Chapter 1

Getting started

This document explains how to install the VirtualBox software (Section 1.1), how to create a virtual machine and how to start it (Section 1.2), how to start the ENRAM data processing (Chapter 2), and finally it provides some background about how the virtual machine is organized (Chapter 3).

Since you are reading this file, it is assumed that you have plugged in the USB drive that contains all the data and software pertaining to the ENRAM project.

1.1 Installing VirtualBox

Use Windows Explorer to browse the USB disk contents. The USB disk should show up as ENRAMUSBDISK in the left hand pane (see figure 1.1). The disk should at least contain the following: (1) a folder 'ENRAMVM' (which contains the virtual hard drive file that we will use in the next section), a file 'readme.pdf' (which is the document you have before you), and a VirtualBox installer for Windows.

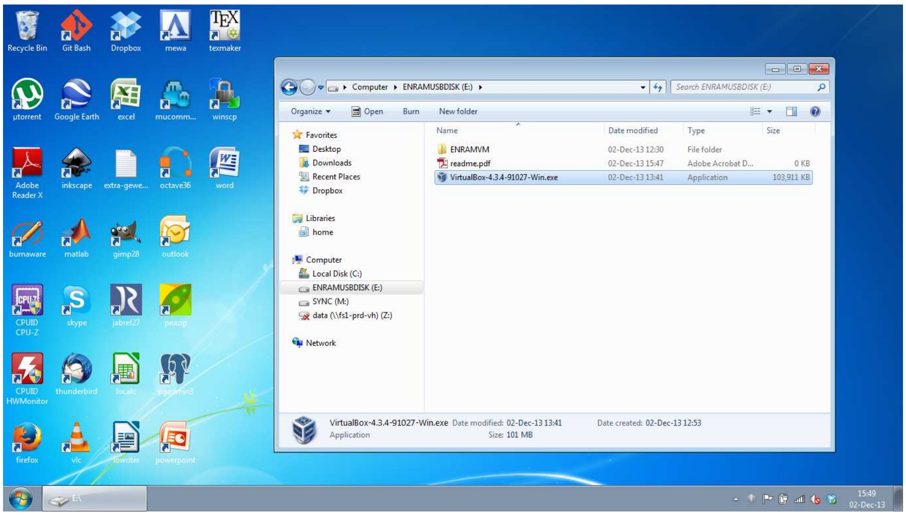

Figure 1.1

Double-click on the VirtualBox installer file ('VirtualBox-4.3.4-91027-Win.exe') to start the setup wizard. A menu will show up (Figure 1.2). Click on the button labeled 'Run'.

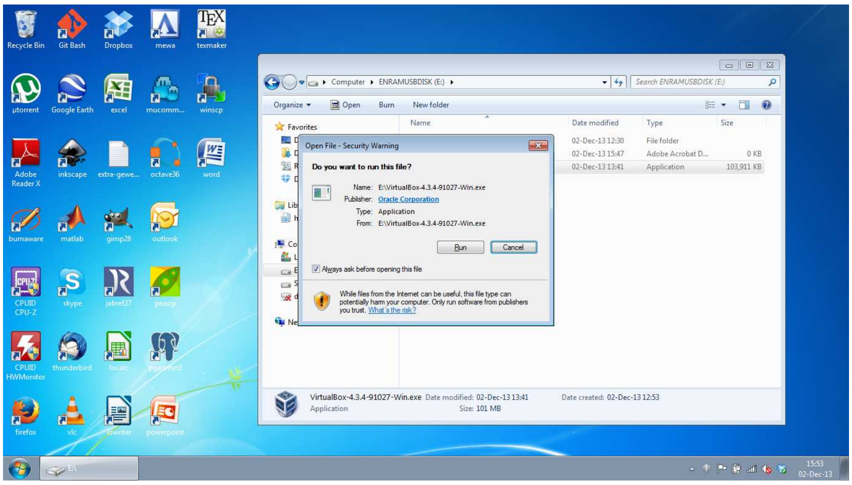

Figure 1.2

The setup wizard program should now start. On the first page of the setup wizard (Figure 1.3), click the button labeled 'Next'.

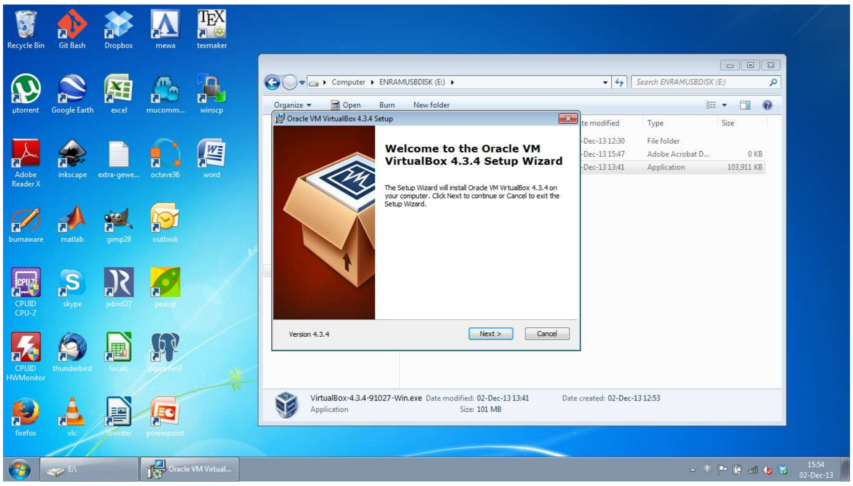

Figure 1.3

On the next page of the setup wizard (Figure 1.4), click the button labeled 'Next'.

