

*The
Mentor Series
from
Luppes Consulting*



Raspberry Pi for the .NET Developer

First Edition

DRAFT

by Lyle Luppes

Raspberry Pi for the .NET Developer

by Lyle Luppés

Version 0.0.1.0

Copyright 2015, Luppés Consulting, Inc.

All Rights reserved. No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form, or by any means, electronic, mechanical, photocopying, recording or otherwise, without prior consent of the publisher.

Editor: Amanda Ford

Technical Reviewers: Lyle Luppés

Product names, logos, brands, and other trademarks featured or referred to within this manuscript are the property of their respective trademark holders.

These trademark holders are not affiliated with the author or any of the author's representatives. They do not sponsor or endorse the contents, materials, or processes discussed within this book.

While every precaution has been taken in the preparation of this book, the author and publishers assume no responsibility for errors or omissions, or for damages resulting from the use of the material in this book or the related software.

ISBN-COMINGSOON (Luppés Consulting)

Table of Contents

Introduction	v
Time to Buy Some Goodies	1
Installing the Operating System	2
Formatting the SD Card	2
Initial Power Up	3
Initial Configuration of Your Pi	4
It's Alive!	5
Remote Desktop	7
Enable Copy/Paste in TightVNC	8
Initial Setup	10
Auto-Start TightVNC	11
Applying System Patches and Updates	13
Your Personal Toolkit	14
A Better Browser	14
File Transfer Utility	14
Personal Touches	16
Installing .NET Mono	17
Install Mono (the wrong way)	17
Install Mono (the right way)	17
Running a C# WinForms Program	19
Installing ASP.NET vNext	20
Update Your Certificates	20
Install KVM	21
Install Libuv	21
Running Your First ASP.NET MVC6 Page	22

Raspberry Pi for the .NET Programmer

Writing an ASP.NET v5 / MVC6 Application.....	24
Conclusion.....	25
Appendix	26
About the Author	27

Introduction

The Raspberry Pi has been a great hit, selling over 5 million devices since its introduction in 2012. People have used it in schools to teach children about programming, and hobbyists have been building lots of interesting projects with it. One of the great things about the Pi ecosystem is that everything is very inexpensive. The latest Raspberry Pi 2 has a 900 MHz quad-core processor for only \$35 -- wow! You can buy all of your components and have a fully functioning computer for less than \$100.

However, one of the problems (although some people might say that's an advantage!) is that it's a Linux computer and uses languages that are not familiar to the millions of experienced .NET developers out there right now. Sure -- we could all learn Python or Scratch or whatever the latest small device language is... but why?

What if we could use the .NET languages we know and love and use most of our installed code base and then deploy that on these little tiny computers? There are millions of .NET programmers out there with billions of lines of tested code in their project libraries. I believe you can use that code easily on the Pi, and I'll show you how you can do it in just a few short steps in this guide.

CHAPTER 1

"Unfortunately, no one can be told what the Matrix is. You have to see it for yourself." Morpheus - The Matrix – 1999

Just like the Matrix, you need to see this for yourself, and as I said in the intro, getting started is relatively inexpensive. Here what I would recommend that you start with for your shopping list:

Time to Buy Some Goodies

Raspberry Pi 2 Model B (1 Gig)	
http://www.mcmelectronics.com/product/83-16530	\$ 35.00
32G microSD Card (You could go smaller, but why? To save \$5?)	
http://www.amazon.com/gp/product/B003WGJYCY/	\$ 13.99
Mini Wireless Keyboard (It's really cool!)	
http://www.amazon.com/gp/product/B00KF9IVKC/	\$ 13.99
USB WiFi Adapter (Less wires is better, right?)	
http://www.amazon.com/gp/product/B003MTTJOY/	\$ 14.99
Plastic Case	
http://www.amazon.com/gp/product/B00OBBFP0O/	\$ 9.49
Total	\$ 87.46

There are also a couple of things that you will need that you probably already have laying around your office:

- A USB power supply with a micro USB cable (an old phone charger will probably work just fine – you need a 5V adapter).
- (Optional) You will need a spare monitor of some kind that supports HDMI. You don't really need a dedicated monitor – you only really need it during the setup, so you can share with your computer if needed. If your monitor only has a DVI connection, you can buy an HDMI to DVI adapter

HDMI – DVI Adapter Cable

<http://www.amazon.com/gp/product/B001TH7T2U/> \$ 7.99

Once you have these components, you're all set to go.

CHAPTER 2

Installing the Operating System

"Mama always said life was like a box of chocolates. You never know what you're gonna get." Forrest Gump

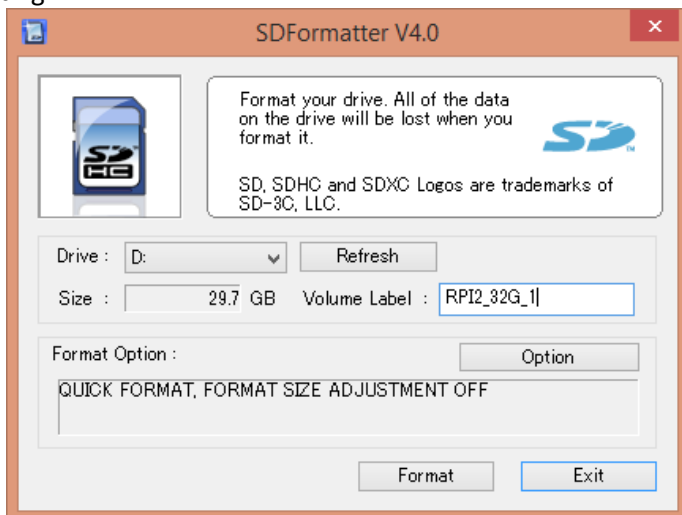
Once you have that little box of Raspberry in your hot little hands, it won't do any until you install an operating system – these little computers don't even have a hard drive. To get rolling, you will need a formatted microSD Card. You might think you could just format it in Windows Explorer, but you are better off using a utility that is designed to format SD cards optimally.

