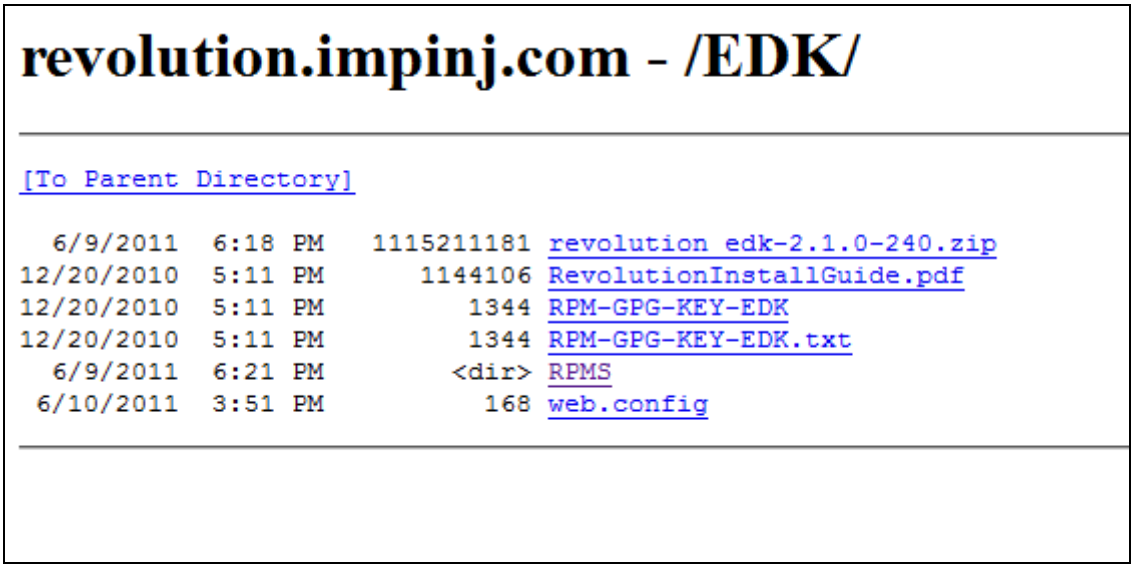
How to setup development environment for Impinj Readers, with Qt Framework in Windows 7

1. Download the Impinj development environment from: http://revolution.impinj.com/EDK/

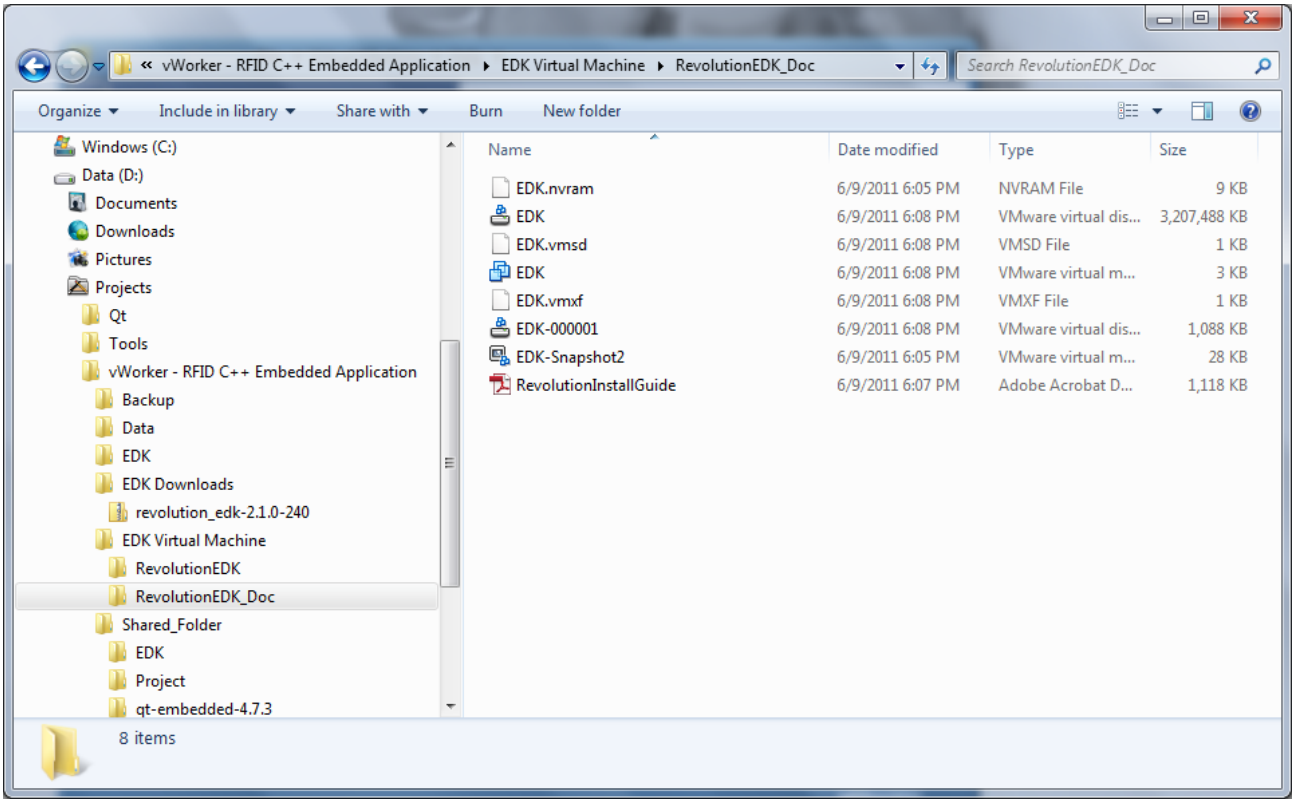


revolution.impinj.com - /EDK/

[To Parent Directory]

6/9/2011	6:18 PM	1115211181	revolution_edk-2.1.0-240.zip
12/20/2010	5:11 PM	1144106	RevolutionInstallGuide.pdf
12/20/2010	5:11 PM	1344	RPM-GPG-KEY-EDK
12/20/2010	5:11 PM	1344	RPM-GPG-KEY-EDK.txt
6/9/2011	6:21 PM	<dir>	RPMS
6/10/2011	3:51 PM	168	web.config

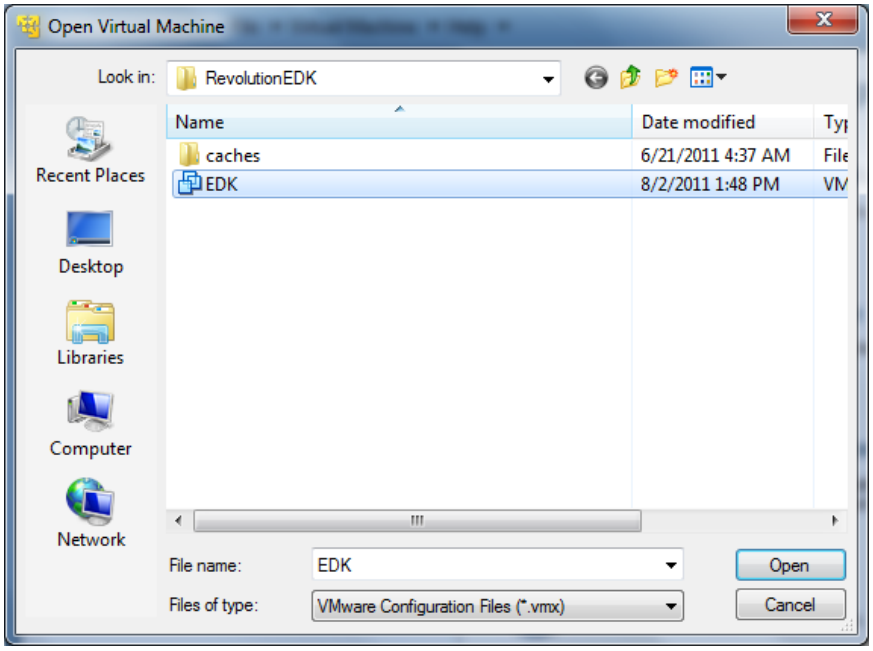
2. Unzip the EDK to your local disk. In my case I used the following folder scheme:



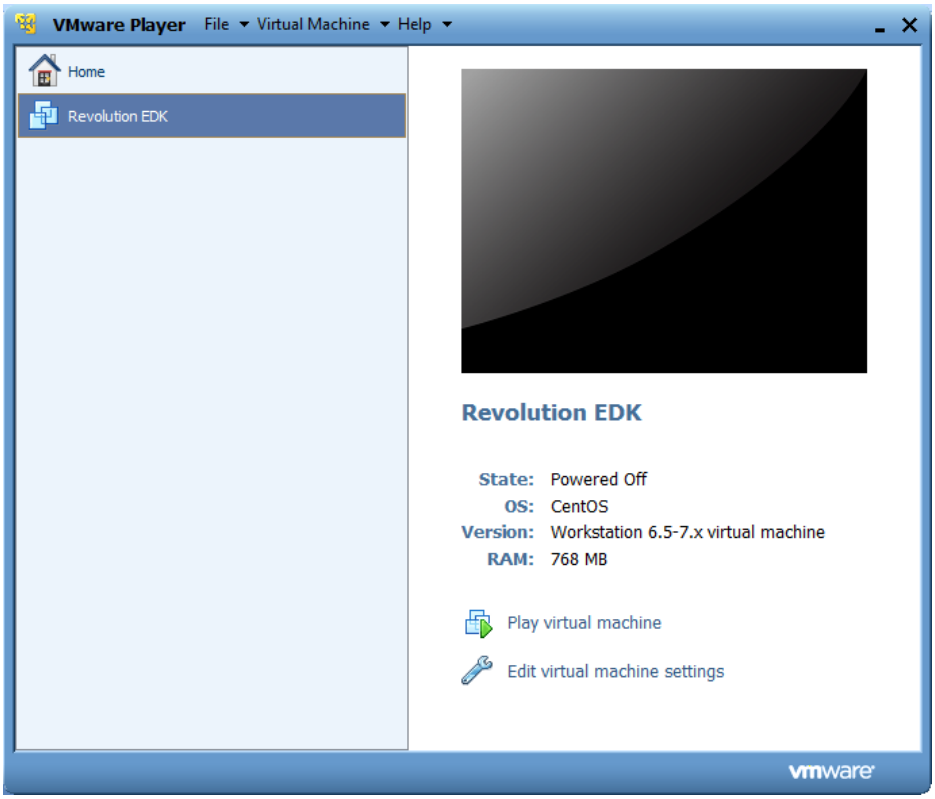
3. Download and install the free VMWare Player from : http://downloads.vmware.com/d/info/desktop_downloads/vmware_player/3_0