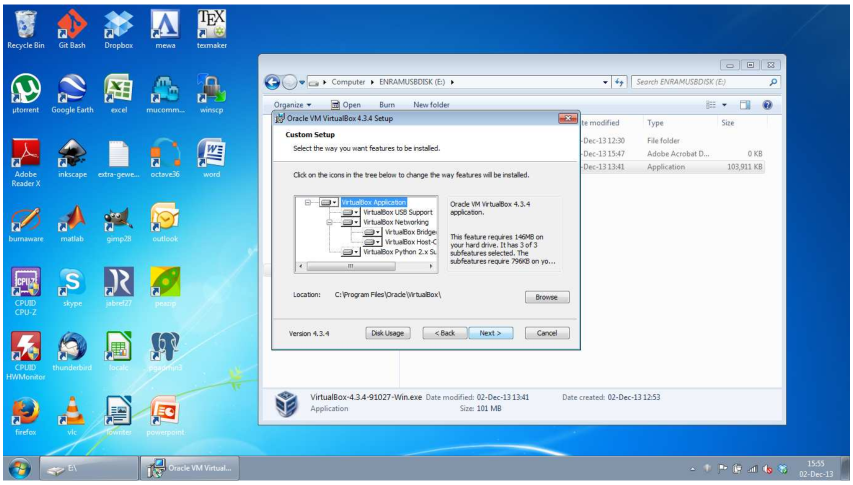

Figure 1.4

On the next page of the setup wizard (Figure 1.5), check the items as you like. Then click the button labeled ‘Next’.

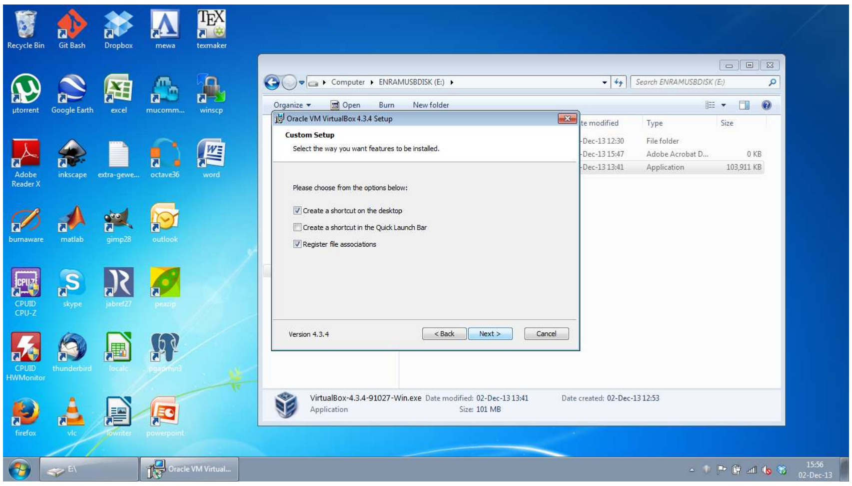

Figure 1.5

Make sure you are not currently doing something that requires network access (like downloading a big file). Then, on the next page of the setup wizard (Figure 1.6), click the button labeled ‘Yes’.

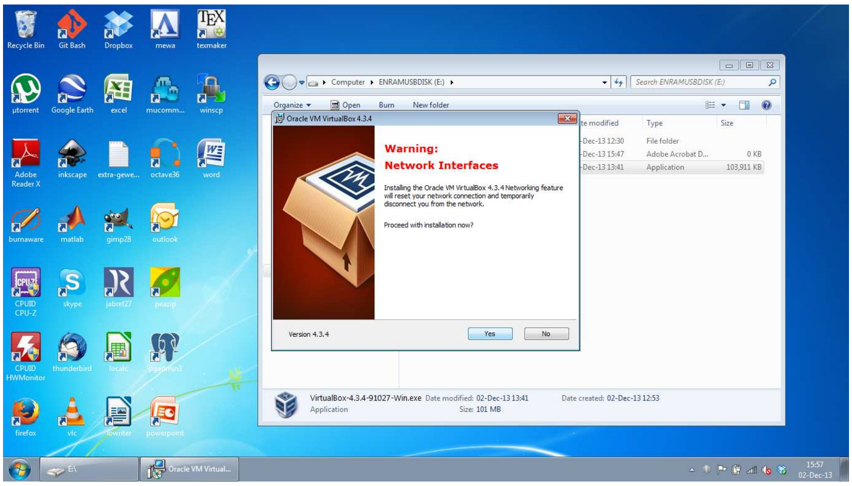

Figure 1.6

On the next page of the setup wizard, click the button labeled ‘Install’.