You can download a free formatter from:

https://www.sdcard.org/downloads/formatter_4/

Formatting the SD Card

Install and open the utility, insert and select your microSD card, and begin formatting.

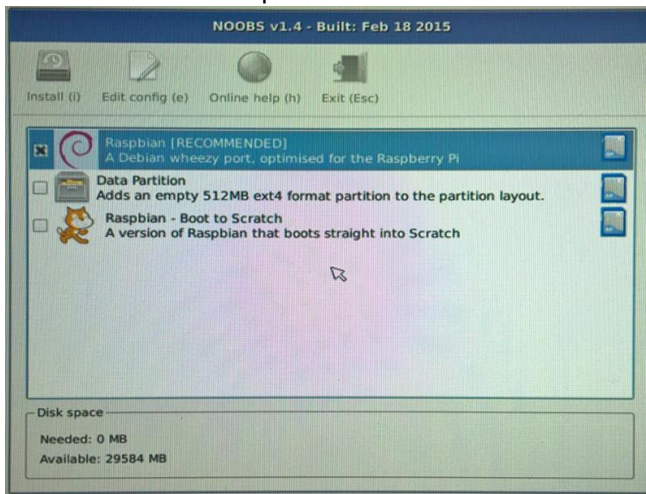


Picking your Operating System

There are several operating versions available at <http://www.raspberrypi.org/downloads/>. If you feel like you must be a real geek, go ahead and install one of the many Unix variants. Be aware though that as of the time of this document, not all of them work on the Raspberry Pi 2, so I'd stick with Raspbian for now. If you just want to get started quickly, simply download the NOOBS install - it's already prepared to make this quick and easy. All you have to do is unzip the downloaded file to your PC, then copy all of the files on to your newly formatted SD card and you are ready to boot up your Pi.

Initial Power Up

Connect all of your new hardware, put microSD card with the extracted NOOBS files into your Pi, and then power it up. You will see a NOOBS installation menu - choose Raspbian from that menu:



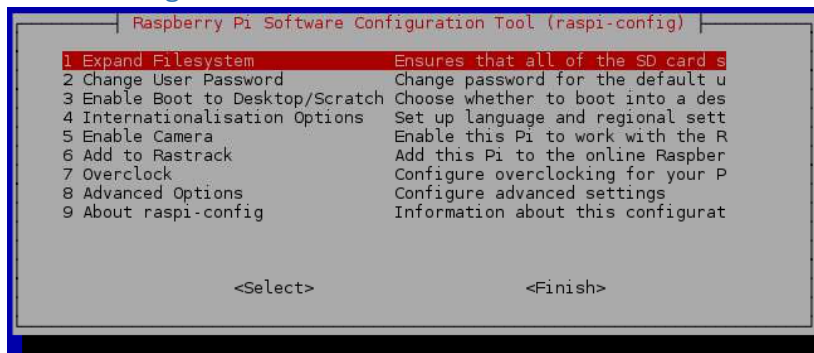
The installer defaults to the Great Britain locale (because the Pi was created there), so you may want to change the language and keyboard to US instead of GB. The one issue I found with using the GB keyboard that made me change it is that I could not find a way to enter the "\" character. (Maybe there is a way, but I couldn't find it!)

Once you click "Install" the Pi will go to work rebuilding your SD card with the operating system.



After a few minutes, you'll be ready to reboot and configure your Raspberry. The first time it boots up you will automatically be greeted with the Config Tool.

Initial Configuration of Your Pi



You don't have to set most of these because NOOBS takes care of most of it, but you will want to change these three options:

- #2 - Change User password - the default User Id is "pi" and password is "rasberry". Since every noob knows this, if you want any kind of security, you really should change this (you can always do it later -- just don't forget!)
- #3 - Enable Boot to Desktop - Select "Desktop Login as pi at the graphical desktop", unless you always want to have to type a command at the command line to start the user interface.
- #4 - If you changed the language to English on the install screen, you will need to select the Internationalisation Options and install the

Raspberry Pi for the .NET Programmer

"en-US.UTF8 UTF8" locale, and then change the default keyboard on the follow up screen. It will then take a minute or two to generate the proper locale files.

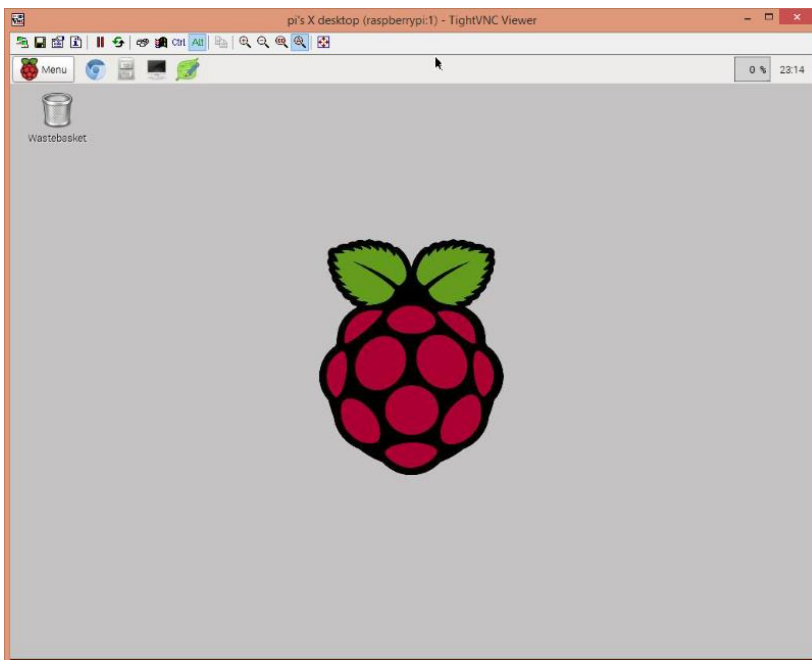
Tab down to Finish and press enter and your Pi will reboot.

