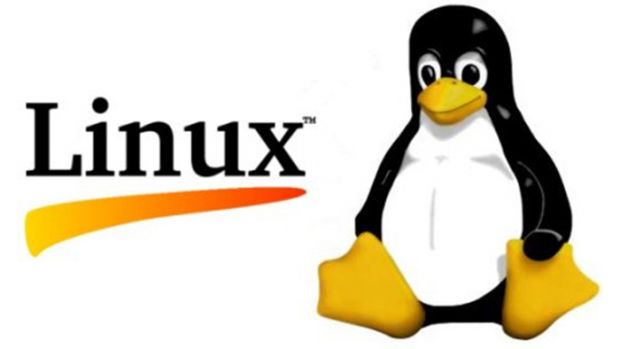
Janek’s quick and dirty guide to…



Contents

[Chapter 1 – Basic terminal navigation 3](#_Toc521535413)

[Navigation: 3](#_Toc521535414)

[Viewing folder contents: 3](#_Toc521535415)

[Folder creation & deletion: 3](#_Toc521535416)

[Using a previous command: 4](#_Toc521535417)

[Chapter 2 – Vim 5](#_Toc521535418)

[Making a new file or editing an already existing one: 5](#_Toc521535419)

[Saving a file without closing vim: 6](#_Toc521535420)

[Saving a file and closing vim: 6](#_Toc521535421)

[Closing vim and not saving any changes: 6](#_Toc521535422)

[Skipping to a certain line: 6](#_Toc521535423)

[Setting up vim “correctly” 7](#_Toc521535424)

[Vim helpful hints: 7](#_Toc521535425)

[Deleting an entire line: 7](#_Toc521535426)

[Highlighting/Selecting parts: 7](#_Toc521535427)

[Pressed ctrl-S by mistake? Everything is stuck? 7](#_Toc521535428)

[Chapter 3 –File permissions 8](#_Toc521535429)

[How to change the permissions for an executable/program: 8](#_Toc521535430)

[Chapter 4 – BASH scripts basics 9](#_Toc521535431)

[Chapter 5 – Installing XMING and PUTTY on windows 10 10](#_Toc521535432)

## Chapter 1 – Basic terminal navigation

Navigation: cd

cd ~/Desktop/MyFolder/Stuff/

*This will navigate the terminal to /home/\*Username\*/Desktop/MyFolder/Stuff/*

*Note: The ~ key removes the need to type /home/\*Username\*/ each time.*

cd ../../

*Move back two folders from the current location, in this example it will result in you being in ~/Desktop*

Viewing folder contents: ls

*ls (el and s, not a capital eyeee)*

*This prints the folders and files in the current directory*

ls

ls -l

*ls -l (dash el for list) this prints the contents in a list format with more information, such as its permissions and date created.*

ls ~/Desktop/

*Prints the contents of ~/Desktop/*

### Folder creation & deletion:

mkdir MyNewFolder

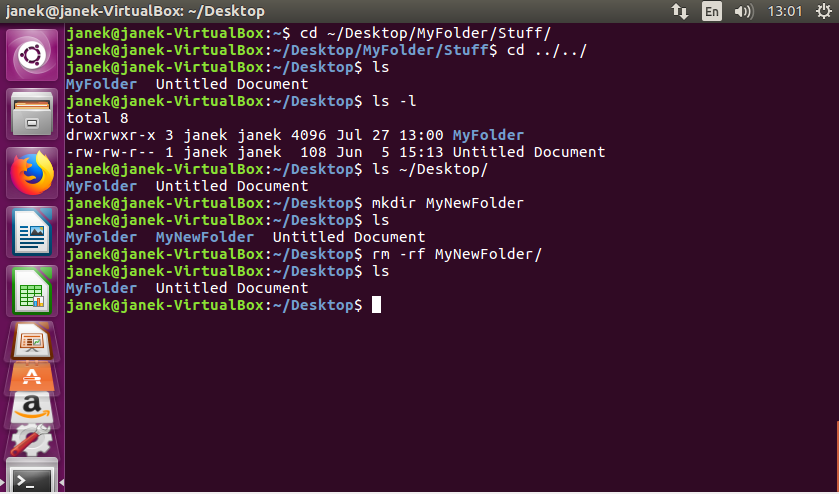
*Makes a new directory in the current working location called MyNewFolder*

rm -rf MyNewFolder

**Note, if any of the above commands say you don’t have permission, add the command “sudo” before the rest, it will probably ask for your password after.**

