

Peculiarity of terminal session applications testing

Written By:
Natalia Novotnaya,
Team Leader of QA Team, Apriorit Inc.

1. Introduction	1
2. Problem statement	1
3. Nuances and main points of testing the application using terminal sessions	2
3.1. Application functioning for different users	2
3.2. Testing client and server cross-platform support	3
3.3. Testing with different RDP versions	4
3.4. Testing multiuser scenarios	4
3.5. Check time of terminal session Log in/	5
3.6. Stress test set for terminal session applications	5
4. Conclusion	6
5. Useful Links	6
Appendix A: Some knowledge about RDP	7
1. Protocol RDP	7
2. Features	8
3. Deploying Remote Desktop	11
Appendix B: Samples of the Test Cases	13
1. Testing multiuser scenarios	13
2. Application functioning for the different users	15
3. Stress test set for the application with terminal	16

1. Introduction

In this article we will try to help those who are going to test client-server application that uses terminal sessions. Not only QA specialists but also developers will find some useful information in this article – at least it is some details that should be taken into account when developing in order not to be in the testers black list of “Bug makers”.

To perform such testing one should know RDP and MSTSC – the brief review of them is given in the appendix. More detailed information can be found by links mentioned in the end of the article and by asking a question to the Mr. Google.

2. Problem statement

So let’s consider the scheme of an application that we are going to test.

In my practice the most of applications that used terminal sessions had client-server architecture. Thus there are always server and client part and the installation for both of them.

We also know information about what kind of client we have: thin or thick one. In the case of the thin client all operations are performed by the server, while thick clients do everything themselves.

There is also a question of whose resources will be used – it depends on the concrete design of the application that uses terminal sessions. Resources can be not only memory but also some software: drivers, codecs, coding etc.

A lot of programs include RDP functions, others improve some of the standard functions. And while testing will be different for the different applications there are some basic tests that can be used for all.

Work schemes of applications using terminal session

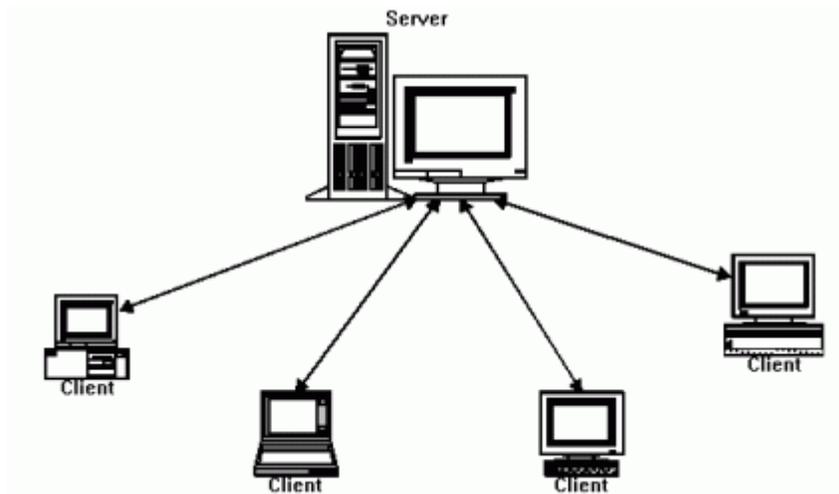


Figure 1

3. Nuances and main points of testing the application using terminal sessions

First we should check the requirement list:

- List of the OS supported by the server and client.
- How application depends on MSTSC.
- What resources will be shared on the server?
- Whose resources – client or server – are used for operation execution?
- What users can work with the application (all or just one)?

3.1. Application functioning for different users

When application supports an option of installation for all users in the system it can be really surprising to find out that it doesn't start or work for the other user (different

from the user who's installed it) especially for that who doesn't have the administrator wrights.

This checking can be performed in the Installation Testing block as well as in the Functional Testing.

The most probable bugs here are writing to the wrong registry branch (for example HKEY_CURRENT_USER) or to the wrong directory, or wrong component registration.

For example our application should work for the Guest user. Then we include such a test:

Test Name: Permissions _Connect to Server as Guest

Pre-conditions:

Client is a client machine where the client part of tested Application is installed

Server is a server machine where the server part of tested Application is installed.

Step name	Description	Expected result
Step 1	Connect to Server from Client via RDP by Guest user	The terminal session starts successfully on Client.
Step 2	Start an Application and check its basic functionality	An Application starts and works correctly according to Guest permissions.

