

# Accessing Universe (Cosmos and Eric): Telnet, FTP, ssh and scp, the r-commands, X-Windows (eXceed and XFree86), VNC and Gnome Sessions

## Installation and Configuration of SSH on MS Windows:

A separate document is available<sup>1</sup> which describes in detail how to download and install the SSH client from SSH Communications on Microsoft Windows machines. Configuration for login to and file-transfer to/from Universe (Cosmos in particular) is also covered.

---

## 1. Telnet and FTP

At the time of writing, Telnet and FTP can be used to access Universe (both Cosmos and Eric) from on campus. However, both Telnet and FTP will shortly be blocked, so that SSH/SCP [Page 1] must be used from all locations.

(Access is already blocked from off-campus, from where SSH/SCP must be used).

### 1.1. About Telnet and FTP

During authentication (logging in) of both a Telnet and FTP session a user's password is sent over the network in clear text. Most of UMIST's network is switched and so this lack of security does not represent a significant risk. However, when accessing Cosmos and Eric from outside of UMIST, e.g., via dail-up, there *is* a real possibility that a third-party could recover both the username and corresponding password from an authentication attempt, thus compromising a user's account.

---

## 2. The r-commands: rlogin, rexec and rsh

The Berkeley r-commands cannot be used to access Universe since they are a security nightmare. (Currently rlogin and rsh can be used from Universe to access other hosts.)

---

## 3. SSH, SCP and SFTP

SSH and friends are a protocol suite of network connectivity tools that encrypt all traffic, including passwords, in order to eliminate eavesdropping, connection hijacking, and other network-level attacks.

SSH, SCP and SFTP clients may be freely downloaded from the Internet (see details below) for both MS Windows and Unix (Solaris and Linux).

SSH, or Secure Shell, is roughly-speaking, encrypted Telnet, i.e., offers a secure method of logging on to a remote machine; SCP and SFTP offer secure methods of copying files to and from remote machines, SFTP being similar in functionality and usage to the more familiar FTP.

---

<sup>1</sup> ../\_ianjr/sshdoc.html

- N.B. *To access Universe, a recent client which supports the keyboard-interactive mode of authentication will be necessary (and this must be selected as the preferred method, or at least on the list of methods attempted). Clients not supporting this mode will not be able to authenticate.*

For each client mentioned below, the number is given of a version which is known to authenticate to Cosmos and Eric successfully; it is expected that later versions will also work.

### 3.1. Solaris and Linux

There are two SSH (and SCP/SFTP) clients which are in common use: that from SSH Communications Security and the OpenSSH client. Both are simple but effective command-line utilities.

#### 3.1.1. SSH Communications Security (v3.2)

This client is available for free download from [www.ssh.com](http://www.ssh.com)<sup>2</sup> (follow the links to: Download, Secure Shell for Workstations, Non-commercial downloads): the source is available so that binaries can be compiled up for Linux and Solaris (experience shows that the source builds with no problems using GCC via the usual three steps, `./configure`, `make` and `make install`).

#### Usage for ssh

```
local_prompt> ssh <remotehost> -l <username_on_remotehost>
```

for example

```
powell> ssh cosmos.umist.ac.uk -l mnopqrs1
PAM authentication
Password:
Last login: Tue Dec 10 2002 09:50:52 from powell.csu.umist.
...
...
cosmos>
```

The first time you connect via ssh you may see something like this:

```
The authenticity of host 'cosmos (130.88.99.10)' can't be established.
DSA key fingerprint is 1f:eb:8b:2a:6c:38:04:30:ea:b9:e2:12:db:53:d9:7c.
Are you sure you want to continue connecting (yes/no)?
```

Assuming you believe that you have truly connected to cosmos, then answer yes! You will not see this question again.