TIP: If you miss Option 3 and end up at the command line instead of the desktop, you can always start the GUI with the command "startx".

TIP: If you want to get back to this screen later, just open a command prompt and type "sudo raspi-config"

It's Alive!

Congratulations - You now have a working Raspberry Pi! Granted, it doesn't do much yet, but it's a start. You should be seeing the home screen:



The first thing you will need to do is to get your network connection running. If you connected an Ethernet cable hard line, it should be already up and working already. You can open a web browser and browse to any website to verify this.

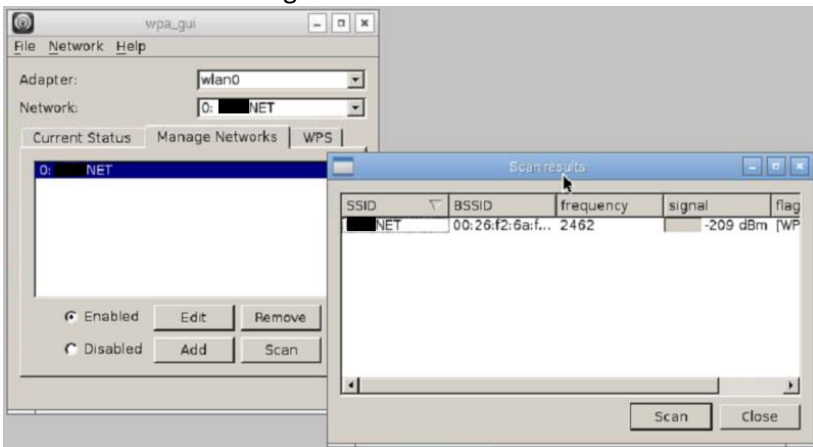
Raspberry Pi for the .NET Programmer

If you want to use WiFi, you will have to do a bit of configuration to get it up and running.

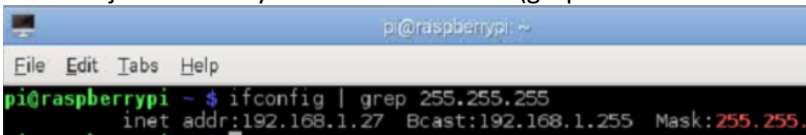
Open the Menu -> Preferences -> WiFi Config menu item. On the "wpa_gui" window that pops up, do the following:

- Select the Manage Networks Tab.
- Select the Enabled radio button
- Click Scan button to open Scan Results popup window
- Select your network and then enter in your wireless key in the PSK textbox on the popup screen.

It should look something like this:



Before you leave this screen, you will want to find your IP Address (you'll want that later), located on the Current Status tab. If you want to find it use a command line, use the "ifconfig" tool (a cousin to the Windows ipconfig command). I usually add a "grep 255.255" command to filter it down to just the line you are interested in. (grep is a Unix search utility)



Once you have an IP address, open your web browser and go to some website to test out your connection.

If you want, you can change your Pi to use a Static IP address, which might make it easier down the line. I don't usually bother with this as I haven't had an

Raspberry Pi for the .NET Programmer

issue using DHCP on my home network because it almost always keeps the address, and I can always look it up if I need to.

Remote Desktop

I've found that the easiest way to work on the Pi is to do it with your normal desktop and use a remote desktop program to access to your Pi. The standard Remote Desktop program in Windows won't work, but there is an easy (and free!) program available called TightVNC. You will need to install the server on your Pi, and then run a client on your Windows Desktop.

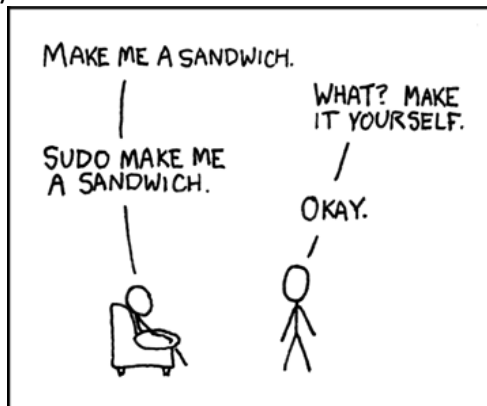
To install TightVNC on your Pi, open a command line window and enter the following commands:

```
sudo apt-get install tightvncserver autocutsel  
tightvncserver
```

Here is an explanation of what the commands do:

sudo	asks the system for elevated rights
apt-get	is the Advanced Packaging Tool command (kinda like NuGet)
install	is the action for apt-get, which followed by the packages that you want apt-get to install
tightvncserver	is the name of the program you want to run

My favorite explanation of what the sudo command does is from Randall Munroe's always excellent xkcd comic:

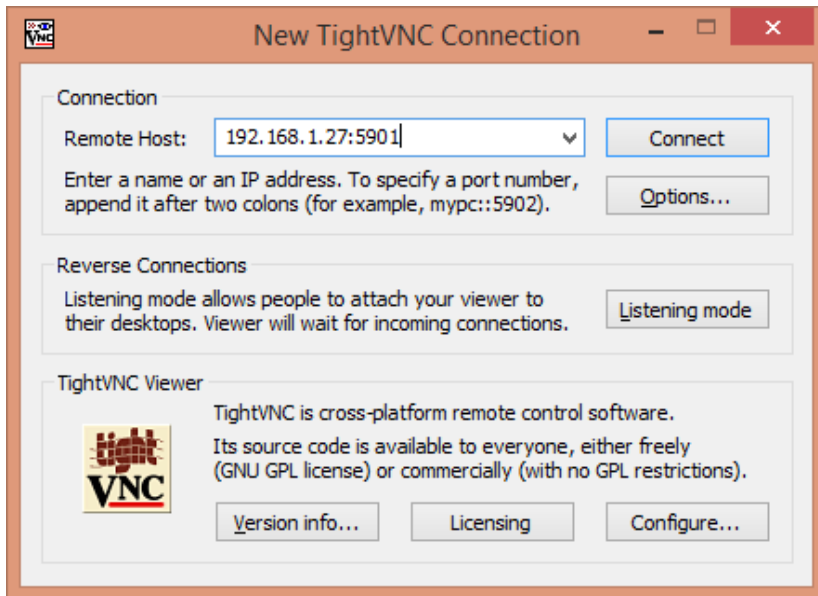


<http://xkcd.com/149/>

Raspberry Pi for the .NET Programmer

When you issue the "tightvncserver" command, and the VNC Server starts for the first time, you will be prompted to supply a password, and then each time you start your VNC client you will be asked for this password. You can safely ignore and say No to the view-only password. Note the message "New 'X' desktop is raspberrypi:1". The "1" tells you that VNC is listening on port 5901. If you run this command again, it would start another program and reply with "raspberrypi:2", which would mean it was running on port 5902.

To install the TightVNC Client on your Windows Desktop, download the program from <http://www.tightvnc.com/download.php> and load it on your system. To connect to your Pi, enter a Remote Host of your IP address plus the ":5901" and click Connect, then you will be prompted to enter your VNC password.



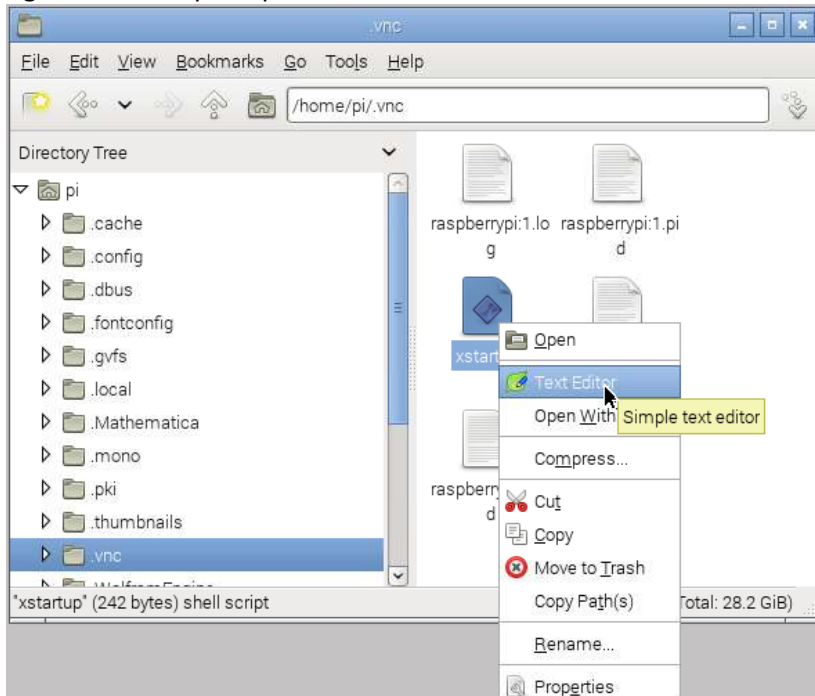
You should now have a remote session open from your Pi on your desktop so you can use your normal mouse and keyboard and start the system update patching process.

Enable Copy/Paste in TightVNC

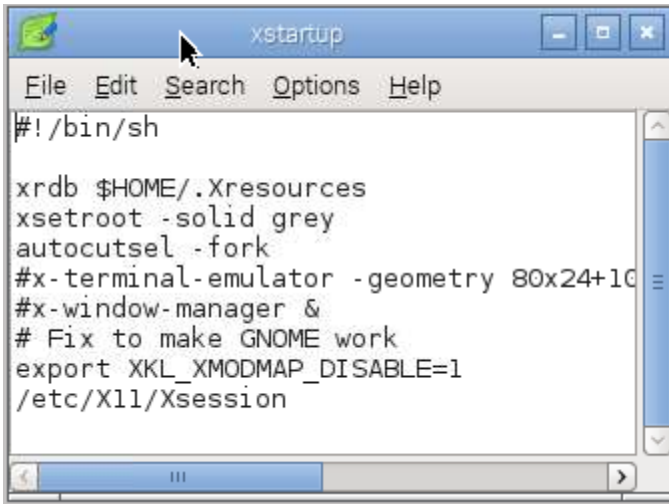
By default, one feature that is not enabled in TightVNC (that I think is sorely needed!) is the ability to copy and paste from your desktop into

Raspberry Pi for the .NET Programmer

and out of your Pi. To enable that, you will have to install one small feature. If you followed my instructions earlier when you apt-get installed tightvncserver, you added another package named "autocutsel". To enable it, you will need to edit the pi\.vnc\xstartup file and add one extra line of configuration. The ".vnc" directory is a hidden directory, which is indicated by having a period as the first character of the directory name. Note that this directory actually won't exist until you run tightvncserver for the first time. Open the File Manager, and turn on hidden files under the view menu, then Navigate to the .vnc\xstartup and right click and open up with the text editor.



Find the line "xsetroot -solid grey" line and then add a new line after it that contains the "autocutsel -fork" command.



Save the file and you should be all set. Stop and restart TightVNCServer and you should now be able to copy and paste from your desktop to your Pi to your heart's content.

Initial Setup

When you open up a command line window the first thing I usually do is to enlarge the small font in that window (Edit-Preferences-Style) to 14 points. I do this because I've found that when I have a remote terminal window open on my Surface 3, that default 10 point font is TINY!