Surely, expected result will be described accordingly to the requirements and application functionality.

3.2. Testing client and server cross-platform support

The hidden danger in such testing can be in the case when server part of the application is installed not on the server operating system (Windows2000, XP, Vista, Seven + a lot of different SPs). So it's better to clear it out at once in order to be prepared. If requirements include such option you should test the corresponding cases with special attention.

The most common situation is when the server part is installed on the server OS (Windows Server2000\2003\2008 + a lot of SPs) and the client has not the server OS.

It is convenient to use two dimensional tables to control this testing process:

client\server	Server 2003	Server 2008	Windows XP	Windows Vista	Windows Seven	...
Windows 2000						
Windows XP						
Windows Vista						
Windows Seven						

.....						
-------	--	--	--	--	--	--

In Configuration Testing you can consider single OS as well as OS with different RDP versions. We will tell about it in the next section.

3.3. Testing with different RDP versions

Testing of applications that are the add-ons for MSTSC depends much on the RDP version (5.0, 5.1, 5.2, 6.0 or 6.1).

In this case we should include function checking for different clients into the Functional Testing. Fortunately version of RDP depends on the OS. Thus checking with RDP versions can be included into the Configuration Testing.

client\server	Server 2003	Server 2008	Windows XP	Windows Vista	Windows Seven	...
Windows 2000						
Windows XP RDP 5.0						
Windows Vista RDP 6.0						
Windows Seven RDP 6.1						
.....						

3.4. Testing multiuser scenarios

One of the key points in the client-server application functioning is the options to work by 1:1, 1:M, M:1, M:M schemas.

When testing these schemas you should check:

- Optimal usage of resources of the client terminal and server.
- Correct usage of the shared resources: libraries, memory, files.
- System performance.

Let's consider the case when several clients are connected to the server.

Here can be the danger when server part takes almost all machine resources (CPU Usage, PF usage, Network Utilization etc.). As the result client can wait for the server answer for a long time. The average time of the answer for different client number should be checked.

Another point is the critical number of the clients connected. Of course the server is not made from gum – so there is such critical client number when the system will just

crash. There is always such number even for the brilliant performance system – but if it is less than 10 it is worth thinking about fact.

Sometimes application uses another scheme when two clients connect to the different servers from the one terminal. This scheme is commonly used for the thick client.

Such test cases can be the basic ones here:

- Maximum allowed count of sessions to one server
- Two sessions from two clients to one server
- Two sessions from one client to two servers
- Two sessions from one client to one server

The usage of the shared resources can be performed both by the application inside itself and by the user. So we should check both variants.

When several users work with the same file several dangers can appear:

- Access denied;
- Waiting till another user finishes his operations;
- Server cannot deal with such loading.

System performance is especially important for the corporate systems with the great number of users. So research of the system maximal performance should be performed thoroughly.

3.5. Check time of terminal session Log in/

The important usability factor is the way how application effects on the time of logging in/off the terminal session. This time should not be much longer than it is without our application installed.

3.6. Stress test set for terminal session applications

Test set for the Stress Testing can be formed from the common stress tests for the client-server applications plus some specific ones:

- Disconnect caused by hardware;
- Disconnect caused by software (enable\disable network).

These tests are double because they should be performed both on the client and server sides.

And more:

- Server reboot;
- Terminal session disconnect (close, disconnect, kill process);
- Terminal session closure (using log off function from the terminal session).

The last point is the additional peculiarity of the terminal session application testing.

4. Conclusion

As the final thought I want mention that the clear requirements make the testing process much more effective and easy. The testing of applications that use terminal sessions is not something very special or something to be afraid of. To perform it you should be acquainted with the client-server application testing – and learn some details.

5. Useful Links

1. Description of starting work with RDP
<http://www.microsoft.com/windowsxp/using/mobility/getstarted/remotefintro.mspix>
2. What is Terminal Service:
 - 2.1. <http://technet.microsoft.com/en-us/library/cc784146.aspx>
 - 2.2. [http://technet.microsoft.com/en-us/library/cc786615\(WS.10\).aspx](http://technet.microsoft.com/en-us/library/cc786615(WS.10).aspx)
3. Download RDP client:
<http://www.microsoft.com/downloads/details.aspx?FamilyId=26F11F0C-0D18-4306-ABCF-D4F18C8F5DF9&displaylang=en>
4. Protocol description: http://en.wikipedia.org/wiki/Remote_Desktop_Protocol
5. Feature list of the new RDP <http://blog.itechtalk.com/2009/windows-7-rdp-features>

Appendix A: Some knowledge about RDP

1. Protocol RDP.

Remote Desktop Connection provides remote control above any computer, running under the Microsoft Windows system in a TCP/IP local area network or the Internet. The user can see a remote desktop on his own machine, use its resources and devices according to settings and assigned permissions and have mouse and keyboard control above the remote computer.