#### Usage for scp

```
local_prompt> scp <file_to_send> <username@remote.host>:<remote_path>/<remote_filename>
```

for example

---

<sup>2</sup> <http://www.ssh.com>

## Accessing Universe (Cosmos and Eric)

```
talby> scp printtest.txt.ps mnopqrs1@cosmos.umist.ac.uk:tmp
warning: Need basic cursor movement capability, using vt100
.
.
Keyboard-interactive:
PAM authentication
Password:
printtest.txt.ps          | 15kB | 15.4 kB/s | TOC: 00:00:01 | 100%
talby>
```

### Usage for sftp

```
local_prompt> sftp <username@remote.host><remotehost>
```

For example:

```
talby> sftp mpciish2@cosmos.umist.ac.uk
Warning: Need basic cursor movement capability, using vt100
.
.
Keyboard-interactive:
PAM authentication
Password:
sftp>
```

Once authenticated, sftp behaves, from the user point of view, just like any other FTP client.

The first time you connect you may see something like this:

```
...Host key not found from database.
Key fingerprint:
xilim-dukak-focam-havac-curuf-fosur-nyvac-bedef-rugys-zaler-byxex
You can get a public key's fingerprint by running
% ssh-keygen -F publickey.pub
on the keyfile.
Are you sure you want to continue connecting (yes/no)?
```

Responding with yes results in the

```
Host key saved to /home/isd/.ssh2/hostkeys/key_22_cosmos.umist.ac.uk.pub
host key for cosmos.umist.ac.uk, accepted by isd Thu Mar 20 2003 13:19:44
```

You will not see this again.

### Authentication Method

The default authentication configuration may need to be changed (either to prevent authentication attempts by a method which will fail, or to ensure that the correct method is attempted).

The configuration is usually contained within `/etc/ssh2/ssh2_config` — change this

```
AllowedAuthentications    publickey,password
```

(or similar) to this

```
AllowedAuthentications    publickey,keyboard-interactive,password
```

### 3.1.2. OpenSSH (v3.1p1-6)

This client is available for free download from [www.openssh.org](http://www.openssh.org)<sup>3</sup>: binaries in RPM format are available for Linux; in addition the source is available so that binaries can be compiled up for Solaris. Binaries are also available from the RedHat site and its mirrors. Solaris binaries are available from [sunfreeware.com](http://sunfreeware.com)<sup>4</sup>.

Usage is as for the SSH Communications Security client.

The configuration is usually contained within `/etc/ssh/ssh_config`. According to the man page, the default authentication methods are, in order, `hostbased`, `publickey`, `keyboard-interactive` and `password`. Should the client initially attempt authentication by a method which fails, or fail to authenticate, edit the configuration file to change the defaults to include `keyboard-interactive` is tried and tried before failing methods, e.g:

```
Host *
ForwardX11 yes
PreferredAuthentications publickey,keyboard-interactive
```

For more information see the man pages at the OpenSSH website<sup>5</sup>.

## 3.2. MS Windows

### 3.2.1. SSH Communications Security (v3.2)

This client is available for free download from [www.ssh.com](http://www.ssh.com)<sup>6</sup>. The default configured authentication method will fail — choose *keyboard-interactive*.

### 3.2.2. PuTTY (v0.52 beta)

This client is available from free download from [www.chiark.greenend.org.uk](http://www.chiark.greenend.org.uk)<sup>7</sup>. The default configured authentication method will fail — choose *keyboard-interactive*

---

## 4. Graphical Access and X-Windows: XFree86, eXceed

X-Windows is the foundation (protocol) upon which most GUIs on Unix boxes are based. As you would expect processes (applications) running on the local machine can display in a window (or windows) on the local screen. In addition *processes running on a remote machine can display on the local box* — to achieve this an X-server must be running on both machines. Thus X-Windows can be used to have processes which are running on Cosmos, for example an `xterm`, an editor and an image-viewer, all display on the local desktop/screen.

X-servers are also available for MS Windows. *X-clients* — processes displaying a window on the desktop, whether on the machine local to the process or on a remote machine — can display their windows on the MS Windows desktop in one of two modes (see below).

---

<sup>3</sup> <http://www.openssh.org>

<sup>4</sup> <http://sunfreeware.com>

<sup>5</sup> <http://www.openssh.org/manual.html>

<sup>6</sup> <http://www.ssh.com>

<sup>7</sup> <http://www.chiark.greenend.org.uk/~sgtatham/putty/>

## **4.1. An Introduction to Using X-Windows**

An excellent introduction to using X-Windows is available in PDF from [www.rice.edu](http://www.rice.edu)<sup>8</sup>.