As you read through this manual, you will see a lot of commands and files that will set up your Pi. I've included them in this document so that you can see what they do, but you probably don't want to type them all manually. To help you out, I've published all the scripts out to a GitHub repository:

<https://github.com/llupes/pisharp>

To use these script files, first fetch the first master file from the repository entering this command in the command line window on your Pi.

`wget https://raw.githubusercontent.com/llupes/pisharp/master/getScripts`

At this point you will have a file, but the operating system doesn't know that it's a script, so you have to go tell it to give the file execute permissions by running the "chmod" command.

`chmod 755 getScripts`

Raspberry Pi for the .NET Programmer

Finally, run that command file using the "sh", which will in turn fetch the other files and automatically mark them as executable.

```
sh getScripts
```

Here is an explanation of what the commands do:

wget	program that retrieves files from an http* source
chmod 755	marks a file as executable
sh	starts a command line interpreter to run some commands

Auto-Start TightVNC

At this point, you have the TightVNCServer installed and running, but if you reboot, it won't start up until you enter the "tightvncserver" command again. If you want it to start up automatically (and you do!), do the following.

Create a new file in the init.d directory. This is protected directory, so you have to use a sudo command to start your editor and get elevated privileges. At this point most tutorials will spell out a "sudo nano..." command for you. Why? Because that's the Unix admin way to do it, with the ubiquitous and obtuse Nano editor. But we are .NET programmers and we know there are better tools available, so let's use them! Leafpad is the alter ego on the Pi of NotePad, so let's use that. Enter the command:

```
sudo leafpad
```

Paste or type the following text into the file, and save that file as "/etc/init.d/vncboot".

```
### BEGIN INIT INFO
# Provides: vncboot
# Required-Start: $remote_fs $syslog sudo networking
# Required-Stop: $remote_fs $syslog sudo networking
# Default-Start: 2 3 4 5
# Default-Stop: 0 1 6
# Short-Description: Start/Stop TightVNC at boot
### END INIT INFO

export USER='pi'
```


Raspberry Pi for the .NET Programmer

```
case "$1" in
start)
echo "Starting VNC Server"
su $USER -c '/usr/bin/vncserver :1 -geometry 1280x800'
;;
stop)
echo "Stopping VNC Server"
pkill Xtightvnc
;;
*)
echo "Usage: /etc/init.d/tightvncserver {start|stop}"
exit 1
;;
esac
exit 0
```

Tip: You can fetch the contents of the file from GitHub and then simply copy it to the directory using these commands: (if you ran the getScripts command earlier you already have it, so you can skip the first line)

```
wget https://raw.githubusercontent.com/lluppess/pisharp/master/1_vncboot
sudo cp /home/pi/1_vncboot /etc/init.d/vncboot
```

Then you'll need to mark this as an executable script:

```
sudo chmod 755 /etc/init.d/vncboot
```

You can now start or stop the service manually by running this command. Make sure you do this at least once so that you set up the remote access password:

```
sudo /etc/init.d/vncboot start
sudo /etc/init.d/vncboot stop
```

I'm not sure why, but it seems to need a reboot here before you do the next update command or it doesn't work... so you should reboot your Pi now by executing the command: "sudo reboot"

However, just having that file in the directory won't make it auto-start. To do that, you will have to enter the command:

```
sudo update-rc.d vncboot defaults
```

Reboot your Raspberry Pi, and you should be able to connect across the network using the VNC Viewer and now you won't need an extra keyboard, mouse, and monitor for your Pi cluttering up your desktop anymore!

Applying System Patches and Updates

Like any new operating system, there are always patches and updates that you need to install when you first install an operating system.

Open a command prompt (LXTerminal) and run these commands:

```
sudo apt-get update
sudo apt-get dist-upgrade
sudo apt-get install htop autoconf automake screen
sudo rpi-update
sudo reboot
```

Tip 1: If you ran getScripts earlier, you already have this file. If not, you can fetch these commands from GitHub using this command (enter the wget all on one line):

```
wget https://raw.githubusercontent.com/lluppes/pisharp/master/2_update_os
chmod 755 2_update_os
```

Execute the command file with this command:

```
sh 2_update_os
```

Tip 2: If the script stops and asks you to confirm with a "Y/n", make sure you enter an UPPER case Y, otherwise sometimes the script just stops.

Here is an explanation of what the commands do:

sudo	asks the system for elevated rights
apt-get	is the Advanced Packaging Tool command
update	will download software package lists
dist-upgrade	updates the software packages (this different than just using "upgrade" as it retrieves all the dependencies also)
rpi-update	will update the Raspberry firmware
install	will install a new program
"htop autoconf automake screen" – are useful utilities	

Once everything is updated, reboot with a "sudo reboot" command.

CHAPTER 3

Your Personal Toolkit

Have you noticed that meetings go smoother without any knowledge or expertise? Scott Adams - Dilbert Cartoon

Now that you have the base system up and running, it's time to configure some tools and utilities to make your life easier before we continue.

