

How to Remotely Access Java Tools for CSE 1020

By Steven Castellucci (v2, January 2012)

You don't always need to be in the PRISM lab to work on your 1020 assignments. Working on your own computer is convenient if the lab is busy, closed, or if you live far from the University. To do so, you must first install software on your computer. Which software you install depends on how you wish to work.

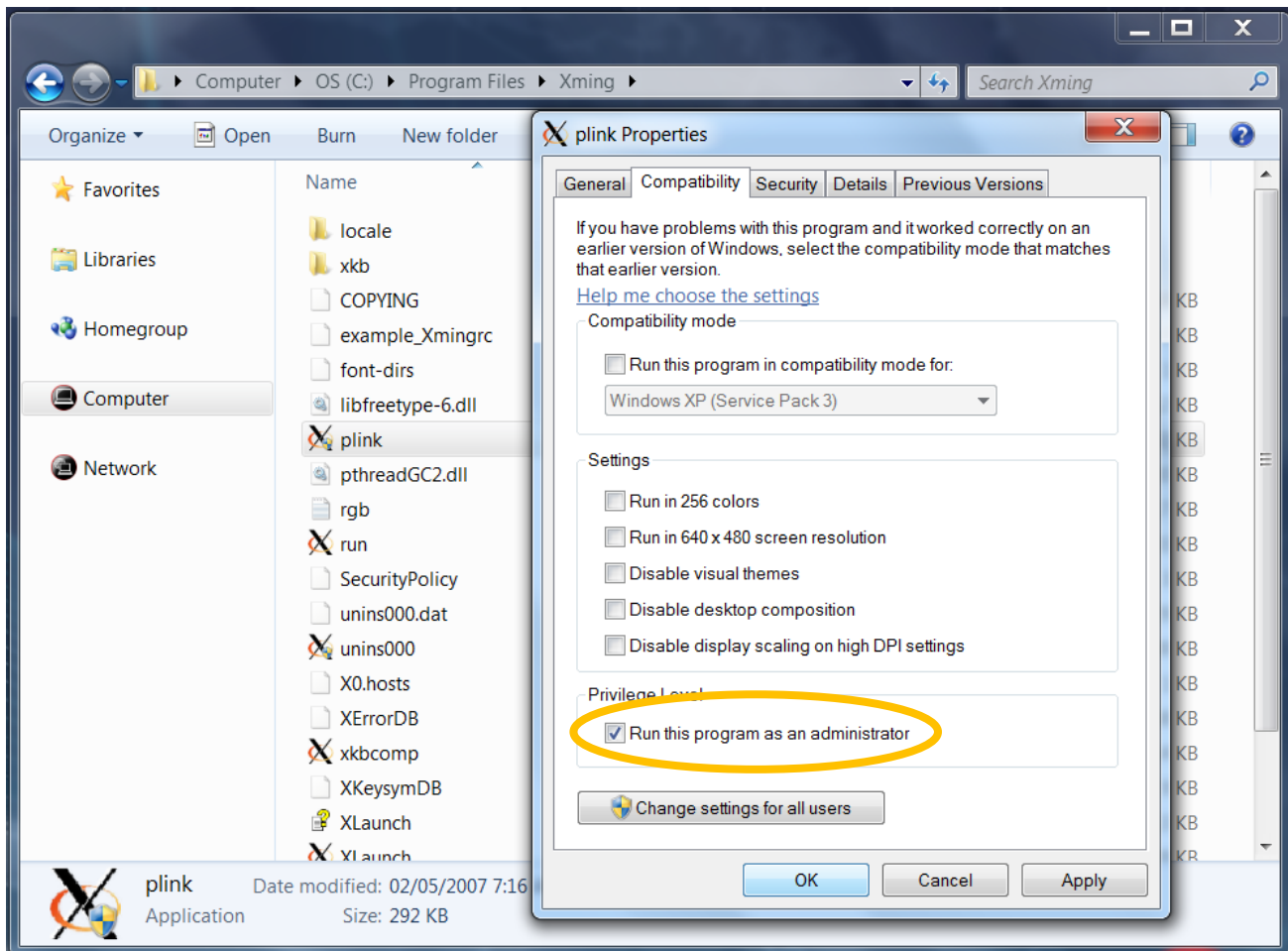
Using an Internet Connection

By connecting via a high-speed internet connection, you will be able to use the resources of the CSE server remotely.

