server.
/f	Starts Remote Desktop Connection in full-screen mode.
/w:<width>	Specifies the width of the Remote Desktop window.
/h:<height>	Specifies the height of the Remote Desktop window.
/public	Runs Remote Desktop in public mode. In public mode, passwords and bitmaps aren't cached.
/multimon	Configures the Remote Desktop Services session monitor layout to be identical to the current client-side configuration.
/l	Enumerates the monitor attached to the local PC and the ID associated with each monitor. The monitor ID can be used to populate the selected monitors RDP file setting.
/edit <connectionfile>	Opens the specified **.rdp** file for editing.
/restrictedAdmin	This mode won't send your credentials to the remote PC, which can protect you if you connect to a compromised device. Connections made

Parameter	Description
	from the remote PC might not be authenticated by other PCs, which impact application functionality and compatibility. The **/admin** parameter is implied.
	These below options are specifically used with the **Remote Desktop Services Manager** or **Remote Desktop Session Hosts**, typically on **Windows Server** editions. They are designed for administrators to monitor, assist, or take control of user sessions remotely
/remoteGuard	This mode prevents credentials from being sent to the remote PC, which can help protect your credentials if you connect to a compromised device. Unlike Restricted Administrator mode, Remote Guard also supports connections made from the remote PC by redirecting all requests back to your device.
/prompt	Prompts you for your credentials when you connect to the remote PC.
/shadow:<sessionID>	Specifies the ID of the session to shadow.
/control	Allows control of the session when shadowing.
/noConsentPrompt	Allows shadowing without user consent.
/?	Displays help at the command prompt.

Remarks

- Default.rdp is stored for each user as a hidden file in the user's **Documents** folder.
- User created **.rdp** files are saved by default in the user's **Documents** folder, but can be saved anywhere.
- To span across monitors, the monitors must use the same resolution and must be aligned horizontally (that is, side-by-side). There's currently no support for spanning multiple monitors vertically on the client system.

Examples

To connect to a session in full-screen mode, type:

```
mstsc /f
```

or

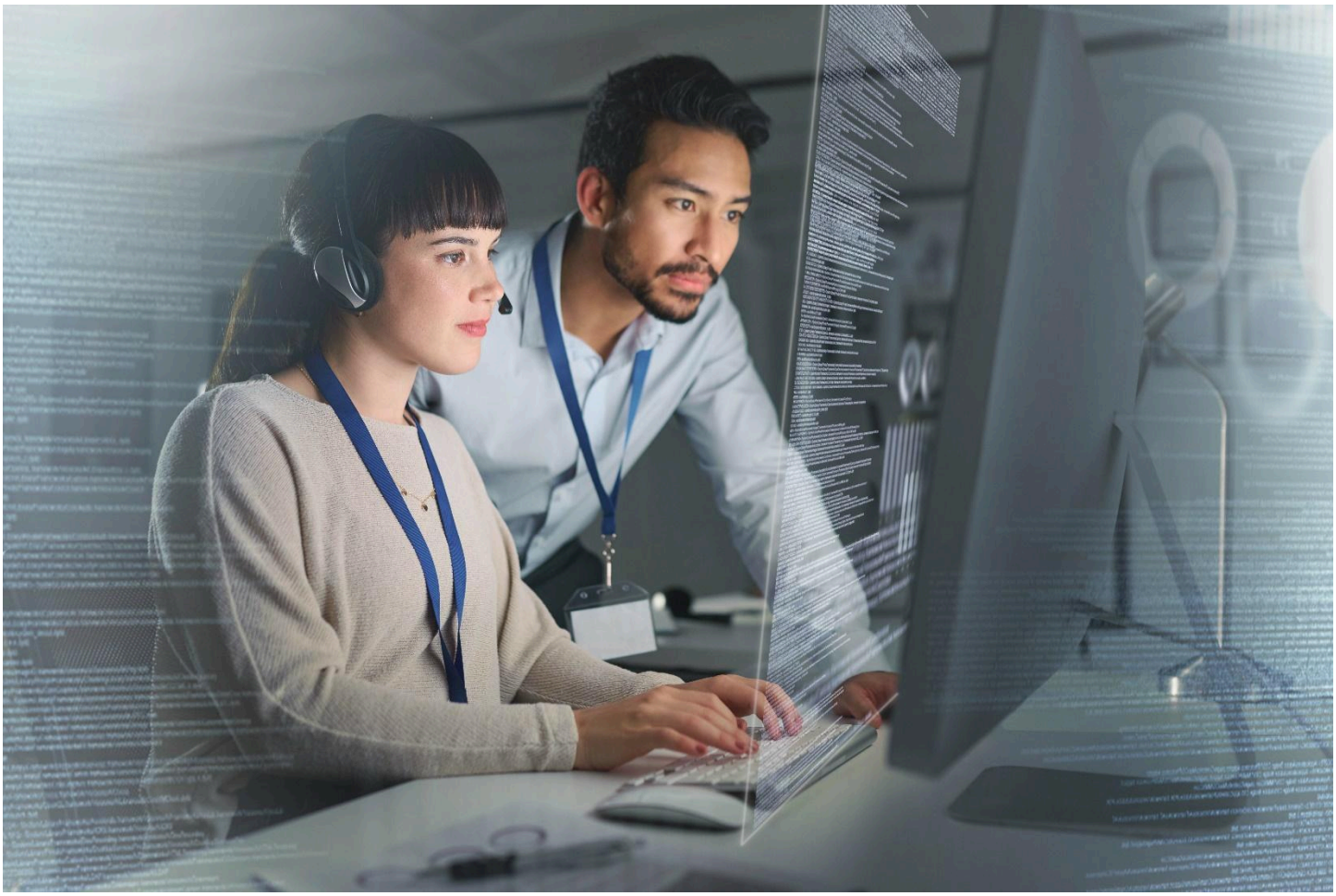
```
mstsc /v:computer1 /f
```

To assign width/height, type:

```
mstsc /v:computer1 /w:1920 /h:1080
```

To open a file called *filename.rdp* for editing, type:

```
mstsc /edit filename.rdp
```



Understanding Remote Desktop Protocol (RDP)

Remote Desktop Protocol (RDP) is the underlying technology that powers Remote Desktop. It's a complex protocol handling everything from user authentication to display rendering. To truly understand RDC, a deep dive into RDP is essential.

- **Microsoft's Official Documentation:** While not the most in-depth, it's a good starting point for understanding the basics of RDP.
    - **Remote Desktop Services Overview:**

How RDC Works and Functions in the Operating System

To grasp how RDC operates within the Windows OS, these resources can be valuable:

RDP Architecture and OS Interaction

RDP operates on a layered architecture, which includes:

1. **Transport Layer**: This handles the transmission of data between the client and server, typically using the TCP/IP protocol for secure and reliable communication.
2. **Core Layer**: This is the heart of RDP, managing the flow of data, including user inputs and graphical output, through various stages such as compression, encryption, and framing. This layer ensures that the data from the server is presented on the client device, and user actions are transmitted back to the server.
3. **Security Layer**: RDP incorporates advanced encryption techniques to secure the data being transmitted, ensuring that sensitive information remains protected during remote sessions

   MS Learn

   NETWORK ENCYCLOPEDIA

Interaction with the Operating System

On the server side, RDS integrates with the Windows OS to manage the remote sessions. The OS handles user authentication, session state management, and resource allocation (e.g., CPU, memory). RDP abstracts much of the underlying network complexity, allowing developers to create applications that leverage remote desktop services without needing to handle the details of network communication MS Learn

RDS components such as the Remote Desktop Gateway (RD Gateway) and RD Web Access further extend the protocol by providing secure, encrypted channels for accessing remote services, even across complex network topologies NETWORK ENCYCLOPEDIA

To learn more about RDP's layered architecture and its interaction with the OS, check out the articles on Microsoft Learn MS Learn and other related resources.

Here are some discussions from Reddit's r/sysadmin and Spiceworks regarding Remote Desktop Protocol (RDP) and Windows kernel issues:

1. **Kernel and RDP Interactions**: Discussions frequently revolve around how kernel-related issues can impact RDP sessions. For instance, faults in KERNELBASE.dll have been known to cause RDP failures, especially in environments using multi-factor authentication like PingID .
2. **Event ID 41 (Microsoft-Windows-Kernel-Power)**: This error is often discussed when diagnosing random shutdowns or failures in RDP connections. It's typically related to power interruptions, but can also affect systems during remote sessions .
3. **Authentication Failures and Kernel Updates**: Kernel updates, particularly those related to security patches (e.g., KB5018419), have been identified as causes for RDP authentication issues, such as blocked connections to Domain Controllers .
4. **Black Screen on RDP Sessions**: Another recurring topic is the "Black Screen of Death" during RDP sessions, which can be traced to driver issues involving the kernel, specifically related to the Remote Desktop Input Driver .

These resources provide insights into the complex interplay between the Windows kernel and RDP, particularly when troubleshooting RDP failures.

Common RDC Issues and Troubleshooting

RDC, while powerful, is prone to issues. Here's where to find solutions:

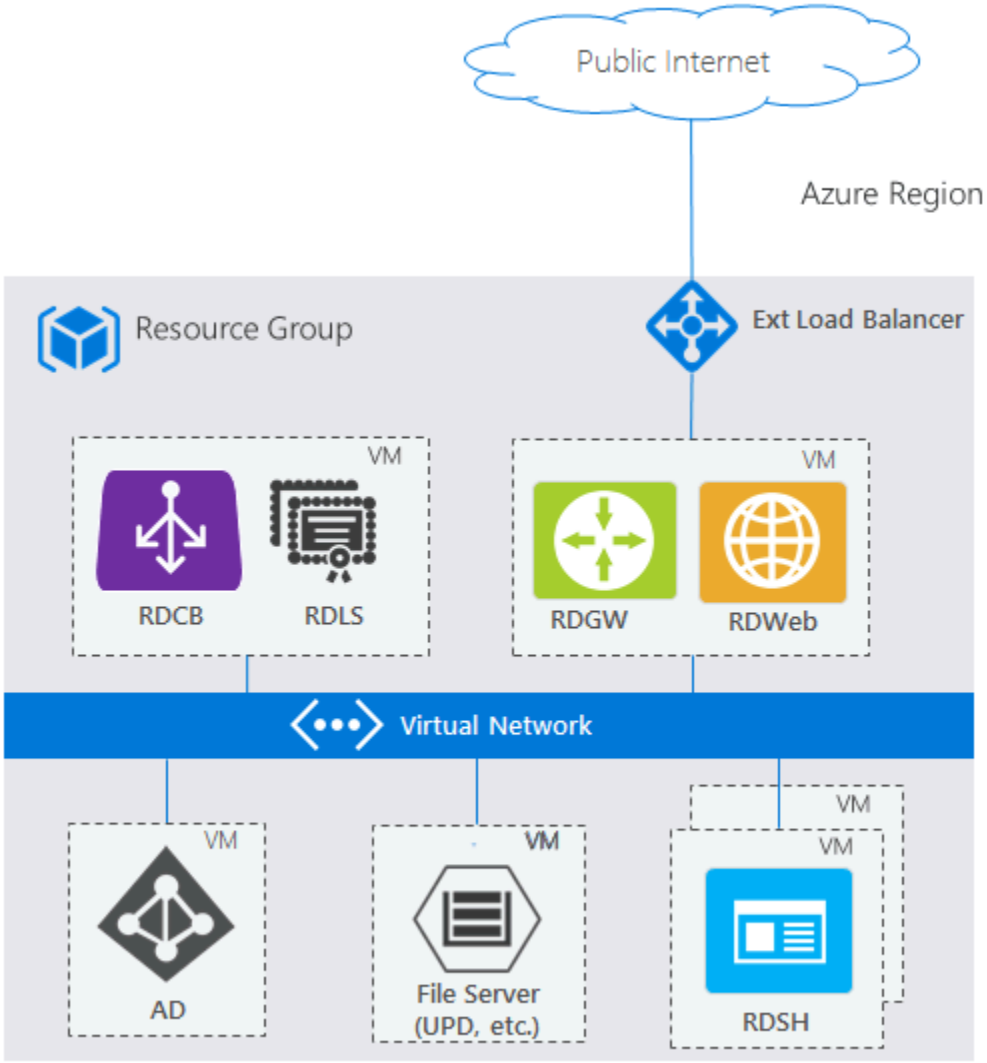
- **Microsoft Support:**
    - Microsoft's support page often has solutions to common RDC problems.
    - **How to use Remote Desktop:** https://support.microsoft.com/en-us/windows/how-to-use-remote-desktop-5fe128d5-8fb1-7a23-3b8a-41e636865e8c