## **4.2. XFree86**

Most Linux distributions (and other free Unices) come with XFree86, a free, open-source version of X-Windows. Usually a remote process (e.g., on Cosmos) will display in its own window on a Linux machine's desktop.

A port of XFree86 to MS Windows has been part of CygWin<sup>9</sup> for some time now. When XFree86 is started a single large window is started on the MS Windows desktop — this is the X-desktop. All X-clients display their windows within this one large window. Usually a local window-manager (e.g., TWM) or environment (such as KDE with its window-manager, KWM) manages this X-desktop.

## **4.3. eXceed**

eXceed<sup>10</sup> is a commercial X-server for MS Windows machines — UMIST has a site license for this software. eXceed has two major modes: it may be configured to start one large window within which all X-clients display their windows (this will allow a remote machine, e.g., Cosmos, to manage your local X-desktop) or, more usually, eXceed has X-clients display their windows directly on the MS Windows desktop. (This second mode is usually faster.)

---

# **5. Secure X Sessions via SSH**

Given an encrypted connection via `ssh`, one can launch X-clients back to the local desktop in the usual way [Page 4], i.e., by ensuring the `DISPLAY` environment variable is correctly set and, if necessary, executing an `xhost` command on the local machine, but this would be perverse! `ssh` can be used to encrypt your X-Windows sessions.

## **5.1. MS Windows**

Coming soon...

### **5.1.1. Hummingbird eXceed**

### **5.1.2. XFree86 (CygWin)**

## **5.2. Unix (Solaris and Linux)**

The usual (unencrypted) method of launching an X11 session from a remote host back to a local one is:

---

<sup>8</sup> <http://www.rice.edu/Computer/Documents/Unix/unix2.pdf>

<sup>9</sup> <http://www.cygwin.com>

<sup>10</sup> <http://www.hummingbird.com>

## Accessing Universe (Cosmos and Eric)

```
local_host> telnet remote.host
Trying <ipaddress>
Connected to <remote_host>.
Escape character is '^]'.
Welcome to the UMIST General Purpose Unix Service
login: <username>
Password:
Last login: Tue Dec 10 09:50:53 from talby.csu.umist.
...
remote_host> export DISPLAY=<localhost>:0
remote_host> xterm &
remote_host>
```

This kind of X11 session connects, unencrypted, onto port 6000 on the local host

```
[mpciish2@cosmos:~]$ netstat | grep powell
cosmos.umist.ac.uk.telnet powell.csu.umist.ac.uk.36404 6610 0 10136 0 ESTABLISHED
cosmos.umist.ac.uk.42630 powell.csu.umist.ac.uk.6000 11424 0 8760 0 ESTABLISHED
```

One can instead have ssh handle the X11 session for you via *port-forwarding*. To achieve this you must either start your SSH connections with the either the +x or +X options, e.g., `ssh mpciish2@cosmos.umist.ac.uk +X` (see the SSH man page for details) or ensure your client default configuration is set up for this, for example (SSH Communications, `/etc/ssh2/ssh2_config`),

```
Tunneling
  ForwardX11 yes
```