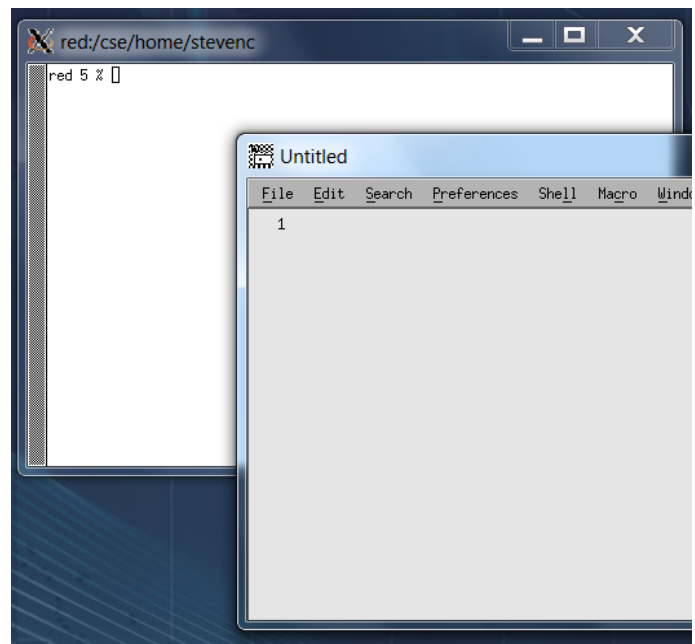
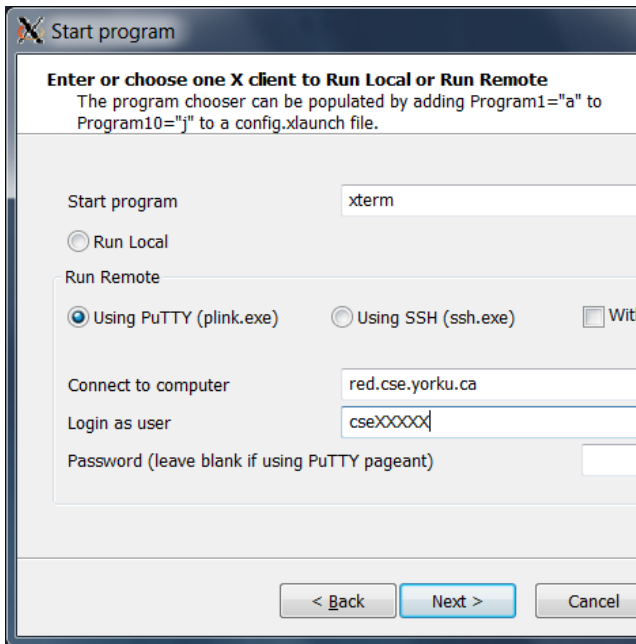
Windows

Download and install [Xming](#), then perform the following:

1. Navigate to the installation directory (C:\Program Files\Xming by default). Right-click on `plink.exe` and select "Properties". Select the "Compatibility" tab and check the option "Run this program as an administrator". This is required to prevent Xming from crashing.