4. Now its time to configure and play the virtual machine. Run your installation of VMWare Player.

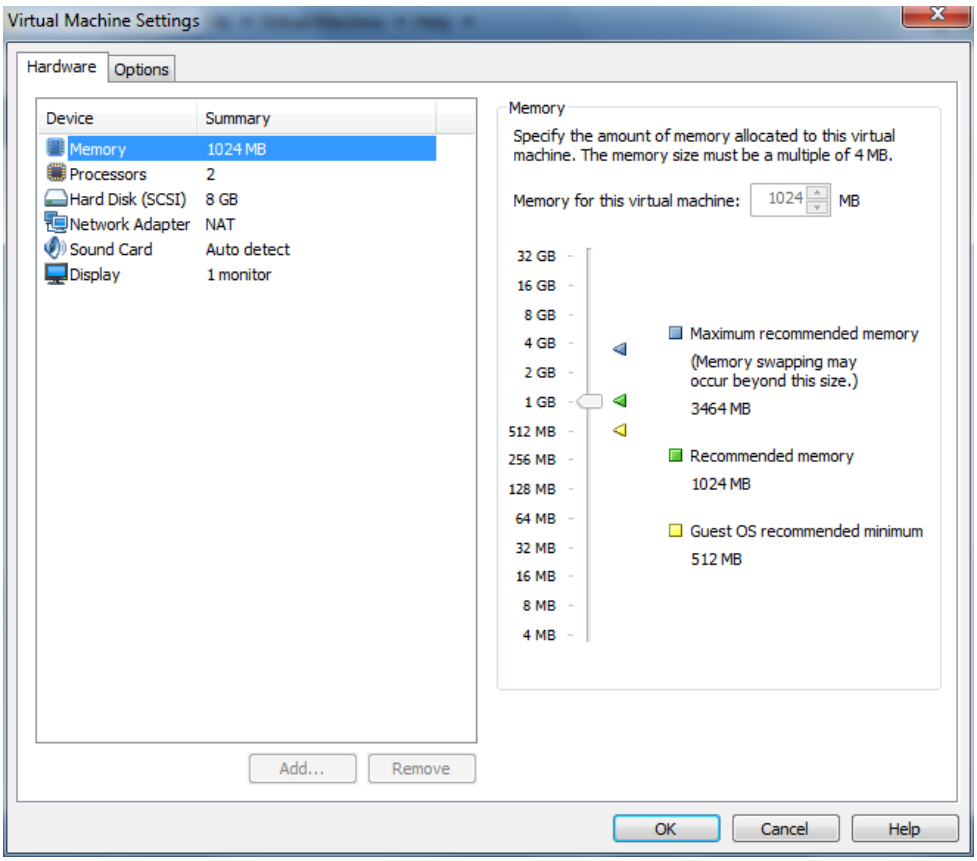
5. From menu select File > Open Vitual Machine. And select your virtual machine in the same folder you unzipped before.

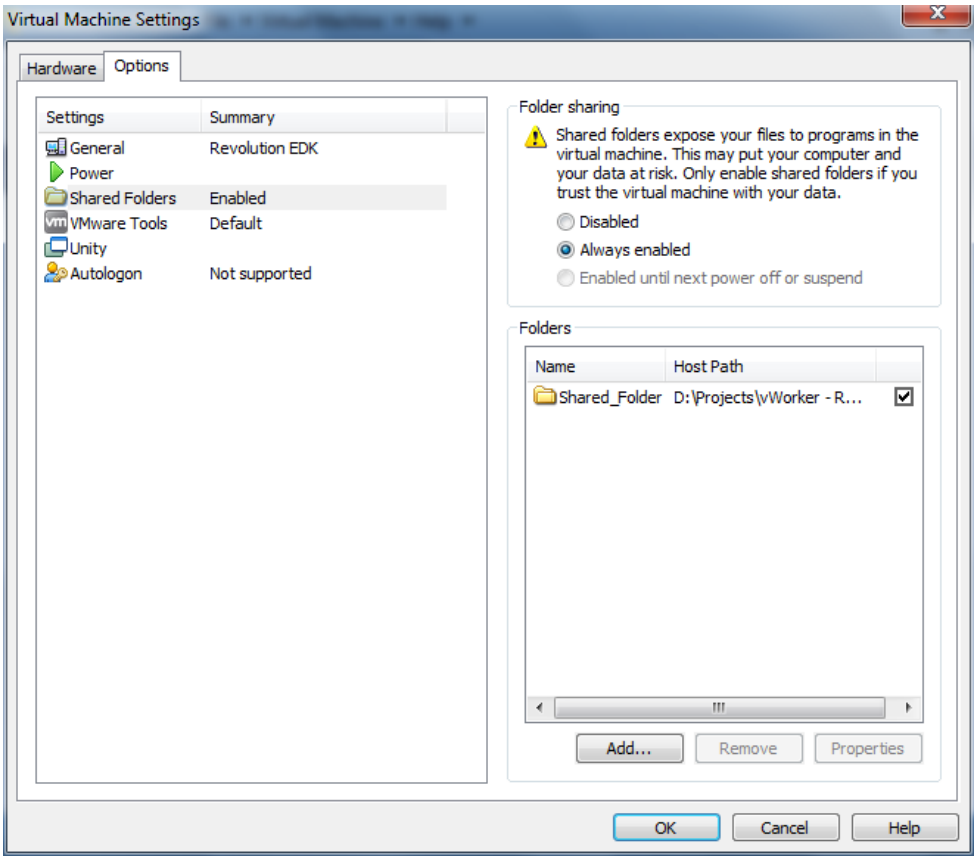


6. After selecting Open, the virtual machine will be added to the library.



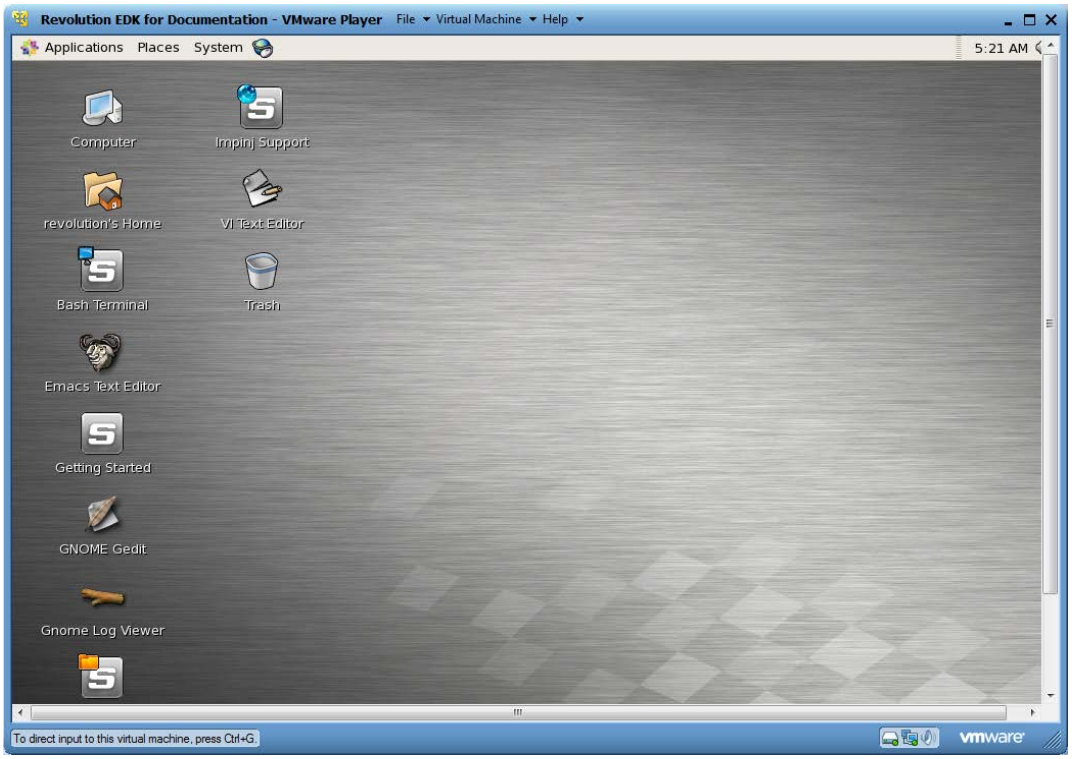
7. Now go to ‘Edit Virtual Machine Settings’, and set variables as follows.



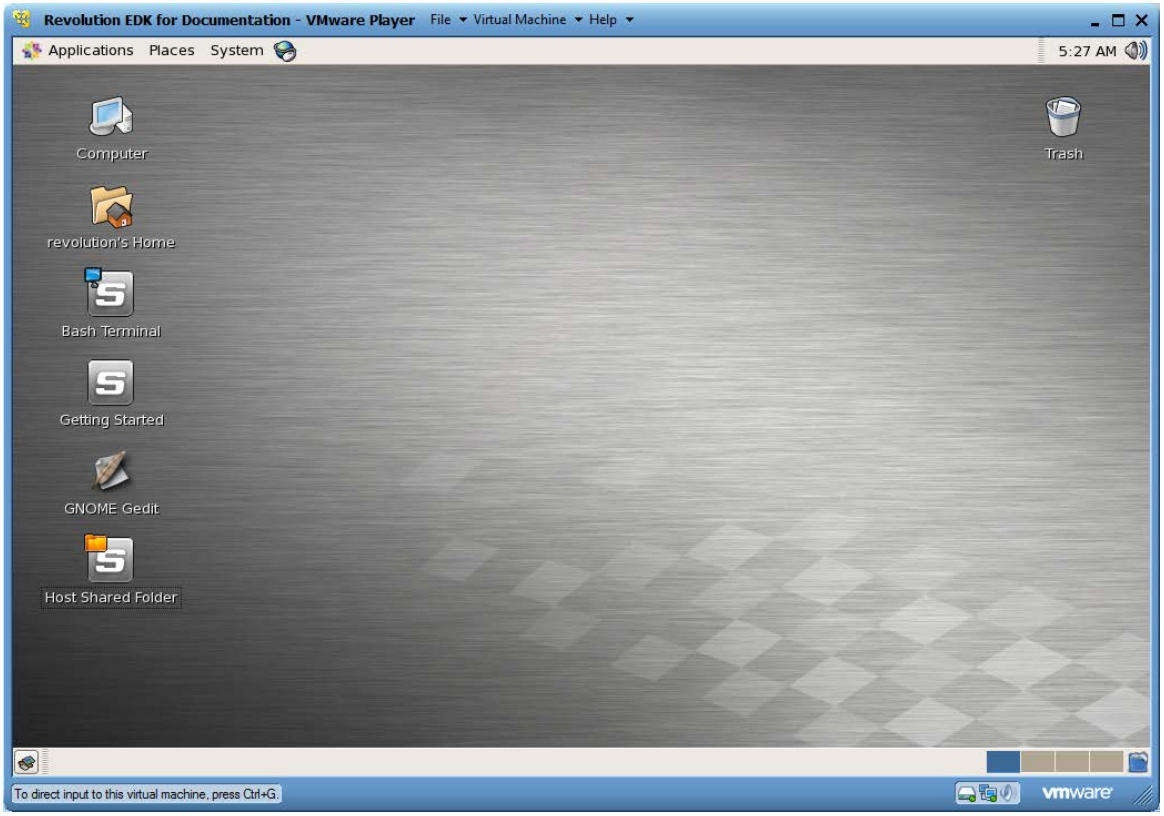


8. Press Ok and return to the main window. Now press ‘Play virtual machine’. The virtual machine will start booting up. Wait until you get the following screen.