or (OpenSSH, `/etc/ssh/ssh_config`),

```
Host *
  ForwardX11 yes
```

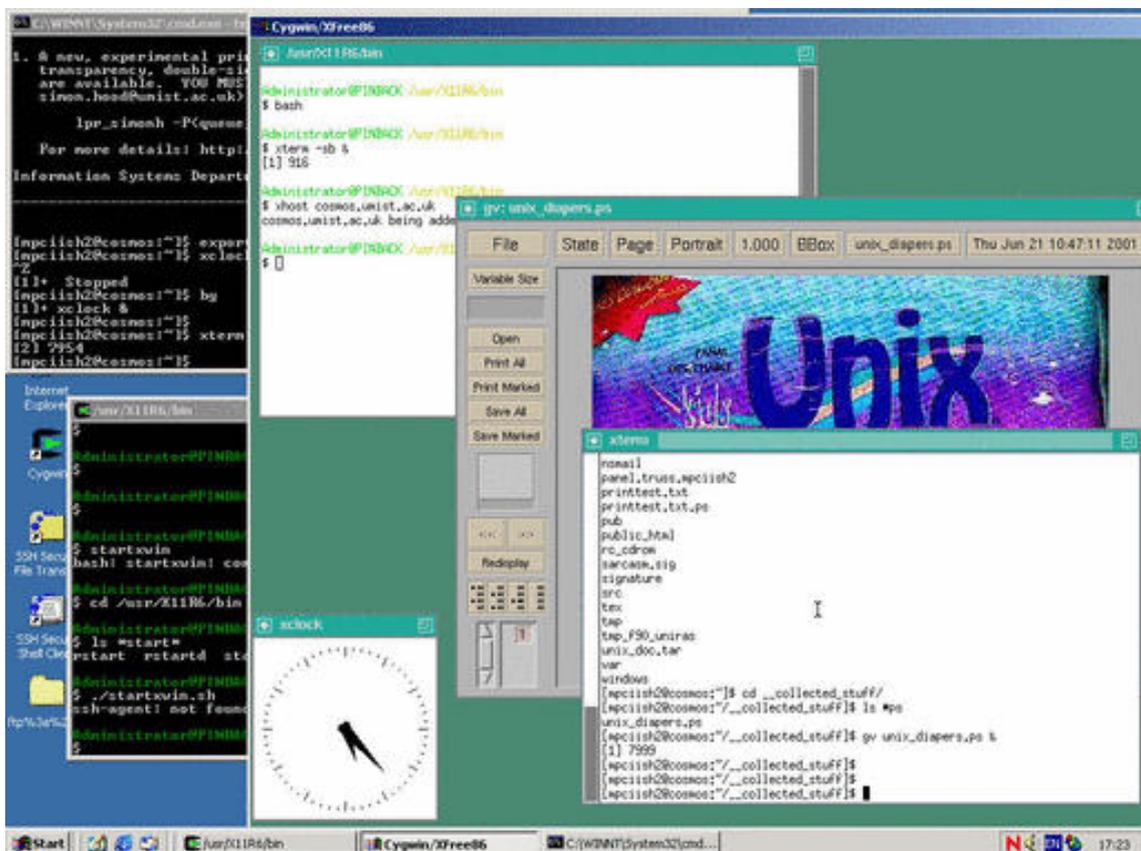
Then connect as usual, but *do not* manually set the DISPLAY environment variable — echoing the value should now show something slightly different from usual, for example `cosmos:10.0`. After starting a new X11 session (e.g., launching Emacs back from Cosmos to your local screen) we can confirm that Emacs is running through the ssh connection on port 22 (not the usual X11 port, 6000):

```
cosmos> echo $DISPLAY
cosmos:10.0
cosmos> emacs &
cosmos> netstat | grep talby
cosmos.umist.ac.uk.22 talby.csu.umist.ac.uk.36236 63360 0 10136 0 ESTABLISHED
cosmos>
```

Of course one *can* start an unencrypted X11 session after initially connecting via ssh by manually setting DISPLAY should one wish!

## 6. Having Cosmos Manage Your Local X Desktop on a MS Windows Machine

Using XFree86, or using eXceed in single-window mode, one can arrange for a remote X-server (e.g., that on Cosmos) to manage the local X-desktop. The upshot is that, menus, drag-and-drop and other GUI functionality will be available, just as if you were sat at the remote server's console — this is in addition to your own, local MS Windows WIMP environment (see figure — click to enlarge). In particular one can have Cosmos manage a Gnome Session [Page 10] on your local X-desktop.