This connection is implemented by RDP protocol.

There are many versions of RDP. The most popular are the following: 5.0, 5.1, 5.2, 6.0, and 6.1[1]. There is a new version 6.1.7000 for Windows 7[2].

2. Features

Main Remote Desktop features are:

2.1. Display features:

- Color quality can be changed in range [8, 32]-bit
- Remote desktop size supports range from 640 x 480 to 1280 x 1024 pixels and full screen.
- Multiple monitors are supported. The Remote Desktop can be spread across several monitors up to 4096 x 2048 pixels.

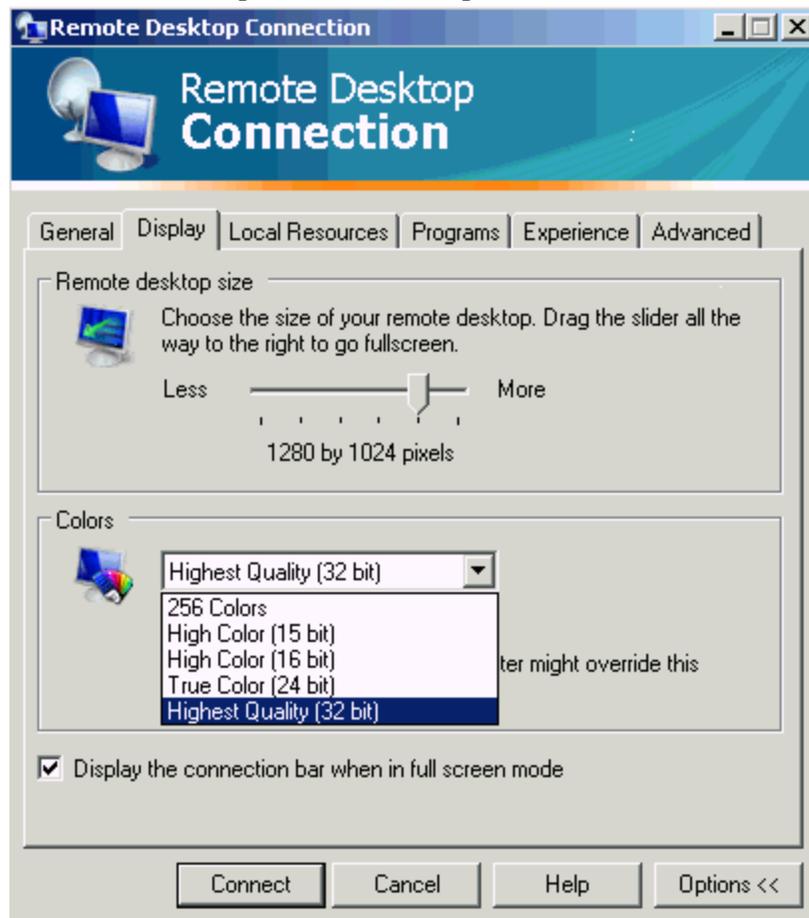


Figure 2

2.2. Local devices and resources redirection:

- Sound Redirection feature allows users to have the remote desktop sound on their local computer.
- Drive Redirection feature allows users to use their local drives on a remote desktop within the terminal session.
- Smart cards features

- Port Redirection feature allows to get access to local serial and parallel ports within the terminal session.
- Redirect Plug and Play devices.
- Printer Redirection feature allows using local printer as locally or network shared printer within the terminal session.
- Clipboard Sharing feature allows copy\past operations between the remote computer and the local computer