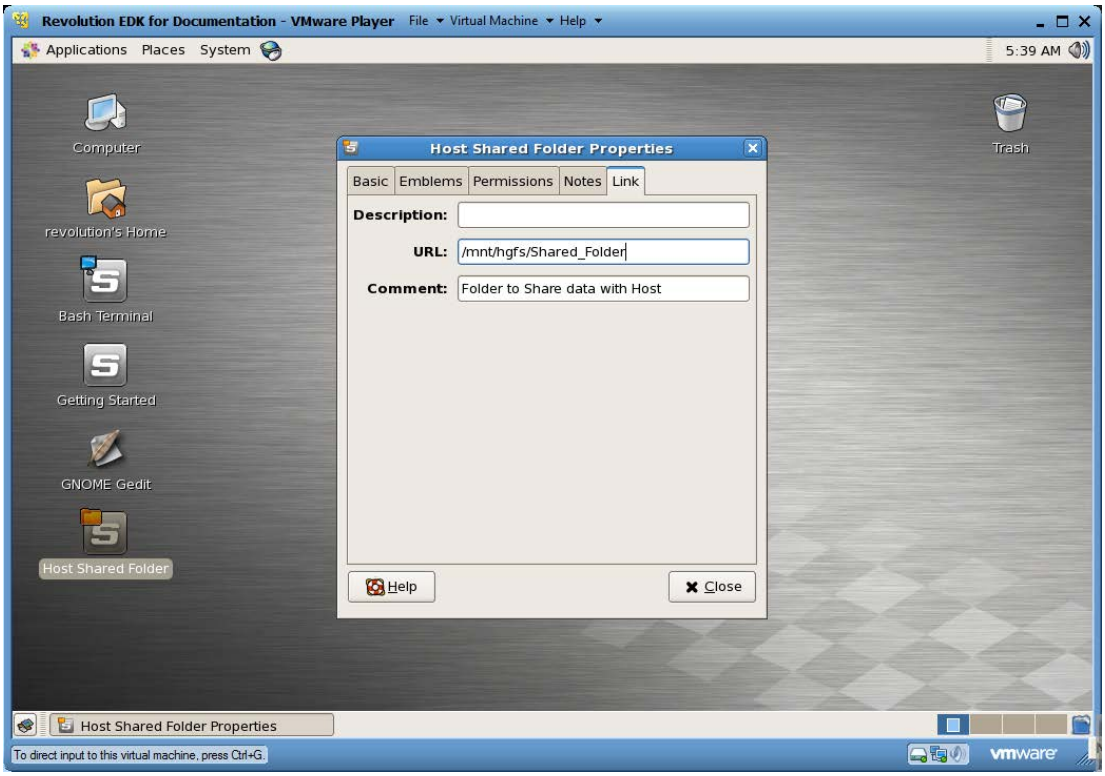
9. Enter username “**revolution**”, and proceed to the main screen.



10. Now we will configure the environment inside the virtual machine. For starters, you can have a look at basic documentation by clicking the icon named ‘Getting Started’. We will get rid of some irrelevant icons from the desktop.



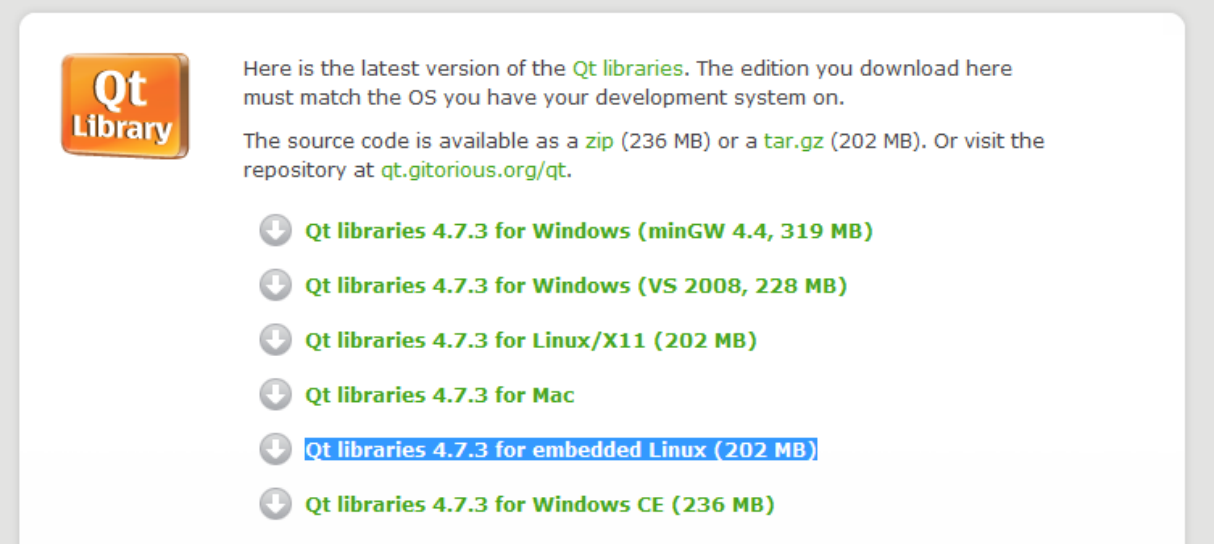
11. Now edit the Host Shared Folder icon, and change its properties to reflect the correct shared folder path.



Double click the icon afterwards to check that it opens the folder without giving errors. We will need this folder a lot during the course of development.

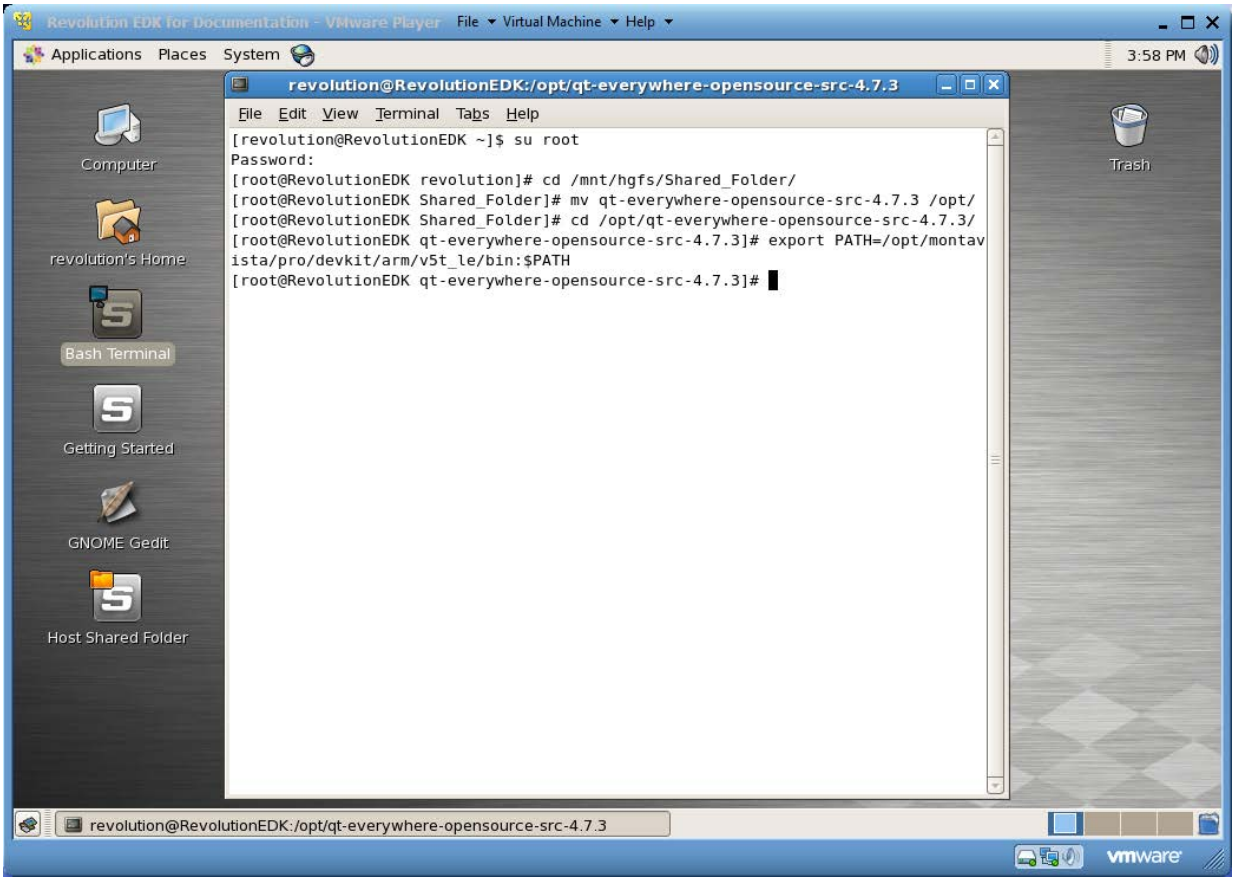
12. Since I decided to use a framework to develop the project, so that a lot of convenience classes are available for easy development. For that I selected my favorite Qt framework, now owned by Nokia. Go to the following link in the host and download the "Qt libraries 4.7.3 for embedded Linux (202 MB)"