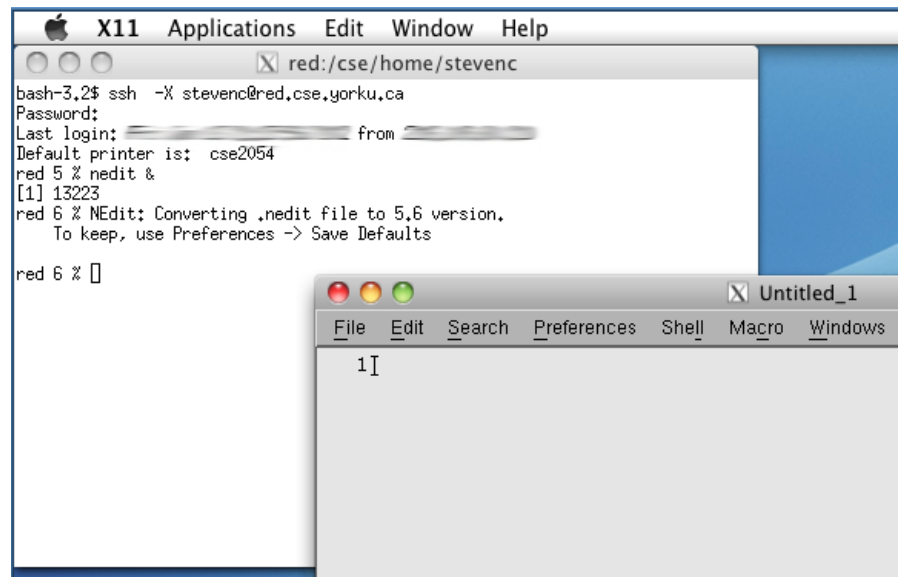


2. Create an Xming connection to the CSE server. Start XLaunch from the Xming program group. Select “Multiple windows”, then click “Next”. Select “Start a program”, then click “Next”. Select “Using PuTTY”, enter `red.cse.yorku.ca` as the server and your username, then click “Next”. Click “Next” again. Click “Save configuration” and save the file “CSE.xlaunch” to your desktop. Double-click the file to connect to the CSE server. If you are prompted to save the server’s key certificate, type “yes” and press Enter.