Standard RDS deployment architectures

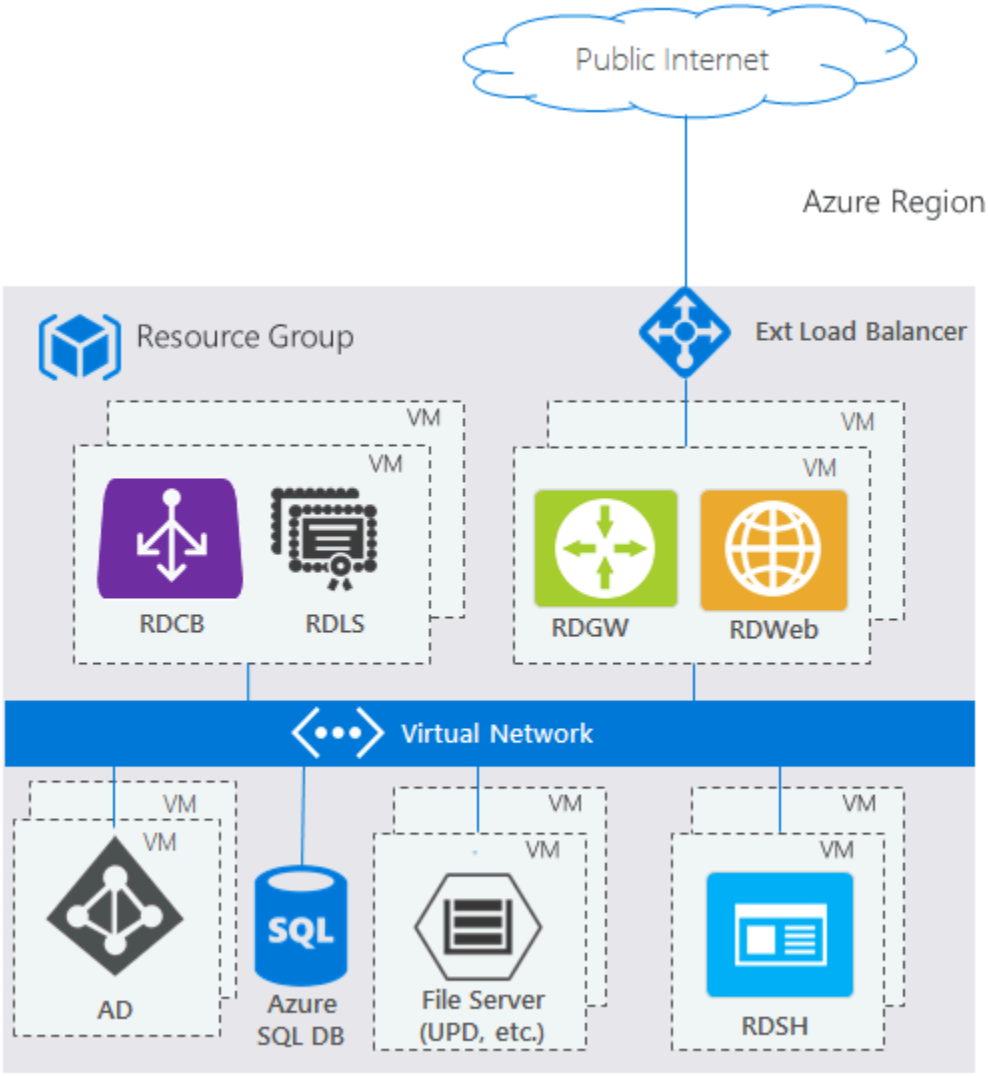
Remote Desktop Services has two standard architectures:

- **Basic deployment** – This contains the minimum number of servers to create a fully effective RDS environment
- **Highly available deployment** – This contains all necessary components to have the highest guaranteed uptime for your RDS environment

Basic deployment



Highly available deployment



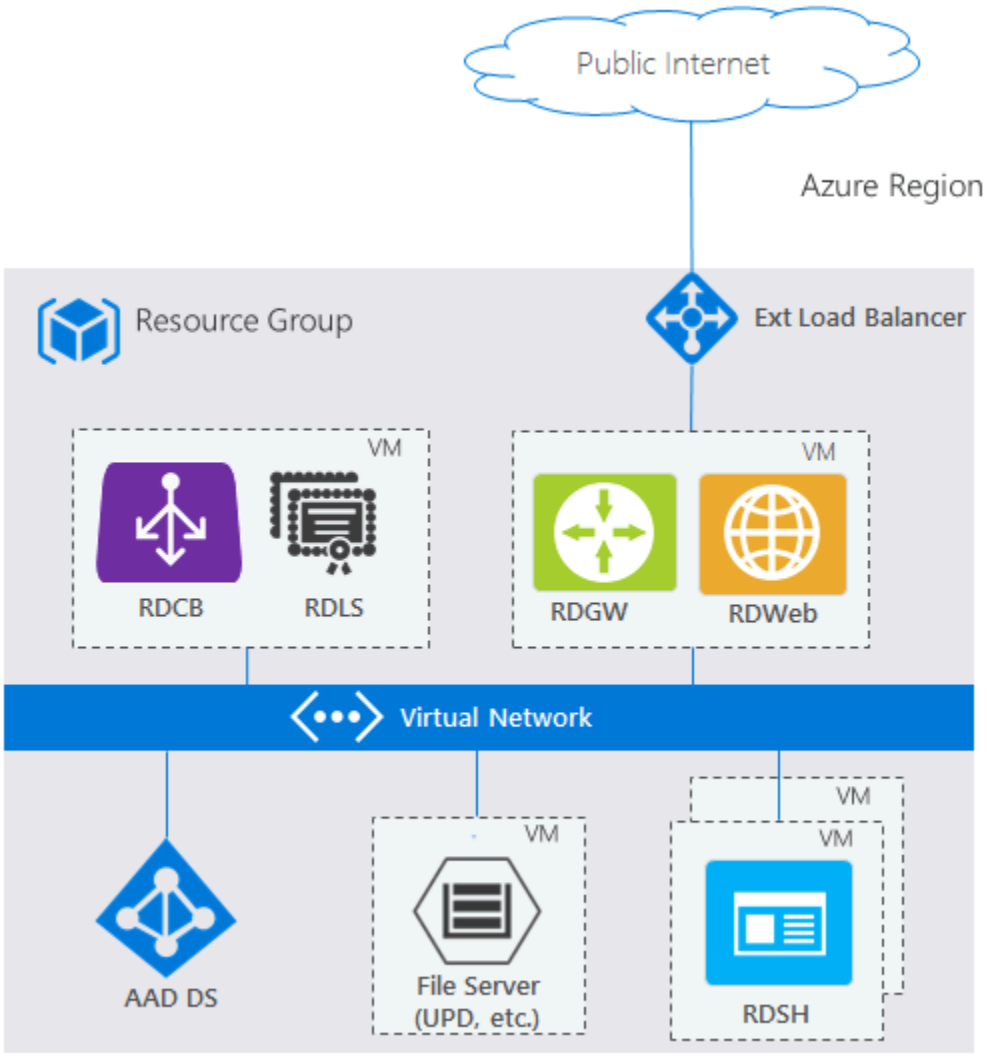
RDS architectures with unique Azure PaaS roles

Though the standard RDS deployment architectures fit most scenarios, Azure continues to invest in first-party PaaS solutions that drive customer value. Below are some architectures showing how they incorporate with RDS.

RDS deployment with Microsoft Entra Domain Services

The two standard architecture diagrams above are based on a traditional Active Directory (AD) deployed on a Windows Server VM. However, if you don't have a traditional AD and only have a Microsoft Entra tenant—through services like Office365—but still want to leverage RDS, you can use Microsoft Entra Domain Services to create a fully managed domain in your Azure IaaS environment that uses the same users that exist in your Microsoft Entra tenant. This removes the

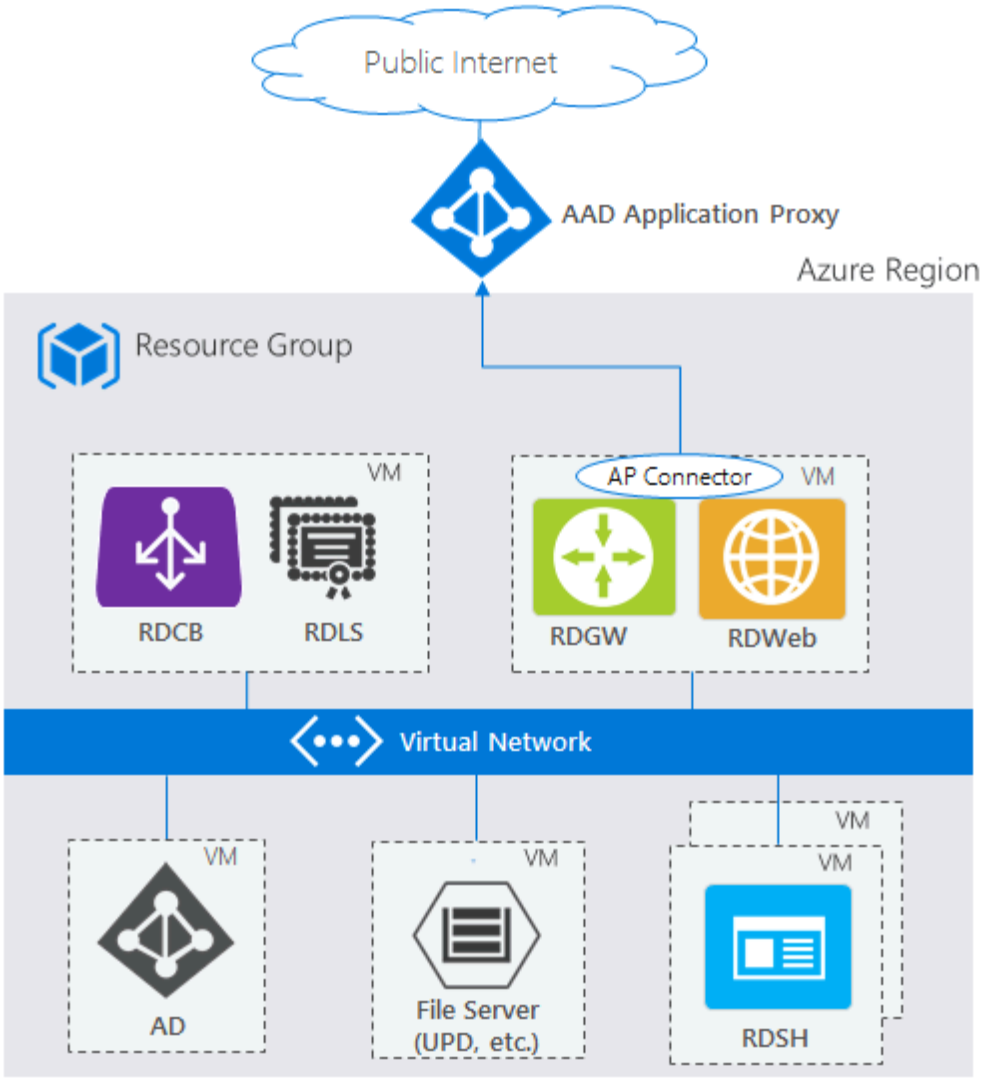
complexity of manually syncing users and managing more virtual machines. Microsoft Entra Domain Services can work in either deployment: basic or highly available.



RDS deployment with Microsoft Entra application proxy

The two standard architecture diagrams above use the RD Web/Gateway servers as the Internet-facing entry point into the RDS system. For some environments, administrators would prefer to remove their own servers from the perimeter and instead use technologies that also provide additional security through reverse proxy technologies. The Microsoft Entra application proxy PaaS role fits nicely with this scenario.

For supported configurations and how to create this setup, see how to publish Remote Desktop with Microsoft Entra application proxy.



Supported configurations for Remote Desktop Services

- 07/03/2024
-

In this article

Applies to: Windows Server 2025, Windows Server 2022, Windows Server 2019, Windows Server 2016

When it comes to supported configurations for Remote Desktop Services environments, the largest concern tends to be version interoperability. Most environments include multiple versions of Windows Server. For example, you may have an existing RDS deployment running an earlier version of Windows Server but want to upgrade to a later version of Windows Server to take advantage of the new features. The question then becomes, which RDS components can work with different versions and which need to be the same?

So with that in mind, here are basic guidelines for supported configurations of Remote Desktop Services in Windows Server.

Note

Make sure to review the system requirements for Windows Server.

Best practices

- Use the most recent version of Windows Server for your Remote Desktop infrastructure (the Web Access, Gateway, Connection Broker, and license server). Windows Server is backward-compatible with these components. Meaning a Windows Server 2022 RD Session Host can connect to a 2025 RD Connection Broker, but not the other way around.

- For RD Session Hosts - all Session Hosts in a collection need to be at the same level, but you can have multiple collections. For example, you can have a collection with Windows Server 2019 Session Hosts and one with Windows Server 2025 Session Hosts.
- An RDS license server can only process client access licenses (CALs) from the same or previous versions of Windows Server. Meaning, if you upgrade your RD Session Host to Windows Server 2025, you also need to upgrade the license server.
- Follow the upgrade order recommended in Upgrading your Remote Desktop Services environment.
- If you are creating a highly available environment, all of your Connection Brokers need to be at the same OS level.

RD Connection Brokers

Starting in Windows Server 2016, there's no restriction for the number of Connection Brokers you can have in a deployment when using Remote Desktop Session Hosts (RDSH) and Remote Desktop Virtualization Hosts (RDVH). The following table shows which versions of RDS components work in a highly available deployment with three or more Connection Brokers.

3+ Connection Brokers in HA	RDSH or RDVH 2025	RDSH or RDVH 2022	RDSH or RDVH 2019	RDSH or RDVH 2016
Windows Server 2025 Connection Broker	Supported	Supported	Supported	Supported
Windows Server 2022 Connection Broker	N/A	Supported	Supported	Supported
Windows Server 2019 Connection Broker	N/A	N/A	Supported	Supported
Windows Server 2016 Connection Broker	N/A	N/A	N/A	Supported

Support for graphics processing unit (GPU) acceleration

Remote Desktop Services support systems equipped with GPUs. Applications that require a GPU can be used over the remote connection. Additionally, GPU-accelerated rendering and encoding can be enabled for improved app performance and scalability.