http://qt.nokia.com/downloads

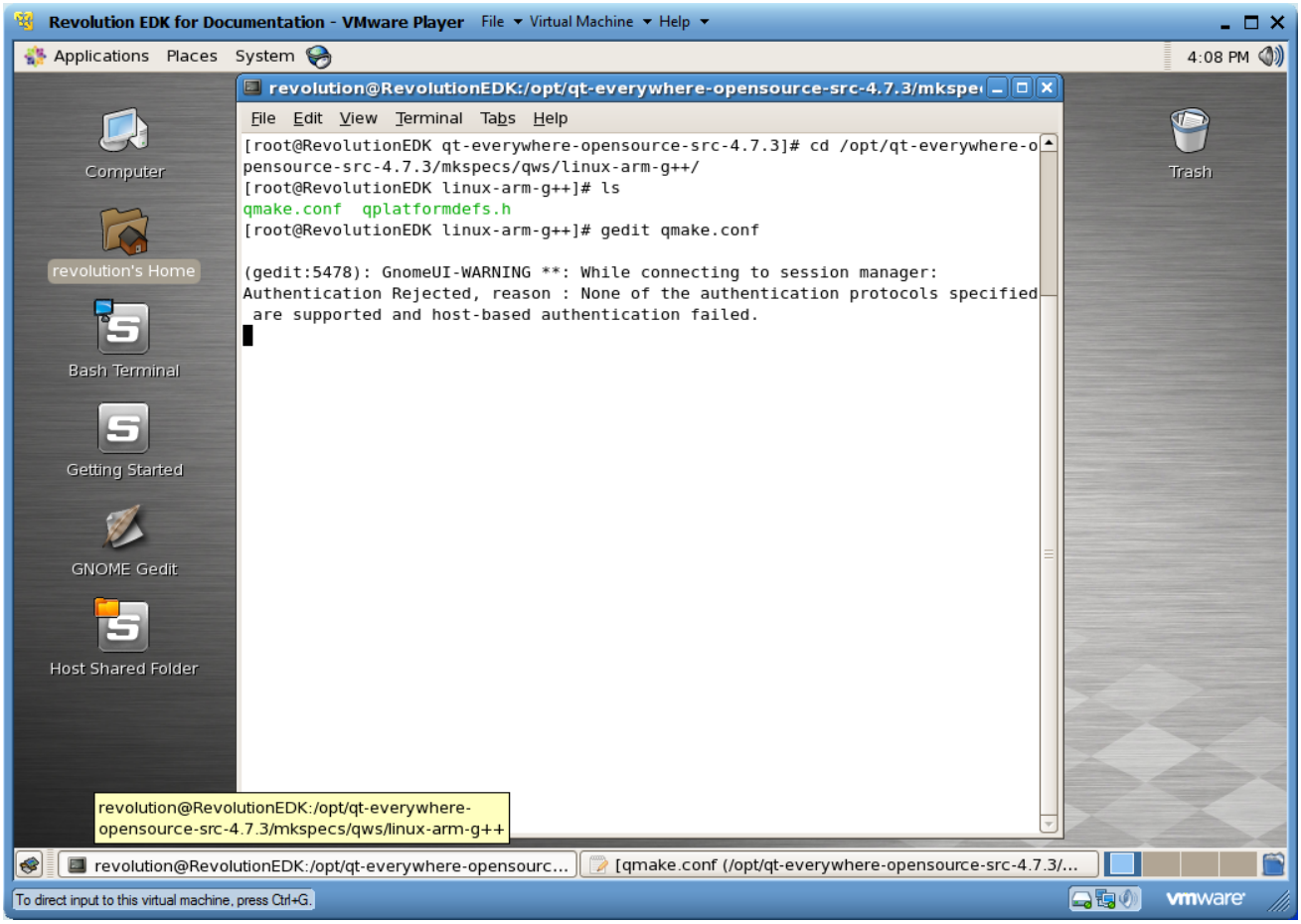


Unzip the downloaded archive to the Shared_Folder using 7Zip utility or your favorite archive handling program. Extraction can take a while so please be patient.

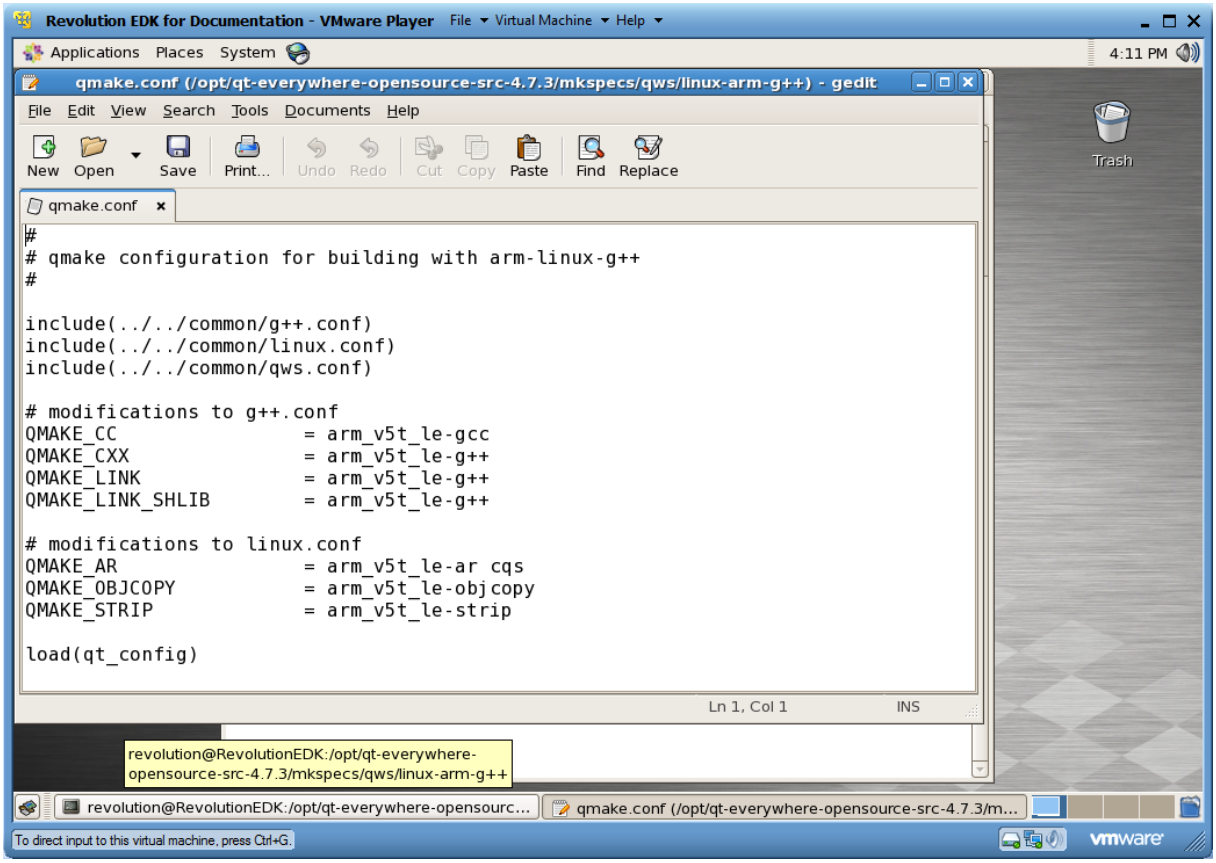
13. Now go to the virtual machine and open a Bash Terminal, and issue following commands on the command line. Root password is 'edk'.



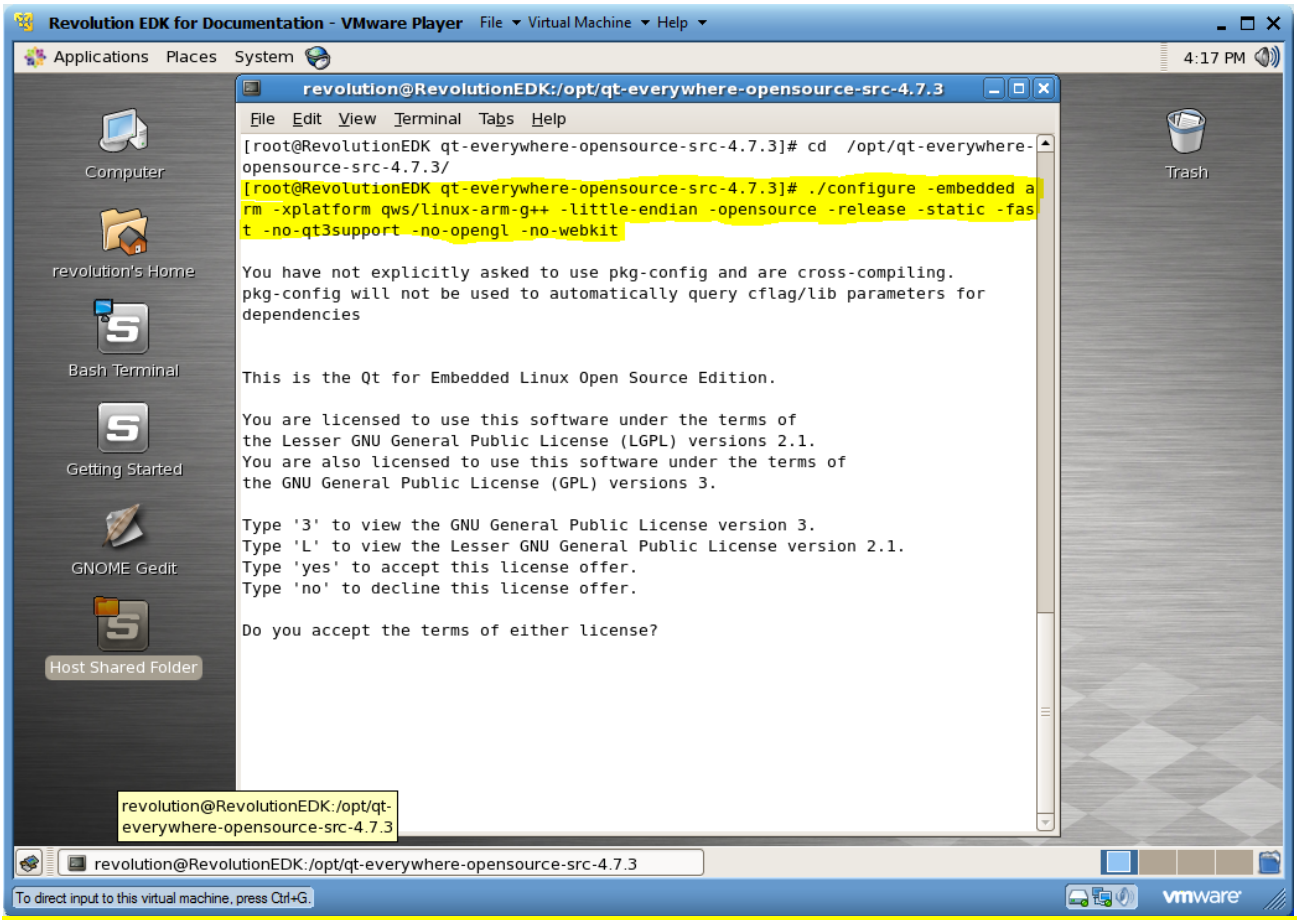
14. Then use following commands.