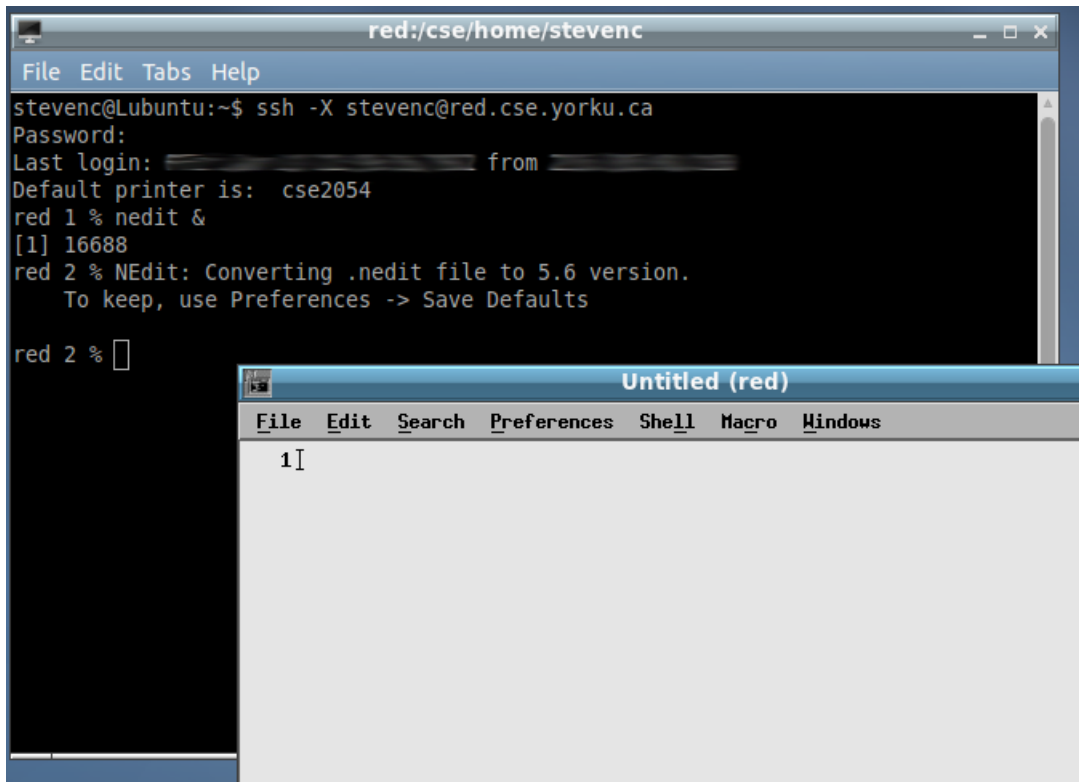
OSX

Start the X11 client found in the Utilities group in Applications. In the terminal window that appears, type `ssh -X cseXXXXX@red.cse.yorku.ca`, replacing “cseXXXXX” with your CSE login. If you are prompted to save the server’s key certificate, type “yes” and press Enter.



Linux

Open a terminal window and type `ssh -X cseXXXXX@red.cse.yorku.ca`, replacing “cseXXXXX” with your CSE login. If you are prompted to save the server’s key certificate, type “yes” and press `Enter`.



The image shows two overlapping windows from a Linux desktop environment. The top window is a terminal window titled "red:/cse/home/steven". The terminal output shows a user logging in via SSH, followed by running the 'nedit' command. The terminal text is as follows:

```
steven@Lubuntu:~$ ssh -X steven@red.cse.yorku.ca
Password:
Last login: _____ from _____
Default printer is: cse2054
red 1 % nedit &
[1] 16688
red 2 % NEdit: Converting .nedit file to 5.6 version.
      To keep, use Preferences -> Save Defaults
red 2 % █
```

The bottom window is an NEdit text editor window titled "Untitled (red)". The editor's menu bar includes "File", "Edit", "Search", "Preferences", "Shell", "Macro", and "Windows". The main editing area contains the number "1" followed by a cursor.

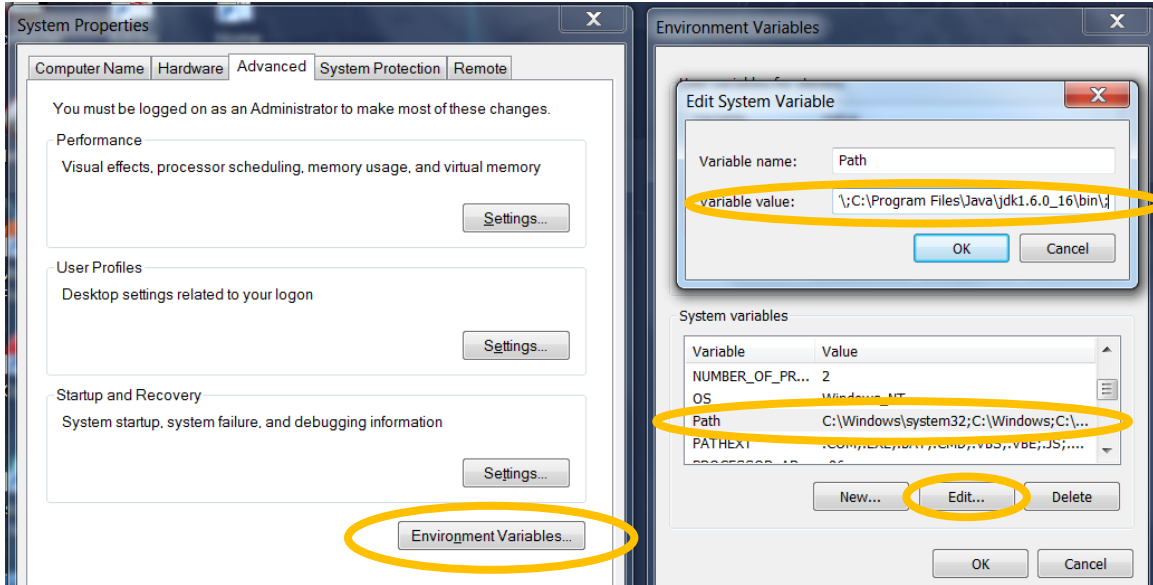
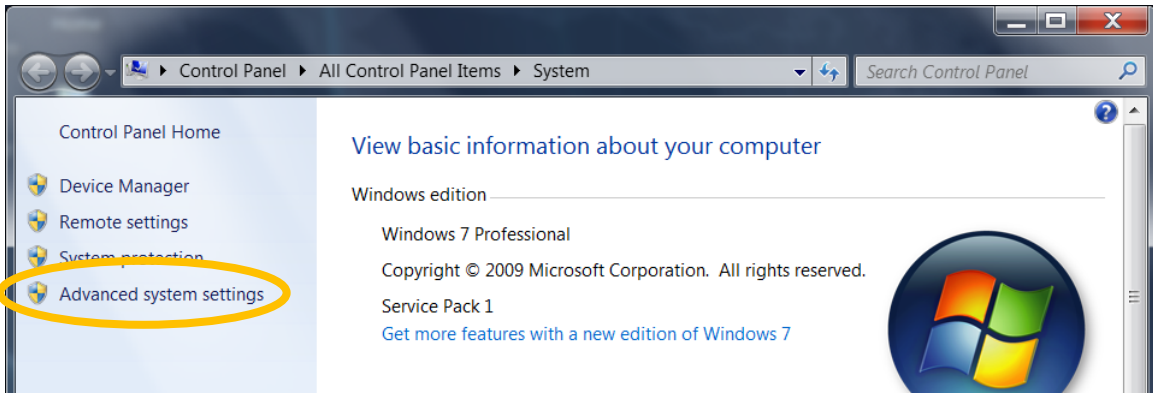
Working Offline

