

Table of Contents

This is the Lbd group hints page on working cross-platform.....	1
Preamble.....	1
Local Gateway.....	1
Non-Unix Host.....	1
- Dual Booting.....	1
- Virtual Machines.....	2
Windows host.....	2
- Unix compiled for windows.....	2
- SSH with an x-server.....	3
- VNC - remote desktop system.....	3
Useful programs for windows/linux.....	4

This is the Lbd group hints page on working cross-platform

Preamble

Some users prefer Unix, some prefer windows, some prefer Mac. The key thing is to think about which is best for you, and then maximise your productivity by using the many available tools to work cross-platform when you need to.

- X-servers, Virtual machines, SLC5, dual-booting, VNC [↗](#) and NoX [↗](#) are all possible alternatives.
- Virtual Machines [↗](#), SLC5 and dual-booting laptops let you install LHCb software locally, but then you must maintain the software distribution yourself.
- CernVM is a virtual machine with lhcb software pre-installed, but with a minimal SLC5 kernel. You will find you are missing several unix commands and that it is in most cases unsatisfactory as a full desktop environment.
- Connecting up your SLC5 or VM to afs allows you to use the afs-installed software locally, but it is difficult to obtain a good performance from afs without a lot of tinkering.
- VNC [↗](#) and NoX [↗](#) are both remote-desktop-equivalents for unix-based platforms

Local Gateway

Lbd has it's own local SLC5 gateway on the CERN network running LHCb software and afs, with a NoX [↗](#) server. You can find instructions for this at CernLbdGateway. This can be used on any host, and gives you a great cross-platform experience, you can always see the same desktop, and don't need to worry about forgetting the name of the machine.

Non-Unix Host

There are five basic ways to use Linux/UNIX commands on your windows system.

To natively run LHCb software

- Dual Boot into Linux and your favourite OS (windows/Mac)
- Run a virtual machine, either CernVM or Ubuntu

To remotely run LHCb software

- Use a secure shell connection, to a UNIX machine once running an X server
- Use a remote connection, to a UNIX machine once running a NoX [↗](#)/VNC [↗](#) server +ssh port forwarding
- Install a compiled set of binaries for using UNIX commands directly under windows

- Dual Booting

Can be problematic, especially for very new or very very old laptops, but if it works it is one of the best options. However you may find it difficult to share files between OS-es, and find it annoying that you need to hibernate/resume to run your favourite programs. You also can't cut/paste between dual boots, of course.

- Virtual Machines

ThomasRuf keeps a fantastic description of virtual machines [here](#) and [here](#)

- Virtual machines have almost all the advantages of dual booting, without the drawbacks of having to restart to switch, and usually simplify cut/pasting and file sharing.
- CernVM is a virtual machine with lhcb software pre-installed, but with a minimal SLC5 kernel. You will find you are missing several unix commands and that it is in most cases unsatisfactory as a full desktop environment.

Windows host

Windows comes without any nice X-server, and without any nice command-line interface. If you want to avoid having to pay for your X-server on your private machine, here are some alternatives.

- Unix compiled for windows

Cygwin [is](#) a compiled set of UNIX binaries and scripts which runs well on the Win32 kernel (no evidence of compatibility with 64 bit core as of yet). It will be slower than using UNIX directly i.e. through dual booting, but allows you to use UNIX and windows at the same time.

Cygwin is FREE!

The Cygwin installer is tricky to use, you need around 1.5GB free to install and 1GB for temporary files downloaded during installation. Cygwin, (which is a different compilation to Cygwin-X), also comes with an X-server.

To install correctly, with an X-server and all features:

- Download the installer for Cygwin [is](#)
- Run the installer to "download without installing", this will help you later. The files created can be deleted later.
- Select a directory with a lot of space for these files ~1GB is enough
- **DON'T** install everything, it doesn't work
- **DON'T** use the default installation, only very basic things are added
- Select from the list of components those that you'd like to get,
 - ◆ do this by changing the "skip" or "default" to "full"
 - ◆ make sure you select "shells", "graphics" and "X11".
 - ◆ Remember what you selected! It will be needed later.
- Switch the view to the "Not Required, Skip, Not installed" view using the button at the top right
 - ◆ Select any component with X11 in the name that you missed.
- **DON'T** go back to the package view, as this will reset everything to default
- Next then should download the required files
- Run the installer again, with "install from local directory"
- Select only the components you downloaded earlier
- Select all the components you downloaded earlier
- Switch the view to the "Not Required, Skip, Not installed" view using the button at the top right
 - ◆ Select everything, as this is a list of the downloaded files you have missed
- **DON'T** go back to the package view, as this will reset everything to default
- Really you should install to your C-drive as long as you have enough space.
- Just click OK, continue, whatever on any and all errors that appear during installation
- Run the Cygwin program once it has installed. This will initially establish local home directories

To find your X-server:

- Either: Use the command `startx` in the Cygwin window (this will open a new x terminal and x-server)
- OR: find the batch file `startxwin.bat` in `C:\Cygwin\usr\X11R6\bin` and make a shortcut to it
 - ◆ this will open just one (bash) window

- SSH with an x-server

SSH stands for secure shell. A secure shell connection allows you to log into a Unix system (as if you were sitting at the desk) and use the resources as per usual.