*Remove this folder without questioning me!! (the -rf does that), if you’re just deleting a file omitting -rf works fine.*

Below is a combined example of all the above commands:



### Using a previous command:

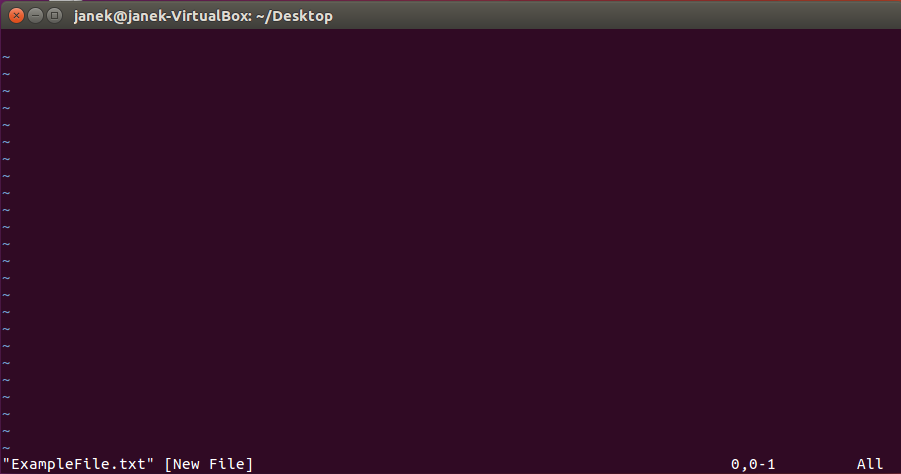
Up arrow and down arrow keys are used for navigating previous commands.

## Chapter 2 – Vim

### Making a new file or editing an already existing one:

vim ExampleFile.txt

**You’ll be greeted with something like below**



**Try pressing a few keys, nothing working? Press ‘a’ – this puts vim into edit mode**

**Note, there are other keys to put vim into edit mode other than ‘a’, but….so what.**

**The bottom left hand corner should now show:**



**You can now write whatever! don’t forget to save as you go! Vim doesn’t autosave for you like Microsoft word does (at least not on default), if your computer crashes or if you quit without saving your work is gone for good!**

**Note, whenever I type a command while in vim I first press the escape key, I will write this as ‘esc’ from now on. Make sure you don’t type ‘esc’!**

### Saving a file without closing vim:

**esc** :w

### Saving a file and closing vim:

**esc** :wq

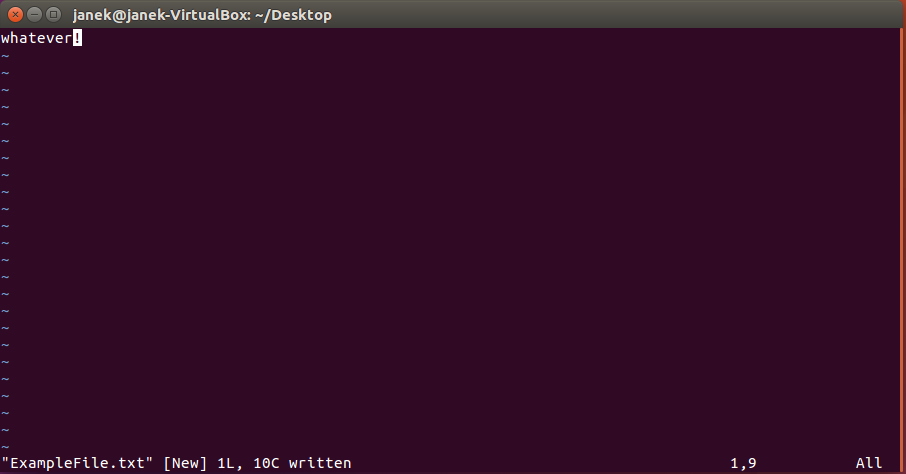
Closing vim and not saving any changes: **(This is useful!)**

**esc** :q!

### Skipping to a certain line:

**esc** :12

**Below is an example, I wrote whatever! And saved the file using :w**



## Setting up vim “correctly”

**Everything up till now is probably enough, but I will show you how to edit the .vimrc file, which contains the settings vim uses each time it is started up, in here you can set colours (which is very useful with coding!!!) and auto-indents for each line (You want 4x spaces, not tabs)**