Remote Desktop Services Session Hosts and single-session client operating systems can take advantage of the physical or virtual GPUs presented to the operating system in many ways, including the Azure GPU optimized virtual machine sizes, GPUs available to the physical RDSH server, and GPUs presented to the VMs by supported hypervisors.

See Which graphics virtualization technology is right for you? for help figuring out what you need. For specific information about DDA, check out Plan for deploying Discrete Device Assignment.

GPU vendors may have a separate licensing scheme for RDSH scenarios or restrict GPU use on the server OS, verify the requirements with your favorite vendor.

GPUs presented by a non-Microsoft hypervisor or Cloud Platform must have drivers digitally-signed by WHQL and supplied by the GPU vendor.

Remote Desktop Session Host support for GPUs

The following table shows the scenarios supported by different versions of RDSH hosts.

Feature	**Windows Server 2016**	**Windows Server 2019**	**Windows Server 2022**	**Windows Server 2025**
Use of hardware GPU for all RDP sessions	Yes	Yes	Yes	Yes
H.264/AVC hardware encoding (if supported by the GPU)	Yes	Yes	Yes	Yes
Load balancing between multiple GPUs presented to the OS	No	Yes	Yes	Yes
H.264/AVC encoding optimizations for minimizing bandwidth usage	No	Yes	Yes	Yes
H.264/AVC support for 4K resolution	No	Yes	Yes	Yes

VDI support for GPUs

The following table shows support for GPU scenarios in the client OS.

Feature	Windows 7 SP1	Windows 8.1	Windows 10
Use of hardware GPU for all RDP sessions	No	Yes	Yes
H.264/AVC hardware encoding (if supported by the GPU)	No	No	Windows 10 1703 or later
Load balancing between multiple GPUs presented to the OS	No	No	Windows 10 1803 or later
H.264/AVC encoding optimizations for minimizing bandwidth usage	No	No	Windows 10 1803 or later
H.264/AVC support for 4K resolution	No	No	Windows 10 1803 or later

RemoteFX 3D Video Adapter (vGPU) support

Note

Because of security concerns, RemoteFX vGPU is disabled by default on all versions of Windows starting with the July 14, 2020 Security Update and removed starting with the April 13, 2021 Security Update. To learn more, see KB 4570006.

Remote Desktop Services supports RemoteFX vGPUs when VM is running as a Hyper-V guest on Windows Server. The following guest operating systems have RemoteFX vGPU support:

- Windows 11
- Windows 10
- Windows Server in a single-session deployment only

Discrete Device Assignment support

Remote Desktop Services supports Physical GPUs presented with Discrete Device Assignment from Hyper-V hosts running Windows Server 2016 or later. See Plan for deploying Discrete Device Assignment for more details.

VDI deployment – supported guest operating systems

Windows Server RD Virtualization Host servers support the following guest operating systems:

- Windows 11 Enterprise

- Windows 10 Enterprise

Note

- Remote Desktop Services doesn't support heterogeneous session collections. The OSes of all VMs in a collection must be the same version.
- You can have separate homogeneous collections with different guest OS versions on the same host.
- The Hyper-V host used to run VMs must be the same version as the Hyper-V host used to create the original VM templates.

Single sign-on

RDS in Windows Server supports two main SSO experiences:

- In-app (Remote Desktop application on Windows, iOS, Android, and Mac)
- Web SSO

Using the Remote Desktop application, you can store credentials either as part of the connection info (Mac) or as part of managed accounts (iOS, Android, Windows) securely through the mechanisms unique to each OS.

To connect to desktops and RemoteApps with SSO through the inbox Remote Desktop Connection client on Windows, you must connect to the RD Web page through Internet Explorer. The following configuration options are required on the server side. No other configurations are supported for Web SSO:

- RD Web set to Forms-Based Authentication (Default)
- RD Gateway set to Password Authentication (Default)
- RDS Deployment set to "Use RD Gateway credentials for remote computers" (Default) in the RD Gateway properties

Note

Due to the required configuration options, Web SSO is not supported with smartcards. Users who login via smartcards might face multiple prompts to login.

For more information about creating VDI deployment of Remote Desktop Services, check out Supported Windows 10 security configurations for Remote Desktop Services VDI.

Using Remote Desktop Services with application proxy services

You can use Remote Desktop Services with Microsoft Entra application proxy. Remote Desktop Services doesn't support using Web Application Proxy.

Supported Windows security configurations for Remote Desktop Services VDI

- Article
- 07/03/2024
-

In this article

1. Remote Credential Guard
2. Shielded VMs and Encryption Supported VMs

Applies to: Windows Server 2025, Windows Server 2022, Windows Server 2019, Windows Server 2016

Windows and Windows Server have new layers of protection built into the operating system to:

- Safeguard against security breaches
- Help block malicious attacks
- Enhance the security of virtual machines, applications, and data.

Note

- Features like Credential Guard may have performance implication on user density. Ensure to test your scenarios. Learn more about other considerations for credential guard configuration.
- Make sure to review the Remote Desktop Services supported configuration information.

The following table outlines which of these new features are supported in a VDI deployment using RDS.

VDI collection type	Managed pooled	Managed personal	Unmanaged pooled	Unmanaged personal
Credential Guard	Yes	Yes	Yes	Yes
Device Guard	Yes	Yes	Yes	Yes
Remote Credential Guard	No	No	No	No
Shielded & Encryption Supported VMs	No	No	Encryption supported VMs with extra configuration	Encryption supported VMs with extra configuration

Remote Credential Guard

Remote Credential Guard is only supported for direct connections to the target machines and not for the ones via Remote Desktop Connection Broker and Remote Desktop Gateway.

Note

If you have a Connection Broker in a single-instance environment, and the DNS name matches the computer name, you may be able to use Remote Credential Guard, although this isn't supported.

Shielded VMs and Encryption Supported VMs

Shielded VMs aren't supported in Remote Desktop Services VDI.

For leveraging Encryption Supported VMs:

- Use an unmanaged collection and a provisioning technology outside of the Remote Desktop Services collection creation process to provision the virtual machines.
- User Profile Disks aren't supported as they rely on differential disks

Plan and design your Remote Desktop Services environment

-

A highly scalable Remote Desktop deployment requires the use of specific patterns and practices. Designing for optimal performance and scale-out is key. Use the scenarios below to help you envision, architect, and continually refine your deployment.

Use the following information to plan and design your deployment:

- Build anywhere
- Network guidance
- Access from anywhere
- High availability
- MultiFactor Authentication
- Secure data storage
- GPU acceleration
- Connect from any device
- Choose how you pay

Be sure to also review the Desktop Hosting Reference Architecture, which provides an overview of the Remote Desktop architecture and helps you plan a hybrid RDS environment that includes Azure infrastructure.

Terminal Services Overview

Terminal Services in Windows refers to the support for multiple interactive user sessions on a single machine. This system was introduced to enable users to remotely access a desktop environment or applications running on a Windows server or client system. Terminal Services transmits the graphical user interface (GUI) of an application or desktop to a remote user, while the remote user's input is sent back to the server for processing.

How Processes Work with Remote Users

Windows uses **Terminal Services Sessions** to manage user interactions. Each user who connects to a terminal server via Remote Desktop or Fast User Switching is assigned a session with a unique ID. Sessions start from **Session 0**, which is reserved for system processes, including services. Interactive users (whether local or remote) are assigned **Session 1** and higher, depending on the order of their login.

When a user logs into a Windows machine remotely, the terminal server creates a new session for the user. The GUI is transmitted to the client device, while input (keyboard, mouse, etc.) is sent back to the server. Only one user session can be active and visible at a time, either the local console session or a remote session.

For each session, Windows creates its own **Window Station**, which manages input, output, and window handling. Only one window station per session, typically called **WinSta0**, can be associated with the GUI (i.e., capable of displaying user interfaces and receiving input). Each window station contains **desktops**, and desktops contain the visual elements of the session (windows, icons, etc.).

The Role of Desktops

In Windows, a **desktop** is a logical surface where user interface components (like windows and dialogs) are displayed. Each session's window station contains one or more desktops, with only one visible at a time.

For example, within a session, desktops might include **the Default desktop**, **the Screen-saver desktop**, and the Winlogon desktop (which is displayed during the Ctrl+Alt+Delete sequence).

Desktops play an important role in separating user interfaces for security reasons. For example, even within the same session, the Winlogon desktop is restricted to system-level processes and cannot be accessed by user applications, ensuring that sensitive actions like password input are handled securely.

Windows Handling of Remote vs. Local Sessions

There is a distinction between **local sessions** (where a user is physically at the machine) and **remote sessions** (where the user is accessing the machine via a network).

- Local sessions use the console session, which is tied to the physical monitor, keyboard, and mouse attached to the machine.
- Remote sessions are handled through the terminal services infrastructure, where a virtual session is created, and input/output is redirected over the network to the remote client.

Windows Server systems are optimized to handle multiple remote sessions concurrently, while Windows client systems (such as Windows 10) typically only allow one user to be logged in at a time, whether local or remote. When a remote user logs in, the local console session is typically locked, and vice versa.

In contrast, Windows Server systems can be configured to support multiple simultaneous remote sessions, enabling multiple users to access the system remotely for different tasks.

Troubleshooting Terminal Services

Several common problems arise with Terminal Services, especially concerning connectivity, session management, and resource allocation. Here are some troubleshooting steps that can be useful:

1. **Session Management**:
    - Ensure that the system is not running out of available session slots. Windows Server, by default, limits the number of remote connections, and exceeding this limit can prevent new users from logging in. Tools like Task Manager or Sysinternals' **Process Explorer** can display the current sessions and their IDs.
2. **Network Connectivity**:
    - Verify that the Remote Desktop Protocol (RDP) service is running and that firewalls are not blocking the RDP port (typically port 3389). Also, check for any network issues between the client and the server.
3. **Session Disconnects and Crashes**:
    - If users are frequently disconnected or experience crashes, it could be due to session resource exhaustion. Check for high CPU or memory usage within the terminal session using **Resource Monitor** or **Task Manager**. Memory leaks in applications running within the session can also cause these issues.
4. **Session 0 Isolation**:

- Since Windows Vista, **Session 0 Isolation** has been implemented to enhance security. This means services running in Session 0 cannot interact with the user's desktop, which can lead to issues where legacy services try to display UI elements that are no longer visible to the user. In these cases, the **Interactive Services Detection Service** can help mitigate the problem by alerting the user to these hidden dialogs.

5. **Application Compatibility**:
    - Applications that were not designed to run in multi-user environments can sometimes conflict with terminal services. Issues like file locking or improper resource management are common in these cases. Testing applications for compatibility with Terminal Services should be part of any deployment process.
6. **Logon Sessions and Token Issues**:
    - Use Sysinternals' **LogonSessions** utility to list all active logon sessions and their associated processes. This can help identify issues where sessions are not terminating correctly or where token permissions are causing conflicts.

Conclusion

Terminal Services is a critical component of the Windows operating system that enables remote access and multi-user capabilities. It uses sessions, window stations, and desktops to separate and manage resources securely for each user. Troubleshooting Terminal Services often involves session management, network troubleshooting, and dealing with specific application behaviors that arise in a multi-user environment.

Troubleshoot Remote desktop disconnected errors

In this article

This article helps you understand the most common settings that are used to establish a Remote Desktop session in an enterprise environment, and provides troubleshooting information for Remote desktop disconnected errors.

Original KB number: 2477176

Note

This article is intended for use by support agents and IT professionals.

Remote Desktop Server

A Remote Desktop Session Host server is the server that hosts Windows-based programs or the full Windows desktop for Remote Desktop Services clients. Users can connect to an RD Session Host server to run programs, to save files, and to use network resources on that server. Users can access an RD Session Host server from within a corporate network or from the Internet.

Remote Desktop Session Host (RD Session Host) was formerly known as the Remote Desktop server role service, and Remote Desktop Session Host (RD Session Host) server was formerly known as Remote Desktop server.

Remote connections for administration

Remote Desktop supports two concurrent remote connections to the computer. You do not have to have Remote Desktop Services client access licenses (RDS CALs) for these connections.

To allow more than two administrative connections or multiple user connections, you must install the RD Session Host Role and have appropriate RDS CALs.

Symptom 1: Limited Remote Desktop session or Remote Desktop Services session connections

When you try to make a Remote Desktop Connection (RDC) to a remote computer or to a Remote Desktop server (Terminal Server) that is running Windows Server 2008 R2, you receive one of the following error messages:

Remote Desktop Disconnected.
This computer can't connect to the remote computer.
Try connecting again. If the problem continues, contact the owner of the remote computer or your network administrator.

Also, you are limited in the number of users who can connect simultaneously to a Remote Desktop session or Remote Desktop Services session. A limited number of RDP connections can be caused by misconfigured Group Policy or RDP-TCP properties in Remote Desktop Services Configuration. By default, the connection is configured to allow an unlimited number of sessions to connect to the server.

Symptom 2: Port assignment conflict

You experience a port assignment conflict. This problem might indicate that another application on the Remote Desktop server is using the same TCP port as the Remote Desktop Protocol (RDP). The default port assigned to RDP is 3389.

Symptom 3: Incorrectly configured authentication and encryption settings

After a Remote Desktop server client loses the connection to a Remote Desktop server, you experience one of the following symptoms:

- You cannot make a connection by using RDP.
- The session on the Remote Desktop server does not transition to a disconnected state. Instead, it remains active even though the client is physically disconnected from the Remote Desktop server.

If the client logs back in to the same Remote Desktop server, a new session may be established, and the original session may remain active.