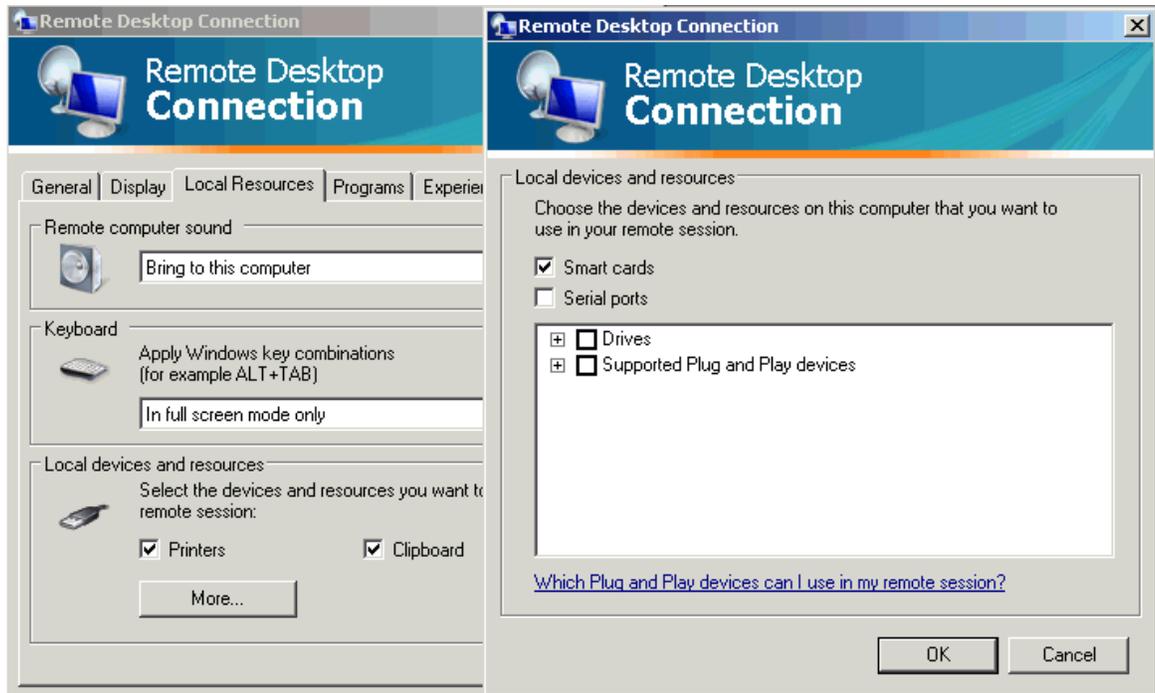


Figure 3

2.3. Remote Programs starting feature:

This feature allows to start remote programs from specified folder within terminal session