Navigate to your home directory

**Hint: cd ~/**

vim .vimrc

**Note: This file won’t be visible with a usual ls command, it’s a hidden file. It might already exist or not, if it exists empty it and follow this.**

You now want to type in the following:



**Save and exit this file, vim will be that little bit better now! For more funky things to add to this file refer to the internet and examples** [**http://vim.wikia.com/wiki/Example\_vimrc**](http://vim.wikia.com/wiki/Example_vimrc)

## Vim helpful hints:

### Deleting an entire line:

Press ‘d’ twice

### Highlighting/Selecting parts:

Press ‘v’ and use the arrow keys to select

### Pressed ctrl-S by mistake? Everything is stuck?

Ctrl-q

## Chapter 3 –File permissions

**Every file on your machine can have a combination of the following permissions:**

* **Read – This program can read stuff**
* **Write – This program can write stuff**
* **Execute – This program can execute on this system**
* **Probably others but these are the main ones.**

**Examples:**

* **RWE – This program can read, write and execute.**
* **RE – This can read and execute but it may not write to anything!**
* **RW – This can execute and write to disk, but it cannot read anything.**
* **E – Can only execute, has no permission to read or write to files.**

### How to change the permissions for an executable/program:

sudo chmod ‘permissions’ ‘name of file’

**The ‘permissions’ part is made up of 3 numbers, the first number deals with the user, then group, then other.**

**The numbers follow the following structure:**

* **4 = Read**
* **2 = Write**
* **1 = Execute**
* **0 = No permission**

**Usage example:**

sudo chmod 755 file.sh

**7: 4 + 2 + 1, Read + Write + Execute**

**5: 4 + 1, Read + Execute**

**5: 4 + 1, Read + Execute**

**For your purposes sticking to 755 is fine, or 700 if you’re super concerned.**

## Chapter 4 – BASH scripts basics

**Bash scripts are interpreted, this means the interpreter (in this example bash will be doing that) interprets what you have written line by line and runs those ‘commands’, this has the benefit of being able to interact with system programs easily, but the negative that any mistake will only be picked up at runtime (when that line is being interpreted to be run), so if it takes an hour to finish the first task then the second task is wrong – well tough titties gotta edit your file or start again.**

**There are MANY other things you can do in bash scripts, such as if statements, while loops, run external programs, run other scripts, all of which would take up their own pdf. The following is just a few examples.**

**Our first script**



**Breakdown:**

**Line 1: This is telling the OS that the following lines will need to be interpreted by ‘bash’.**

**Line 2: Print ‘Hello World!’ to the console**

**Making the script executable**

 **(ignore the white rectangle)**

**Running the script**



**Modified version:**

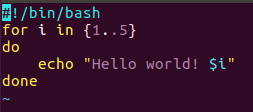
**Breakdown:**

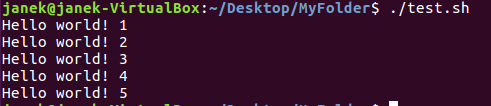
**Line 2: this is a ‘for’ loop, it will loop 5 times, starting at 1 and ending at 5**

**Line 3: ‘do’ the following**

**Line 4: same as last script**

**Line 5: Everything past this point is not inside the for loop.**

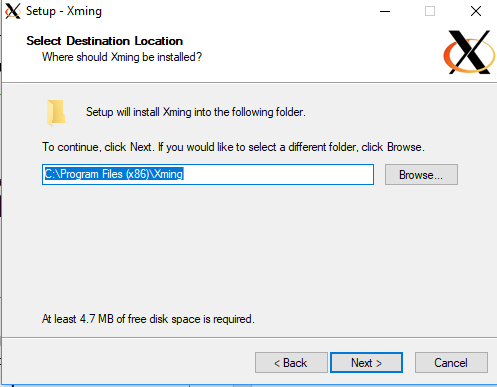




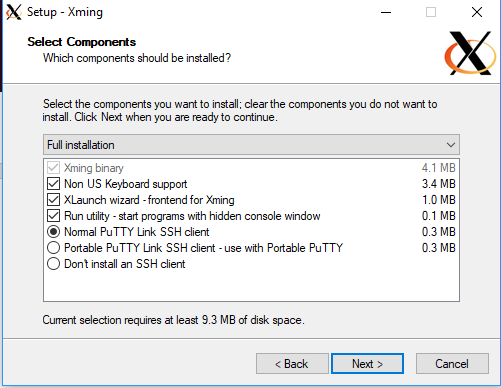
## Chapter 5 – Installing XMING and PUTTY on windows 10