*A Cosmos-managed X-desktop (showing two xterms — one local, the other logged into Cosmos — a clock and Ghostview displaying a picture) running within one large window on XFree86 installed with CygWin on a Windows 2000 machine. A MS Windows' command line and a CygWin bash shell are also visible (on the MS Windows desktop).*

The scenario illustrated in the figure was achieved by using XFree86:

1. After installation of CygWin and XFree86<sup>11</sup> on the (local) MS Windows machine, start XFree86 (e.g., by running the `startwin.bat` batch file).
2. Ensure Cosmos has permission to access your local X-desktop (e.g., on this X-desktop, within an `xterm` accessing *the local machine* type: `xhost cosmos.umist.ac.uk`).
3. Ensure the `DISPLAY` environment variable on Cosmos is set correctly, i.e., to the IP address of your local machine together with the correct X-desktop numeric value (e.g., within the `xterm`, `telnet` or (better) `ssh` to Cosmos, login and then type: `export DISPLAY=130.88.<sub-net>.<host-ip>:0.0`).
4. Finally, at the Cosmos prompt, start a window-manager — this will be displayed on your local X-desktop (e.g., on Cosmos type: `/usr/openwin/bin/twm &`).

A similar scenario can be achieved by using `eXceed` in single-window mode (see the separate documentation<sup>12</sup> on `eXceed`).

---

## 7. Having Cosmos Manage a Second Local Desktop on Your Local Unix Box via VNC

On a Unix box running X-Windows, the traditional way to make use of software running on a remote server such as Cosmos is to run X Windows locally (in conjunction with a local window-manager) and have applications running on the remote machine display in individual windows on the local screen. Usually such applications are started by use of the command line within a `telnet` (or `SSH`) session, or an `xterm`.

One can arrange for Cosmos (or other remote X-server) to manage a local desktop on a MS Windows machine [Page 6] — can one have Cosmos do the same on a local Unix box? Yes — in two ways. One can either start the local X-server without a local window-manager and have Cosmos manage the local (main) desktop — this would mean that you would not have access to your local GUI environment, so is not to be recommended — or one can have Cosmos manage a *second* X-desktop which appears as one large window on the first (cf. using XFree86 to do this [Page 6] on a MS Windows box, above).

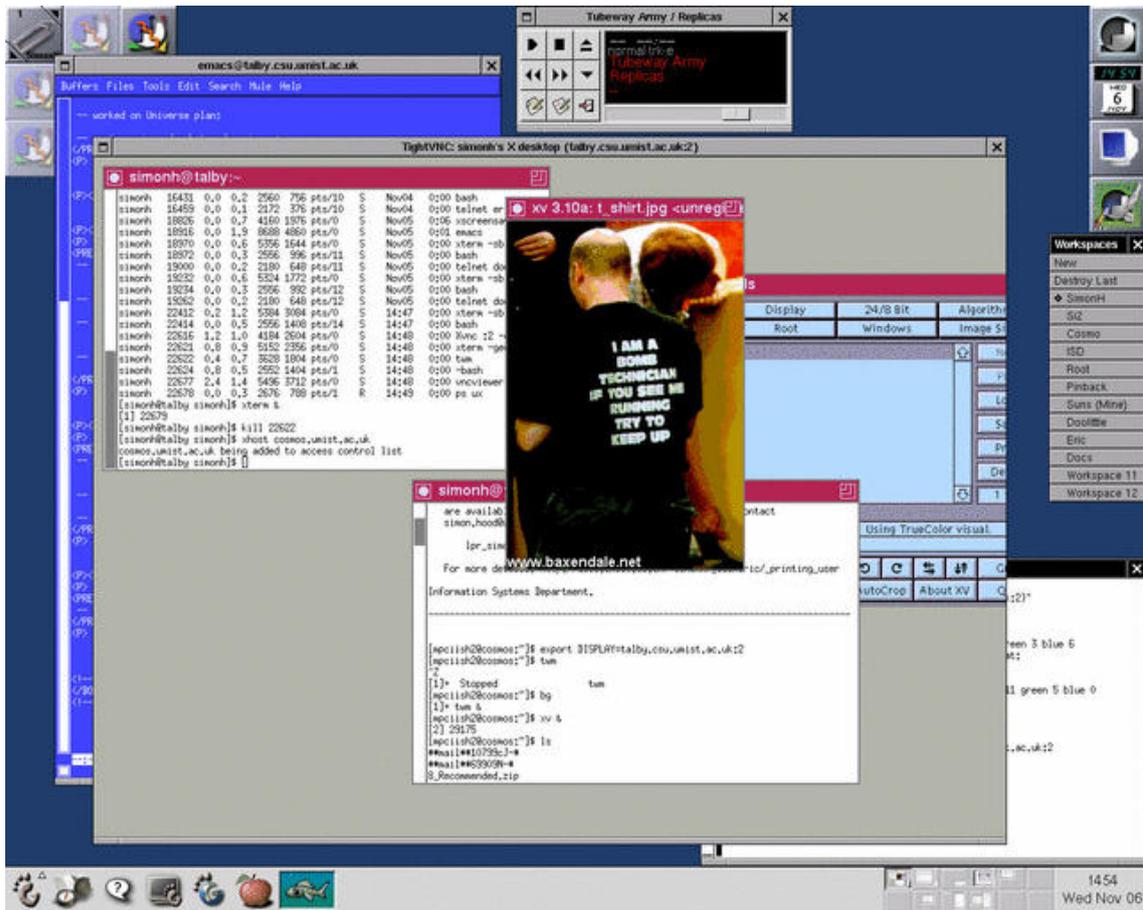
The upshot is that, menus, drag-and-drop and other GUI functionality will be available, just as if you were sat at the remote server's console. This is in addition to your own, local WIMP environment — *see figure* (click to enlarge).