Also, you receive one of the following error messages:

- Error message 1

Because of a security error, the client could not connect to the Terminal server. After making sure that you are logged on to the network, try connecting to the server again.

- Error message 2

Remote desktop disconnected. Because of a security error, the client could not connect to the remote computer. Verify that you are logged onto the network and then try connecting again.

Symptom 4: License certificate corruption

Remote Desktop Services clients are repeatedly denied access to the Remote Desktop server. If you are using a Remote Desktop Services client to log on to the Remote Desktop server, you may receive one of the following error messages.

- Error message 1

Because of a security error, the client could not connect to the Terminal server. After making sure that you are logged on to the network, try connecting to the server again.

- Error message 2

Remote desktop disconnected. Because of a security error, the client could not connect to the remote computer. Verify that you are logged onto the network and then try connecting again.

- Error message 3

Because of a security error, the client could not connect to the Terminal server. After making sure that you are logged on to the network, try connecting to the server again.
Remote desktop disconnected. Because of a security error, the client could not connect to the remote computer. Verify that you are logged onto the network and then try connecting again.

Additionally, the following event ID messages may be logged in Event Viewer on the Remote Desktop server.

- Event message 1

Output

- Event ID: 50

Event Source: TermDD

Event Description: The RDP protocol component X.224 detected an error in the protocol stream and has disconnected the client.

- Event message 2

Output

- Event ID: 1088

Event Source: TermService

Event Description: The terminal services licensing grace period has expired and the service has not registered with a license server. A terminal services license server is required for continuous operation. A terminal server can operate without a license server for 90 days after initial start up.

- Event message 3

Output

- Event ID: 1004

Event Source: TermService

Event Description: The terminal server cannot issue a client license.

- Event message 4

Output

- Event ID: 1010

Event Source: TermService

Event Description: The terminal services could not locate a license server. Confirm that all license servers on the network are registered in WINS/DNS, accepting network requests, and the Terminal Services Licensing Service is running.

- Event message 5

Output

- Event ID: 28
- Event Source: TermServLicensing
- Event Description: Terminal Services Licensing can only be run on Domain Controllers or Server in a Workgroup. See Terminal Server Licensing help topic for more information.
-

Resolution for Symptom 1

To resolve this problem, use the following methods, as appropriate.

Verify Remote Desktop is enabled

1. Open the **System** item in Control Panel. To start the System tool, click Start, click **Control Panel**, click **System**, and then click **OK**.
2. Under **Control Panel Home**, click **Remote settings**.
3. Click the **Remote** tab.
4. Under **Remote Desktop**, select either of the available options, depending on your security requirements:
    - **Allow connections from computers from computers running any version of Remote Desktop (less secure)**
    - **Allow connections from computers only from computers running Remote Desktop with Network Level Authentication (more secure)**

If you select **Don't allow connections to this computer** on the **Remote** tab, no users will be able to connect remotely to this computer, even if they are members of the Remote Desktop Users group.

Verify Remote Desktop Services Limit number of connections policy

1. Start the Group Policy snap-in, and then open the Local Security Policy or the appropriate Group Policy.
2. Locate the following command:

Local Computer Policy > **Computer Configuration** > **Administrative Templates** > **Windows Components** > **Remote Desktop Services** > **Remote Desktop Session Host** > **Connections Limit number of connections**

3. Click **Enabled**.
4. In the **RD Maximum Connections allowed** box, type the maximum number of connections that you want to allow, and then click **OK**.

Verify Remote Desktop Services RDP-TCP properties

Follow these steps, depending on your operating system version.

Setting via Remote Desktop Services Configuration

Configure the number of simultaneous remote connections allowed for a connection:

1. On the RD Session Host server, open Remote Desktop Session Host Configuration. To open Remote Desktop Session Host Configuration, click **Start**, point to **Administrative Tools**, point to **Remote Desktop Services**.

2. Under **Connections**, right-click the name of the connection, and then click **Properties**.
3. On the **Network Adapter** tab, click **Maximum connections**, enter the number of simultaneous remote connections that you want to allow for the connection, and then click **OK**.
4. If the **Maximum connections** option is selected and dimmed, the **Limit number of connections** Group Policy setting has been enabled and has been applied to the RD Session Host server.

Verify Remote Desktop Services Logon rights

Configure the Remote Desktop Users Group.

The Remote Desktop Users group on an RD Session Host server grants users and groups permission to remotely connect to an RD Session Host server. You can add users and groups to the Remote Desktop Users group by using the following tools:

- Local Users and Groups snap-in
- The **Remote** tab in the **System Properties** dialog box on an RD Session Host server
- Active Directory Users and Computers snap-in, if the RD Session Host server is installed on a domain controller

You can use the following procedure to add users and groups to the Remote Desktop Users group by using the Remote tab in the System Properties dialog box on an RD Session Host server.

Membership in the local Administrators group, or equivalent, on the RD Session Host server that you plan to configure, is the minimum required to complete this procedure.

Add users and groups to the Remote Desktop Users group by using the Remote tab

1. Start the System tool. To do this, click Start, click Control Panel, click the **System** icon, and then click **OK**.
2. Under **Control Panel Home**, click **Remote settings**.
3. On the **Remote** tab in the **System Properties** dialog box, click **Select Users**. Add the users or groups that have to connect to the RD Session Host server by using Remote Desktop.

Note

If you select the **Don't allow connections to this computer** option on the **Remote** tab, no users will be able to connect remotely to this computer, even if they are members of the Remote Desktop Users group.

Add users and groups to the Remote Desktop Users group by using Local Users and Groups snap-in

1. Click **Start**, click **Administrative Tools**, and then click **Computer Management**.
2. In the console tree, click the **Local Users and Groups** node.
3. In the details pane, double-click the **Groups** folder.
4. Double-click **Remote Desktop Users**, and then click **Add**.
5. In the **Select Users** dialog box, click **Locations** to specify the search location.
6. Click **Object Types** to specify the types of objects that you want to search for.
7. In the **Enter the object names to select (examples)** box, type the name you want to add.
8. Click **Check Names**.
9. When the name is located, click **OK**.

Note

- You can't connect to a computer that's asleep or hibernating, so make sure the settings for sleep and hibernation on the remote computer are set to Never. (Hibernation isn't available on all computers.) For information about making those changes, see Change, create, or delete a power plan (scheme).
- You can't use Remote Desktop Connection to connect to a computer using Windows 7 Starter, Windows 7 Home Basic, or Windows 7 Home Premium.
- Members of the local Administrators group can connect even if they are not listed.

Resolution for Symptom 2

Important

This section, method, or task contains steps that tell you how to modify the registry. However, serious problems might occur if you modify the registry incorrectly. Therefore, make sure that you follow these steps carefully. For added protection, back up the registry before you modify it. Then, you can restore the registry if a problem occurs. For more information about how to back up and restore the registry, see How to back up and restore the registry in Windows.

To resolve this problem, determine which application is using the same port as RDP. If the port assignment for that application cannot be changed, change the port assigned to RDP by changing the registry. After you change the registry, you must restart the Remote Desktop Services service. After you restart the Remote Desktop Services service, you should verify that the RDP port has been changed correctly.

Remote Desktop server listener availability

The listener component runs on the Remote Desktop server and is responsible for listening for and accepting new Remote Desktop Protocol (RDP) client connections, thereby allowing users to establish new remote sessions on the Remote Desktop server. There is a listener for each Remote Desktop Services connection that exists on the Remote Desktop server. Connections can be created and configured by using the Remote Desktop Services Configuration tool.

To perform these tasks, refer to the following sections.

Determine which application is using the same port as RDP

You can run the netstat tool to determine whether port 3389 (or the assigned RDP port) is being used by another application on the Remote Desktop server:

1. On the Remote Desktop server, click **Start**, click **Run**, type *cmd*, and then click **OK**.
2. At the command prompt, type netstat -a -o and then press Enter.
3. Look for an entry for TCP port 3389 (or the assigned RDP port) with a status of **Listening**. This indicates another application is using this port. The PID (Process Identifier) of the process or service using that port appears under the PID column.

To determine which application is using port 3389 (or the assigned RDP port), use the tasklist command-line tool along with the PID information from the netstat tool:

1. On the Remote Desktop server, click **Start**, click **Run**, type *cmd*, and then click **OK**.
2. Type tasklist /svc and then press Enter.
3. Look for an entry for the PID number that is associated with the port (from the netstat output). The services or processes that are associated with that PID appear on the right.

Change the port assigned to RDP

You should determine whether this application can use a different port. If you cannot change the application's port, you must change the port that is assigned to RDP.

Important

We recommend that you do not change the port that is assigned to RDP.

If you have to change the port assigned to RDP, you must change the registry. To do this, you must be a member of the local Administrators group, or you must have been granted the appropriate permissions.

To change the port that is assigned to RDP, follow these steps:

1. On the Remote Desktop server, open Registry Editor. To open Registry Editor, click **Start**, click **Run**, type *regedit*, and then click **OK**.
2. If the **User Account Control** dialog box appears, verify that the action it displays is what you want, and then click **Continue**.
3. Locate and then click the following registry subkey:

HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Remote Desktop server\WinStations

RDP-TCP is the default connection name. To change the port for a specific connection on the Remote Desktop server, select the connection under the **WinStations** key:

1. In the details pane, double-click the **PortNumber** registry entry.
2. Type the port number that you want to assign to RDP.
3. Click **OK** to save the change, and then close Registry Editor.

Restart the Remote Desktop Services service

For the RDP port assignment change to take effect, stop and start the Remote Desktop Services service. To do this, you must be a member of the local Administrators group, or you must have been granted the appropriate permissions.

To stop and start the Remote Desktop Services service, follow these steps:

1. On the Remote Desktop server, open the Services snap-in. To do this, click **Start**, point to **Administrative Tools**, and then click **Services**.
2. If the **User Account Control** dialog box appears, verify that the action it displays is what you want, and then click **Continue**.
3. In the **Services** pane, right-click **Remote Desktop Services**, and then click **Restart**.
4. If you are prompted to restart other services, click **Yes**.
5. Verify that the **Status** column for the Remote Desktop Services service displays a **Started** status.

Verify that the RDP port has changed

To verify that the RDP port assignment has been changed, use the netstat tool:

1. On the Remote Desktop server, click **Start**, click **Run**, type *cmd*, and then click **OK**.
2. At the command prompt, type netstat -a then press Enter.
3. Look for an entry for the port number that you assigned to RDP. The port should appear in the list and have a status of **Listening**.

Important

Remote Desktop Connection and the Terminal server Web Client use port 3389, by default, to connect to a Remote Desktop server. If you change the RDP port on the Remote Desktop server, you will have to modify the port used by Remote Desktop Connection and the Remote Desktop server Web Client. For more information, see Change the listening port for Remote Desktop on your computer.

Verify that the listener on the Remote Desktop server is working

To verify that the listener on the Remote Desktop server is working correctly, use any of the following methods.

Note

RDP-TCP is the default connection name and 3389 is the default RDP port. Use the connection name and port number specific to your Remote Desktop server configuration.

- Method 1

Use an RDP client, such as Remote Desktop Connection, to establish a remote connection to the Remote Desktop server.

- Method 2

Use the qwinsta tool to view the listener status on the Remote Desktop server:

1. On the Remote Desktop server, click **Start**, click **Run**, type *cmd*, and then click **OK**.
2. At the command prompt, type *qwinsta*, and then press Enter.
3. The RDP-TCP session state should be **Listen**.

- Method 3

Use the netstat tool to view the listener status on the Remote Desktop server:

1. On the Remote Desktop server, click **Start**, click **Run**, type *cmd*, and then click **OK**.
2. At the command prompt, type netstat -a then press Enter.
3. The entry for TCP port 3389 should be **Listening**.

- Method 4

Use the telnet tool to connect to the RDP port on the Remote Desktop server:

1. From another computer, click **Start**, click **Run**, type *cmd*, and then click **OK**.
2. At the command prompt, type telnet <servername> 3389 , where <servername> is the name of the Remote Desktop server, and then press Enter.

If telnet is successful, you receive the telnet screen and a cursor.

If telnet is not successful, you receive the following error message:

Connecting To **servername**... Could not open connection to the host, on port 3389: Connect failed

The qwinsta, netstat, and telnet tools are also included in Windows XP and Windows Server 2003. You can also download and use other troubleshooting tools, such as Portqry.

Resolution for Symptom 3

To resolve the issue, configure authentication and encryption.

To configure authentication and encryption for a connection, follow these steps:

1. On the RD Session Host server, open Remote Desktop Session Host Configuration. To open Remote Desktop Session Host Configuration, click **Start**, point to **Administrative Tools**, point to **Remote Desktop Services**, and then click **Remote Desktop Session Host Configuration**.
2. Under **Connections**, right-click the name of the connection, and then click **Properties**.
3. In the **Properties** dialog box for the connection, on the **General** tab, in **Security layer**, select a security method.
4. In **Encryption level**, click the level that you want. You can select **Low, Client Compatible, High, or FIPS Compliant**. See Step 4 above for **Windows Server 2003** for **Security layer** and **Encryption level** options.