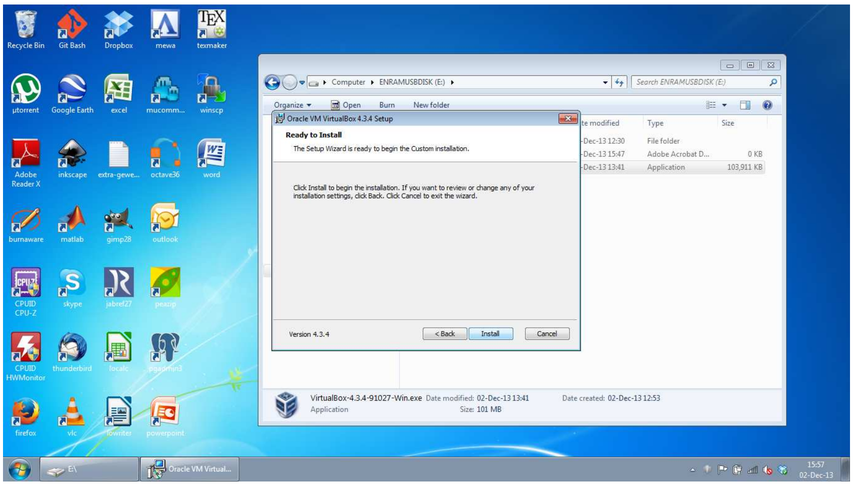

Figure 1.7

Allright! Looks like you just installed VirtualBox. Leave the checkbox checked (Figure 1.8) and click on the button labeled ‘Finish’.

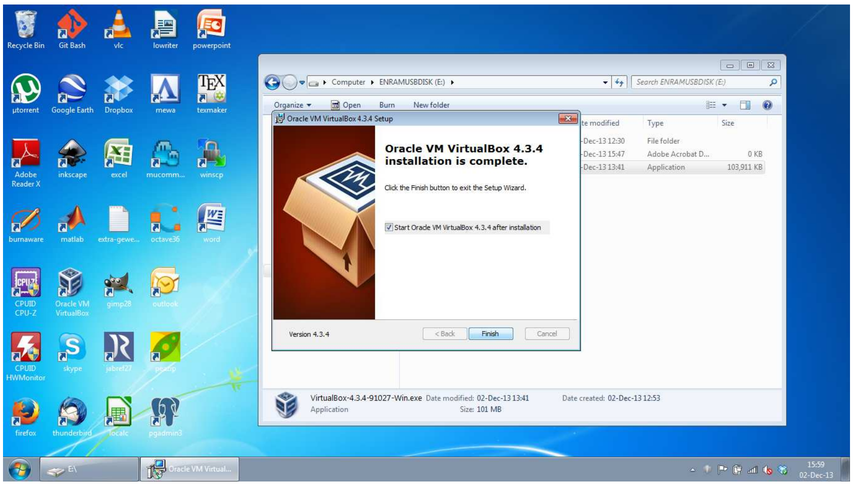

Figure 1.8

1.2 Creating a virtual machine and running it

Before we can access all the goodies that are on the virtual disk, we need to create a so-called *virtual machine*. Start the VirtualBox program if it has not already started. Since this probably is the first time you started VirtualBox, the program will show a welcome message (Figure 1.9). Click on light blue icon in the top left corner labeled 'New'.

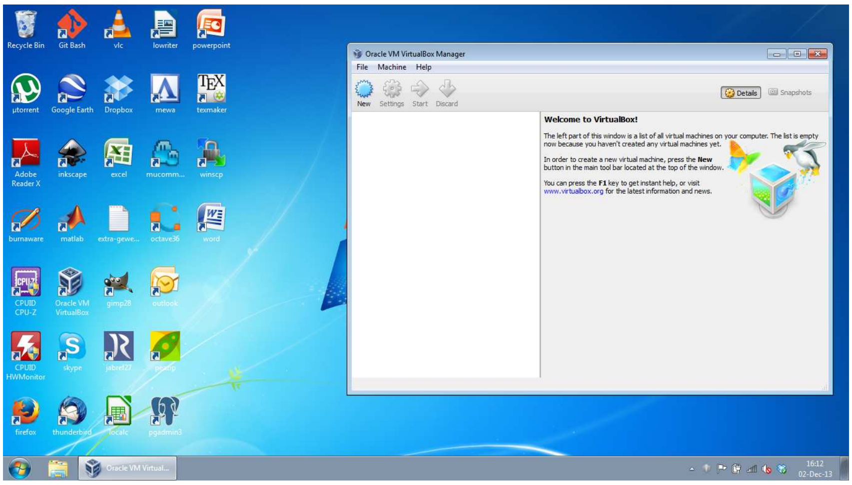

Figure 1.9

In this window (Figure 1.10), specify 'ENRAMVM' as the name of the virtual machine.

Use the drop-down menus to choose the type of operating system ('Linux') and the version ('Ubuntu 64-bit').

Click the button labeled 'Next'.

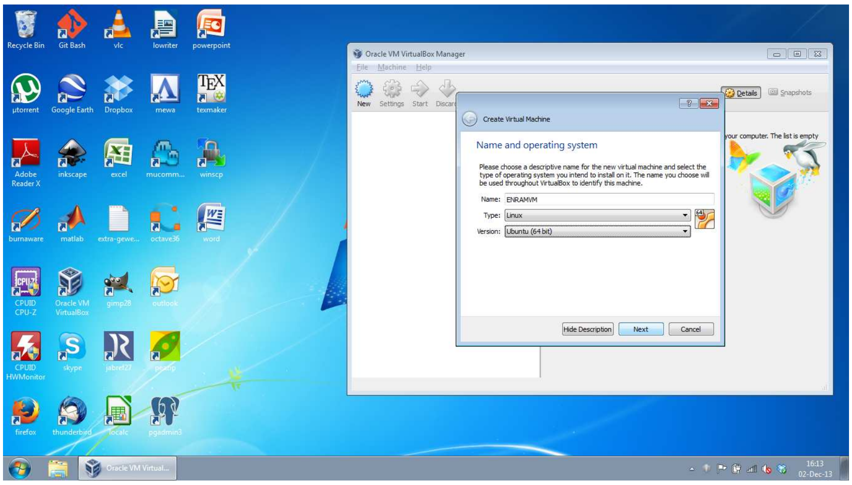

Figure 1.10

In this window, you need to specify the amount of virtual memory that your virtual machine will have. Set it to 4096 MB (Figure 1.11).