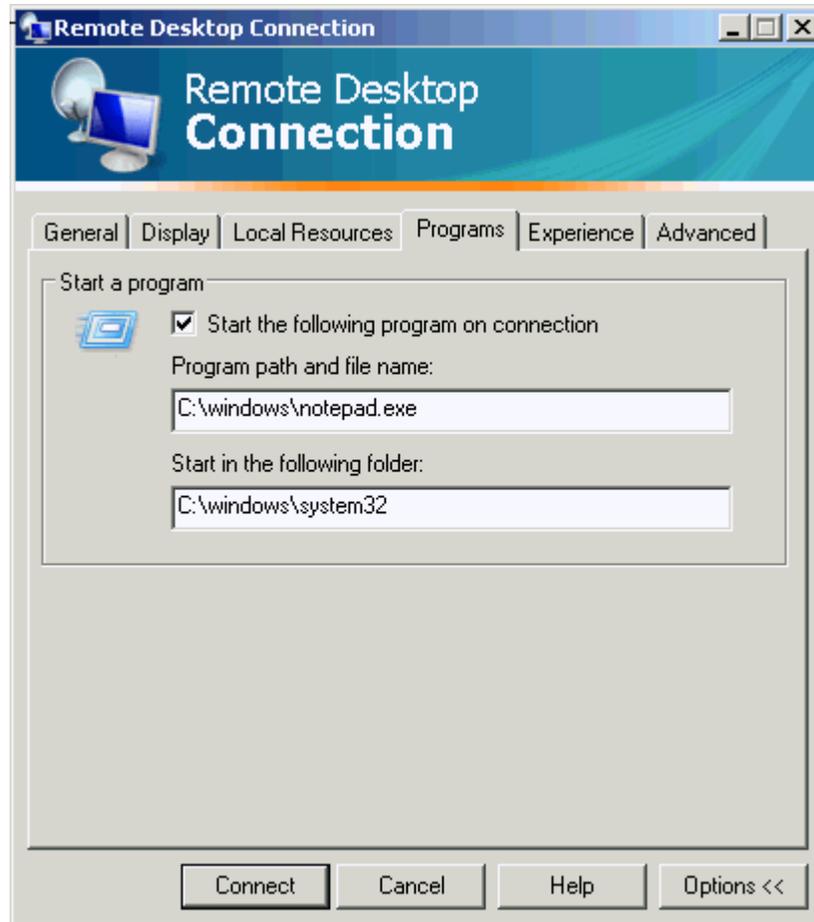


Figure 4

2.4. Advanced Remote Desktop features:

- Server authentication - Security measure that verifies if you are connecting to the correct remote computer or server.
- Support for Transport Layer Security (TLS) 1.0 on both server and client sides (set as default).

TS Gateway servers - A gateway that enables authorized users to connect to remote computers in a corporate network from any computer by using Internet connection with secure HTTPS protocol, using port 443 instead of default port 3389 for Remote Desktop Connection and allowing RDP connections without a VPN or firewall.

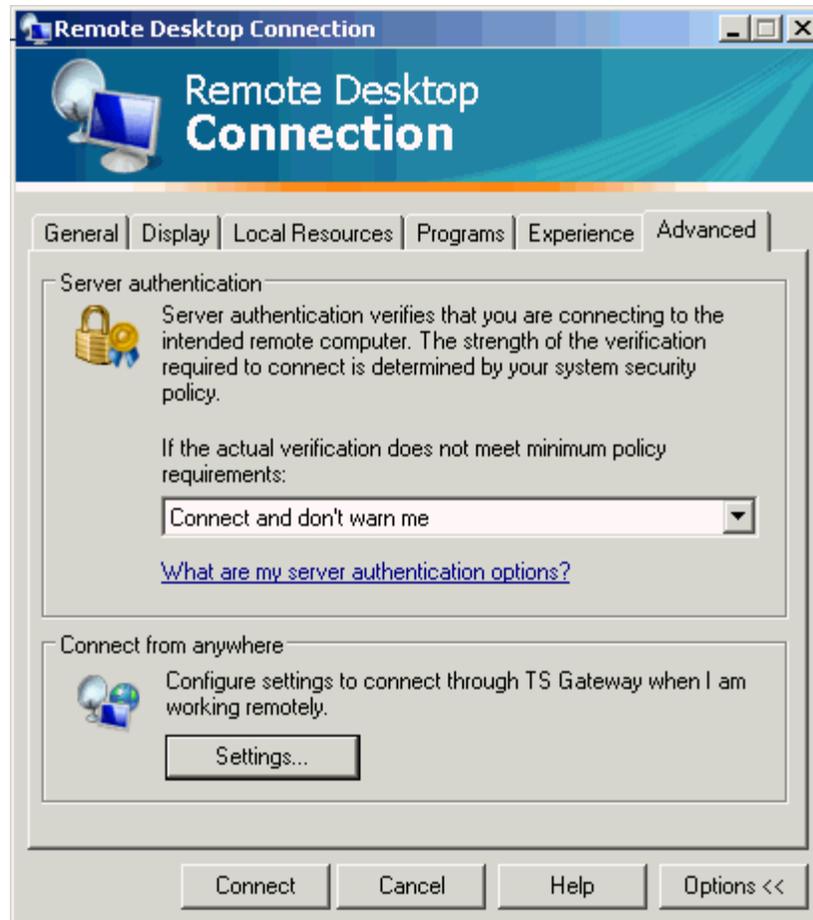


Figure 5

3. Deploying Remote Desktop

- 3.1. Remote Desktop Connection client is a part of all editions of modern Windows OS by default. There are two files `C:\windows\system32\mstsc.exe` and `C:\windows\system32\mstscax.dll` which are executable files of RDC client. Change them to get new version of terminal service client.
- 3.2. Enable Remote Desktop on a remote computer. Check the 'Allow users to connect remotely to this computer' checkbox via System Properties -> Remote tab (see pic5)
- 3.3. Enable users to connect to the remote computer. Add remote desktop users via System Properties -> Remote tab-> Remote Desktop Users.

You must have Administrator's permissions to make these deploying.