In particular one can have Cosmos manage a Gnome Session [Page 10] on your local X-desktop.

---

<sup>11</sup><http://www.cygwin.com>

<sup>12</sup><http://www2.umist.ac.uk/isd/unixnew/cosmos/xwin.htm>



*A Cosmos-managed, second local X-desktop (showing two xterms — one local, one logged into Cosmos — and xv displaying a picture) running running within a VNC viewer running in a large window on the first local X-desktop (on a Linux box).*

The scenario illustrated in the figure was achieved by using VNC:

1. Ensure that a VNC Server and VNC viewer are both installed on the local machine.
2. Start the server, for example:

```
prompt> vncserver -depth 16
```

```
Warning: talby.csu.umist.ac.uk:1 is taken because of /tmp/.X11-unix/X1
Remove this file if there is no X server talby.csu.umist.ac.uk:1
```

```
Warning: talby.csu.umist.ac.uk:2 is taken because of /tmp/.X11-unix/X2
Remove this file if there is no X server talby.csu.umist.ac.uk:2
```

```
New 'X' desktop is talby.csu.umist.ac.uk:3
```

```
Starting applications specified in /home/simonh/.vnc/xstartup
Log file is /home/simonh/.vnc/talby.csu.umist.ac.uk:3.log
```

```
prompt>
```

N.B. The default depth is 8 bits which will deliver poor-looking graphics.

3. Start the vncviewer:

```
vncviewer talby.csu.umist.ac.uk:3
```

A window corresponding to the new desktop (in this case, 3) should open and in this applications specified in the VNC `xstartup` file will appear (this is usually located in `~/.vnc/`) — these should include an `xterm`. Ensure the remote server (e.g., Cosmos) has permission to access your local X-desktop: within the `xterm` type `xhost cosmos.umist.ac.uk`.

N.B. If your VNC `xstartup` file includes a window-manager (it probably will) you'll need to edit this out before starting the VNC viewer, or kill the window-manager after starting the viewer — *you cannot have two window-managers running on the same display*.