Click the button labeled 'Next'.

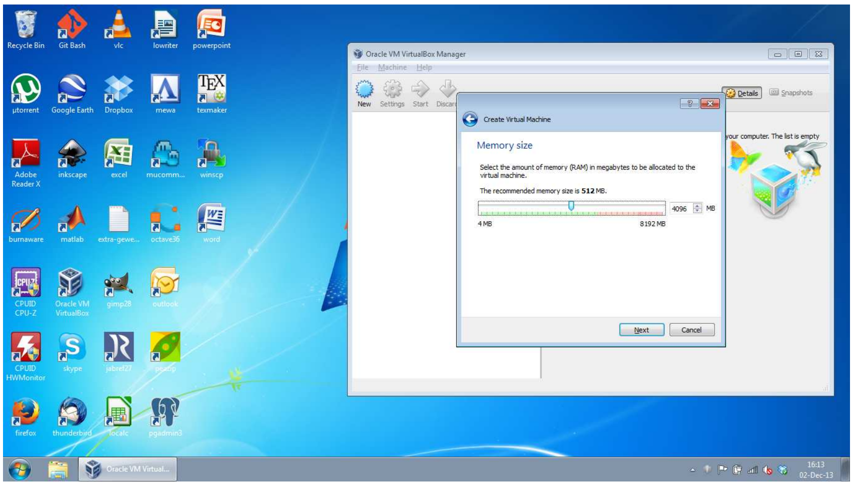

Figure 1.11

In this window, make sure to choose the last option 'Use an existing virtual hard drive file'. Use the little folder icon to the right of the drop-down list to select the virtual hard drive file 'ENRAMVM Clone-disk1.vdi' that is located in folder 'ENRAMVM' on ENRAMUSBDISK.

Then, click the button labeled 'Create'.

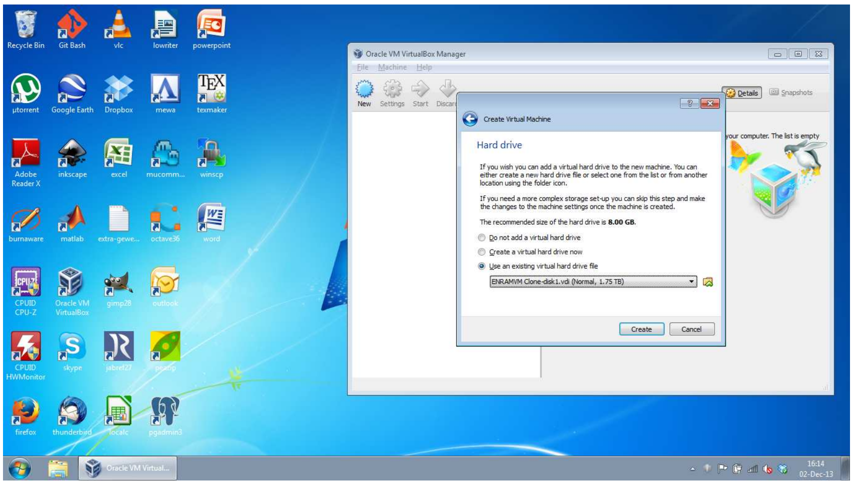

Figure 1.12

Sweet! We have a virtual machine (‘ENRAMVM’; Figure 1.13). Boot up the virtual machine by clicking the icon with the green arrow labeled ‘Start’.

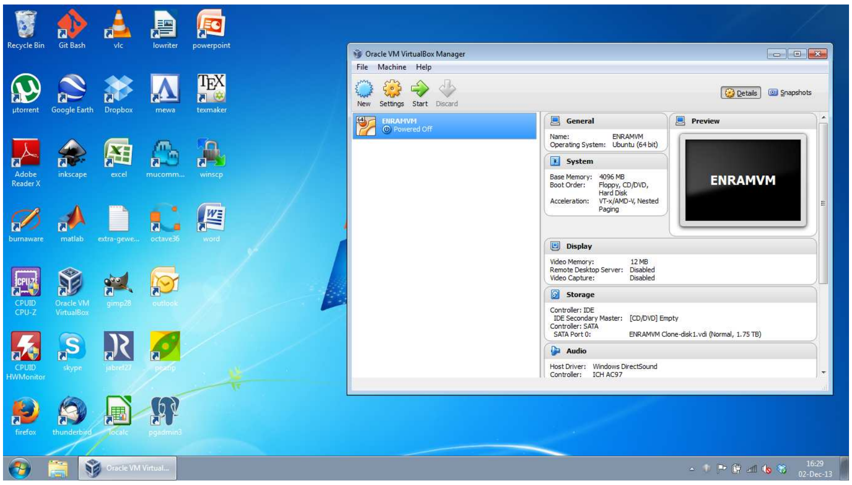

Figure 1.13

A new Window will pop up (Figure 1.14), which initially is just black, but after a couple of seconds, stuff will appear. There will likely also be two warning messages, but you can ignore these by clicking on the little blue icon in the top right corner of the ENRAMVM window.