Note

- To perform this procedure, you must be a member of the Administrators group on the local computer, or you must have been delegated the appropriate authority. If the computer is joined to a domain, members of the Domain Admins group might be able to perform this procedure. As a security best practice, consider using Run as to perform this procedure.
- To open Remote Desktop Services Configuration, click **Start**, click **Control Panel**, double-click **Administrative Tools**, and then double-click **Remote Desktop Services Configuration**.
- Any encryption level settings that you configure in Group Policy override the configuration that you set by using the Remote Desktop Services Configuration tool. Also, if you enable the System cryptography: Use FIPS compliant algorithms for encryption, hashing, and signing Group Policy setting, this setting overrides the **Set client connection encryption level** Group Policy setting.
- When you change the encryption level, the new encryption level takes effect the next time a user logs on. If you require multiple levels of encryption on one server, install multiple network adapters and configure each adapter separately.
- To verify that certificate has a corresponding private key, in Remote Desktop Services Configuration, right-click the connection for which you want to view the certificate, click the **General** tab, click **Edit**, click the certificate that you want to view, and then click **View Certificate**. At the bottom of the **General** tab, the statement, **You have a private key that corresponds to this certificate**, should appear. You can also view this information by using the Certificates snap-in.
- The **FIPS compliant** setting (the **System cryptography: Use FIPS compliant algorithms for encryption, hashing, and signing** setting in Group Policy or the **FIPS Compliant** setting in Remote Desktop server Configuration) encrypts and decrypts data sent from the client to the server and from the server to the client, with the Federal Information Processing Standard (FIPS) 140-1 encryption algorithms, using Microsoft cryptographic modules. For more information, see Terminal Services in Windows Server 2003 Technical Reference.
- The **High** setting encrypts data sent from the client to the server and from the server to the client by using strong 128-bit encryption.
- The **Client Compatible** setting encrypts data sent between the client and the server at the maximum key strength supported by the client.
- The Low setting encrypts data sent from the client to the server using 56-bit encryption.

Additional troubleshooting step: Enable CAPI2 event logs

To help troubleshoot this problem, enable CAPI2 event logs on both the client and server computers. This command is shown in the following screenshot.

Workaround for the issue (You cannot completely disconnect a Remote Desktop server connection) described in Symptom 3

To work around this problem, follow these steps:

1. Click **Start**, click **Run**, type *gpedit.msc*, and then click **OK**.
2. Expand **Computer Configuration**, expand **Administrative Templates**, expand **Windows Components**, expand **Remote Desktop Services**, expand **Remote Desktop Session Host**, and then click **Connections**.
3. In the right pane, double-click **Configure keep-alive connection interval**.
4. Click **Enabled**, and then click **OK**.
5. Close Group Policy Object Editor, click **OK**, and then quit Active Directory Users and Computers.

Resolution for Symptom 4

Important

This section, method, or task contains steps that tell you how to modify the registry. However, serious problems might occur if you modify the registry incorrectly. Therefore, make sure that you follow these steps carefully. For added protection, back up the registry before you modify it. Then, you can restore the registry if a problem occurs. For more information about how to back up and restore the registry, see 322756 How to back up and restore the registry in Windows.

To resolve this problem, back up and then remove the **X509 Certificate** registry keys, restart the computer, and then reactivate the Remote Desktop Services Licensing server. To do this, follow these steps.

Note

Perform the following procedure on each of the Remote Desktop servers.

1. Make sure that the Remote Desktop server registry has been successfully backed up.
2. Start Registry Editor.

3. Locate and then click the following registry subkey:

HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Terminal Server\RCM

4. On the **Registry** menu, click **Export Registry File**.
5. Type exported- Certificate in the **File name** box, and then click **Save*.

Note

If you have to restore this registry subkey in the future, double-click the Exported-parameters.reg file that you saved in this step.

6. Right-click each of the following values, click **Delete**, and then click **Yes** to verify the deletion:
    - **Certificate**
    - **X509 Certificate**
    - **X509 Certificate ID**
    - **X509 Certificate2**
7. Exit Registry Editor, and then restart the server.

References

For more information about Remote Desktop Gateway, see the following articles:

- 967933 Error message when a remote user tries to connect to a resource on a Windows Server 2008-based computer through TS Gateway by using the FQDN of the resource: "Remote Desktop Disconnected"
- 329896 Because of a security error, the client could not connect to the Remote Desktop server
- Group Policy Settings for Remote Desktop Services in Windows Server 2008 R2
- Troubleshooting General Remote Desktop Error Messages

If this article does not help you resolve the problem, or if you experience symptoms that differ from those that are described in this article, visit the Microsoft Support for more information. To search your issue, in the **Search support for help** box, type the text of the error message that you received, or type a description of the problem.

Data collection

If you need assistance from Microsoft support, we recommend you collect the information by following the steps mentioned in Gather information by using TSS for User Experience issues.

How to Troubleshoot Common Remote Desktop Issues

https://techanta.com/how-to-troubleshoot-common-remote-desktop-issues/

Remote Desktop is an essential tool that allows users to access their computers or servers remotely. Whether you're a business professional working from home or an IT administrator managing multiple systems, Remote Desktop enables convenient access to resources from any location. However, like any technology, Remote Desktop is not without its challenges. In this article, we will explore common issues that users may encounter and provide step-by-step troubleshooting solutions.

Table of Contents

Introduction

Remote Desktop, also known as Remote Desktop Protocol (RDP), is a technology developed by Microsoft that enables users to connect to a remote computer or server over a network. It provides a graphical interface for accessing resources, running applications, and performing administrative tasks as if you were physically present at the remote machine.

The importance of Remote Desktop cannot be overstated, especially in today's interconnected world. It allows businesses to facilitate remote work, increase productivity, and streamline IT management. However, encountering issues while using Remote Desktop can be frustrating. Let's dive into some common problems and their solutions.

Common Remote Desktop Issues

Connection Problems

One of the most common issues with Remote Desktop is connection problems. Users may experience difficulties establishing a connection to the remote computer or server. This can be caused by various factors, including network issues, firewall settings, or outdated Remote Desktop clients.

Slow Performance

Another frequent problem is slow performance during a Remote Desktop session. Laggy response times and sluggishness can hinder productivity. Slow performance can be due to network congestion, limited bandwidth, resource-intensive applications running on the remote machine, or inadequate network settings.

Screen Resolution Issues

Sometimes, when connecting to a remote machine, the screen resolution may not match your local display settings. This can result in a distorted or improperly scaled interface, making it difficult to read text or access certain features.

Audio and Printing Problems

Audio and printing problems are common when using Remote Desktop. Sound may not play on the remote machine, or printers connected to the local computer may not be recognized within the remote session. These issues can be caused by driver conflicts, improper settings, or compatibility issues.

Login and Authentication Issues

Login and authentication problems can prevent users from accessing their remote resources. Incorrect credentials, password expiration, or misconfigured Active Directory settings can lead to login failures, preventing a successful Remote Desktop session.

Troubleshooting Steps for Connection Problems

If you encounter connection problems while using Remote Desktop, try the following troubleshooting steps:

Check Network Connection: Ensure that both the local and remote machines have a stable network connection. Check Ethernet or Wi-Fi connectivity and verify that there are no network disruptions.

Verify Remote Desktop Services: Make sure that the remote computer has Remote Desktop services enabled. Go to the System Properties settings and confirm that Remote Desktop is allowed.

Check Firewall Settings: Check the firewall settings on both the local and remote machines. Ensure that the necessary ports (usually port 3389) are open to allow Remote Desktop connections.

Update Remote Desktop Client: If you are using a Remote Desktop client application, ensure that it is up to date. Check for any available updates or patches that might address known connection issues.

By following these steps, you can troubleshoot and resolve most connection problems with Remote Desktop.

Troubleshooting Steps for Slow Performance

To address slow performance during a Remote Desktop session, consider the following steps:

Close Unnecessary Programs: Close any unnecessary programs or background processes running on the remote machine. This frees up system resources and improves performance.

Optimize Network Settings: Adjust network settings to optimize Remote Desktop performance. For example, disabling bandwidth-intensive features like desktop background and font smoothing can enhance speed.

Adjust Display Settings: Lower the color depth or reduce the screen resolution of the remote session. This can significantly improve performance, especially when connecting over limited bandwidth connections.

Disable Visual Effects: Disable visual effects on the remote machine, such as animations and transparent windows. This reduces the processing load and enhances performance.

By implementing these troubleshooting steps, you can mitigate slow performance issues and ensure a smoother Remote Desktop experience.

Troubleshooting Steps for Screen Resolution Issues

If you encounter screen resolution issues during a Remote Desktop session, try the following steps:

Adjust Display Settings: On the remote machine, adjust the display settings to match your local computer's resolution. This ensures that the remote session is displayed correctly on your screen.

Configure Remote Desktop Resolution: Modify the Remote Desktop settings to match your preferred resolution. This can be done through the Remote Desktop client options or the Remote Desktop settings on the remote machine.

Use Scaling Options: If the remote session appears too small or too large, use scaling options available in the Remote Desktop client. Scaling allows you to adjust the size of the remote session window to your preference.

By applying these troubleshooting steps, you can resolve screen resolution issues and achieve a more comfortable viewing experience.

Troubleshooting Steps for Audio and Printing Problems

When encountering audio and printing problems in a Remote Desktop session, follow these steps to resolve the issues:

Check Audio and Printer Settings: Ensure that audio and printer redirection are enabled in the Remote Desktop client options. Verify that the correct audio and printer devices are selected for the remote session.

Update Audio and Printer Drivers: Update the audio and printer drivers on both the local and remote machines. Outdated or incompatible drivers can cause conflicts and prevent audio and printing functionality.

Restart Audio and Printer Services: Restart the audio and printer services on the remote machine. This can resolve temporary glitches or conflicts that may be affecting audio playback or printer recognition.

By following these troubleshooting steps, you can address audio and printing problems and regain full functionality within your Remote Desktop session.

Troubleshooting Steps for Login and Authentication Issues

If you encounter login and authentication issues while trying to establish a Remote Desktop session, consider the following steps:

Verify User Credentials: Double-check the username and password entered for the remote session. Ensure they are correct and match the credentials authorized to access the remote resources.

Reset Passwords: If you suspect a password issue, reset the password for the user account associated with the remote session. Ensure that the new password meets the complexity requirements and is properly synchronized.

Check Active Directory Configuration: Review the Active Directory configuration and ensure that the user account has the necessary permissions to establish a Remote Desktop session. Verify that group policies are correctly applied.

Enable Network Level Authentication: Network Level Authentication (NLA) adds an additional layer of security by authenticating the user before a session is established. Enable NLA on both the local and remote machines to enhance security and prevent unauthorized access.

By following these troubleshooting steps, you can overcome login and authentication issues, allowing a successful Remote Desktop session.

Conclusion

Remote Desktop is a powerful tool that enables remote access to resources and facilitates productivity. However, users may encounter common issues such as connection problems, slow performance, screen resolution discrepancies, audio and printing glitches, and login and authentication challenges. By following the troubleshooting steps provided in this article, you can effectively address these problems and enjoy a seamless Remote Desktop experience. Visit Now.

Frequently Asked Questions (FAQs)

Why is my Remote Desktop connection slow?

Slow Remote Desktop connections can be caused by network congestion, limited bandwidth, or resource-intensive applications running on the remote machine. Optimizing network settings, closing unnecessary programs, and adjusting display settings can help improve performance.

How do I fix audio issues in Remote Desktop?

To resolve audio issues in Remote Desktop, ensure that audio redirection is enabled in the Remote Desktop client options. Update audio drivers on both the local and remote machines and restart audio services on the remote machine if needed.

Can I print from a Remote Desktop session?

Yes, printing from a Remote Desktop session is possible. Ensure that printer redirection is enabled in the Remote Desktop client options. Install the necessary printer drivers on both the local and remote machines for seamless printing functionality.

How do I troubleshoot login issues in Remote Desktop?

To troubleshoot login issues in Remote Desktop, verify that the entered credentials are correct and match the authorized user account. Reset passwords if necessary, check Active Directory configuration, and ensure that Network Level Authentication is enabled.

What is Network Level Authentication in Remote Desktop?

Network Level Authentication (NLA) is a security feature in Remote Desktop that requires the user to authenticate before establishing a session. Enabling NLA enhances security and prevents unauthorized access to remote resources.

Advance Troubleshooting for Remote Desktop Protocol (RDP) in Windows 10/11

Disclaimer: This article includes deep troubleshooting in Windows Environment including the registry troubleshooting. We request you to backup all your personal files, bookmarks, and data files on MS Outlook before we proceed with a troubleshooting mentioned beneath. Once the files are deleted, we would not be able to recover them afterwards. On the Other hand, we may have to repeat some troubleshooting steps to ensure quality, which may appear tedious however that is for a good cause.

https://answers.microsoft.com/en-us/windows/forum/all/advance-troubleshooting-for-remote-desktop/bd334727-f449-477d-af88-73910593dac8

Remote Desktop Protocol (RDP) is a communication protocol developed by Microsoft which provides a user with a software interface to connect and control another computer over a network connection. In other words, Remote Desktop allows users to take control of another device from a distance. A user deploys RDP client software for this purpose, while the other computer must run RDP server software. Remote Desktop server (A computer that to be accessed or controlled) needs to be running a Pro edition of Windows, while a client machine (the device you are connecting from) can be running any edition of Windows (Pro or Home), and it is also applicable for different operating systems. RDP allows users to connect to Remote Desktop Services from Windows Server and remote PCs, use and control desktops and apps that an admin has made available to us. There are clients available for many different types of

devices on different platforms, such as desktops and laptops, tablets, smartphones, and Remote Desktop through a web browser. A web browser on desktops and laptops, helps us connect without having to download and install any software.