15. This will open a gedit session where we will change the names of variables to as follows.

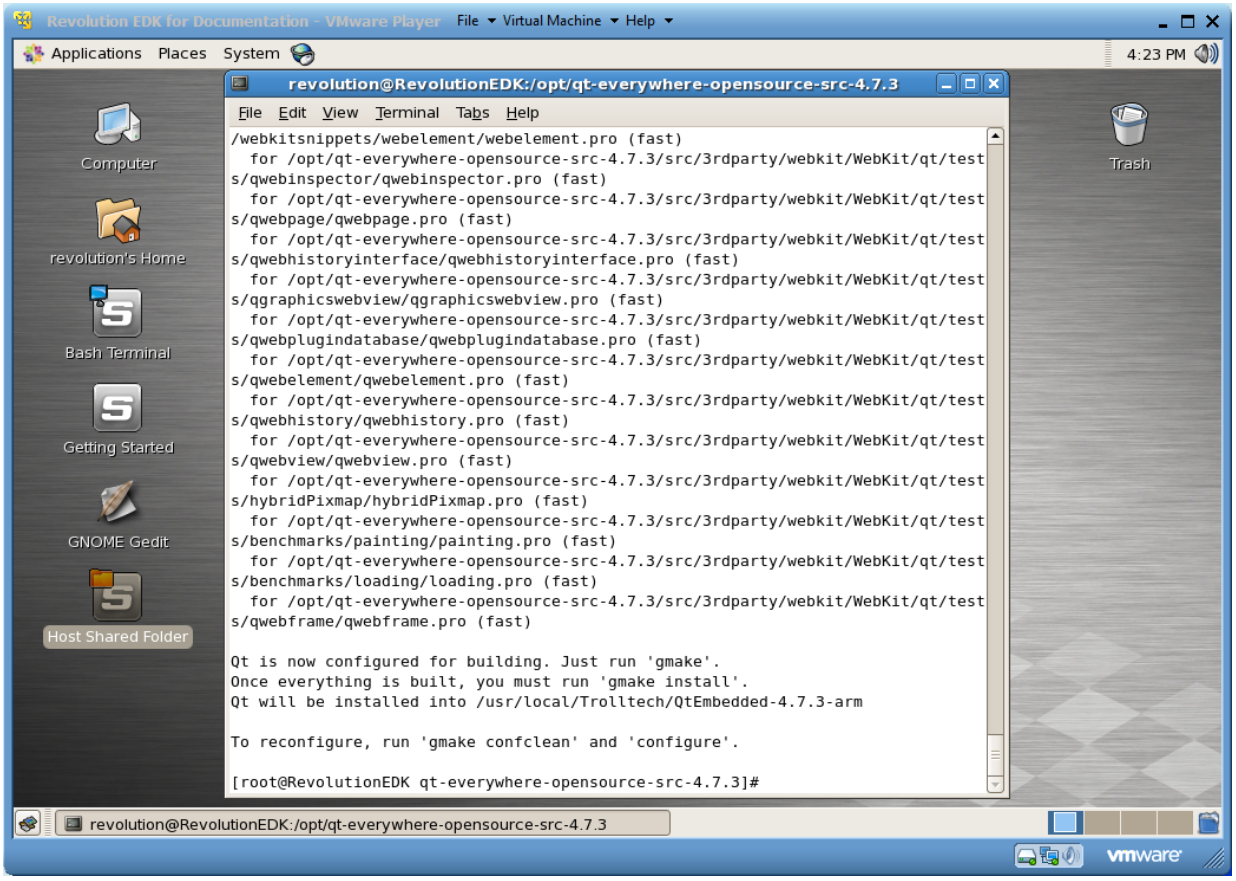


16. Press Save and exit the gedit session. Now we are ready to cross-compile the Qt framework for our impinj speedway hardware. Now on the Bash terminal issue the following command.

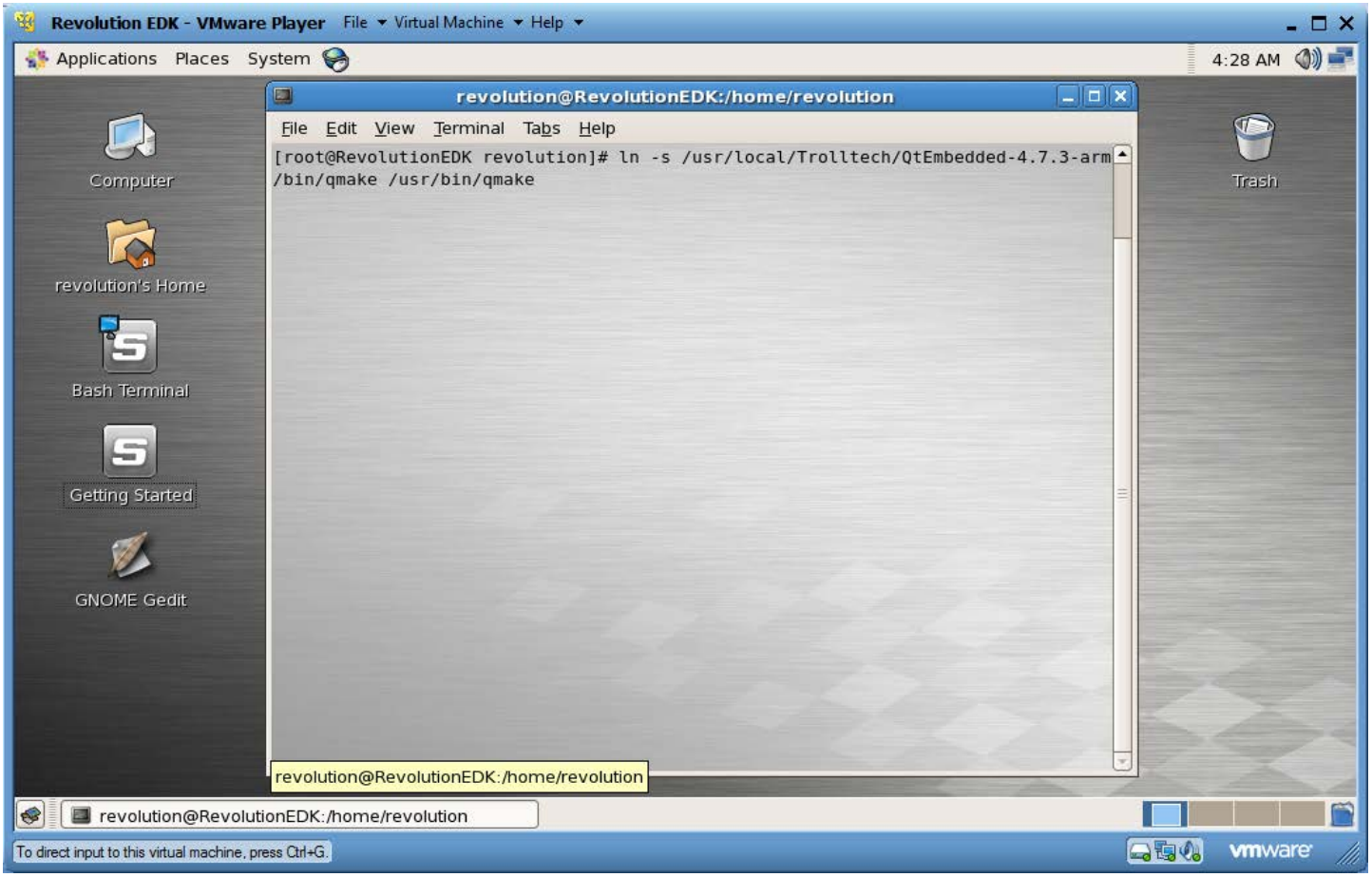


The yellow highlighted command is very important and should be typed carefully with no errors.

17. Type ‘yes’ and press enter. Now please be patient and wait for the process to finish. The process will modify make files for the entire Qt framework source. At the end of the process, the screen should look like this.