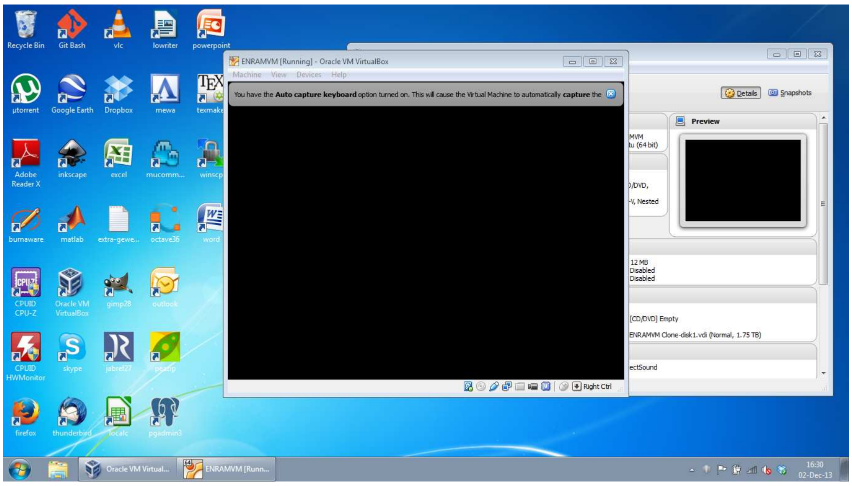

Figure 1.14

After a couple more seconds, your newly created ENRAMVM machine will have finished booting and will be ready for you to use. It will show you a blue desktop with a few icons on it (Figure 1.15).

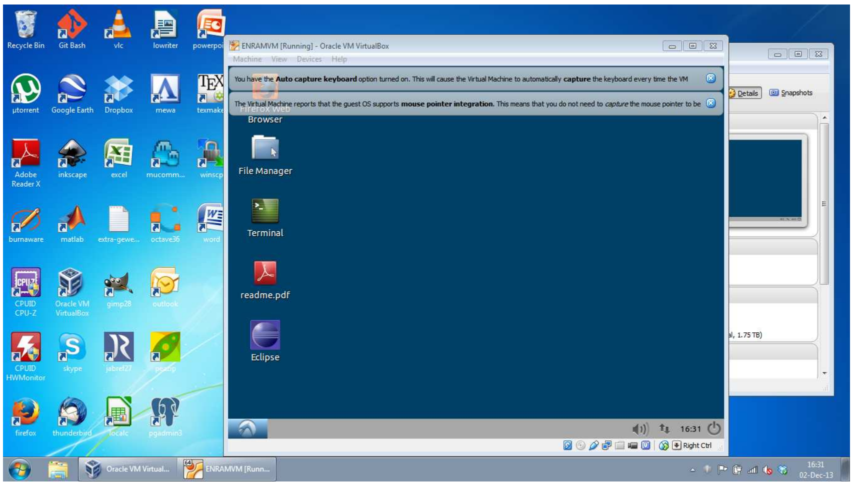

Figure 1.15

Before we do anything else, let's first maximize the virtual screen. You can do this by clicking on the menu item 'View' and then selecting 'Switch to Fullscreen' (Figure 1.16).

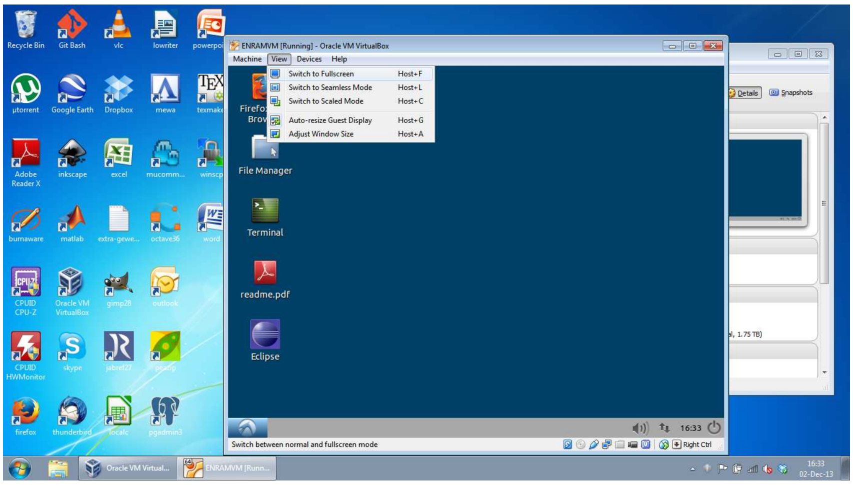

Figure 1.16

Note that you can toggle between fullscreen and windowed mode by simultaneously pressing the letter F button and the Ctrl button on the right hand side of your keyboard (Figure 1.17).

Click the ‘Switch’ button to switch to Fullscreen mode.