A Better Browser

I don't know about you, but when I do any kind of web development, I prefer to start with the Chrome browser. Wouldn't it be awesome if we could get that on the Pi...? Well, you can! There is a version of Chromium (a cousin to Google Chrome) that is available for the Pi. Simply run the command:

```
sudo apt-get install chromium-browser
```

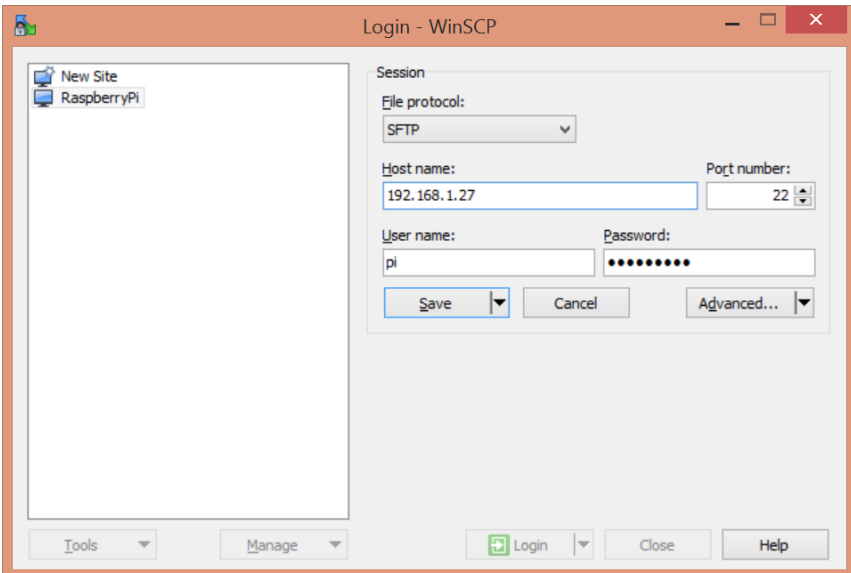
Now a version of Chromium is ready for you to use, just like its big brother on the desktop!

Tip 1: If you ran 2_update_os earlier, this package was already installed for you. If you run it again, that's OK – it will just say it's already installed.

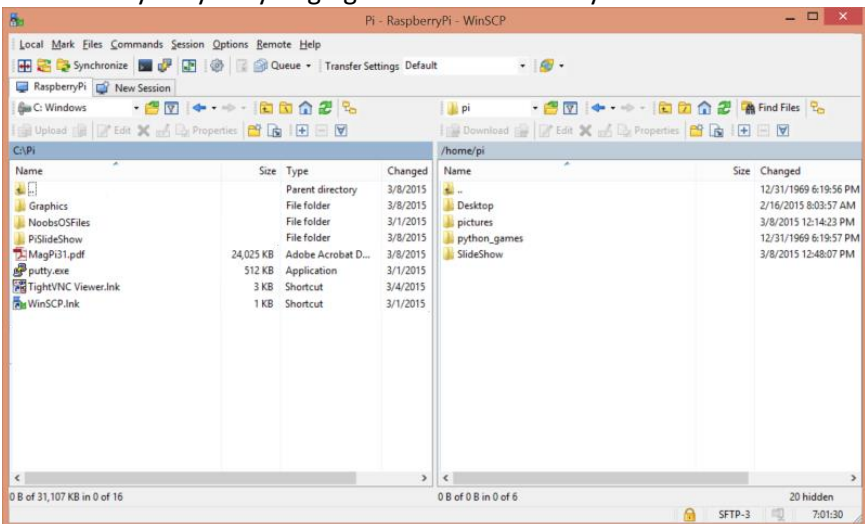
File Transfer Utility

Another utility that you will need to have is a simple way of transferring files, like an FTP client. The program that I like the best is WinSCP. Go to <http://winscp.net/eng/download.php> and download and install the appropriate version of this awesome program. Fire it up, and create a profile for your Pi with your IP address and username/password:

Raspberry Pi for the .NET Programmer



Now you'll have a familiar drag/drop explorer style program that will make it very easy for you to get files on an off of your Pi!

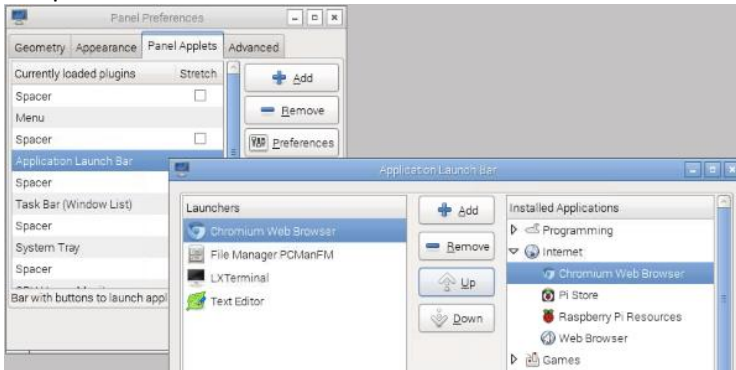


Raspberry Pi for the .NET Programmer

Personal Touches

If you want to customize the menu to have the programs you like at your fingertips, right Click on menu bar and select Panel Settings.

- You can move menu to bottom of screen if you like.
- Change the icons on the menu bar by selecting the Panel Applets – Preferences button. You can add Chrome and Text Editor, and then remove Ephypany, Mathematica, and Wolfrum if you don't plan to use them.



CHAPTER 4

Installing .NET Mono

"He is intelligent, but not experienced. His pattern indicates two-dimensional thinking." Spock in Star Trek: Wrath of Khan

Install Mono (the wrong way)

Installing Mono seems to be (almost) as simple as running the following command:

```
sudo apt-get mono-complete
```

If you do that, check the version number by executing this command

```
mono --version
```

However... this will get you version 3.2.8, which is definitely not the latest. If you use this, you will get a *partially* working Mono that will work for very simple programs and crash when you do something really complex like try to put a textbox on a screen. If you do that, you will not only be intelligent, you'll be experienced. (*Or you can just take my word for it, since I experienced it for you. BTW – that never worked with my teenagers either – they had to make those mistakes for themselves!*)

Install Mono (the right way)

Installing the latest Mono isn't quite as intuitive as the previous way of doing it. You need a few configurations first. The following steps will load the latest build of Mono on your Pi:

```
sudo apt-key adv --keyserver keyserver.ubuntu.com --recv-keys
3FA7E0328081BFF6A14DA29AA6A19B38D3D831EF
echo "deb http://download.mono-project.com/repo/debian wheezy main" |
sudo tee /etc/apt/sources.list.d/mono-xamarin.list
sudo apt-get update
sudo apt-get upgrade
sudo apt-get install mono-complete
```