In this article, we will learn 'How to establish Remote Desktop Connection and How to troubleshoot Remote Desktop' using Windows machines. To make a connection between two devices to be connected and accessed from a distance, both must have Remote Desktop enabled. If you're using an older or other version of Windows, you can find additional information for Enabling Remote Desktop at the Microsoft Support Center. The device to be used to access the content can be running on any version of Windows or even non-Windows operating systems. If you are using a non-Windows operating system, you can download the Microsoft Remote Desktop app to get connected. Once you've downloaded it, enter the name of the PC you want to connect to in the app. The host computer must have Remote Desktop enabled (The computer to be accessed). This feature is only available on Windows 10/11 Pro and above versions however it is not available for home versions.

Troubleshooting Steps for Remote Desktop for Windows machines

When it comes to troubleshooting a Windows machine with respect to Remote Desktop, we need to ensure that the machines are liberated from Network barriers and security glitches which may inherent a condition for a connection. The outgoing and incoming connections can be affected by a presence of security software on your machine. Please validate, whether your firewall is blocking the RDP connection or not and if yes, add it to the whitelist if required.

We can also check if your account has sufficient permissions to start a connection from a source computer. If we have an incorrect configuration of listening ports, corrupt RDC credentials, or some network-related issues, the protocol may get disoriented. Therefore, before we start, we need to ensure that basic settings and requirements are in place to facilitate a smooth connection.

Get started with the Remote Desktop app for Windows | Microsoft Learn

Enable Remote Desktop in Settings

Click on **Search** on a Taskbar in Windows, type **Settings** and open it. Search for **Remote Desktop** in Settings or we can find it under the **System** tab.

Then, Toggle the **Switch** for Remote Desktop, forward to **Enable** it and select the **Confirm** option on the dialog box.

After we enable this, we will be able to use the Remote Desktop feature, and other PCs will also be able to access our PC remotely.

Validate if Remote Desktop Services is Enabled and Running.

Type **services.msc** in a search box on a Taskbar and open Services. Then, double-click **Remote Desktop Services** to bring up its properties window.

Select the **Automatic** option on the Startup type drop-down menu if the service is disabled.

Press the **Apply** button. Now, click the **Start** button and select **OK** to exit the window.

Allow Remote Desktop through Windows Defender Firewall

Open **Control Panel** from the built-in Apps in Windows 10/11, click the view drop-down menu and select the large icons option and select **Windows Defender Firewall**. Then, click **Allow** an app or feature through Windows Defender Firewall on the left side of the Control Panel. Click on **Change** settings to add or change apps permission. It will show a list of apps and features that are allowed for inbound and outbound connections.

Scroll down and check the **Remote Desktop** box for **Private** and **Public** columns. Click **OK** to apply the changes. We must let it pass through the Firewall. If the Windows 10/11 Remote Desktop is working or responding on your PC, validate if your **Windows Defender Firewall** is not blocking it. It is required to validate, how we have configured a firewall policy, it may block some inbound and outbound communications. We need to check Windows Defender Firewall settings to see if Remote Desktop Connection is blocked. If yes, add the app to the allowed list.

Change Your Network Profile

In Windows 10/11, we can choose to make our network profile **Public** or **Private**. On a **Public** network, Windows disables the network discovery feature to hide our computer from other computers. We can try to change our network to **Private** to see if you we can establish a connection with the network discovery feature enabled.

Click on **Search** on a Taskbar in Windows, type **Settings** and open it. Go to **Network and Internet**. In the **Status** tab, check your network status. To change the status, click on the **Properties** and then set your network profile to **Private**. If it is already set to Private, change it to **Public** and validate any improvements.

Reset the Remote Desktop Connection Credentials

When we establish a new Remote Desktop connection for the first time, the client saves the credentials for quick login. However, incorrect credentials can result in the Remote Desktop connection error to the remote computer. A reset of saved credentials can resolve this error.

Type **Remote Desktop Connection** in the Windows search bar and open the **client**. Click the drop-down for **Computer** and choose your **Remote PC**. Click on the **Delete** link under the Username section and click **Yes** to confirm the action.

Once the credentials are reset, **Relaunch** the Remote Desktop Connection client and try to connect again.

Add the Remote PC Address to the Hosts File

Windows' Hosts file contains information to map a connection between an IP address and domain name. There is another way to resolve the Remote Desktop when we can't connect to a remote PC or an error to add the remote IP address to the hosts file in a PC. If we manually add the Remote PC address to the hosts file can help us resolve any issues that may occur because of domain name resolution.

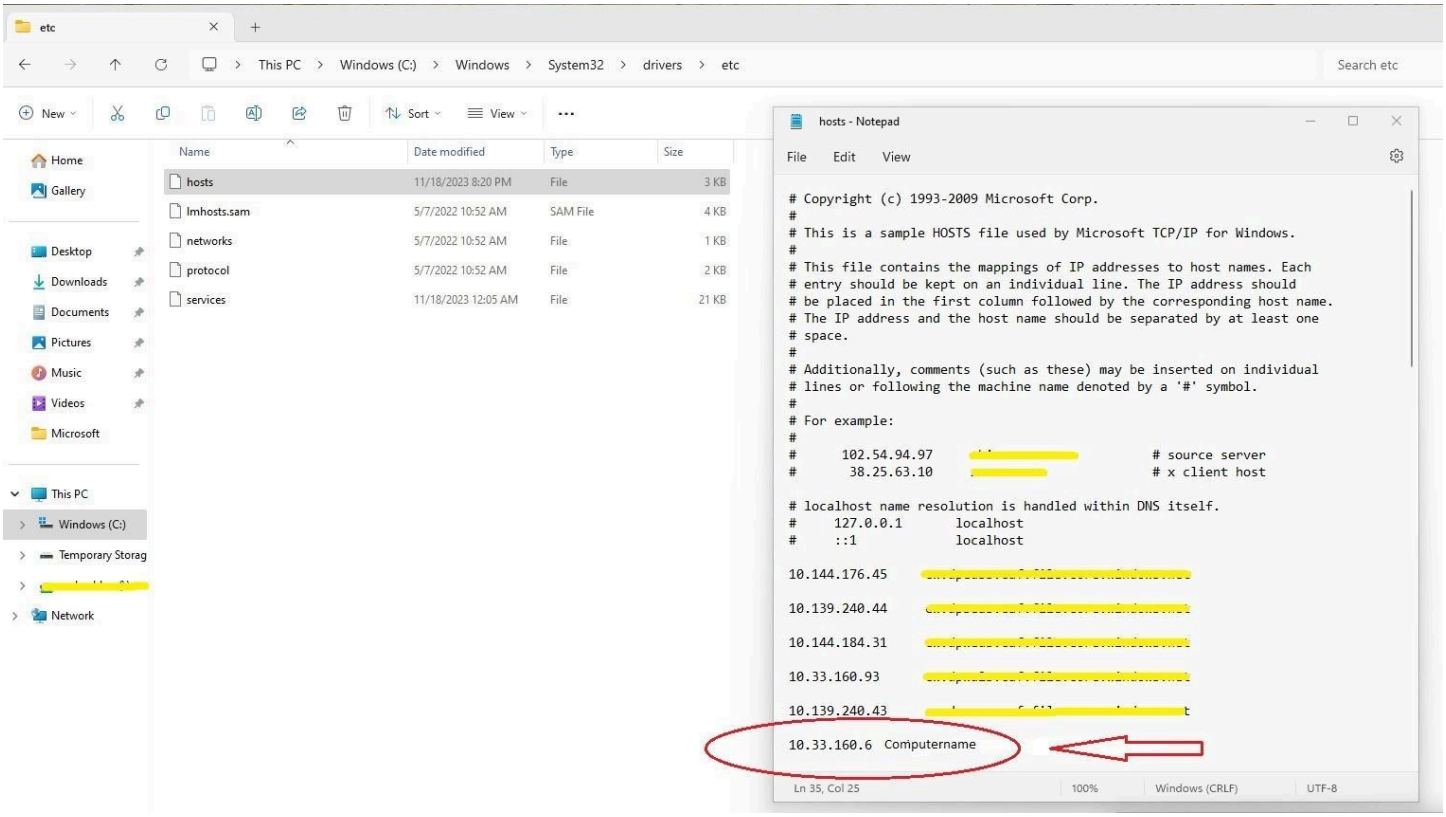
Open **File Explorer** and navigate to the following location:

C:\Windows\System32\drivers\etc

In the **etc** folder, right-click on the hosts file, choose **Open With**, and select **Notepad** from the list of apps.

We will see a few entries of listed IP Address in the hosts file. Then we need to **add the IP address** space and the name of the remote computer we want to connect to and save the file.

Example: 192.168.2.5 Computer2



Reference Image for editing hosts file under C:\Windows\System32\drivers\etc

We need to ensure that we have the correct IP address for the remote computer. On the remote computer, **ipconfig** command using **Command Prompt** will help us find a computer IP. If we are not with the remote computer, we must ask someone at the location to do this for us, then take the IP address.

Similarly, to find out the computer name of the remote device, Click on Windows Menu >> Control Panel >> System and view the computer name and workgroup.

Update Registry Parameters

***Registry Disclaimer**: Please be clear that serious problems might occur if we modify the registry incorrectly and please ensure that you follow these steps carefully. It is recommended that you back-up Windows registry before proceeding with Windows Registry editor. Therefore, we can always restore Windows registry if something goes wrong during the stated operation. For more information about how to back up and restore the registry, please refer a link mentioned below.*

Press the Windows key + R, type **regedit**, and click **OK**.

Follow the path mentioned below.

HKEY_LOCAL_MACHINE\SOFTWARE\Policies\Microsoft\Windows NT\Terminal Services\Client

Now, right-click the **Client** folder and choose **New** > **DWORD (32-bit)** Value.

Type **fClientDisableUDP** as the name of the new item.

Finally, double-click it, set its **Value** data to **1** and click the **OK** button.

Add the **RDGClientTransport** Key to the Registry

There is another way to fix Remote Desktop connection-related issues is to make the Registry Editor to add the RDGClientTransport key. It will force the Remote Desktop Protocol to use RPC/HTTP connection instead of HTTP/UDP.

In the Registry Editor, navigate to the following location.

Computer\HKEY_CURRENT_USER\SOFTWARE\Microsoft\Terminal Server Client

Right-click on the **Terminal Server Client** key and select **New** > **DWORD** (32-bit) Value.

Rename the value as **RDGClientTransport.**

Then, double-click on the newly created values and enter **1** in the Data value field. Click **OK** to save the changes.

Check and Configure RDP Listen Port

RDP uses 3389 as the default listen port. We can configure the listening port using the Registry Editor.

Open the Registry Editor and navigate to the following location:

HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Terminal Server\WinStations\RDP-Tcp

Select the **RDP-Tcp** key. Then, in the right pane, right-click on **PortNumber** and select **Edit**.

Set the value to **3389** with base as Decimal and click **OK**.

Restart your machine.

Update Group Policy

If the issue persists, a Group Policy Object may be blocking the connection to a local computer. We will have to manually enable the service using Group Policy Editor.

Press Windows Key + R to open **Run**. Type **gpedit.msc** and click **OK**. This will open the **Group Policy Editor**.

Note: If we use Windows 10 Home, we need to enable GPE manually before you can access the tool.

Enroll a Windows device automatically using Group Policy - Windows Client Management | Microsoft Learn

In the Group Policy Editor, navigate to the following location:

Computer Configuration\Administrative Templates\Windows Components\Remote Desktop Services\Remote Desktop Session Host\Connections

In Settings, locate and double-click on **Allow users to connect remotely by using Remote Desktop Services**.

Select **Enabled** and click **Apply** and **OK** to save the changes.

Then, close the Group Policy Editor and open **Command Prompt** as administrator.

Type **cmd** in the Windows search on a Taskbar, right-click on Command Prompt, and select **Run as Administrator**.

In the Command Prompt, type **gpupdate /force** and hit **Enter**. This will force the recent changes made to the GPO.

Restart your machine.

Turn off UDP in Group Policy

Press the Windows key + R, type **gpedit.msc**, and click **OK**.

Navigate to the path below.

Computer Configuration > Administration Templates > Windows Components > Remote Desktop Services > Remote Desktop Connection Client

Now, double-click **Turn Off UDP On Client** in the right pane.

Then, tick the **Enabled** radio button and click **OK**.

Restart your machine and test the program.

Additional troubleshooting (Workaround)

Validate the status of the RDP protocol on a remote computer.

To check and change the status of the RDP protocol on a remote computer, use a network registry connection.

Press Windows key + R to open **Run**. In the Run dialog box, type **regedit** and hit Enter to open Registry Editor.

Select **File** in the Registry, then select **Connect Network Registry**.

In the Select Computer dialog box, enter the name of the remote computer. Select Check Names and Click on **OK**.

Then, navigate to the registry key path below:

HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Terminal Server

At the location, on the right pane, double-click the **fDenyTSConnections** key to edit its properties.

To enable RDP, set the Value data of **fDenyTSConnections** from **1** to **0**.

The value of 0 indicates RDP is enabled, while the value of 1 indicates RDP is disabled.

Check the status of the RDP self-signed certificate.