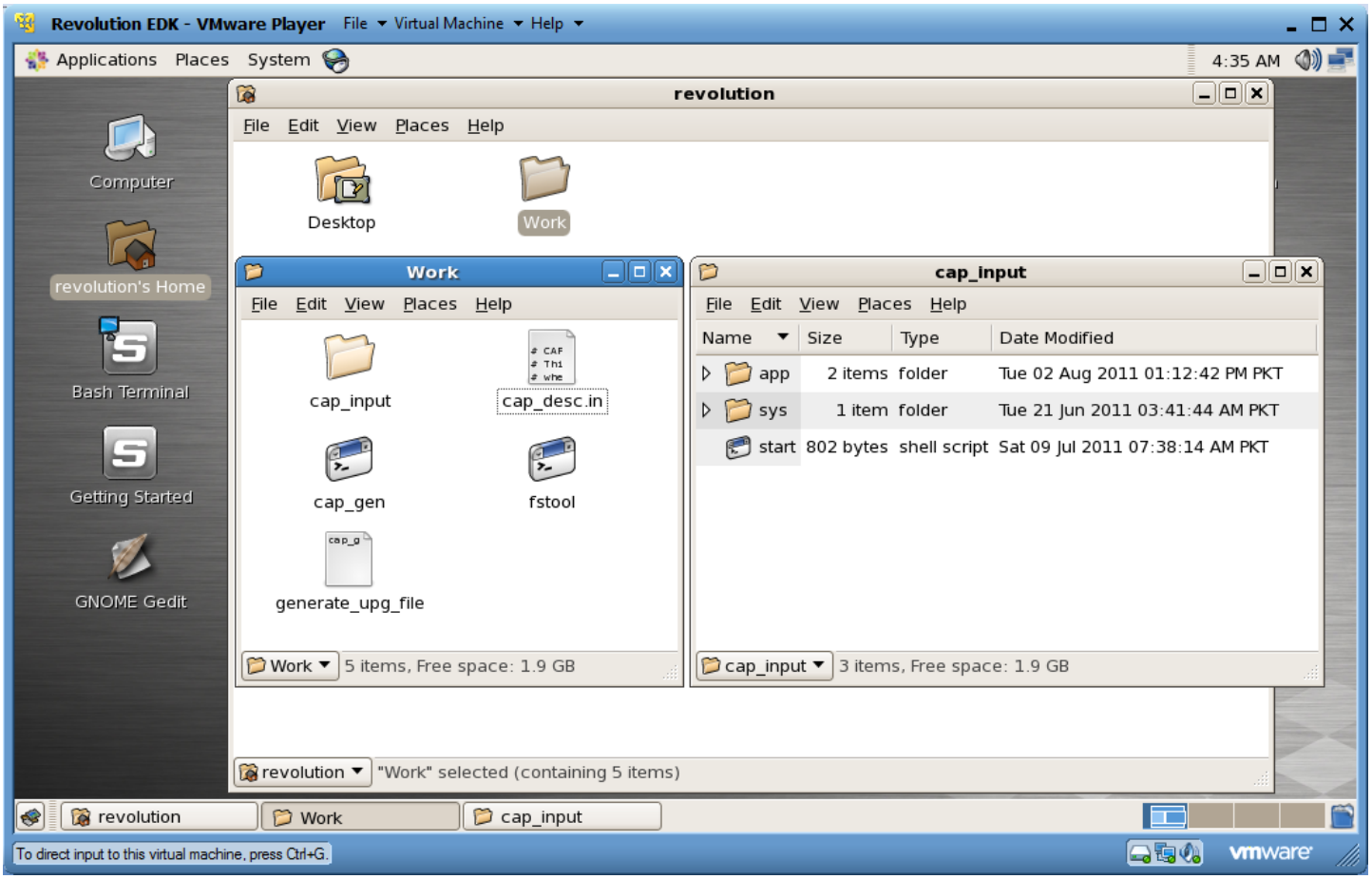


18. Follow the on-screen instructions, and issue '**gmake**' and '**gmake install**' commands. These two commands will take a long time to process. When the process is clean run the following command.



If it says, file exists, then its ok. We only wanted to make sure if for some reason the symbolic link wasn't created, then it should be present.

19. Now we will create our working folders. Open Revolution's Home from desktop icon and create folders as shown below.



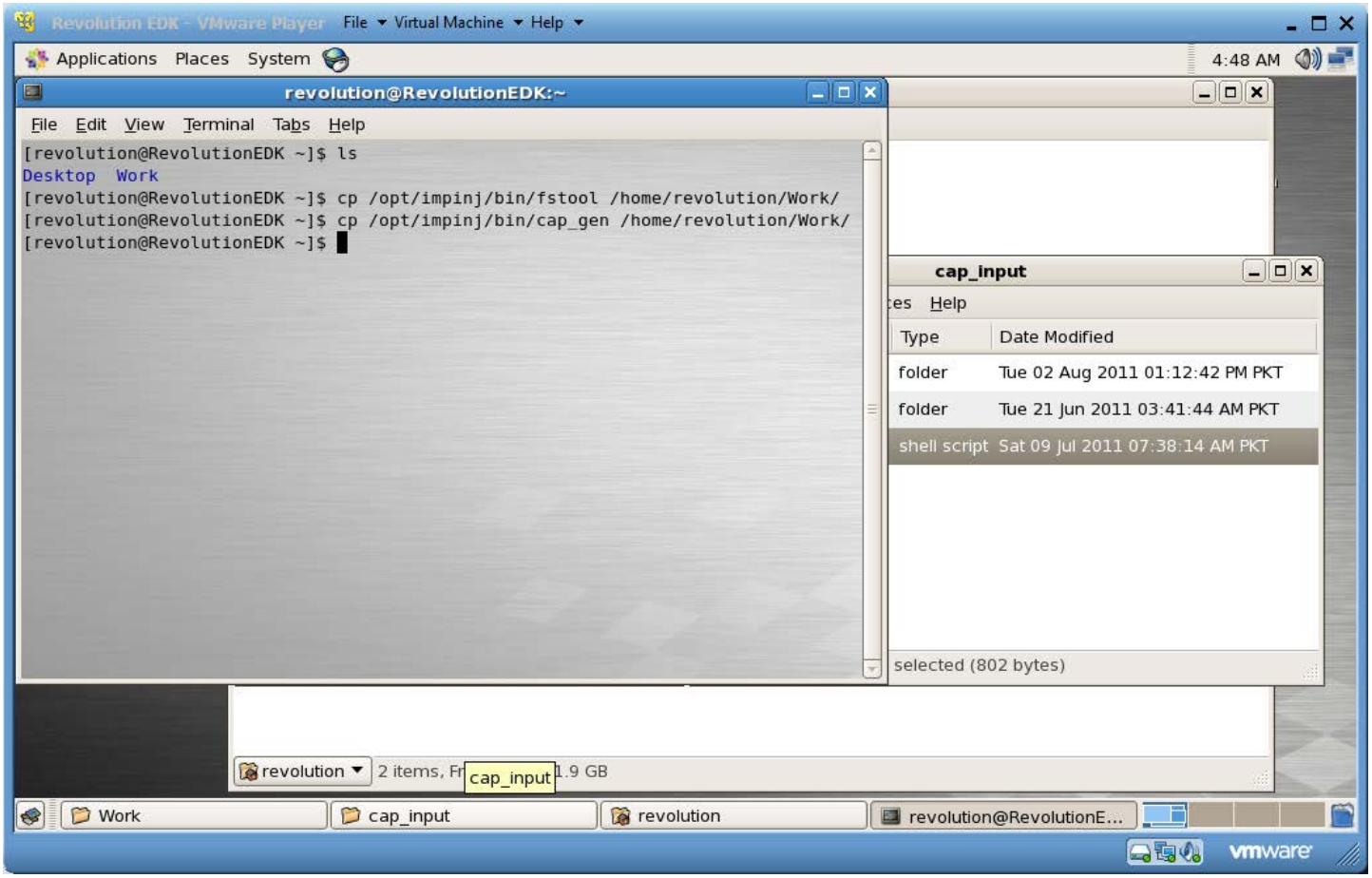
20. In the Work folder, create a file (right click -> Create Document -> Empty File). Rename it to cap_desc.in. Double click to edit the file and copy paste the following lines into it. Then save and exit the editor.

```
# CAP Description File
# This file contains the settings used by the CAP generation tool
# when a CAP partition upgrade file is produced.
[Description]

# Version is a 4 part number in decimal with each part limited to
# 0-255. It is the version of the CAP to be generated.
Version = 0.3.0.0

# Valid Reader Hardware is a 3 part number in decimal representing
# the reader model and major/minor revisions on which the CAP may be
# loaded. Each field may be replaced by a '*' to mean 'all'.
#
# Format = aaa-bbb-ccc
#   aaa - Model number
#   bbb - Major revision
#   ccc - Minor revision
Valid Reader Hardware = *-*-*

# File System Layout is an value used by the reader to determine how
# the CAP partition should be loaded to flash. Currently the only
# supported layout version is 10.
File System Layout = 10

# Input Directory is the top-level directory of the filesystem to
# create. The files under this directory will be available on the
# reader under /cust after the CAP is loaded.
Input Directory = /home/revolution/Work/cap_input
```

21. Now create another file and rename it to generate_upg_file. Double click to edit it and add the following line to it.

```
cap_gen -d cap_desc.in -o cap.upg
```

22. Now do the following on the command line.



Now your work directory is complete.

23. Now in the cap_input directory create a file named “start” and add the following script into it.

```
#!/bin/sh

# Configure NTP time sync
echo -e "server time.datasport.com\r" > /etc/ntp.conf

# Modify TCP parameters
echo 10 > /proc/sys/net/ipv4/tcp_keepalive_time
echo 5 > /proc/sys/net/ipv4/tcp_keepalive_intvl
echo 3 > /proc/sys/net/ipv4/tcp_keepalive_probes

# Preparartion for a dirty way to detect USB storage disconnections
#umount /mnt/usbfs/usbsda1
#mkdir /mnt/usbfs/usbsda1/UsbNotPresent
#mount /dev/sda1 /mnt/usbfs/usbsda1


# Wait until the network is both connected and we have a DNS server
# if we are
while true; do
  netconf | grep -q "connectionStatus='Connected'"
  if [ $? = 0 ]
  then
    netconf | grep -q "ipAddressMode='Static'"
    if [ $? = 0 ]
    then
      #don't wait around for a dns server.  It won't come
      break;
    else
      dnsconf | grep -q Server
      if [ $? = 0 ]
      then
        break;
      fi
    fi
  fi
  sleep 1
done


#if LLA is enabled, wait until mdnsd starts and reset syslog
netconf | grep -q "LLAStatus='enabled'"
if [ $? = 0 ]
then
  while [ ! -f "/var/run/mdnsd.pid" ]
  do
    sleep 1;
  done

  while [ ! -f "/var/run/syslogd.pid" ]
  do
    sleep 1;
  done
```

```
  kill -9 `cat /var/run/syslogd.pid`
  sleep 2
  /sbin/syslogd -m 0 &
  sleep 2
fi

# For firmware 4.6.2.240, start ntpd manually
/usr/sbin/ntpd

(( count = 1 ))

while true ; do
        if [ -f /cust/app/DataOne ] ; then
                if [ -x /cust/app/DataOne ] ; then
                        /cust/app/DataOne
                        /usr/bin/logger –p user.notice \
                        "Restarting DataOne, count $count."
                        echo -e "Restarting DataOne, count $count." >>
/mnt/usbfs/usbsda1/log/log.txt
                        (( count = count + 1 ))
                fi
                sleep 3
        else
                exit 0
        fi
done
```