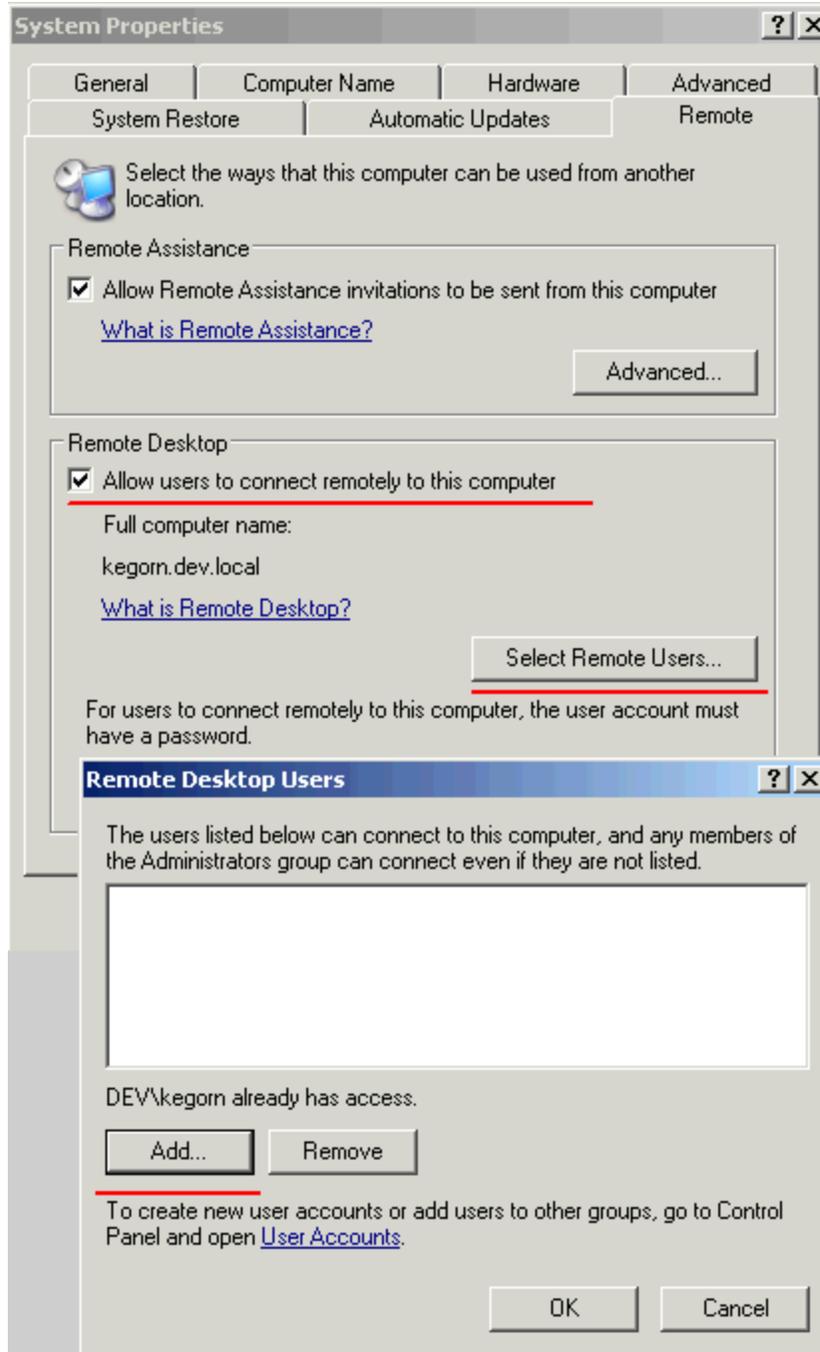


Figure 6

Appendix B: Samples of the Test Cases

1. Testing multiuser scenarios

- **Multi sessions _ Two sessions from one client to one server**

Pre-conditions:

Client1 is a client machine where the client part of the Application is installed.

Server1 is a server machine where the server part of the Application is installed.

Step name	Description	Expected result
Step 1	Connect to Server1 from Client1 via RDP to start Session1	The terminal session1 starts successfully. CPU and Virtual Memory Usage on Client and Server are in norm.
Step 2	Run the Application and check its basic functionality in Session1	The Application starts and works correctly in Session1. CPU and Virtual Memory Usage on Client and Server are in norm.
Step 3	Connect to Server1 from Client1 via RDP to start Session2	Session2 starts successfully. There are two opened remote desktops on Client1. CPU and Virtual Memory Usage on Client and Server are in norm.
Step 4	Run the Application and check its basic functionality in Session1	The Application starts and works correctly in Session2 and Session1 simultaneously. CPU and Virtual Memory Usage on Client and Server are in norm.

