**Using Remote X Clients**  
**You should always use ssh to forward X instead of using below steps. Since Linux’s security is increased, below steps won’t enable you to start X from client.**

Suppose your local network contains two machines. The computer called 192.168.100.101 is a powerful machine that hosts important programs like word processors and data analysis utilities.  
The computer called 192.168.100.102 is a much less powerful system, but it has an adequate monitor  
and keyboard. Therefore, you want to sit at 192.168.100.102 and run programs that are located on  
192.168.100.101. Both systems run Linux. To accomplish this task, follow these steps:  
**1.** Log into 192.168.100.102, and if it’s not already running X, start it.  
**2.** Open a terminal (such as an xterm) on 192.168.100.102.  
**3.** Type **xhost +192.168.100.101** in 192.168.100.102’s terminal. This command tells 192.168.100.102 to accept for  
display in its X server data that originates on 192.168.100.101.  
**4.** Log into 192.168.100.101 from 192.168.100.102. You might use Telnet or Secure Shell (SSH), for instance.  
The result should be the ability to type commands in a shell on 192.168.100.101.

ssh –X root@192.168.100.101

**5.** On 192.168.100.101, type **export DISPLAY=192.168.100.102:0.0** (This assumes that you’re using bash;  
if you’re using tcsh, the command is **setenv DISPLAY 192.168.100.102:0.0**.) This command  
tells 192.168.100.101 to use 192.168.100.102 for the display of X programs. **Note**: IP is the local workstation’s IP where you want the GUI application to be displayed.

DISPLAY: This variable identifies the display used by the X windows server. It’s usually set to :0.0, which indicates the first display on the current computer. You can run multiple X sessions on one computer, in which case each one gets a different DISPLAY number—for instance, :0.0 for the first session and :1.0 for the second session. When you use X in a networked environment, this value may be preceded by the name of the computer at which you’re sitting, as in machine4.luna.edu:0.0. This value is set automatically when you log in, but you may change it if necessary.  
**6.** Type whatever you need to type to run programs at the 192.168.100.101 command prompt. For  
instance, you could type **loffice** to launch LibreOffice. You should see the programs  
open on 192.168.100.102’s display, but they’re running on 192.168.100.101—their computations use 192.168.100.101’s  
CPU, they can read files accessible on 192.168.100.101, and so on.  
**7.** After you’re done, close the programs you’ve launched, log off 192.168.100.101, and type **xhost**  
**-192.168.100.101** on 192.168.100.102. This tightens security so that an intruder on 192.168.100.101 won’t be able to  
modify your display on 192.168.100.102

**Note:**

As an added security measure, many Linux distributions today configure X to ignore true  
network connections. If your distribution is thus configured, the preceding steps won’t work;

when you try to launch an X program from the remote system, you’ll get an error message.  
To work around this problem, you must make an additional change, depending on how X  
is launched:  
**GDM** On older versions of GDM, check the GDM configuration file (typically /  
etc/X11/gdm/gdm.conf), look for the line DisallowTCP=true, and change it to read  
DisallowTCP=false. On newer versions of GDM, edit /etc/gdm/custom.conf and add a line  
that reads DisallowTCP=false to the [security] section (adding the section if required).  
**KDM or XDM** These two XDMCP servers both rely on settings in the Xservers file (in  
/etc/X11/xdm for XDM, and in this location or some other highly variable location for  
KDM). Look for the line that begins with :0. This line contains the command that KDM  
or XDM uses to launch the X server. If this line contains the string -nolisten tcp, remove  
that string from the line. Doing so eliminates the option that causes X to ignore conventional network connections.  
**Special openSUSE Configuration** In openSUSE, you must edit /etc/sysconfig/displaymanager and set the DISPLAYMANAGER\_XSERVER\_TCP\_PORT\_6000\_OPEN option to yes.  
**X Launched from a Text-Mode Login** If you log in using text mode and type **startx**  
to launch X, you may need to modify the startx script itself, which is usually stored in /  
usr/bin. Search this script for the string -nolisten tcp. Chances are that this string will  
appear in a variable assignment (such as to defaultserverargs) or possibly in a direct call  
to the X server program. Remove the -nolisten tcp option from this variable assignment  
or program call.  
Once you’ve made these changes, you’ll need to restart X as described earlier in  
“Running an XDMCP Server.” Thereafter, X should respond to remote access requests.

Another option for running X programs remotely is to use the Virtual Network  
Computing (VNC) system (www.realvnc.com). VNC runs a special X server on the computer that is designed to be used from a distance, and a special VNC client runs on the  
computer at which you sit. You use the client to contact the server directly. This reversal of  
client and server roles over the normal state of affairs with conventional X remote access  
is beneficial in some situations, such as when you’re trying to access a distant system from  
behind certain types of firewalls. VNC is also a cross-platform protocol; it’s possible to  
control a Windows or Mac OS system from Linux using VNC, but this isn’t possible with  
X. (X servers for Windows and Mac OS are available, enabling you to control a Linux  
system from these non-Linux OSs.)