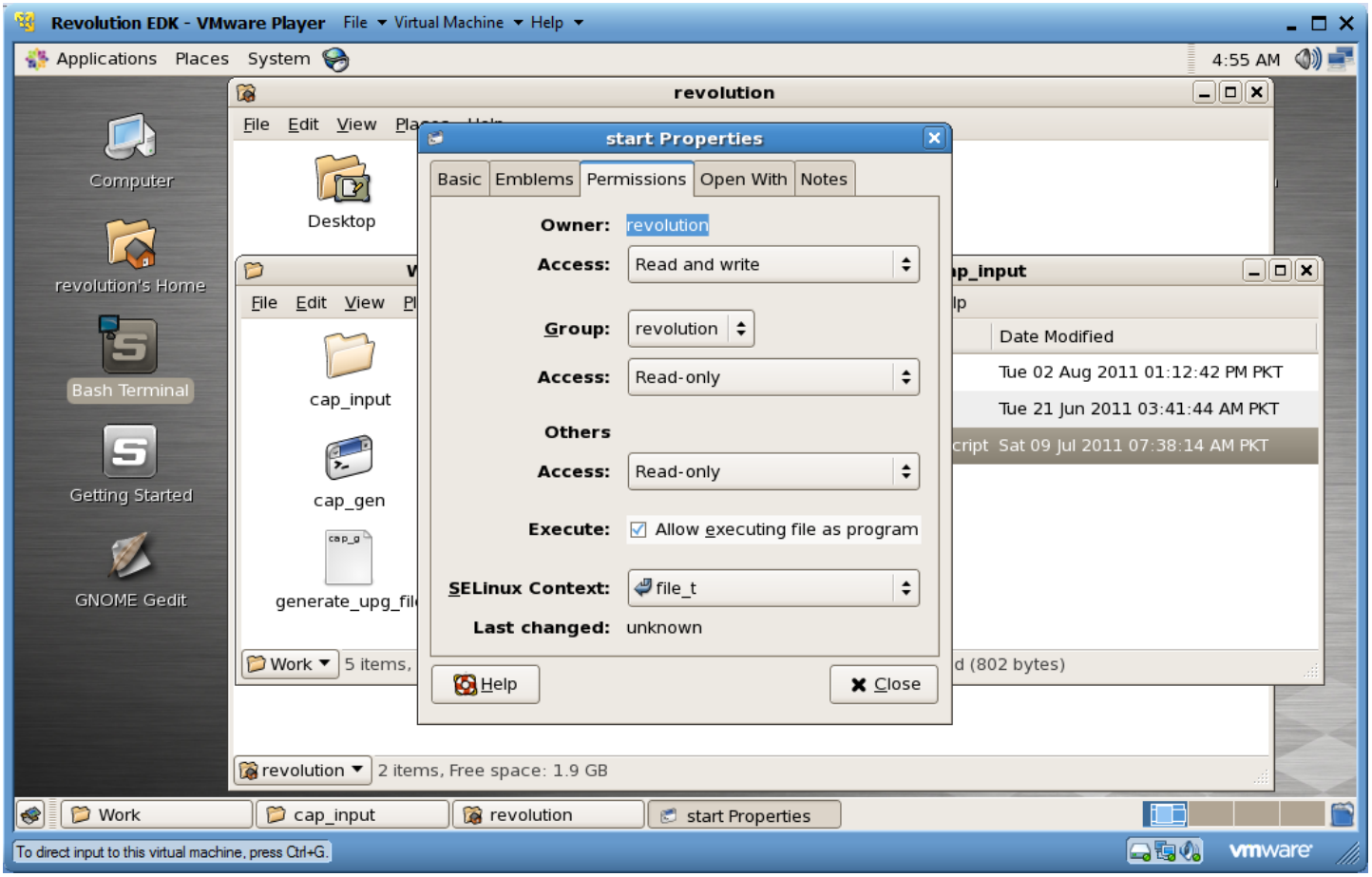
24. Now in the sys folder, create a file named “reader.conf” and addthe following lines into it.

[rshell]
password=developer

[SoftwareFeatures]
StartFTP=yes

25. The remaining app folder is where we will copy our DataOne application and its config files before generating the cap.upg file.

26. Do not forget to change permissions for the start and generate_upg_file files that we just created. Right Click -> Properties -> Permissions -> Allow executing file as a program (set it to checked)

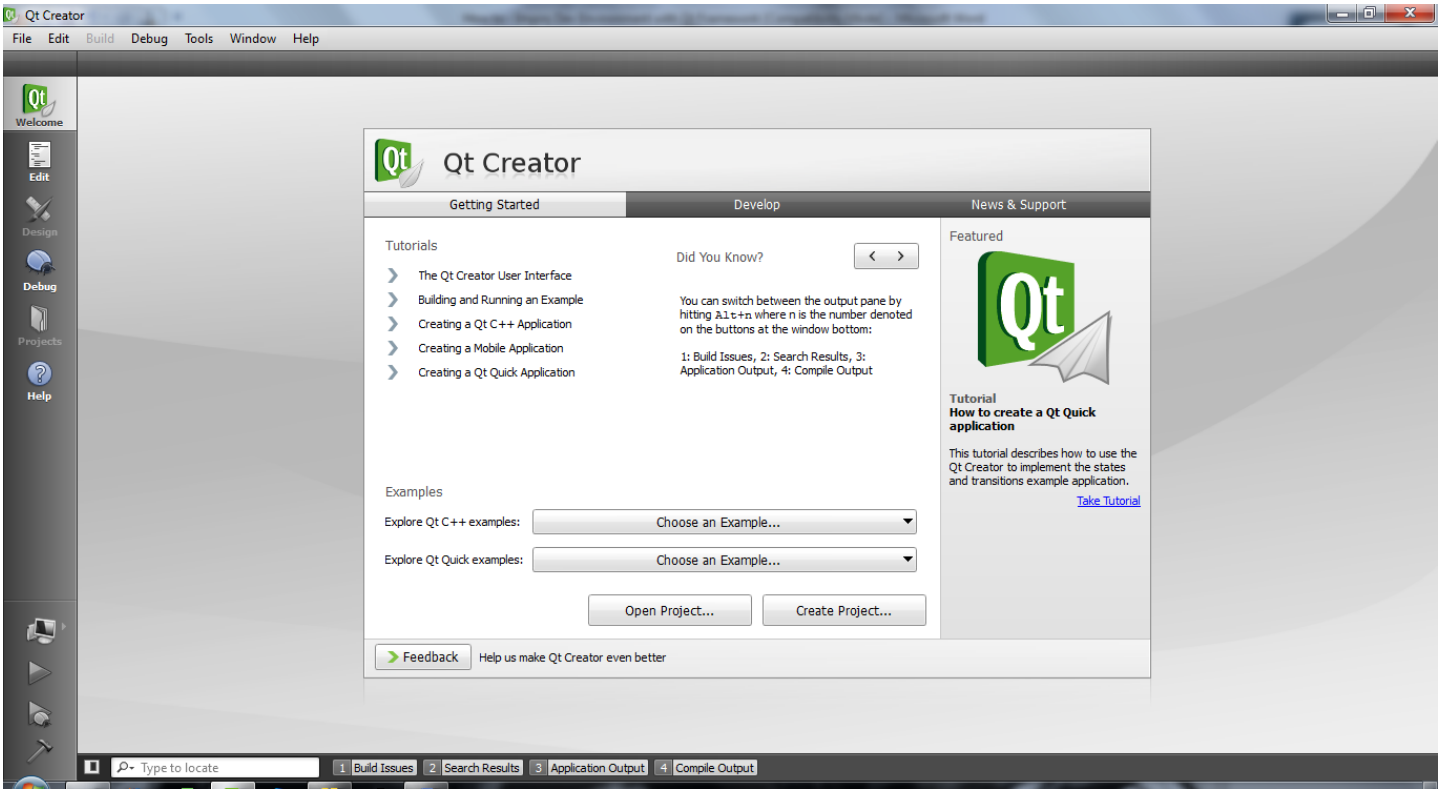


27. Now we need a god quality source code editor. We could use the Emacs, VI or Gedit inside the virtual machine, but they are so primitive and only provide basic functionality. Therefore I decided to use Qt's own native editor called Qt Creator. I tried to recompile it from source for our virtual machine's environment, but I found that it is not compatible with certain OS libraries of this version of CentOS. So, I decided to use the editor on windows and share the source with the virtual machine through shared_folder.

28. Extract the source code package to the disk, and if you are using the same folder architecture as mine, you should see the folders EDK and Project in the Shared_Folder.

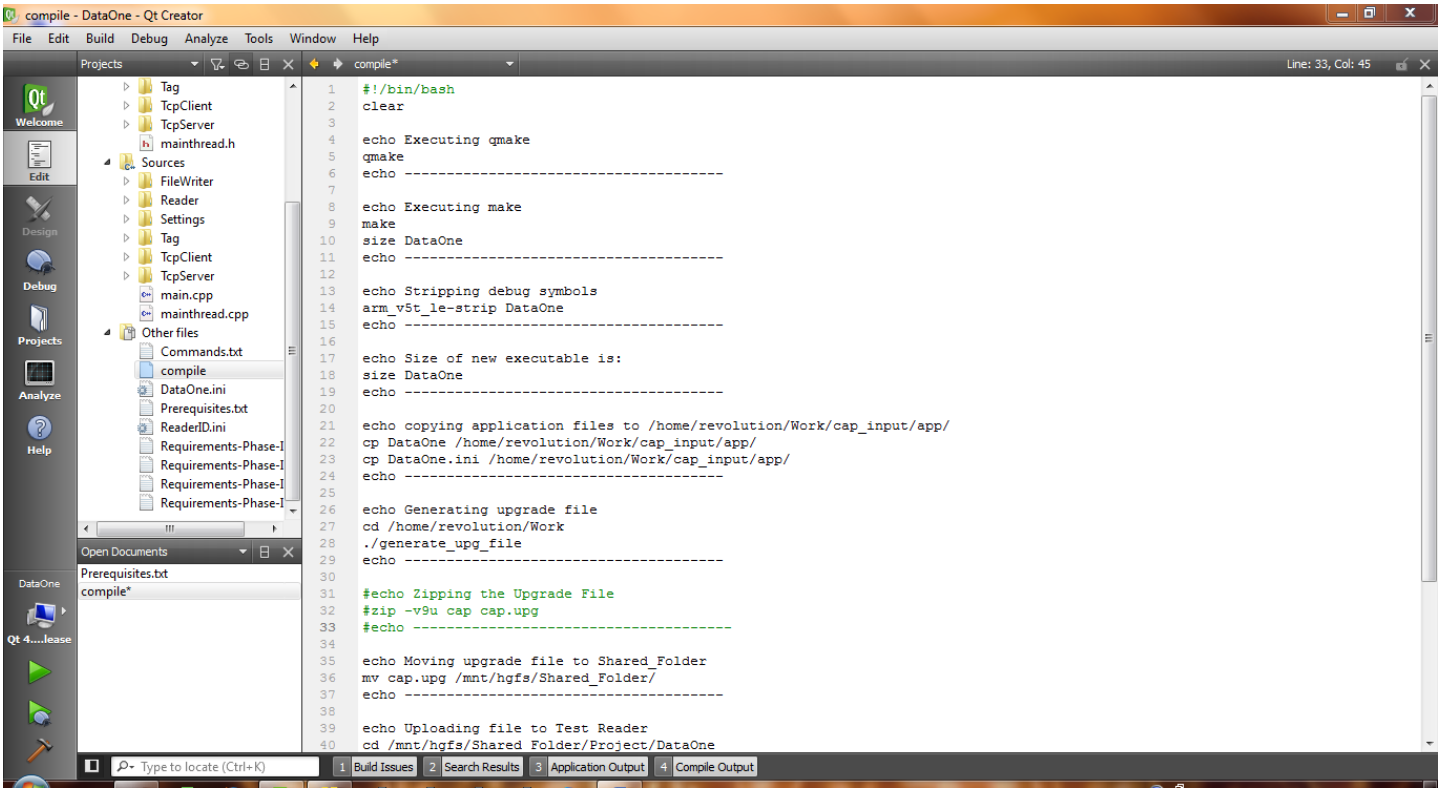
29. Download and install Qt SDK from http://qt.nokia.com. Complete SDK is of considerable size, but installing SDK is better than manually linking the libraries and sources from different locations.