- **Multi sessions _ Two sessions from one client to two servers**

Pre-conditions:

Client1 is a client machine where the client part of the Application is installed.

Server1 is the first server machine where the server part of the Application is installed.

Server2 is the second server machine where the server part of the Application is installed.

Step name	Description	Expected result
Step 1	Connect to Server1 from Client1 via RDP to start Session1	The terminal session starts successfully. CPU and Virtual Memory Usage on Client and Server are in norm.
Step 2	Start an application and check its basic functionality in Session1	The application starts and works correctly in Session1 CPU and Virtual Memory Usage on Client and

		Server are in norm.
Step 3	Connect to Server2 from Client1 via RDP to start Session2	The terminal session starts successfully. There are two opened remote desktops on Client1. CPU and Virtual Memory Usage on Client and Server are in norm.
Step 4	Start the Application and check its basic functionality in Session2	The application starts and works correctly in Session2 and Session1 simultaneous. CPU and Virtual Memory Usage on Client and Server are in norm.

- **Multi sessions _ Two sessions from two clients to one server**

Pre-conditions:

Client1 is the first client machine where the client part of the Application is installed.

Client2 is the second client machine where the client part of the Application is installed.

Server1 is a server machine where the server part of the Application is installed.

Step name	Description	Expected result
Step 1	Connect to Server1 from Client1 via RDP to start Session1	The terminal session starts successfully on Client1. CPU and Virtual Memory Usage on Client and Server are in norm.
Step 2	Start an Application and check its basic functionality in Session1	An Application starts and works correctly in Session1. CPU and Virtual Memory Usage on Client and Server are in norm.
Step 3	Connect to Server1 from Client2 via RDP to start Session2	The terminal session starts successfully on Client2. CPU and Virtual Memory Usage on Client and Server are in norm.
Step 4	Start the application and check its basic functionality in Session2	The application starts and works correctly in Session2 and Session1 simultaneous. CPU and Virtual Memory Usage on Client and Server are in norm.

- **Multi sessions _ Maximum allowed count of sessions to one server**

Pre-conditions:

Client1 is the first client machine where the client part of the Application is installed.

Client2 is the second client machine where the client part of the Application is installed.

Client N is the umpteenth client machine where the client part of the Application is installed.

Server1 is a server machine where the server part of the Application is installed. Maximum allowed count of remote sessions to Server1 is N.

Step name	Description	Expected result
Step 1	Connect to Server1 from Client1 via RDP to start Session1	The terminal session starts successfully on Client1. CPU and Virtual Memory Usage on Client and Server are in norm.
Step 2	Start an Application and check its basic functionality in Session1	An Application starts and works correctly in Session1. CPU and Virtual Memory Usage on Client and Server are in norm.
Step 3	Connect to Server1 from Client2 via RDP to start Session2	The terminal session starts successfully on Client2. CPU and Virtual Memory Usage on Client and Server are in norm.
Step 4	Start an Application and check its basic functionality in Session2	An Application starts and works correctly in Session2 and Session1 simultaneous. CPU and Virtual Memory Usage on Client and Server are in norm.
...
Step N	Connect to Server1 from ClientN via RDP to start SessionN	The terminal session starts successfully on Client N. CPU and Virtual Memory Usage on Client and Server are in norm.
Step N+1	Start an Application and check its basic functionality in Session N	An Application starts and works correctly in Session1, Session2 ... Session N simultaneous. CPU and Virtual Memory Usage on Client and Server are in norm.

2. Application functioning for the different users

- **Permissions _Connect to Server as Guest**

Pre-conditions:

Client is a client machine where the client part of the Application is installed

Server is a server machine where the server part of the Application is installed.

Step name	Description	Expected result
Step 1	Connect to Server from	The terminal session starts successfully on

	Client via RDP by Guest user	Client.
Step 2	Start an Application and check its basic functionality	An Application starts and works correctly according to Guest permissions.

- **Permissions _Connect to Server as User**

Pre-conditions:

Client is a client machine where the client part of the Application is installed

Server is a server machine where the server part of the Application is

installed. There is a User on the server with usual user rights.

Step name	Description	Expected result
Step 1	Connect to Server from Client via RDP by User	The terminal session starts successfully on Client.
Step 2	Start an Application and check its basic functionality	An Application starts and works correctly according to User permissions.