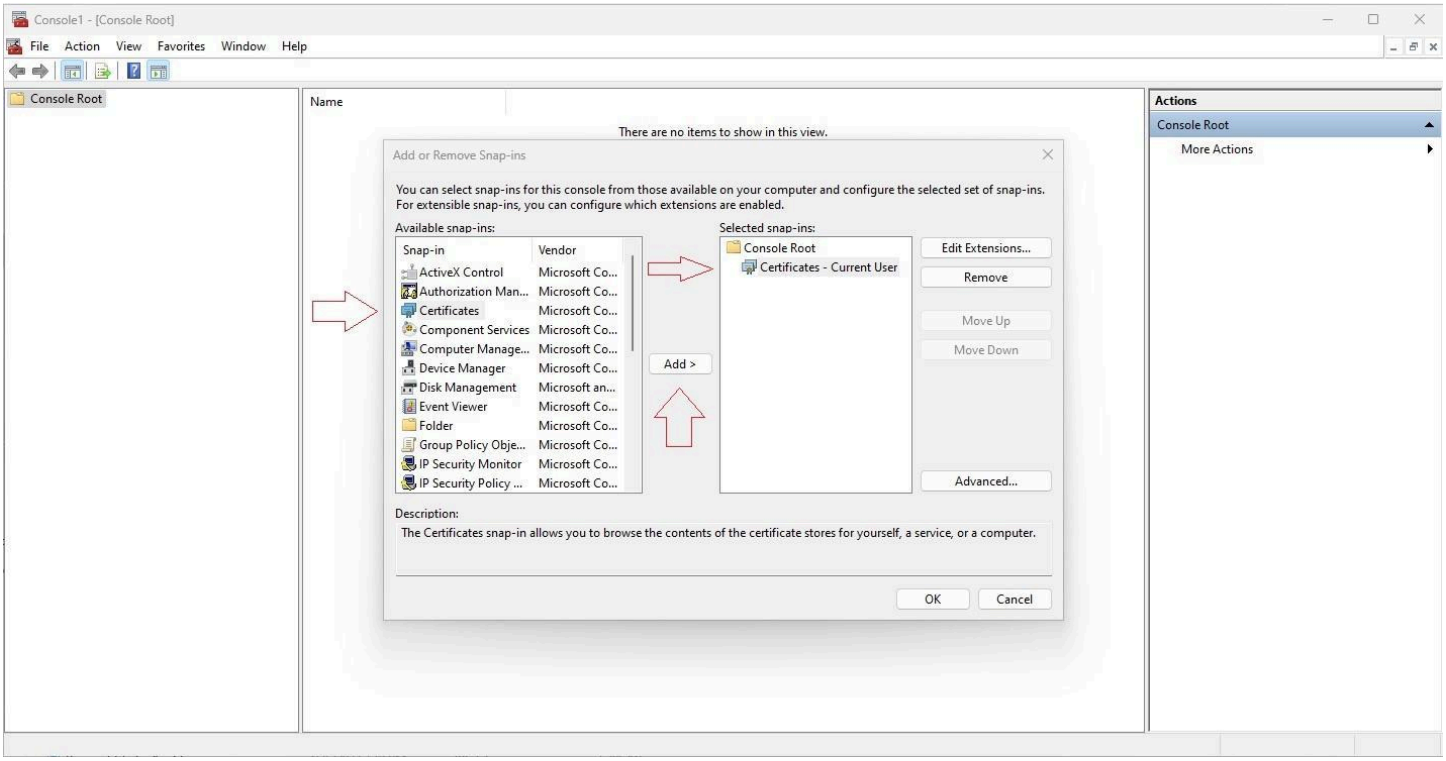
If we still find difficulty in validating a connection using Remote Desktop, check the status of RDP Self-signed Certificate.

Press the Windows key + R to open **Run**. In the Run dialog box, type mmc and hit Enter to open **Microsoft Management Console**.

Click the **File** menu. Select **Add/Remove Snap-in**. Select **Certificates** from the list of snap-ins.

Click **Add**. When you are prompted to select the certificate store to manage, select **Computer account**.

Then, click **Next**. Select the affected computer. Click the **Finish** button and then click **OK**.



Reference image for Microsoft Management Console and directions for adding Certificates.

Now, in the **Certificates** folder under **Remote Desktop**, delete the **RDP self-signed certificate**.

On the affected computer, **Restart** the **Remote Desktop Services** service. **Refresh** the Certificates snap-in.

If the RDP self-signed certificate has not been recreated, check the permissions of the **MachineKeys** folder.

Check the permissions of the MachineKeys folder.

On the affected computer, open File Explorer and navigate to the directory path below:

C:\ProgramData\Microsoft\Crypto\RSA

At the location, right-click on **MachineKeys**, select **Properties**, select **Security**, and then select **Advanced.**

Make sure that the following permissions are configured:

Administrators: **Full control** and Everyone: **Read, Write**

If you find it hard to perform above mentioned troubleshooting and need assistance, please refer a link mentioned below.

Contact Us - Microsoft Support

Additional note

- We need to ensure that the Clipboard redirect function is enabled, to use it on the remote computer. It allows Remote Desktop Connection to copy text from host machine to client machine.

To use this function, open the Remote Desktop Connection dialog box by typing Remote Desktop in your Start menu search bar, select Show Options. Find the Local Resources tab, under Local devices and resources, put a check in the Clipboard box. The option should be turned on by default.

Reference Articles

What's new in Windows client deployment - Windows Deployment | Microsoft Learn

Remote Desktop client - supported configuration | Microsoft Learn

Remote Desktop clients for Remote Desktop Services and remote PCs - Windows Server | Microsoft Learn

Connect to Azure Virtual Desktop with the Remote Desktop client for Windows - Azure Virtual Desktop | Microsoft Learn

This article was last published on Thursday, December 14th, 2023

To identify and resolve port conflicts using tools like netstat and tasklist

To identify and resolve port conflicts using tools like `netstat` and `tasklist`, you can follow these steps. Port conflicts occur when multiple applications try to use the same network port, which can prevent services like Remote Desktop (which uses port 3389 by default) from working properly.

Step-by-Step Instructions

1. **Check for Port Conflicts with `netstat`**

The `netstat` command is used to display active connections and listening ports. To check if port 3389 (or another port) is being used by another process:

Steps:

1. **Open Command Prompt**:
   - Press `Win + R`, type `cmd`, and press Enter.
2. **Run `netstat` Command**:
   - In the Command Prompt, type the following and press Enter:
     ```
     netstat -a -o | findstr :3389
     ```
   - This command will filter the output to show only the entries related to port 3389 (the default RDP port). It will display information such as the protocol (TCP or UDP), the local address, the foreign address, and the PID (Process Identifier) of the process using that port.

Example Output:

```
TCP    0.0.0.0:3389       0.0.0.0:0      LISTENING       1234
```

- In this example, the process with PID `1234` is using port 3389 and is in the `LISTENING` state.

2. **Identify the Process Using the Port with `tasklist`**

After finding the PID, you can use the `tasklist` command to identify which application or service is using that port.

Steps:

1. **Run `tasklist` Command**:

   - In the same Command Prompt, type the following and press Enter:

     ```

     tasklist /svc | findstr 1234

     ```

   - Replace `1234` with the PID you found using `netstat`.

Example Output:

```

svchost.exe                  1234 Services       TermService

```

- In this example, the service `TermService` (which is responsible for Remote Desktop Services) is using port 3389.

3. **Resolve the Port Conflict**

Once you've identified the application or service using the port, you have a few options to resolve the conflict.

Option 1: Stop the Conflicting Process

If the process is not critical and can be stopped:

1. **Stop the Process**:

   - Use the following command to stop the process:

     ```

     taskkill /PID 1234 /F

     ```

   - Replace `1234` with the actual PID. The `/F` flag forces the process to terminate.

Option 2: Change the Port for the Application or Service

If you can't stop the process because it's essential, you can change the port for either the conflicting application or Remote Desktop.

To Change the RDP Port:

1. **Open Registry Editor**:
   - Press `Win + R`, type `regedit`, and press Enter.
2. **Navigate to the RDP Port Configuration**:
   - Go to:
     ```
     HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Terminal Server\WinStations\RDP-Tcp
     ```
3. **Modify the PortNumber Entry**:
   - Find the `PortNumber` key, double-click it, change it to a different port (e.g., 3390), and click OK.
4. **Restart the Remote Desktop Services**:
   - Open the Services management console (press `Win + R`, type `services.msc`, and press Enter).
   - Find `Remote Desktop Services`, right-click it, and select `Restart`【7†source】【6†source】.

4. **Verify the Change**

After resolving the conflict, use `netstat` again to ensure that the correct service is using the appropriate port.

Check the New Port:

1. **Run `netstat` to Verify**:

   ```
   netstat -a -o | findstr :3390
   ```

   - Ensure that the output shows the correct process using the new port without conflicts.

Example Scenario

You are unable to connect to Remote Desktop on a server, and suspect a port conflict is preventing the service from working. Using `netstat`, you discover that port 3389 is being used by a different service. By using `tasklist`, you identify that a non-essential application is occupying the port. You stop that application using `taskkill`, which frees up port 3389 for Remote Desktop. If the application were essential, you could change Remote Desktop to use port 3390 instead, and adjust firewall settings to allow the new port.

Summary of Commands

- **`netstat -a -o | findstr :3389`**: Check which process is using port 3389.
- **`tasklist /svc | findstr <PID>`**: Identify the process or service using the port.
- **`taskkill /PID <PID> /F`**: Terminate the process using the conflicting port.
- **Registry Edit**: Change the RDP port if stopping the process is not an option

By following these steps, you can identify and resolve port conflicts effectively, ensuring that your services, like Remote Desktop, run smoothly.

Understanding the Remote Desktop Protocol (RDP)

This article describes the Remote Desktop Protocol (RDP) that's used for communication between the Terminal Server and the Terminal Server Client. RDP is encapsulated and encrypted within TCP.

Original KB number: 186607

Summary

RDP is based on, and is an extension of, the T-120 family of protocol standards. A multichannel capable protocol allows for separate virtual channels for carrying the following information:

- presentation data
- serial device communication
- licensing information
- highly encrypted data, such as keyboard, mouse activity

RDP is an extension of the core T.Share protocol. Several other capabilities are retained as part of the RDP, such as the architectural features necessary to support multipoint (multiparty sessions). Multipoint data delivery allows data from an application to be delivered in **real time** to multiple parties, such as Virtual Whiteboards. It doesn't require to send the same data to each session individually.

In this first release of Windows Terminal Server, we're concentrating on providing reliable and fast point-to-point (single-session) communications. Only one data channel is used in the initial release of Terminal Server 4.0. However, the flexibility of RDP gives plenty of room for functionality in future products.

One reason that Microsoft decided to implement RDP for connectivity purposes within Windows NT Terminal Server is that it provides an extensible base to build many more capabilities. RDP provides 64,000 separate channels for data transmission. However, current transmission activities are only using a single channel (for keyboard, mouse, and presentation data).

RDP is designed to support many different types of Network topologies, such as ISDN, POTS. RDP is also designed to support many LAN protocols, such as IPX, NetBIOS, TCP/IP. The current version of RDP will only run over TCP/IP. With customer feedback, other protocol support may be added in future versions.

The activity involved in sending and receiving data through the RDP stack is essentially the same as the seven-layer OSI model standards for common LAN networking today. Data from an application or service to be transmitted is passed down through the protocol stacks. It's sectioned, directed to a channel (through MCS), encrypted, wrapped, framed, packaged onto the network protocol, and finally addressed and sent over the wire to the client. The returned data works the same way only in reverse. The packet is stripped of its address, then unwrapped, decrypted, and so on. Finally the data is presented to the application for use. Key portions of the protocol stack modifications occur between the fourth and seventh layers, where the data is:

- encrypted
- wrapped
- framed
- directed to a channel
- prioritized

One of the key points for application developers is that, in using RDP, Microsoft has abstracted away the complexities of dealing with the protocol stack. It allows them to write clean, well-designed, well-behaved 32-bit applications. Then the RDP stack implemented by the Terminal Server and its client connections takes care of the rest.

For more information about how applications interact on the Terminal Server, and what to know when developing applications for a Windows Terminal Server infrastructure, see the following white paper:
Optimizing Applications for Windows NT Server 4.0, Terminal Server Edition

Four components worth discussing within the RDP stack instance are:

- the Multipoint Communication Service (MCSMUX)
- the Generic Conference Control (GCC)
- Wdtshare.sys

- Tdtcp.sys

MCSmux and GCC are part of the International Telecommunication Union (ITU) T.120 family. The MCS is made up of two standards:

- T.122: It defines the multipoint services
- T.125: It specifies the data transmission protocol

MCSMux controls:

- channel assignment by multiplexing data onto predefined virtual channels within the protocol
- priority levels
- segmentation of data being sent

It essentially abstracts the multiple RDP stacks into a single entity, from the perspective of the GCC. GCC is responsible for management of those multiple channels. The GCC allows the creation and deletion of session connections and controls resources provided by MCS. Each Terminal Server protocol (currently, only RDP and Citrix's ICA are supported) will have a protocol stack instance loaded (a listener stack awaiting a connection request). The Terminal Server device driver coordinates and manages the RDP protocol activity. It's made up of smaller components:

- an RDP driver (Wdtshare.sys) for UI transfer, compression, encryption, framing, and so on.
- a transport driver (Tdtcp.sys) to package the protocol onto the underlying network protocol, TCP/IP.

RDP was developed to be entirely independent of its underlying transport stack, in this case TCP/IP. It means that we can add other transport drivers for other network protocols as customers needs for them grow, with little or no significant changes to the foundational parts of the protocol. They're key elements to the performance and extendibility of RDP on the network.