Working offline allows you increased freedom. With a laptop, you could work on your assignments during your commute to and from campus. Working offline is also convenient if you have a slow or problematic internet connection. However, this method can require downloading, installing, and configuring numerous programs.

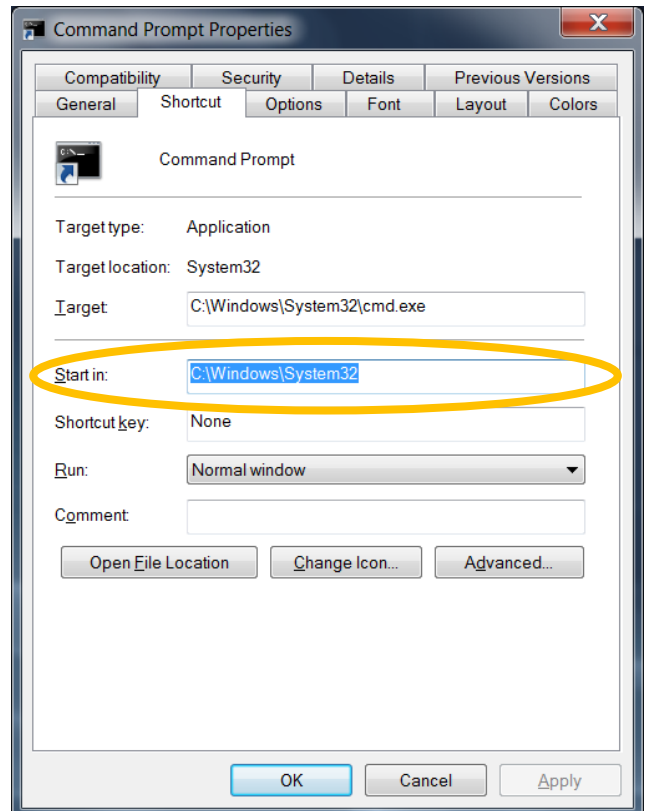
NOTE: Downloading the Java SE Development Kit (JDK) now requires signing-up for a free account from Oracle.