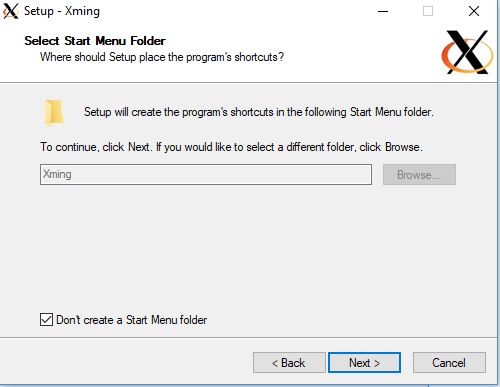
**Download the installer here:** <http://sourceforge.net/project/downloading.php?group_id=156984&filename=Xming-6-9-0-31-setup.exe>

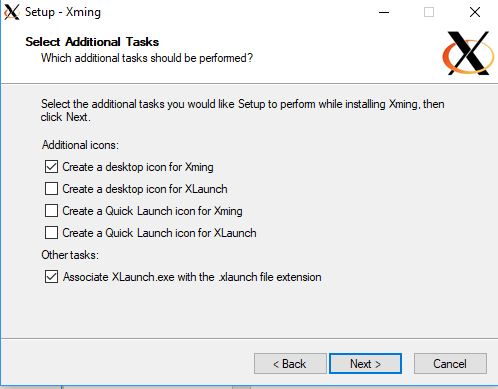


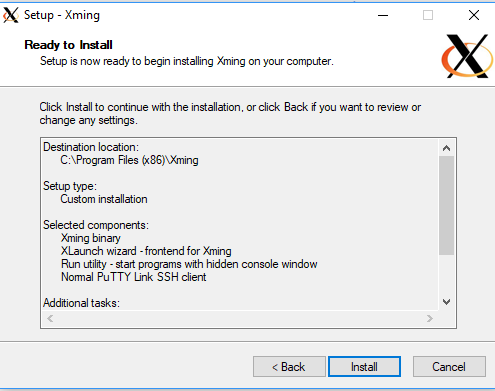


**Probably don’t need “Non US keyboard support” - Optional**

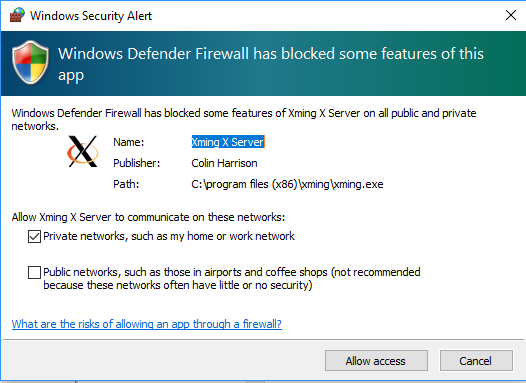








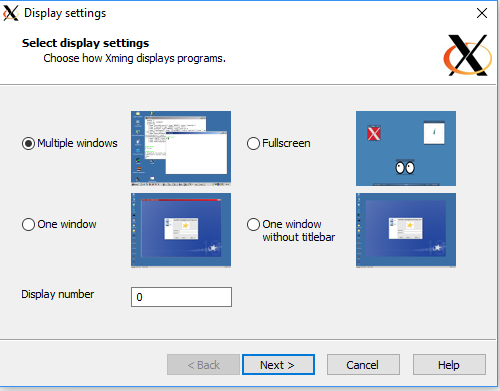
**If this pops up press “Allow access” and at least let Xming use Private networks (ie your home network).**