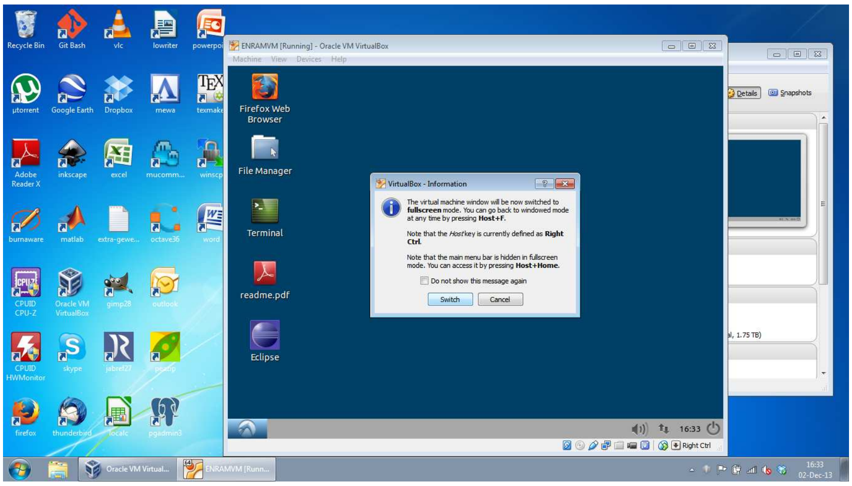

Figure 1.17

You should now see the virtual machine’s desktop fullscreen (Figure 1.18).

Figure 1.18

The next chapter explains how to use the ENRAM software and data, but at some point you'll want to power down the virtual machine, so let's look at that first.

Click on the blue icon in the taskbar in the lower left corner of the screen, and choose 'Logout' (Figure 1.19).

Figure 1.19

In this menu, choose 'Shutdown' to power down the virtual machine (Figure 1.20). After a few seconds, you will be back at you Windows desktop (Figure 1.21).

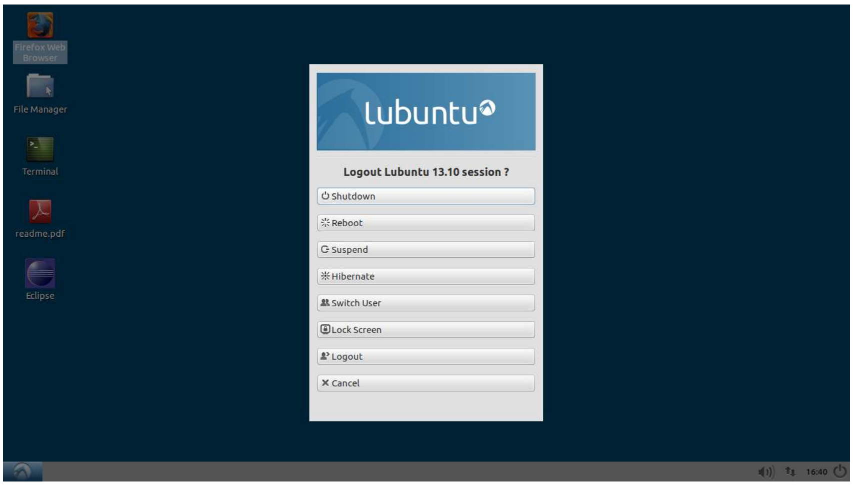

Figure 1.20

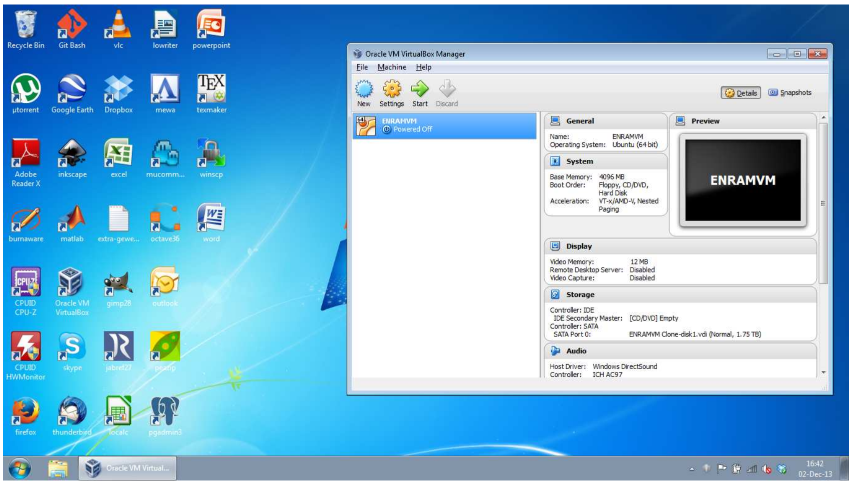

Figure 1.21

Chapter 2

Using the ENRAM software

Make sure you have your virtual machine running (as in Figure 1.18).

Start the File Manager by clicking on the desktop icon labeled ‘File Manager’. In the file manager window, double-click on the directory called ‘enram’ to inspect its contents.

Start a terminal by double-clicking the desktop icon labeled ‘Terminal’. This should bring up a terminal program (Figure 2.1). In the terminal window, use the `cd` command to change directory into the ‘enram’ directory.