Remote Desktop Services protocols

Protocol Name	Description	Short Name
Remote Desktop Protocol: Basic Connectivity and Graphics Remoting	This protocol facilitates user interaction with a remote computer system by transferring graphics display information from the remote computer to the user and transferring input from the user to the remote computer, where the input is injected into the user session. This protocol also provides an extensible mechanism allowing specialized communication between	MS-RDPBCGR

Protocol Name	Description	Short Name
	components on the user computer and components running on the remote computer.	
Terminal Services Gateway Server Protocol	This protocol provides the ability to tunnel RDP communications through a gateway for a connection between an RDP client and an RD Session Host server behind a firewall.	MS-TSGU
Terminal Services: Terminal Server Runtime Interface	This protocol is used for remotely querying and configuring various aspects of an RD Session Host. For example, this protocol can be used to query the number of active sessions running on an RD Session Host.	MS-TSTS
Terminal Services: Workspace Provisioning Protocol	This protocol allows a unified view of user work resources for administrators that have no access to non-managed computers. The protocol is used to transfer information so that the client computer can launch a remote desktop and remote applications on a server or virtual computer.	MS-TSWP

Member protocols used to optimize graphical data, support session management and licensing

Expand table

Protocol Name	Description	Short Name
Remote Desktop Protocol: Desktop Composition Virtual Channel Extension	This protocol supports remote desktop composition (the composition of a sprite tree that represents the desktop, with nodes representing the sprites).	MS-RDPEDC
Remote Desktop Protocol: Graphics Devices Interfaces (GDI) Acceleration Extension	This protocol encodes the drawing operations that produce an image, reducing the bandwidth associated with graphics remoting.	MS-RDPEGDI
Remote Desktop Protocol: Composited Remoting V2	This protocol is used to display the contents of a desktop running on one machine (the server) on a second remote machine (the client) connected to the first via a network.	MS-RDPCR2
Remote Desktop Protocol: NSCodec Extension	This protocol specifies an image codec that can be used to encode screen images by utilizing efficient and effective compression.	MS-RDPNSC
Remote Desktop Protocol: RemoteFX Codec Extension	This protocol specifies a lossy image codec that can be used to encode screen images by utilizing efficient and effective compression.	MS-RDPRFX
Remote Desktop Protocol: Session Selection Extension	This protocol describes the messages exchanged between an RDP client and a server to facilitate the precise targeting of an application-sharing context.	MS-RDPEPS
Remote Desktop Protocol: Licensing Extension	This protocol allows authorized RDP clients or users to connect to an RD Session Host. This extension involves communication between the RDP client, the RD Session Host, and a license server. The RD Session Host can be configured to be in per-device or per-user license mode. Client Access Licenses (CALs) are installed on a license server, and when an RD Session Host requests a license on a client's behalf, the license server issues a license out of its available pool of licenses.	MS-RDPELE

Member protocols used to enable transporting device data or resource data between an RDP client and an RD Session Host server

Protocol Name	Description	Short Name
Remote Desktop Protocol: Clipboard Virtual Channel Extension	This protocol provides basic programmatic access to the clipboard provided by an operating system and ensures that any application has the capability to place data onto the clipboard, extract data from the clipboard, enumerate the data formats available on the clipboard, and register to receive notifications when the system clipboard is updated.	MS-RDPECLIP
Remote Desktop Protocol: File System Virtual Channel Extension	This protocol provides access between the RD Session Host server and the RDP client file system drivers by redirecting all input/output requests and responses between the two.	MS-RDPEFS
Remote Desktop Protocol: Serial and Parallel Port Virtual Channel Extension	This protocol specifies the communication used to enable the redirection of ports between a terminal client and an RD Session Host server. By redirecting ports from the RDP client to the RD Session Host server, applications running on an RD Session Host server can access the remote devices attached to those ports.	MS-RDPESP
Remote Desktop Protocol: Print Virtual Channel Extension	This protocol specifies the communication used to enable the redirection of printers between an RDP client and an RD Session Host server. By redirecting printers from the RDP client to the RD Session Host server, applications running on a server can access the remote devices as if they were local printers.	MS-RDPEPC
Remote Desktop Protocol: Video Capture Virtual Channel Extension	This protocol enables the transfer of video capture data from the RDP client to the RD Session Host. For example, an application running on an RD Session Host can record video capture data. This data will be transferred from the RDP client to the RD Session Host, allowing the application to record from a video capture device installed on the RDP client.	MS-RDPECAM
Remote Desktop Protocol: Smart Card Virtual Channel Extension	This protocol enables client smart card devices to be available, within the context of a single RDP session, to server-side applications.	MS-RDPESC
Remote Desktop Protocol: Remote Programs Virtual Channel Extension	This protocol is a Remote Desktop Protocol (RDP) feature (as specified in [MS-RDPBCGR]) that presents a remote application (running remotely on a remote application integrated locally (RAIL) server) as a local user application (running on the RAIL client computer). Also known as RAIL.	MS-RDPERP
Remote Desktop Protocol: Multiparty Virtual Channel Extension	This protocol enables the remote display of desktop and application content. To effectively implement an application-sharing or collaborative solution, additional information is conveyed to keep the participants apprised of who else is involved, in addition to which applications or windows are being shared.	MS-RDPEMC

Protocol Name	Description	Short Name
Remote Desktop Protocol: Audio Output Virtual Channel Extension	This protocol transfers audio data from the RD Session Host server to the RDP client. For example, when the RD Session Host server plays an audio file, this protocol is used by the RD Session Host server to transfer the audio data to the RDP client. The RDP client can then play the audio.	MS-RDPEA
Remote Desktop Protocol: Dynamic Virtual Channel Extension	This protocol implements a generic connection-oriented communication channel on top of the virtual channel protocol. A dynamic virtual channel (DVC) is established over an existing static virtual channel.	MS-RDPEDYC
Remote Desktop Protocol: Plug and Play Devices Virtual Channel Extension	This protocol specifies the communication used to enable the redirection of plug and play devices between an RDP client and an RD Session Host server.	MS-RDPEPNP
Remote Desktop Protocol: XML Paper Specification (XPS) Print Virtual Channel Extension	This protocol specifies communication between a virtual printer driver installed on an RD Session Host server and a printer driver installed on the RDP client. The primary purpose of this protocol is to acquire printing capabilities and to display a printer-specific user interface on the RDP client.	MS-RDPEXPS
Remote Desktop Protocol: Audio Input Virtual Channel Extension	This protocol enables the transfer of audio data from the RDP client to the RD Session Host. For example, an application running on an RD Session Host can record audio data. This data will be transferred from the RDP client to the RD Session Host, allowing the application to record from an audio device installed on the RDP client.	MS-RDPEAI
Remote Desktop Protocol: USB Devices Virtual Channel Extension	This protocol is used to redirect universal serial bus (USB) devices from a terminal client to the terminal server, which allows the server access to a device that is physically connected to the client as if the device were local to the server.	MS-RDPEUSB
Remote Desktop Protocol: Video Virtual Channel Extension	This protocol enables the transfer of synchronized audio and video data from an RD Session Host to an RDP client. The RDP client can play the audio and video data and synchronize this data using the timing information provided by the protocol.	MS-RDPEV
Remote Desktop Protocol: UDP Transport Extension	This protocol extends the transport mechanisms in RDP to enable network connectivity between the user's machine and a remote computer system over the User Datagram Protocol (UDP).	MS-RDPEUDP
Remote Desktop Protocol: UDP Transport Extension Version 2	This protocol specifies extensions to the transport mechanisms in RDP. It is used to exchange data, for example audio and video, between a remote desktop client and remote desktop server over UDP transport using a URCP based rate control.	MS-RDPEUDP2
Remote Desktop Protocol: Graphics Pipeline Extension	This protocol is used by RDPBCGR to enable smart processing of graphics data.	MS-RDPEGFX
Remote Desktop Protocol: Multitransport Extension	This protocol enables open multiple transports as an extension to RDPBCGR.	MS-RDPEMT
Remote Desktop Protocol: Virtual Channel Echo Extension	This protocol is used to determine network characteristics (such as round-trip time (RTT)) between an RD Session Host and an RDP client.	MS-RDPEECO

Protocol Name	Description	Short Name
Remote Desktop Protocol: Video Optimized Remoting Virtual Channel Extension	This protocol is designed to be run within the context of a Remote Desktop Protocol (RDP) virtual channel established between an RDP Client and an RD Session Host. This protocol extension is applicable when the RD Session Host server is displaying content that it classifies as video and needs to send that video data to the RDP client.	MS-RDPEVOR
Remote Desktop Protocol: Input Virtual Channel Extension	This protocol is used to transfer multitouch input frames (generated by a physical or virtual touch digitizer) or pen input from an RDP client to an RD Session Host server.	MS-RDPEI
Remote Desktop Protocol: Display Control Virtual Channel Extension	This protocol is used to request display configuration changes in a remote session.	MS-RDPEDISP
Remote Desktop Protocol: Authentication Redirection Virtual Channel	This protocol is used to perform authentication protocol operations over a virtual channel in the remote session. This allows the RDP server session to perform network authentication while not having access to the underlying user credentials.	MS-RDPEAR
Remote Desktop Protocol: WebAuthn Virtual Channel Protocol	This protocol provides a way for a user to do WebAuthn operations over the RDP protocol. It enables a server to send a WebAuthn request to a client. The client can then use this request to talk to authenticators (platform as well as cross-platform) and reply with the response.	MS-RDPEWA

Here's a PowerShell script that prompts you for a service name and then displays the service's status, name, display name, and process ID (PID):

How This Script Works:

1. **Prompt for Input**: Uses Read-Host to prompt the user to enter the service name.
2. **Get Service Information**:
   - Uses Get-Service to check if the service exists.
   - Uses Get-WmiObject to fetch detailed information about the service, including the DisplayName, State, and ProcessId.
3. **Display Information**: Outputs the service name, display name, status, and PID to the console.
4. **Error Handling**: If the service is not found, it displays an appropriate message.

++

```
$serviceName = Read-Host "Enter the service name"


# Get the service information using Get-Service
$service = Get-Service -Name $serviceName -ErrorAction SilentlyContinue
```

```
# Check if the service exists
if ($service) {
    # Get detailed service information using Get-WmiObject
    $serviceDetail = Get-WmiObject Win32_Service -Filter "Name='$serviceName'"

    # Display the service details
    Write-Output "Service Name: $($serviceDetail.Name)"
    Write-Output "Display Name: $($serviceDetail.DisplayName)"
    Write-Output "Status: $($serviceDetail.State)"
    Write-Output "Process ID (PID): $($serviceDetail.ProcessId)"
} else {
    Write-Output "Service '$serviceName' not found. Please check the service name and try again."
}
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
```

PowerShell script to enable RDC on a domain computer

It will prompt you for each step.

```
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
# Prompt for input values with descriptive messages
$TargetComputer = Read-Host "Enter the hostname or IP address of the target computer (e.g., Server01 or 192.168.1.10)"
$Domain = Read-Host "Enter the domain name (e.g., Contoso)"
$AdminUser = Read-Host "Enter the username of the domain admin (e.g., Administrator)"

# Step 1: Enable Remote Desktop by modifying the registry value
Write-Host "Enabling Remote Desktop via registry change on $TargetComputer..."
try {
    Invoke-Command -ComputerName $TargetComputer -ScriptBlock {
```

```
        Set-ItemProperty -Path "HKLM:\System\CurrentControlSet\Control\Terminal Server" -Name "fDenyTSConnections" -Value 0
    }
    Write-Host "Success: Remote Desktop enabled via registry." -ForegroundColor Green
} catch {
    Write-Host "Failure: Unable to enable Remote Desktop via registry." -ForegroundColor Red
}

# Step 2: Allow Remote Desktop through the Windows Firewall
Write-Host "Allowing Remote Desktop through Windows Firewall on $TargetComputer..."
try {
    Invoke-Command -ComputerName $TargetComputer -ScriptBlock {
        Enable-NetFirewallRule -DisplayGroup "Remote Desktop"
    }
    Write-Host "Success: Remote Desktop allowed through Windows Firewall." -ForegroundColor Green
} catch {
    Write-Host "Failure: Unable to allow Remote Desktop through Windows Firewall." -ForegroundColor Red
}

# Step 3: Add the default domain admin to the Remote Desktop Users group
Write-Host "Adding $Domain\$AdminUser to the Remote Desktop Users group on $TargetComputer..."
try {
    Invoke-Command -ComputerName $TargetComputer -ScriptBlock {
        Add-LocalGroupMember -Group "Remote Desktop Users" -Member "$using:Domain\$using:AdminUser"
    }
    Write-Host "Success: $Domain\$AdminUser added to Remote Desktop Users group." -ForegroundColor Green
} catch {
    Write-Host "Failure: Unable to add $Domain\$AdminUser to the Remote Desktop Users group." -ForegroundColor Red
```

```
}

# Step 4: Verify Remote Desktop is enabled
Write-Host "Verifying that Remote Desktop is enabled on $TargetComputer..."
try {
    $RDCStatus = Invoke-Command -ComputerName $TargetComputer -ScriptBlock {
        Get-ItemProperty -Path "HKLM:\System\CurrentControlSet\Control\Terminal Server" -Name "fDenyTSConnections"
    }

    if ($RDCStatus.fDenyTSConnections -eq 0) {
        Write-Host "Success: Remote Desktop is enabled on $TargetComputer." -ForegroundColor Green
    } else {
        Write-Host "Failure: Remote Desktop is not enabled on $TargetComputer." -ForegroundColor Red
    }
} catch {
    Write-Host "Failure: Unable to verify the Remote Desktop status on $TargetComputer." -ForegroundColor Red
}

++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
```