From <http://www.mono-project.com/docs/getting-started/install/linux/>

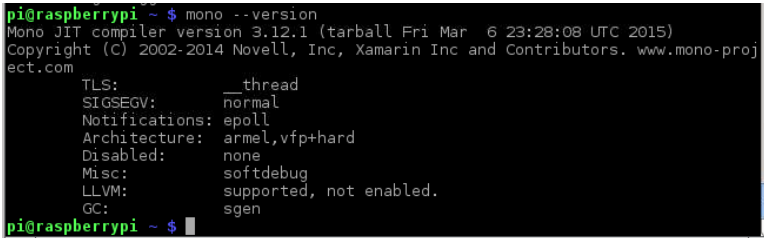
Raspberry Pi for the .NET Programmer

Tip: If you ran getScripts earlier, you already have this file, and can skip the first two commands.

```
wget https://raw.githubusercontent.com/lluppess/pisharp/master/3_install_mono
chmod 755 3_install_mono
sh 3_install_mono
```

When you are done, check the version number with this command, and you should see a 3.12* version:

```
mono --version
```

A terminal window on a Raspberry Pi showing the output of the 'mono --version' command. The output includes the Mono JIT compiler version (3.12.1), copyright information, and various configuration details like TLS, SIGSEGV, Notifications, Architecture, Disabled, Misc, LLVM, GC, and thread support.

```
pi@raspberrypi ~ $ mono --version
Mono JIT compiler version 3.12.1 (tarball Fri Mar  6 23:28:08 UTC 2015)
Copyright (C) 2002-2014 Novell, Inc, Xamarin Inc and Contributors. www.mono-project.com
   TLS:               __thread
   SIGSEGV:           normal
   Notifications:     epoll
   Architecture:      armel,vfp+hard
   Disabled:          none
   Misc:              softdebug
   LLVM:              supported, not enabled.
   GC:                sgen
pi@raspberrypi ~ $
```

CHAPTER 5

Running a C# WinForms Program

"Only a fool argues with badgers, grizzlies, or cooks." Unknown

Running a Windows Forms application on a Pi (or Console App) is actually very simple.

- Create and compile a Windows Forms App
- Use WinSCP to copy the files from your bin folder to the Pi
 - Create a directory on the Pi to hold your files
 - Copy the EXE and any referenced DLL's
 - You don't need the .XML or .PDB files or the vshost.* files
- Switch over to the Pi and double click to run the program
- Voila – it should just work!

I've tested this with a fairly simple forms program that had dialogs, file IO, etc., and had no problems. That's about as far as I've explored so far.

More details coming in an upcoming version...

CHAPTER 6

Installing ASP.NET vNext

"It's no wonder that truth is stranger than fiction. Fiction has to make sense."

Mark Twain

One of the awesome features of the new version of ASP.NET v5 is that it's designed to run on many operating systems, including on a Mac and Linux. However, there are a couple of pieces that you need to install in order to make your new ASP.NET vNext pages work – primarily the KVM and Libuv modules.

Update Your Certificates

In order to get those programs installed properly, you'll need some updated certificate. Update the certificates on your Pi with these commands. You'll be prompted multiple times to confirm the certificates, and they will warn you that they certs are invalid (don't believe it!). Just say "Y" to each certificate and you should have no problems with this.

```
sudo certmgr -ssl -m https://go.microsoft.com
sudo certmgr -ssl -m https://nugetgallery.blob.core.windows.net
sudo certmgr -ssl -m https://nuget.org
sudo certmgr -ssl -m https://www.myget.org/F/aspnetvnext/
mozroots --import --sync
```

Tip: You can do this automatically:

```
wget https://raw.githubusercontent.com/lluppes/pisharp/master/4_install_certs
chmod 755 4_install_certs
sh 4_install_certs
```

Install KVM

To install KVM (the kernel Virtual Machine) use these commands:

```
mkdir ~/sources/  
mkdir ~/sources/aspnet5  
cd ~/sources/aspnet5  
git clone git://github.com/aspnet/home.git  
sh ~/sources/aspnet5/home/kvminstall.sh  
echo " Run this command: source /home/pi/.k/kvm/kvm.sh"  
echo " Then run this command: kvm upgrade"
```

I'm not sure why, but the last two commands fail when I try to include and run them in this batch file, so I had to run them manually...

Tip: You can do this automatically:

```
wget https://raw.githubusercontent.com/lluppess/pisharp/master/5_install_kvm  
chmod 755 5_install_kvm  
sh 5_install_kvm
```

Install Libuv

Libuv is a cross-platform support library which was originally written for NodeJS, designed around the event-driven asynchronous I/O model.

Install it using these commands:

```
wget http://dist.libuv.org/dist/v1.0.0-rc1/libuv-v1.0.0-rc1.tar.gz  
tar -xvf libuv-v1.0.0-rc1.tar.gz  
cd libuv-v1.0.0-rc1/  
./gyp_uv.py -f make -Duv_library=shared_library  
make -C out  
sudo cp out/Debug/lib.target/libuv.so /usr/lib/libuv.so.1.0.0-rc1  
sudo ln -s libuv.so.1.0.0-rc1 /usr/lib/libuv.so.1
```

Tip: You can do this automatically:

```
wget https://raw.githubusercontent.com/lluppess/pisharp/master/6_install_libuv  
chmod 755 6_install_libuv  
sh 6_install_libuv
```