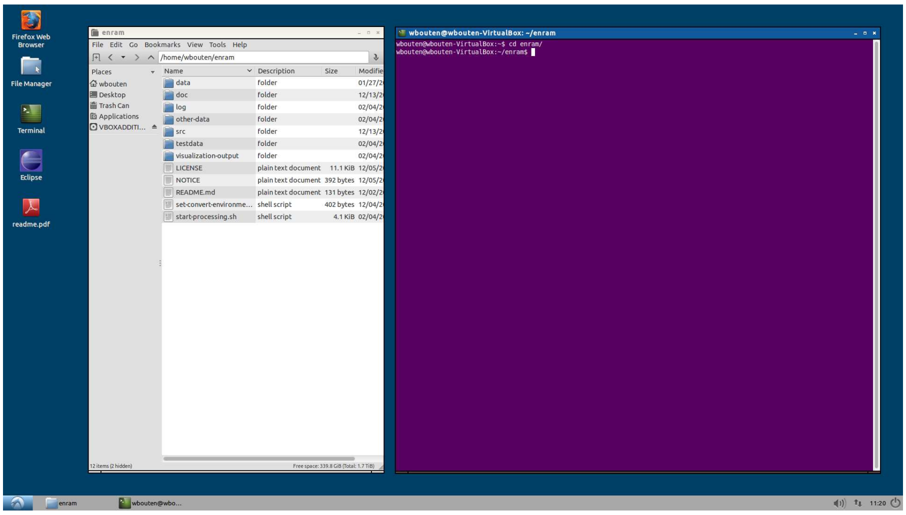

Figure 2.1

You can now start the ENRAM workflow as follows. Type:
`.  ./start-processing.sh`

(But make sure to type it exactly as it is displayed here, including the leading dot).

The terminal will then first ask you whether you want an interactive session or you want to start processing in batch mode; then it will ask you to select either the test data set or the full data set (Figure 2.2).

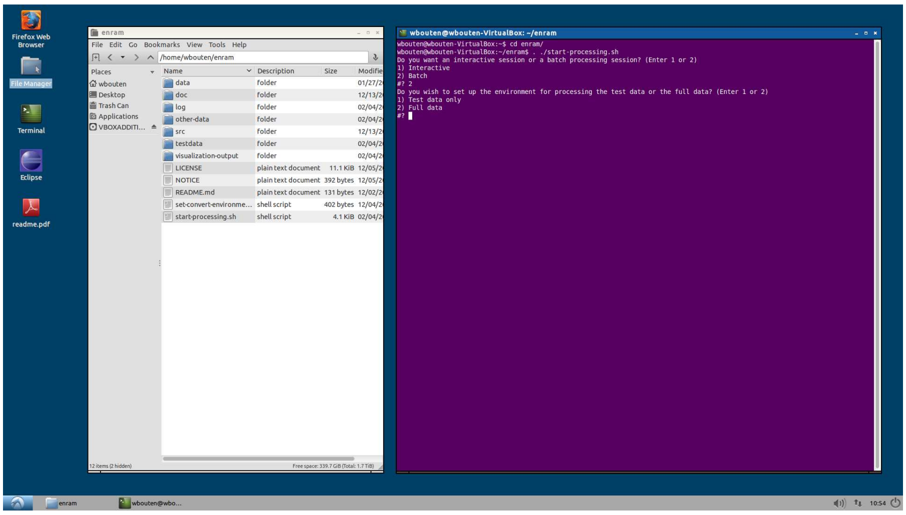

Figure 2.2

Answer '1' or '2' and press Enter at each question. The terminal will now set up the necessary environment variables.

If you chose to process the data in batch mode, you can use the Firefox web browser to view the batch program's feedback (Figure 2.3) while the terminal program is running. Just navigate to 'file:///home/wbouten/enram/log/stderr.txt' and 'file:///home/wbouten/enram/log/stdout.txt'. Note that these locations have been added to the bookmarks, so if you just type 'stderr' or 'stdout' in Firefox's URL bar, it should suggest the corresponding files automatically.

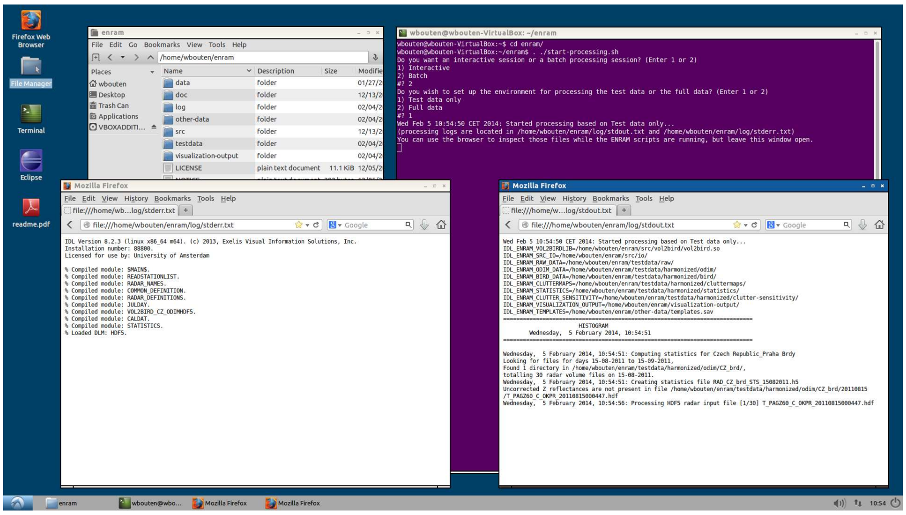

Figure 2.3

Chapter 3

Organization of the software

3.1 Original problem

Software has been developed by Adriaan Dokter and Martin de Graaf during previous projects. Dokter's software is written in the C programming language, whereas most of De Graaf's work is in IDL. The two components work together to read radar signals from ODIM-HDF5 formatted files containing radar sensor data, most importantly reflectivities and radial velocities. The software can use these data to visualize the intensity of bird migration around a radar station, either as PPI's[1] or as altitude profiles. The software further includes algorithms to calculate cluttermaps for a given radar station.