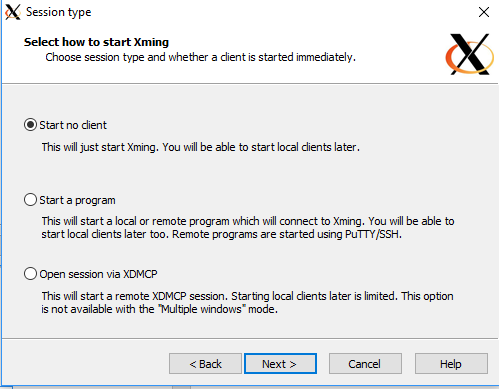


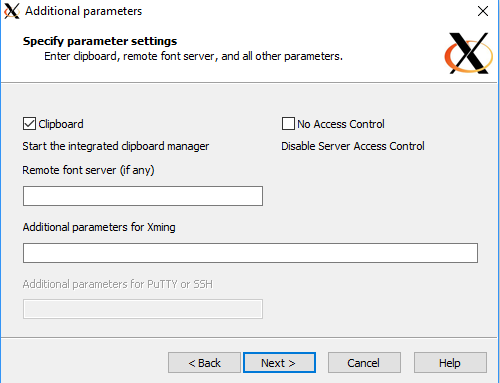
**Go here:**

**C:\Program Files (x86)\Xming**

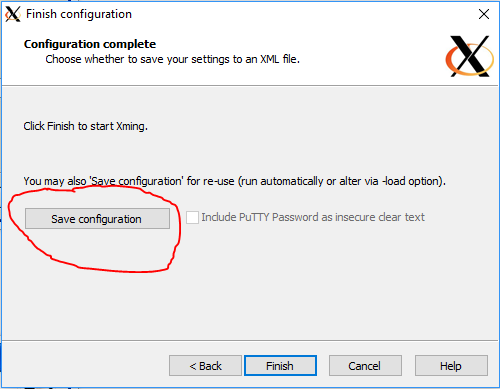
**And open “Xlaunch.exe”**







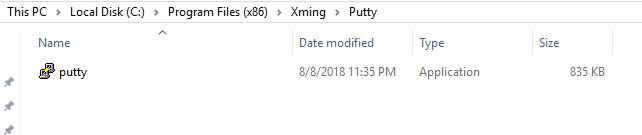
**Click on “Save configuration”, save it in C:\Program Files (x86)\Xming**



**Setting up PUTTY**

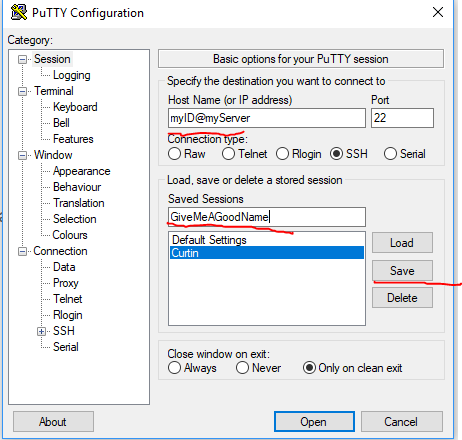
**Download the following:** <https://the.earth.li/~sgtatham/putty/latest/w64/putty.exe>

**Make a new folder in C:\Program Files (x86)\Xming called Putty and place it in there**



**Type in the command you normally use to communicate with your server, so it should follow the form myID@myServer (You don’t type in ‘ssh’ here!).**

**Then give it a good memorable name like “ChemServer” and hit save.**



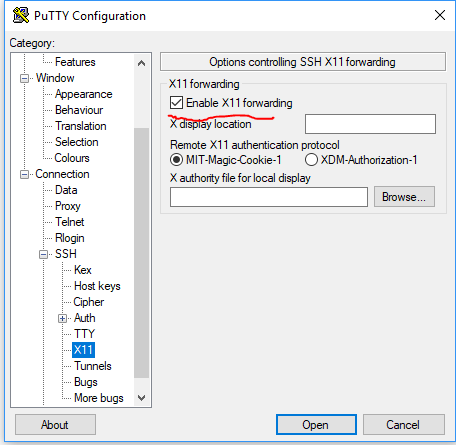
**After this click on the new session in the list (Above Curtin is selected) and press “Load”.**

**Now find “Connection”, “SSH”, click on the plus next to that, and click on X11.**

**Here you need to tick “Enable X11 forwarding”**

**Go back to the previous screen by scrolling up and pressing the “Session” text.**

**Click the “ChemServer” thing again and press SAVE.**



**Now before we go any further with PUTTY make sure XMING is actually running! So go to**

**C:\Program Files (x86)\Xming**

**and open Xming.exe (You will probably notice nothing! No news is good news here!)**

**Now go back to PUTTY and press on Open, making sure your ChemServer preference was selected and loaded.**

**You will now be greeted by a login system, log in using the details for the chem server you were given. This is just like a terminal you use on Ubuntu.**

**Test the graphics capabilities by typing “xclock” you should see a clock!**