Raspberry Pi for the .NET Programmer

Running Your First ASP.NET MVC6 Page

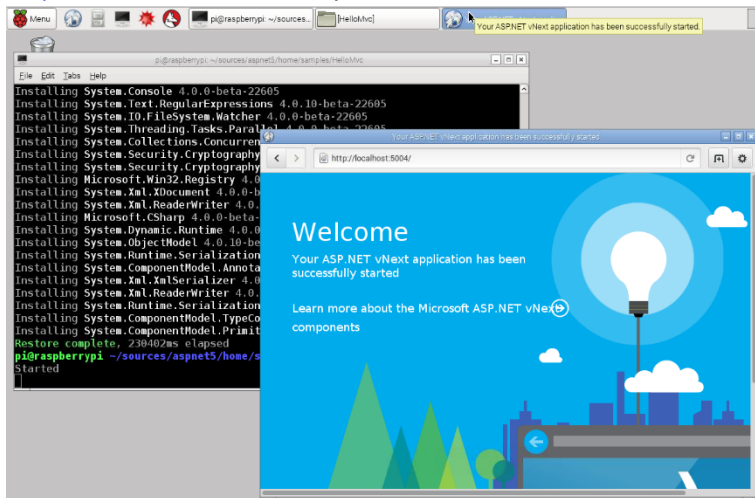
After installing and configuring the app, run these commands:

```
cd /home/pi/sources/aspnet5/home/samples/HelloMvc
unset RUNLEVEL
kpm restore
k kestrel
```

The unset RUNLEVEL command is there because I keep getting the error "System.ArgumentException: An element with the same key already exists in the dictionary". If you run the command "printenv" you'll see that you have two rows that say runlevel=. The command "unset RUNLEVEL" will remove the second one, then you can run the k kestrel command and it should work.

There must be a way to get rid of this, but I haven't found it yet!

If everything worked and was installed properly, you should see the word "Started" in your command window. Open a browser and navigate to <http://localhost:5004/> and you should see the default welcome screen.



Raspberry Pi for the .NET Programmer

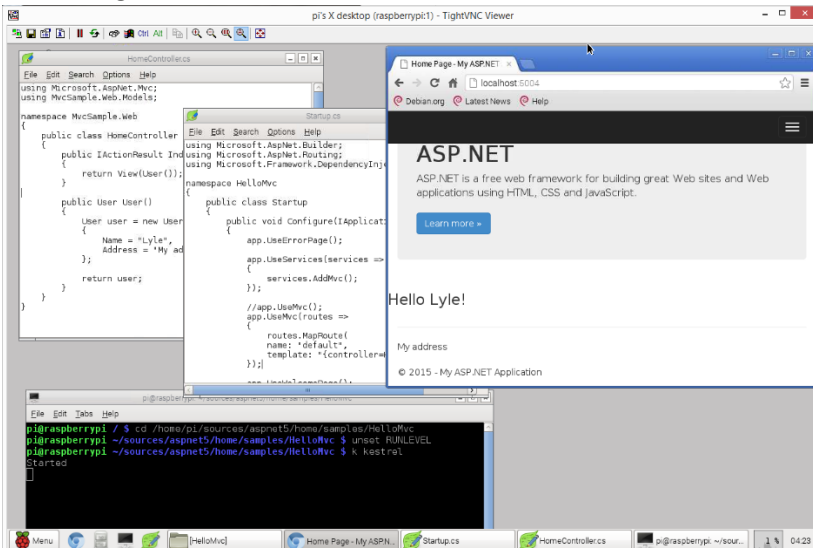
To replace the welcome screen with the more typical index page, you have to add a route in the startup.cs. Edit the HelloMVC/startup.cs and comment out the UseMvc() line and add a new route:

```
//app.UseMvc();
app.UseMvc(routes =>
{
    routes.MapRoute(
        name: "default",
        template: "{controller=Home}/{action=Index}/{id?}");
});
```

If you want some customized content, edit the HelloMVC HomeController.cs to have User.Name equal to your name.

```
User user = new User() {
    Name = "Lyle",
    Address = "My address"
};
```

Go back to your command window and hit Ctl-Z to stop the process. It's stopped but actually still running so you will need to type in the command "kill %1" to really kill it. Once it's gone, type in "k kestrel" command to restart it and then refresh your browser, and you should end up with something like this:



CHAPTER 7

Writing an ASP.NET v5 / MVC6 Application

Coming soon in an upcoming release... more details about what's new in MVC6 and tips on how to write an MVC web application for this platform...

Conclusion

*"No more training do you require. Already know you, that which you need."
Yoda in "Star Wars VI - Return of the Jedi"*

You did it! At this point, you should have a fully functioning MVC6 web server for less than \$100! OK – maybe not really fully functioning, but it certainly has some potential in the new Internet of Things device world.

I hope this guide helped - I know it would have been helpful to me when I started doing this!

Lyle

Appendix

Here are some other interesting and helpful blogs that I've found along the way:

<http://www.hanselman.com/blog/HowToRunASPNET5Beta3OrGoLangOnARaspberryPi2.aspx>

<http://schlapsi.com/2014/08/aspnet-vnext-running-on-mac-os-x/>

<http://www.raspberry-sharp.org/>

<http://www.neil-black.co.uk/raspberry-pi-beginners-guide>

<http://taoofmac.com/space/dev/dotNET/ARM>

<http://j.tlms.be/2014/11/running-asp-net-on-a-raspberry-pi-with-mono-and-owin/>

<http://raspberryywebserver.com/>

About the Author

Lyle Luppé has been developing applications for over 28 years, starting on mainframes and AS/400's, then moving to client-server and Windows applications, and then on to web development. After working at very large companies and very small consulting firms, he started his own business in 2000, doing just about everything from building servers to wiring networks to being the DBA. He enjoys doing a wide variety of tasks, but likes pure creative development the best. Lyle wrote the book "Delivering Mobile-Friendly Websites with MVC4" in 2012, and has presented at several conferences on that topic. He has also taught other classes at corporations and at the college level, and enjoys finding chances to share what he is learning with others.