

Virtual Network Computing

Virtual Network Computing (VNC) is a remote display system which allows you to view a computing 'desktop' environment not only on the machine where it is running, but from anywhere on the Internet and from a wide variety of machine architectures. This package contains a client which will allow you to connect to other desktops running a VNC server.

Configuration

You don't have to configure much on the server side (server = the computer which will run the session, so perhaps your Linux desktop pc), but you can control what applications to run in your VNC session. The startup script for VNC is `~/ .vnc/xstartup`. (the directory `.vnc` will be created the first time you run a vnc session, or when you install one of our template configurations mentioned below, but if you want to create your own setup before starting a session, make sure to create the directory first)

Note: if you want to run KDE Plasma in your vnc session, this is the setup:

```
#!/bin/sh
unset SESSION_MANAGER
vncconfig -iconic &
startplasma-x11 &
```

For Gnome, replace the last line with `gnome-session`. Templates for the various desktops are available for installation. Run one of these commands to get the version of your liking:

```
SfinxUpdate xstartup-kde
SfinxUpdate xstartup-gnome (*)
SfinxUpdate xstartup-cinnamon (*)
SfinxUpdate xstartup-lxde
SfinxUpdate xstartup-mate
SfinxUpdate xstartup-openbox
SfinxUpdate xstartup-xfce
```

(*) these desktops are often too heavy for a vnc session, since by default, they have 3D graphical desktop effects enabled, that need to be emulated in software, slowing down your session (and the host computer) without added benefits. Use at your own risk!

So in general, a more lightweight xstartup would be preferable! See [VNC desktop issues](#) for details.

Starting a session

A VNC session is started using the command **vncserver**. The first time you run this command, it will ask you for a password, as an additional security measure (this password can later be changed using the `vncpasswd` command, or by removing `~/ .vnc/passwd`)

Suppose your desktop is named **eendracht.strw.leidenuniv.nl** (so, replace this name with the name of your desktop wherever it is appropriate). The vncserver command will then print out a message like this:

```
New 'eendracht:1 (jansen)' desktop is eendracht:1
Starting applications specified in /home/jansen/.vnc/xstartup
Log file is /home/jansen/.vnc/eendracht:1.log
```

Remember the desktop name/number (e.g. eendracht:1), you will need it later to connect, since there can be more than one VNC session on the same computer.

Security

VNC will ask you to set a password. But for greater security, you may also want to restrict access to the running session, using the `-localhost` option on the vncserver command. With that option in place, only the local host can connect to the session (still requiring a password as well); for remote access to the session, you will have to connect through ssh, as explained below in the viewer-part of this page. Without `-localhost`, machines on the same network as yours will be able to connect with just the plaintext password.

Starting the viewer

From the Linux commandline, you can start the viewer using `vncviewer eendracht:1` (with of course the name and session number that you got when starting vncserver). If you run the viewer on the same machine as the server, you can omit the hostname, and just use the `:1` to tell vncviewer which session to connect to.

There is a graphical interface to start the viewer and set options, this is called `tsclient` (or Terminal Service Client in the menu). This same program can also interface to Windows terminal services so make sure to select “VNC” as the protocol.

For accessing a VNC session running on another computer (either at the observatory, or somewhere else in the world), our Fedora desktop computers come with a list of VNC viewers to choose from:

- vncviewer: the basic commandline viewer (part of TigerVNC)
- vinagre: Gnome VNC,RDP viewer.
- remmina: Gnome VNC, RDP, NX and SSH client. Very configurable, reliable and fast

There are of course also VNC viewers for other operating systems, eg MacOS and Windows:

- Mac OS X has a built-in VNC viewer, named Screen Sharing from Finder, in the Go menu, Connect to server
- [RealVNC VNC Connect](#) viewer for almost any OS
- [Chicken, VNC viewer for Mac](#)
- WinVNC
- TightVNC (Java, works on any current OS)

Starting the viewer from outside, through ssh

You cannot directly connect to a vncserver from outside the institute, since VNC itself is only minimally encrypted, and not secure enough to be exposed to the net. The solution is then to combine VNC with a SSH tunnel. And since many of our servers and compute nodes and clusters are not directly reachable from outside, you may need to set up a ssh ProxyJump (see [SSH tips and tricks](#)).

Now we could try to describe how to set up such a tunnel, but the mechanism is known to vncviewer, so usually it is as easy as adding `-via` and a hostname to handle the ssh setup. Now, since all our Linux computers are reachable through ssh, it is usually best to use the same computer for ssh that is running your vnc session. The command to start the viewer will then be:

```
vncviewer -via eendracht.strw.leidenuniv.nl :1
```

(no hostname necessary before the `:1`, once ssh has connected to the host, everything else is local)

Just in case you cannot connect directly to the machine where you run your VNC session, you can connect through the machine `ssh.strw.leidenuniv.nl`:

```
vncviewer -via ssh.strw.leidenuniv.nl eendracht:1
```

So, this sets up an ssh connection through `ssh.strw.leidenuniv.nl` and then starts viewing the VNC session at `eendracht:1`.

VNC over SSH tunnels for other VNC viewers

Many other VNC clients have options to set up the SSH tunnel. See the documentation for your VNC viewer for further details.

Setting up port forwarding manually

In cases where you cannot directly use the ssh tunnel from vncviewer (e.g., you're not on Linux), you will have to set up the tunnel yourself.

The most important part to know is, that SSH forwards TCP ports, and for VNC sessions, the TCP port in use is `5900 + the VNC session number`, so it will be `5901` for the `:1` session. The other important thing to know is, what a tunnel actually does: it creates an encrypted connection between a local port on your computer, to a port on the remote computer. Once that is done, you want to point your VNC client to the **local** port, and let SSH take care of the rest.

Example for commandline ssh on Mac OS X or Linux, in the example where your vnc runs on a computer called `eendracht`, and you have to connect through the ssh gateway `ssh.strw.leidenuniv.nl` (and your vnc is on `:1` so the port is `5901`):

```
ssh -NfL 5901:eendracht:5901 username@ssh.strw.leidenuniv.nl
```

This forwards local port 5901 to port 5901 on the remote computer. So now you can connect locally with:

```
vncviewer :1
```

Setting up port forwarding through a graphical user interface

If you would like to configure port forwarding through a gui, good choices are: putty for Windows and Linux, bitvise for Windows; or ssh tunnel manager for Mac OS X

See also

- [Real VNC](#) (the VNC implementation in use at our computers)
- [Set up VNC on MAC OS X](#)
- [Set up VNC on MAC OS X \(nederlandstalig\)](#)
- [VNC desktop issues](#) Some things just don't work well over VNC

From:

<https://helpdesk.physics.leidenuniv.nl/wiki/> - **Computer Documentation Wiki**

Permanent link:

<https://helpdesk.physics.leidenuniv.nl/wiki/doku.php?id=vnc>

Last update: **2025/10/07 12:19**