X11R6 is the current UNIX standard graphics interface, it is the protocol used to give you graphics such as windows, buttons, etc. X11 port forwarding is the name for the forwarding method/process by which graphics can be packaged and displayed over a connection, i.e. for ssh. To do this you need a compatible server at each end, Linux, MacOS, or windows running an X-server.

The best X server is eXceed, but this is expensive. Alternatively use Cygwin as above.

Once you have such a server, you should use X11 port forwarding, either by `ssh -X` in Cygwin/UNIX or by changing your options in the ssh client you have chosen.

[puTTY](#), [Cygwin](#), [SSH](#), provide ssh clients.

- VNC - remote desktop system

VNC stands for Virtual Network Computing and is a simple way of exporting your desktop for remote access anywhere with an internet connection.

If you're sick and tired of none-graphical interfaces and you want all the functionality of sitting in front of the machine itself, or even if you have a home PC you'd like to log into remotely from a unix machine Virtual Network Computing VNC is the way to go.

VNC is FREE!!

You need:

- A VNC client (not mandatory) installed on your local system
 - ◆ Try [TightVNC](#) or [RealVNC](#) (both are good). I'd recommend TightVNC.
 - ◆ UNIX -> try the Java version of TightVNC
- Java installed on your local system, and plugged into your browser of choice
- An SSH client (mandatory)
 - ◆ [puTTY](#), [Cygwin](#), [SSH](#), provide ssh clients.

Then:

- Run a `vncserver` on the HOST machine
 - ◆ **UNIX** -> open a terminal and type `> vncserver :XX` where XX is the display number you'd like to use (pick something between 0 and 99)
 - ◆ **Windows** -> Start the VNC server that you downloaded, should automatically start display 0 or 1 (port 5900 or 5901).
- Set up an SSH tunnel on the client machine
 - ◆ **UNIX** -> open a terminal and type `> ssh -X -L 59##:127.0.0.1:59XX username@hostname.com` where XX is the display number of the host server you set up (i.e.

5900+the number, 5901 for display :1) and ## is the port number you'd like to forward to (pick something between 0 and 99)

- ◆ Windows -> Cygwin, open a terminal and type `> ssh -X -L 59##:127.0.0.1:59XX username@hostname.com` where XX is the display number of the host server you set up (i.e. 5900+the number, 5901 for display :1) and ## is the port number you'd like to forward to (pick something between 0 and 99)
- ◆ **Windows** -> PuTTY, start a putty session.
 - ◇ Use the hostname of the computer at which you established the host,
 - ◇ In the SSH->Tunnels configure a new tunnel, Source Port: 59##, Destination port: 127.0.0.1:59XX, where XX is the display number of the host server you set up (i.e. 5900+the number, 5901 for display :1) and ## is the port number you'd like to forward to (pick something between 0 and 99)
 - ◇ Save the session as something memorable in the Session configuration
- Run a viewer on the client machine
 - ◆ Anywhere -> Most machines are not equipped with a viewer, so use a web-interface. Open a browser with Java and go to `http://hostname:58XX/` where XX is the display number of the host server you set up, or `http://ipaddress:58XX/` where ip address is the ip address of the host machine. This will only work for networks and machines without a million layers of protection. If you're going from windows server to linux client, this is the way to go.
 - ◆ UNIX: TightVNC java -> use the command `> java VncViewer HOST localhost PORT 59##`
 - ◆ Windows -> Run your client viewer to connect to localhost:## where ## is the port that you forwarded to in the previous step

Want your usual desktop?

- Windows server-> automatically gives you control of the active desktop
- UNIX/linux server -> edit the `~/ .vnc/xstartup` script (remove the comments from the two obvious places)

For more information on making the connection more secure, see here [↗](#)

Useful programs for windows/linux

- WinSCP [↗](#) Secure Copying, nice interface.
- GIMP [↗](#), (java) image package, particularly useful for converting TO postscript/eps for windows
- Emacs [↗](#) for windows
- Ghost View [↗](#) for windows, particularly useful for viewing postscript
- MiTeX [↗](#) latex compiler for windows
- Eclipse fantastic all-singin-all-dancing IDE and TeX editor (java) for any platform
- JaxoDraw [↗](#) (java) draw Feynman diagrams and export as postscript
- TightVNC [↗](#) (java) Remote Desktop across platforms
- NoX [↗](#) x-forwarding-free server

-- RobLambert - 15-Nov-2010

This topic: Main > CernLbdPlatform

Topic revision: r2 - 2010-11-16 - RobLambert



Copyright &© 2008-2020 by the contributing authors. All material on this collaboration platform is the property of the contributing authors.

Ideas, requests, problems regarding TWiki? Send feedback