

UW Computer Science and Engineering
Starting UNIX X Clients From Windows XP

revised September 2008 - Warren Jessop

Prerequisites

You'll need a PC running XP or Vista.

The Goal

By following the instructions in the next section you will have accomplished the following:

- Installed software (X Server and an SSH Client) on your PC.
- Created a shortcut on your desktop to start X and SSH, configured in a way that you can use your PC to safely initiate and display X client programs running on remote machines. This shortcut is similar to those that appear on the instructional lab PCs in 002, 006 and 022.

Installation Instructions

Install PuTTY

PuTTY (**putty.exe**),¹ is an SSH client available for download at <http://www.chiark.greenend.org.uk/~sgtatham/putty/>. Like other clients featuring the SSH2 protocol, it allows you to display remote X clients safely on your PC by using a feature called “X tunneling”. On the website you will see an installer file (the name ends in “installer.exe”) that you should choose.

Install Reflection X

Reflection X, the PC X server, is available on a CD from the Support Office, CSE 260. Double-click “setup.exe” on this CD, then choose the defaults (with the possible exception of allowing all users) and hit “next” until it’s done.

Create the Shortcut on Your Desktop

In the following example we'll assume you want to create an icon that will start an X session on UNIX host **attu**. However, the directions are general and will show you how to create a new shortcut that will start a secure X session on any UNIX host. It does not really matter if you've used either Reflection X (the supported CSE X server for Windows) or **PuTTY** before—but of course they need to be installed on your computer.

Here's one way to create an icon (shortcut) that will start an X session on **attu**:

¹ Previously this documentation featured SSH Secure Shell (or **sshclient.exe**), available in the UWICK kit, <http://www.washington.edu/computing/software/uwick>; however, **Sshclient** has some drawbacks when used at home, particularly with home routers.

1. Copy `\\cseexec\cs\nt\dist-area\miscellaneous\reflection\pconnect.cmd` to your desktop using one of the Windows tools. (If you need to know more about `\\cseexec\cs`, see <http://www.cs.washington.edu/lab/sw/uwcsentdfs.html>) You only need to do this once, and you don't necessarily have to copy it to your desktop, but make sure that the full path name of the location you copy it to is entered into the shortcut in step 4 below.
2. Move the cursor to the desktop and click the right mouse button.
3. Choose New->Shortcut from the pop-up menus.
4. Click "Browse...", scroll down to 'pconnect.cmd', choose it, then click OK.
5. Append "attu" (or perhaps the full pathname, "attu.cs.washington.edu", e.g. if this is on your home machine) to the text in the "Type the location of the item:" box. E.g. if the text reads "`C:\Documents and Settings\jouser\Desktop\pconnect.cmd`" then after this step it will read "`C:\Documents and Settings\jouser\Desktop\pconnect.cmd` attu'.
6. Optional: If you have previously created and saved a PuTTY "session", you can append the name of the session after the hostname. E.g. if your saved session is called "whitebg" you could put this in the "location" box: "`C:\Documents and Settings\jouser\Desktop\pconnect.cmd` attu whitebg'.
7. Click Next.
8. Type "SSH-X to attu" in the "Type a name for this shortcut:" box.
9. Click finish. That's it.

What Happens when you Click on the Shortcut

Note: the *first* time you click on the shortcut both Reflection and PuTTY are going to issue a number of annoying popups that you should dispatch as follows:

- Do **not** run the Reflection optimizer. It takes way too long and does practically nothing.
- If you are asked about connecting to an XDMCP host, just cancel.
- Do **not** use the reflection client wizard.
- Allow Reflection to run, i.e. remove the block.
- **Do** cache the host key for PuTTY, so that it does not ask every time.

These popups should not happen again, and the normal sequence will happen:

1. **Reflection-X** will start (but only if it is not already started).
2. **Putty** will start and will pop up a UNIX terminal window for a remote UNIX host - we'll use "attu" for this specific example. It may also pop up a "PuTTY Security Alert" window; if this is the first time you have logged into attu via **PuTTY** you can safely hit Yes.
3. You enter your CSENetID and Kerberos password in response to the prompts, which will log you into attu.
4. You get a UNIX prompt in the "attu - PuTTY" window.

Now, assuming you've gotten this far, you have several choices:

- If you just want to start X clients on `attu`, you can go ahead and do that, e.g.:

```
xterm &
```

to start an xterm.

- If you want to start X clients on a different host, say ‘`notattu`’, use `ssh` on the original **PuTTY** window:

```
ssh -n -Y notattu xterm &
```

Using **PuTTY** has these advantages over other methods:

- You will have an initial Kerberos ticket on the UNIX machine.
- Your UNIX X environment (i.e., `$DISPLAY`) will be set up correctly—that’s provided you have not set it in your "dotfiles", e.g. `.cshrc`. The value of your `DISPLAY` variable should always be

```
localhost:n.0
```

where n is some number greater than 0.

- X network traffic between clients on the UNIX host (the one you logged into using PuTTY) and your X display will be encrypted.

Caveats

Use Port Forwarding

If you don’t use SSH-style port forwarding, tunneling or IPSec, all X traffic is unencrypted. What this means in plain English is: if you want to start X clients on other UNIX hosts, DON’T use `rlogin` or `telnet` or `xrsh`. Instead, use ‘`ssh -n host xterm &`’ from one of the existing UNIX shells.

Problems With Tcl and the Tk Toolkit

If you use a UNIX X application that makes use of the “Tk” toolkit and that uses the Tk “send” command, take note. For, example, `exmh` is such an application, and it uses the “send” command to start `xterm` or editor X clients, e.g. `emacs`; in this case the error generated by `exmh` reads:

```
X server insecure (must use xauth-style authorization); command ignored.
```

The reasons for this behavior are complex and have to do with the way the Tk “send” command ensures security. We will only deal here with what’s needed to get “send” to work and hence get around the problem.

The solution is to give Reflection X a copy of your UNIX **.Xauthority** file (this is found in your UNIX home directory). This file has what’s called a “magic cookie” inside. Note that the magic cookie is only created *after* your SSH-X session has been established, so Reflection X should not be given the cookie before then.

A Windows program, **copyrxauth**, has been written that will copy your **.Xauthority** file to a folder that Reflection expects it to be in—and rename it to **RXAUTH**, another of Reflection’s expectations. **Copyrxauth** takes one parameter, the name of your UNIX home directory. For example, for a user called “jouser” this would normally be something like **/homes/iws/jouser** on the instructional UNIX systems, or **/homes/gws/jouser** for most staff research accounts.

Taking the latter example: once jouser has established an ssh connection to a research UNIX system, she would open up a command window on her Windows workstation and enter:

```
o:\nt\dist-area\miscellaneous\reflection\copyrxauth /homes/gws/jouser
```


Table of Contents

- Prerequisites..... 1**
- The Goal..... 1**
- Installation Instructions..... 1**
 - Install PuTTY..... 1
 - Install Reflection X..... 1
 - Create the Shortcut on Your Desktop..... 1
- What Happens when you Click on the Shortcut
..... 2**
- Caveats..... 3**
 - Use Port Forwarding..... 3
 - Problems With Tcl and the Tk Toolkit..... 3
- 5**