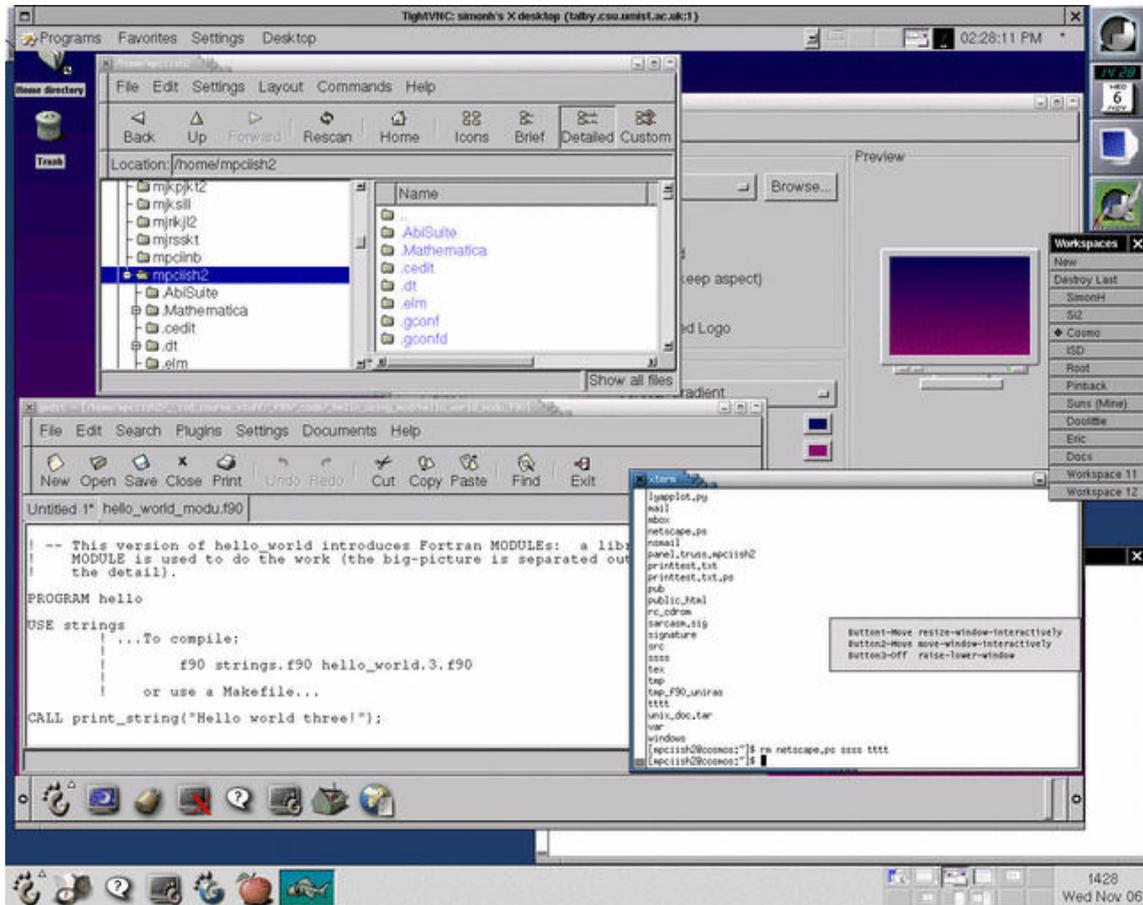
4. On the remote server (e.g., Cosmos), set the `DISPLAY` environment variable to correspond to this new desktop (in this case, `talby.csu.umist.ac.uk:3`) and start a graphical session — here we start an `xterm` and `TWM`, a lightweight/minimal window-manager:

```
cosmos> export DISPLAY=talby.csu.umist.ac.uk:3
cosmos> /usr/openwin/bin/xterm &
cosmos> /usr/openwin/bin/twm &
```

Graphical functionality in this desktop, including menus, are now those for Cosmos.

## 8. The gnome-session

At the time of writing (November, 2002) the GNOME<sup>13</sup> environment (v1.4) has just been downloaded, compiled and installed on Cosmos, though it has not been fully configured. This means that one can have Cosmos manage a gnome-session in a window (sub-desktop) on your local (main) desktop.



*A Cosmos-managed GNOME-session (showing an xterm, the GMC file-manager, the gedit editor and also the GNOME configuration tool) running on a second local X-desktop running within a VNC viewer running in a large window on the first local X-desktop which is running the local GNOME environment, on a Linux box!*

This is possible via:

- On a MS Windows box [Page 6], use XFree86 (or eXceed in one-window mode).
- Running VNC on a Unix box already running X as described above [Page 8].

To achieve this scenario, follow the instructions from the appropriate section (follow one of the above two links), but rather than simply starting a window-manager on the remote server (e.g., Cosmos) start a Gnome Session:

```
/usr/local/bin/gnome-session &
```

<sup>13</sup><http://www.gnome.org>

This may take some time to start the first time it is run as significant configuration takes place — this configuration is stored within `~/ .gnome` so subsequent starts are much faster.