- **Permissions _Connect to Server as Administrator**

Pre-conditions:

Client is a client machine where the client part of the Application is installed

Server is a server machine where the server part of the Application is

installed. There is an Administrator user on the server with usual administrator rights.

Step name	Description	Expected result
Step 1	Connect to Server from Client via RDP by Administrator	The terminal session starts successfully on Client.
Step 2	Start an Application and check its basic functionality	The application starts and works correctly according to Administrator permissions.

3. Stress test set for the application with terminal

- **Stress testing _Close a RDP session**

Pre-conditions:

Client is a client machine where the client part of the Application is installed

Server is a server machine where the server part of the Application is installed.

Step name	Description	Expected result
Step 1	Connect to Server from Client via RDP	The terminal session starts successfully on Client.
Step 2	Start an Application.	The Application starts and works correctly.

Step 3	Close RDP session while Application is working.	“Disconnect Windows session” message appears.
Step 4	Click the OK button.	The RDP session is closed.
Step 5	Again connect to Server from Client via RDP.	The terminal session starts successfully on Client.
Step 6	Start an Application and check its basic functionality	An Application starts and works correctly according to Administrator permissions.

- **Stress testing _ Disable network**

Pre-conditions:

Client is a client machine where the client part of the Application is installed

Server is a server machine where the server part of the Application is installed.

Step name	Description	Expected result
Step 1	Connect to Server from Client via RDP	The terminal session starts successfully on Client.
Step 2	Start an Application.	The application starts and works correctly.
Step 3	Disable network while Application is working.	Network is disabled. RDP session soon closes.
Step 4	Again connect to Server from Client via RDP.	The terminal session starts successfully on Client.
Step 5	Start The application and check its basic functionality	The application starts and works correctly according to Administrator permissions.

- **Stress testing _ Unplug a network cable**

Pre-conditions:

Client is a client machine where the client part of the Application is installed

Server is a server machine where the server part of the Application is installed.

Step name	Description	Expected result
Step 1	Connect to Server from Client via RDP	The terminal session starts successfully on Client.
Step 2	Start an Application.	The application starts and works correctly.
Step 3	Unplug a network cable while Application is working.	Network is disabled. The RDP session is closes in some time.
Step 4	Again connect to Server from Client via RDP	The terminal session starts successfully on Client.
Step 5	Start The application and check its basic functionality	The application starts and works correctly according to Administrator permissions.

- **Stress testing _ Disconnect a RDP session**

Pre-conditions:

Client is a client machine where the client part of the Application is installed

Server is a server machine where the server part of the Application is installed.

Step name	Description	Expected result
Step 1	Connect to Server from Client via RDP	The terminal session starts successfully on Client.
Step 2	Start an Application.	The application starts and works correctly.
Step 3	Disconnect a RDP session while Application is working.	The RDP session is closed.
Step 4	Connect to Server from Client via RDP.	The terminal session starts successfully on Client.
Step 5	Start The application and check its basic functionality	The application starts and works correctly according to Administrator permissions.

- **Stress testing _ Log off a RDP session**

Pre-conditions:

Client is a client machine where the client part of the Application is installed

Server is a server machine where the server part of the Application is installed.

Step name	Description	Expected result
Step 1	Connect to Server from Client via RDP	The terminal session starts successfully on Client.
Step 2	Start an Application	The application starts and works correctly
Step 3	Log off from RDP session while Application is working	The Log off Windows message appears
Step 4	Click the Log off button	The RDP session is closed
Step 5	Connect to Server from Client via RDP again.	The terminal session starts successfully on Client.
Step 6	Start the application and check its basic functionality	The application starts and works correctly according to Administrator permissions.

- **Stress testing _ Restart a Server**

Pre-conditions:

Client is a client machine where the client part of the Application is installed

Server is a server machine where the server part of the Application is installed.

Step name	Description	Expected result
Step 1	Connect to Server from Client via RDP	The terminal session starts successfully on Client.
Step 2	Start an Application	The application starts and works correctly
Step 3	Restart Server from RDP session while Application is working	Server restarts successfully. The RDP session is closed.
Step 4	Connect to Server from Client via RDP again.	The terminal session starts successfully on Client.
Step 5	Start the application and check its basic functionality	The application starts and works correctly according to Administrator permissions.