

Installation and Configuration of OCaml for Windows 10

By: Brandon Kim, Jonathan Morley,
Francis Oyebanjo, and BT Rappaport

Table of Contents

<u>Introduction</u>	2
<u>Installing OPAM and OCaml</u>	3
<u>Checkpoint</u>	7
<u>Installing Extra Packages (OCamlfind, OUnit, rlwrap)</u>	8
<u>Ensuring Correct Installation and Configuration</u>	11
<u>Troubleshooting Errors</u>	12
<u>Glossary</u>	16

Introduction

This manual is for OCaml beginners who would like to use OCaml on a Windows 10 computer. OCaml is not very compatible for installation on a Windows computer but is made simple with Bash on Ubuntu on Windows, a brand new Windows 10 feature that enables direct use of Linux command-line tools. However, the directions available are usually not Windows 10 specific and assume the audience is already well versed in Linux tools. This manual provides beginners to OCaml and Linux with a clear and orderly set of instructions for installing and configuring OCaml on Windows 10.

OCaml is a heavy-duty programming language that supports functional, imperative and object-oriented styles. OCaml's safety, advanced type system, and automatic memory management are a few features that make it a highly effective programming tool. Even though OCaml is a powerful language, it is also simple and easy to use, making it an ideal language to expose to students in undergraduate computer science courses. In this manual we will install OCaml via OPAM, a package manager (collection of tools that automates the process of installing, upgrading, configuring, and removing programs from a computer) for OCaml.

Requirements/ Materials

- Computer running Windows 10 (64-bit) with build 14393 or later
- Administrative access of the computer
- Bash on Ubuntu on Windows installed and setup
 - https://msdn.microsoft.com/en-us/commandline/wsl/install_guide
- Internet access with decent download speed

Important Notes

- Copy and paste (to paste simply right click in bash where you want to paste) the commands from the manual into Bash, especially if they are long. Bash is sensitive and anything but the exact command will be an error. If you enter something incorrectly, press "Ctrl-c" to kill the process and try again.
- Make sure that what appears in each of the images is the same as on your screen before continuing. If any errors occur, refer to the troubleshooting section of the instructions.
- If you are prompted for your password at any point during installation, type your password and hit enter. You will not see anything appear as you type. This is a Bash feature to ensure possibly onlookers cannot even determine your password length.
- It is not essential that you understand the commands. You need only to copy-paste them. However, if you want a deeper understanding there is a glossary at the end.

Installing OPAM and OCaml

At this point it is assumed you already have Bash on Ubuntu on Windows installed. If you do not, go to the requirements section for a link to Bash on Ubuntu on Windows installation instructions.

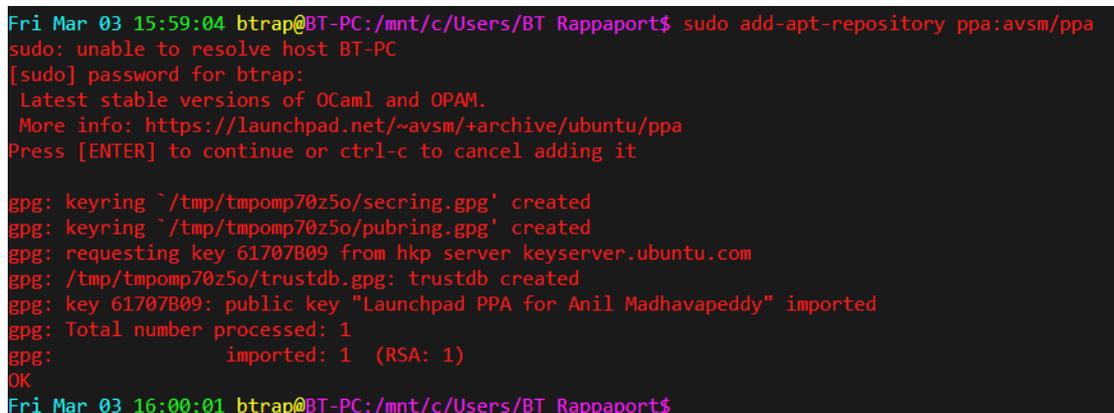
In order to install Ocaml and its package manager OPAM follow these instructions:

1. Open Bash on Ubuntu on Windows: **Go to the start menu. Type “bash”. Click “Bash on Ubuntu on Windows”**. Wait for your prompt to appear. The prompt is the line that ends with a \$.
2. Enter the following command (type or copy/paste it in and then press enter):
sudo add-apt-repository ppa:avsm/ppa

This will add the avsm Personal Package Archive (PPA), a software repository that contains OCaml and its accompanying software, to your list of downloadable sources.

Note: You may be prompted to enter your password to proceed. This is a precautionary measure to ensure only the owner of the computer changes important configurations. The default timeout is 15 minutes.

The command should result in something that resembles the following image:



```
Fri Mar 03 15:59:04 btrap@BT-PC:/mnt/c/Users/BT Rappaport$ sudo add-apt-repository ppa:avsm/ppa
sudo: unable to resolve host BT-PC
[sudo] password for btrap:
Latest stable versions of OCaml and OPAM.
More info: https://launchpad.net/~avsm/+archive/ubuntu/ppa
Press [ENTER] to continue or ctrl-c to cancel adding it

gpg: keyring `~/tmp/tmpomp70z5o/secring.gpg' created
gpg: keyring `~/tmp/tmpomp70z5o/pubring.gpg' created
gpg: requesting key 61707B09 from hkp server keyserver.ubuntu.com
gpg: /tmp/tmpomp70z5o/trustdb.gpg: trustdb created
gpg: key 61707B09: public key "Launchpad PPA for Anil Madhavapeddy" imported
gpg: Total number processed: 1
gpg:          imported: 1 (RSA: 1)
OK
Fri Mar 03 16:00:01 btrap@BT-PC:/mnt/c/Users/BT Rappaport$
```

Image of Bash Window Installing the avsm PPA.

3. Enter the following update code: **sudo apt-get update**

This will update all of your software packages, getting all the updates from the repository you just added.

The command output should look similar to the image below:

```
Fri Mar 03 22:19:11 btrap@BT-PC:/mnt/c/Users/BT Rappaport$ sudo apt-get update
sudo: unable to resolve host BT-PC
[sudo] password for btrap:
Ign http://archive.ubuntu.com trusty InRelease
Ign http://ppa.launchpad.net trusty InRelease
Get:1 http://archive.ubuntu.com trusty-updates InRelease [65.9 kB]
Hit http://ppa.launchpad.net trusty Release.gpg
Get:2 http://security.ubuntu.com trusty-security InRelease [65.9 kB]
Hit http://ppa.launchpad.net trusty Release
Get:3 http://archive.ubuntu.com trusty-backports InRelease [65.9 kB]
Hit http://ppa.launchpad.net trusty/main amd64 Packages
Hit http://archive.ubuntu.com trusty Release.gpg
Hit http://ppa.launchpad.net trusty/main Translation-en
Get:4 http://archive.ubuntu.com trusty-updates/main amd64 Packages [956 kB]
Get:5 http://security.ubuntu.com trusty-security/main amd64 Packages [589 kB]
Get:6 http://archive.ubuntu.com trusty-updates/restricted amd64 Packages [16.4 kB]
Get:7 http://archive.ubuntu.com trusty-updates/universe amd64 Packages [400 kB]
```

Image of Bash window installing updates.