---

## **9. Accessing Universe (Cosmos and Eric) from the UMIST PC Clusters**

The SSH/SFTP clients from SSH Communications are installed and configured to access Cosmos on all ISD PC Clusters.

### **9.1. Simple Telnet-Like Access**

To obtain simple Telnet-like access to Cosmos click on the “Start” button and then choose:

```
Programs ->  
Unix Connectivity ->  
SSH Secure Shell
```

This will start the SSH Communications SSH client.

### **9.2. Using SSH with X-Windows (Hummingbird eXceed)**

If you require graphical access via X-Windows to Cosmos click on the “Start” button and then choose

```
Programs ->  
Unix Connectivity ->  
Secure Shell Client - Xwindows
```

which starts the SSH Communications SSH client and also starts the eXceed X-server on your PC. You should now be able to launch X-clients from Cosmos onto your local desktop — these will be tunnelled through your SSH connection.

### **9.3. File Transfer**

If you need to move files to or from Cosmos click on the “Start” button and then choose:

```
Programs ->  
Unix Connectivity ->  
Secure File Transfer Client
```

which starts the SSH Communications SFTP client — this is a drag and drop, encrypted/secure FTP client.

---

## **10. FAQ: Problems? Look here first**

### **Contents:**

- When I telnet to Eric why can't I get a login prompt? [Page 12]
  - I'm using SSH, but I can't login/authenticate! Why? [Page 13]
  - I'm using SSH, but I can't display Eric applications on my screen [Page 14] anymore — why? (And it used to work with Telnet!)
-

## **When I telnet to Eric why can't I get a login prompt?**

I telnet to Eric and see either this...

```
telnet eric
Trying 130.88.97.1...
Connected to eric.
Escape character is '^]'.
Connection closed by foreign host.
```

...or this</DOTS>

```
telnet eric
Trying 130.88.97.1...
telnet: connect to address 130.88.97.1: Connection timed out
```

What's wrong?

Telnet and FTP access to Eric have been blocked for some time now. You must use SSH login (and SCP or SFTP to transfer files). SSH and friends are described above [Page 1].

## I'm using SSH, but I can't login/authenticate! Why?

When I try to login/authenticate using SSH I am repeatedly asked for my password, but I still can't get in. I see this...

```
mpciish2's password:
mpciish2's password:
mpciish2's password:
warning: Authentication failed.
Disconnected; no more authentication methods available (No further
authentication methods available.).
```

...or this...

```
Keyboard-interactive:
PAM authentication
Password:
Keyboard-interactive:
PAM authentication
LDAP Password:
simonh's password:
.
.
warning: Authentication failed.
Disconnected; no more authentication methods available (No further
authentication methods available.).
```

In the first case, the most likely explanation is that your SSH client is either too old to support *keyboard-interactive* authentication or has not been configured to use this method. Keyboard-interactive authentication is described above [Page 1]. When logging in via the keyboard-interactive method you should see something like this (note the two lines “Keyboard-interactive” and “PAM authentication”):

```
ssh eric -l mpciish2
-----
UMIST IT Regulations
.
.
-----
Keyboard-interactive:
PAM authentication
Password:
Authentication successful.
.
.
```

In the second case it is likely that you are using an incorrect user name. Remember that to login to a host via SSH, you need to specify the username: `ssh mpciish2@eric.umist.ac.uk` or `ssh eric.umist.ac.uk -l mpciish2`. If you do not explicitly state the username on the remote machine then the local username is assumed.

**I'm using SSH, but I can't display Eric applications on my screen anymore — why?**

With Telnet, X11 connections are made separately (usually on port 6000); this is facilitated by setting the DISPLAY environment variable on the remote machine (this may be done automatically) and giving the remote machine permission to access your local screen (usually via the xhost command — this may also be done automatically). One can do the same thing with SSH, though it is not recommended as this defeats the security offered by the encrypted (SSH) connection. Instead one should “tunnel” the X11 connection through the SSH connection. This procedure is detailed above [Page 5]

**About this page:**

Produced from the SGML: /home/isd/public\_html/\_cosmeric/\_accessing\_cosmos/\_reml\_grp/index.reml

On: 11/4/2003 at 16:3:46

Options: reml2 -l long -o tex -p single