Windows

1. Download and install the [Java SE Development Kit \(JDK\) from Oracle](#). Also download the [TYPE](#) package to eCheck your work. Save the `type.jar` file in the `jre\lib\ext` subdirectory of your JDK installation. By default, this is `C:\Program Files\Java\jdk1.6.0_19\jre\lib\ext` (assuming version 1.6u19 of the JDK). You will still need an internet connection to have your eCheck grade recorded (if applicable), but you will be able to check the correctness of your program offline.
2. Next, right-click “Computer” on the Desktop (or Start Menu) and select “Properties”. Select “Advanced system settings”, then “Environment Variables...”. Find and select the system variable “Path”. Click on “Edit...”. At the end of the existing value, type a semicolon (“;”), the path to the Java bin folder (using the above example, `C:\Program Files\Java\jdk1.6.0_19\bin`), then another semicolon. Click “OK” three times to close all the dialogs.

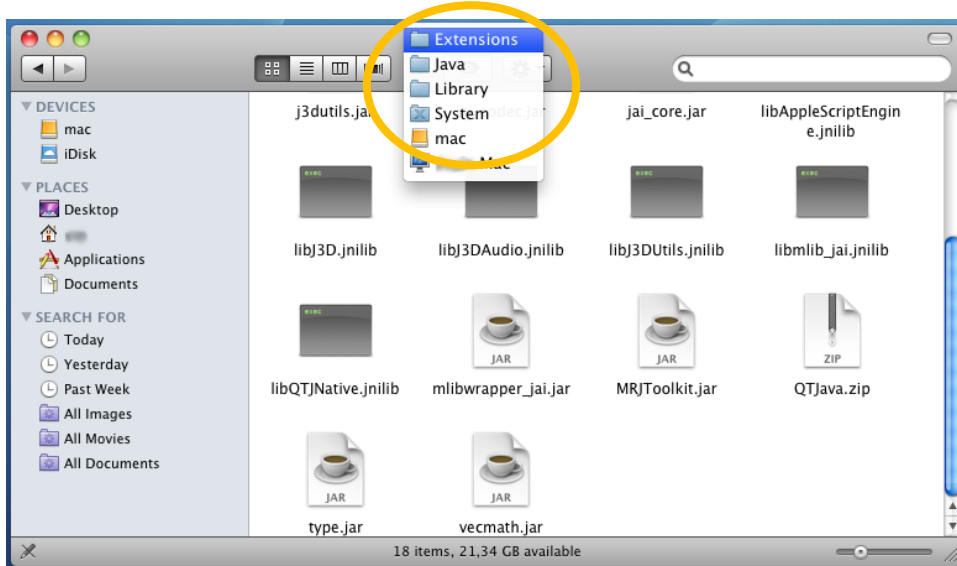


3. Create a folder (anywhere) to store all of your Java source code files.
4. Next, create a shortcut to the Command Prompt. Right-click on the Desktop and select "New", then "Shortcut". Type "cmd" as the location, then click "Next". Type a name for the shortcut (your choice), then click "Finish". Right-click on the new shortcut and select "Properties". In the "Start in" field, type the path to the folder you created in Step 3. Click "OK".
5. To check your installation, double-click on the shortcut and enter `javac`. If you see a list of options, the JDK is configured correctly. Otherwise, check that your `PATH` environment variable was set correctly.
6. Finally, you will need an editor to create Java programs. In the lab, you will typically use NEdit, but it is not available for Windows. The textbook CD contains a copy of [Crimson](#). Other free editors include [ConTEXT](#), [iEdit](#), [Notepad++](#), and [PSPad](#).



OS X

Luckily, OS X comes pre-installed with the Java SE Development Kit (JDK). Simply download the [TYPE](#) package to eCheck your work. Save the `type.jar` file in the `/System/Library/Java/Extensions` subdirectory of your JDK installation. The Terminal screen can be accessed from the Utilities folder in Applications.



Linux

Download and install the [Java SE Development Kit \(JDK\) from Oracle](#). Also download the [TYPE](#) package to eCheck your work. Save the `type.jar` file in the `jre\lib\ext` subdirectory of your JDK installation.

You will need to update your path environment variable as shown below. The example assumes the JDK was installed in `/usr/lib` (change it as appropriate). If you use the `bash`, `ksh`, or `sh` shell, modify the `~/ .bashrc`, `ENV`, or `~/ .profile` file, respectively.