30. After the installation is complete, run the QtCreator from Start Menu -> All Programs -> Qt SDK. Qt Creator looks like this.

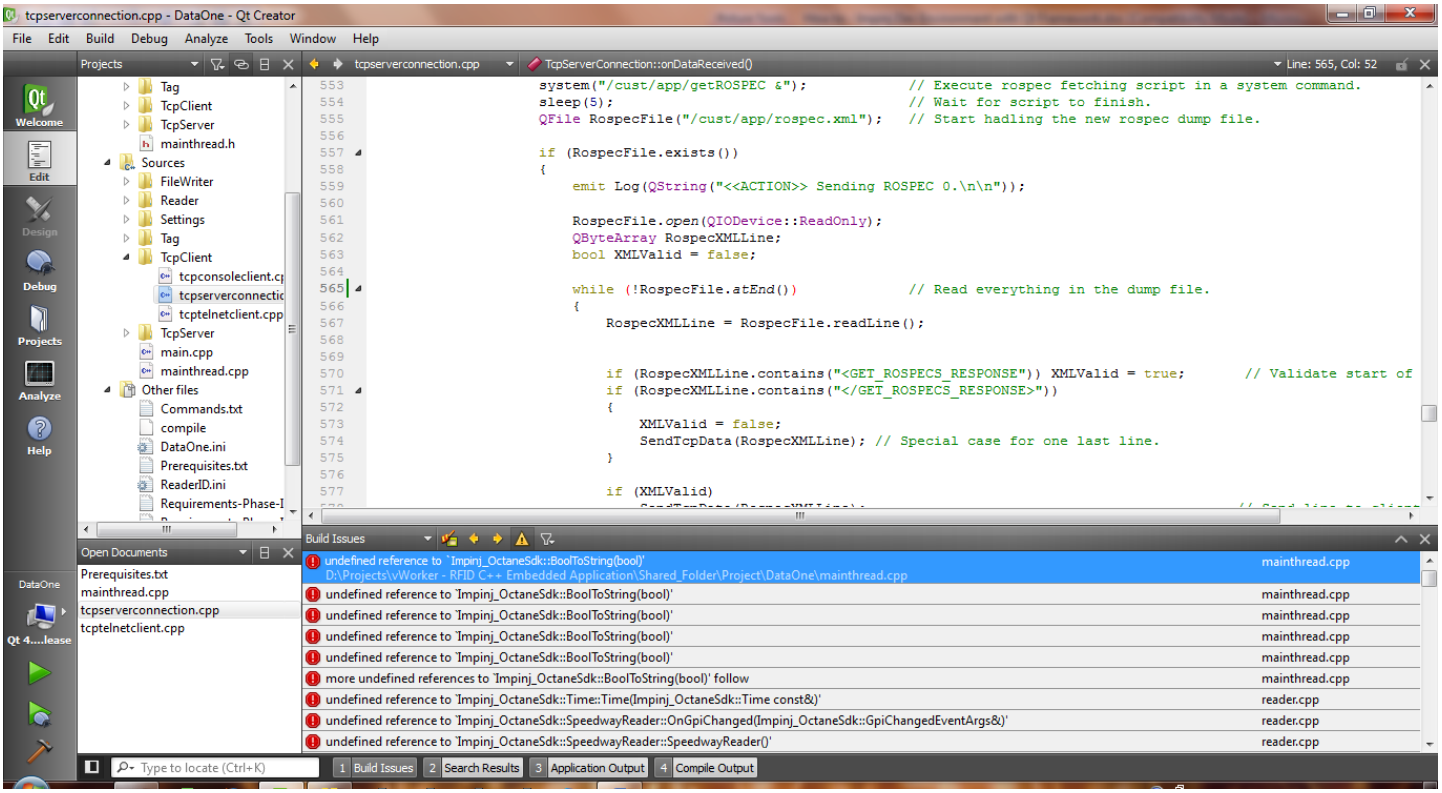


31. Click on the Open Project button and navigate to ..\Shared_Folder\Project\Speedway_Application_Qt\Speedway for opening our application project. If at first opening, the Qt creator says that it has found user configuration from some other machine, then select NO. It will create new environment settings according to your machine. But actually they have no effect on our project as we are not using those settings.

32. Now you can edit the source code and associated files. I have created a script file for automating the compile and generating the upgrade file as follows. This script will be executed from terminal inside EDK virtual machine.



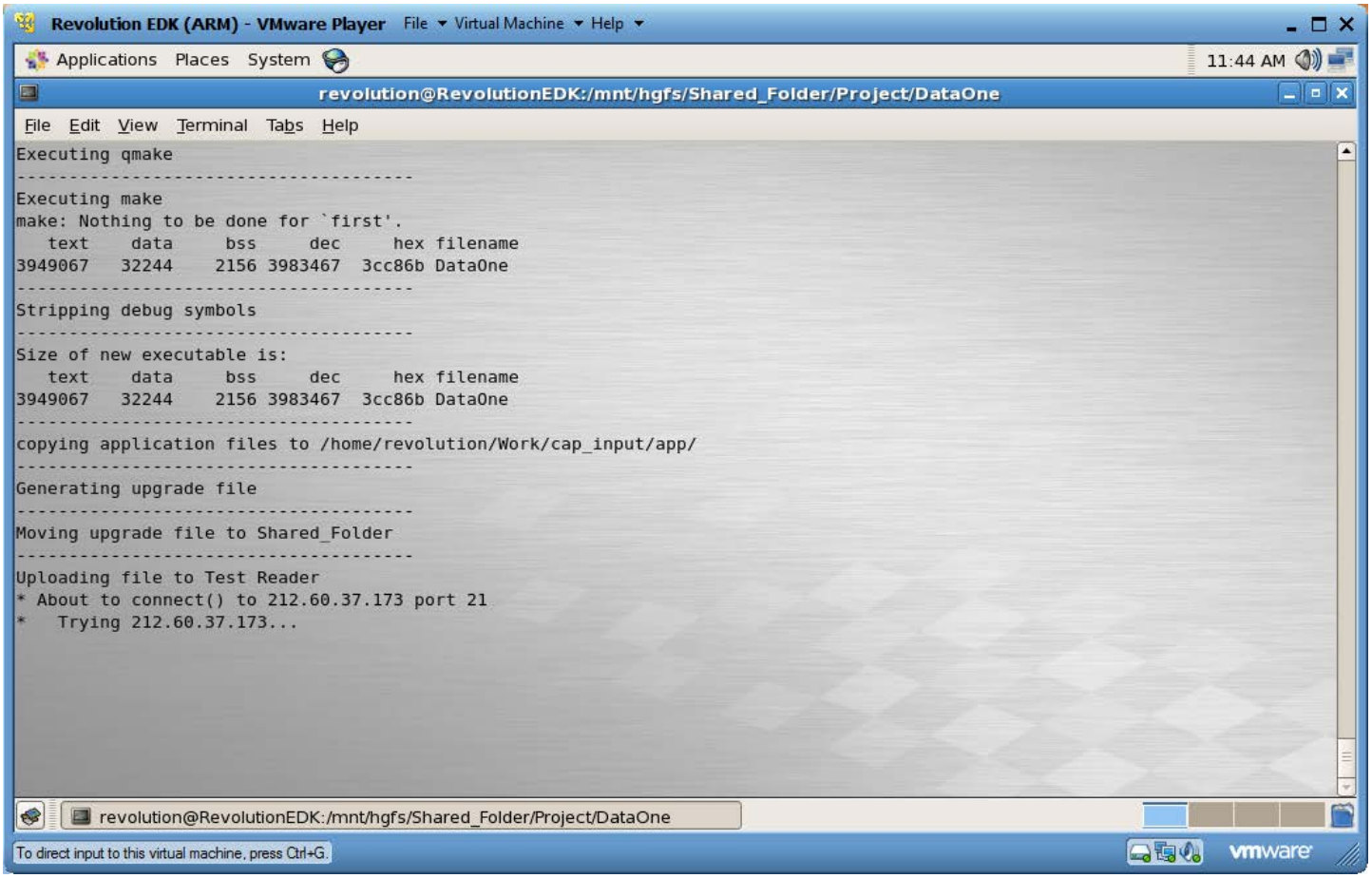
33. After editing the source code in Qt Creator, you can hit the hammer button on bottom left and it will try to build the project. It will indicate all build issues in the following way.



The red icons show that there were errors in the build process. But, if the output looks like in the above picture, it is OK. Because we only wanted to check the syntax errors here. We will actually compile the project inside the virtual machine, because, relevant impinj and Octane SDK libraries are inside the virtual machine and not on windows.

34. Once you are satisfied with your code, save the project (File -> Save All). Then go inside the virtual machine and execute compile script from terminal.

[revolution@RevolutionEDK DataOne]$./Compile



Voila….. you have successfully compiled the application, generated the upgrade file and copied it to the Shared_Folder. This script is also configured to upload the new executable to reader through FTP. Also, you can upgrade your reader from its HTTP interface.

35. Please read the file Prerequisites.txt in the main folder of the project. This will help you understand what you need to know to start with this project.

Good luck…
The End.