Currently, data from the following stations are included:

Table 3.1: List of radar stations

station code	**full name**
CZBRD	Brdy-Praha, Czech Republic
CZSKA	Skalky, Czech Republic
FIANJ	Anjalankoski, Finland
FIIKA	Ikaalinen, Finland
FIKOR	Korpo, Finland
FIKUO	Kuopio, Finland

[1] http://en.wikipedia.org/wiki/Plan_Position_Indicator

Table 3.1: List of radar stations

station code	full name
FILUO	Luosto, Finland
FIUTA	Utajärvi, Finland
FIVAN	Vantaa, Finland
FIVIM	Vimpeli, Finland
FRABB	Abbeville, France
FRALE	Aléria, France
FRAVE	LAvesnois, France
FRBLA	Blaisy-Haut, France
FRBOL	Bollène, France
FRBOR	Bordeaux, France
FRBOU	Bourges, France
FRCAE	Falaise, France
FRCHE	Cherves, France
FRCOL	Collobrières, France
FRGRE	Grèzes, France
FRMCL	Montclar, France
FRMOM	Momuy, France
FRMTC	Montancy, France
FRNAN	Nancy, France
FRNIM	Nîmes, France
FRTRO	Arcis-sur-Aube, France
HRBIL	Bilogora, Croatia
IEDUB	Dublin, Ireland
IESHA	Shannon, Ireland
NLDBL	De Bilt, The Netherlands
NLDHL	Den Helder, The Netherlands
NOAND	Andoya, Norway
NOBML	Boemlo, Norway
NOHAS	Hasvik, Norway
NOHGB	Haegebostad, Norway

Table 3.1: List of radar stations

station code	**full name**
NOHUR	Hurum, Norway
NORSA	Rissa, Norway
NORST	Rost, Norway
NOSTA	Stad, Norway
PLBRZ	Brzuchania, Poland
PLGDA	Gdansk, Poland
PLLEG	Legionowo, Poland
PLPAS	Pastewnik, Poland
PLPOZ	Poznań, Poland
PLRAM	Ramża, Poland
PLRZE	Rzeszów, Poland
PLSWI	Świdwin, Poland
SEANG	Ängelholm, Sweden
SEARL	Stockholm-Arlanda, Sweden
SEASE	Ase, Sweden
SEHUD	Hudiksvall, Sweden
SEKIR	Kiruna, Sweden
SEKKR	Karlskrona, Sweden
SELEK	Leksand, Sweden
SELUL	Luleå, Sweden
SEOSU	Östersund, Sweden
SEOVI	Örnsköldsvik, Sweden
SEVAR	Vara, Sweden
SEVIL	Vilebo, Sweden
SILIS	Lisca, Slovenia
SKKOH	Kojsovska hola, Slovakia
SKMAJ	Maly Javornik, Slovakia

3.2 Directory structure

The following table lists a selection of the most important files and directories as used in the 'ENRAMVM' Virtual Machine.

Location (relative to `/home/wbouten/enram/`)	**Description**
`data/`	contains the radar data and derived products
`data/harmonized/bird/`	...
`data/harmonized/cluttermaps/`	...
`data/harmonized/clutter-sensitivity/`	...
`data/harmonized/odim/`	radar data for all stations for the period 15-Aug-2011 to 16-Sep-2011 in ODIM-HDF5 format.
`data/harmonized/statistics/`	contains output from `src/process/cluttermap.pro`
`data/raw/`	radar data for most stations for the period 15-Aug-2011 to 16-Sep-2011 as provided by individual weather services
`data/stations.txt`	configuration file containing a list of stations that are part of the data set to be analyzed
`doc/`	some relevant literature
`doc/readme/`	the latex files needed for compiling this document
`log/stderr.txt`	standard error log
`log/stderr.txt`	standard output log
`other-data/`	miscellaneous data (currently not used)
`src/`	source code main directory
`src/io`	country-specific i/o routines written in IDL
`src/lib`	programs and functions from external libraries, written in IDL
`src/process/bird_call.pro`	IDL program that ...
`src/process/cluttermap.pro`[1]	IDL program that ... Output is written to `data/harmonized/cluttermaps/`.
`src/process/clutter_sensitivity.pro`[1]	IDL program that ...

Location (relative to /home/wbouten/enram/)	Description
src/process/flysafe2.pro[1]	IDL program that ...
src/process/statistics.pro	IDL program that calculates statistics based on which ... Output is written to data/harmonized/statistics/
src/raw-to-odim-conversion-tools	(incomplete) collection of conversion scripts, mostly written in Bash and C, with which the initial conversion was performed to convert the data from data/raw to the ODIM-HDF5 format in data/harmonized/odim.
src/visualization/make_bird_density_profiles.pro[1]	IDL program that creates figures of bird density as a function of time and altitude.
src/visualization/make_colored_donut_maps.pro[1]	IDL program that creates a figure in which the bird density around each radar is plotted as a colored donut.
src/visualization/make_ppi_time_space_bird_rain.pro[1]	IDL program that creates a map of distributed PPI's of rain and of bird density, at given time intervals.
src/visualization/multi_ppi.pro[1]	IDL program that visualizes multiple PPI's.
src/visualization	figure-generating programs
src/vol2bird	stores Adriaan Dokter's C language program
testdata/	subset of data/, used in integration testing
visualization-output/	directory where all visualization programs place their output figure files

[1]This program is part of the main data processing pipeline, as defined in the Bash script start-processing.sh.