4. Once that last command is done, Copy and Paste the following command and click enter to install the most up to date version of OPAM and Ocaml:

sudo apt-get install ocaml ocaml-native-compilers camlp4-extra opam

This will result in the following output:

```
Fetches 21.0 kB in 8s (2,501 B/s)
Reading package lists... Done
Sun Mar 05 11:04:20 btrap@BT-PC:/mnt/c/Users/BT Rappaport$ sudo apt-get install ocaml ocaml-native-compilers camlp4-extra opam
sudo: unable to resolve host BT-PC
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following package was automatically installed and is no longer required:
  os-prober
Use 'apt-get autoremove' to remove it.
The following extra packages will be installed:
  camlp4 ocaml-base ocaml-base-nox ocaml-compiler-libs ocaml-interp ocaml-nox
  opam-docs
Suggested packages:
  tc18.5-dev tk8.5-dev ocaml-doc tuareg-mode ocaml-mode
Recommended packages:
  leedit readline-editor
The following NEW packages will be installed:
  camlp4 camlp4-extra ocaml ocaml-base ocaml-base-nox ocaml-compiler-libs
  ocaml-interp ocaml-native-compilers ocaml-nox opam opam-docs
0 upgraded, 11 newly installed, 0 to remove and 36 not upgraded.
Need to get 0 B/36.3 MB of archives.
After this operation, 244 MB of additional disk space will be used.
Do you want to continue? [Y/n] y
Selecting previously unselected package ocaml-base-nox.
(Reading database ... 49052 files and directories currently installed.)
Preparing to unpack .../ocaml-base-nox_4.02.3-2ppa4~trusty_amd64.deb ...
Unpacking ocaml-base-nox (4.02.3-2ppa4~trusty) ...
Selecting previously unselected package ocaml-compiler-libs.
Preparing to unpack .../ocaml-compiler-libs_4.02.3-2ppa4~trusty_amd64.deb ...
Unpacking ocaml-compiler-libs (4.02.3-2ppa4~trusty) ...
Selecting previously unselected package ocaml-interp.
Preparing to unpack .../ocaml-interp_4.02.3-2ppa4~trusty_amd64.deb ...
Unpacking ocaml-interp (4.02.3-2ppa4~trusty) ...
Selecting previously unselected package ocaml-nox.
Preparing to unpack .../ocaml-nox_4.02.3-2ppa4~trusty_amd64.deb ...
```

Image of Bash Window installing OPAM and OCaml

5. Run the command **sudo apt-get install make**. This will install make, something OPAM needs in order to work properly.

```
Thu Mar 09 19:00:28 btrap@BT-PC:/mnt/c/Users/BT Rappaport$ sudo apt-get install make
sudo: unable to resolve host BT-PC
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following package was automatically installed and is no longer required:
  os-prober
Use 'apt-get autoremove' to remove it.
Suggested packages:
  make-doc
The following NEW packages will be installed:
  make
0 upgraded, 1 newly installed, 0 to remove and 36 not upgraded.
Need to get 0 B/119 kB of archives.
After this operation, 328 kB of additional disk space will be used.
Selecting previously unselected package make.
(Reading database ... 50516 files and directories currently installed.)
Preparing to unpack ../make_3.81-8.2ubuntu3_amd64.deb ...
Unpacking make (3.81-8.2ubuntu3) ...
Processing triggers for man-db (2.6.7.1-1ubuntu1) ...
Setting up make (3.81-8.2ubuntu3) ...
Thu Mar 09 19:00:48 btrap@BT-PC:/mnt/c/Users/BT Rappaport$
```

Picture of **sudo apt-get install make**

If you already have make installed Bash will look like this, which is perfectly fine:

```
Thu Mar 09 18:58:27 btrap@BT-PC:/mnt/c/Users/BT Rappaport$ sudo apt-get install make
sudo: unable to resolve host BT-PC
[sudo] password for btrap:
Reading package lists... Done
Building dependency tree
Reading state information... Done
make is already the newest version.
The following package was automatically installed and is no longer required:
  os-prober
Use 'apt-get autoremove' to remove it.
0 upgraded, 0 newly installed, 0 to remove and 36 not upgraded.
Thu Mar 09 18:58:43 btrap@BT-PC:/mnt/c/Users/BT Rappaport$
```

Picture of **sudo apt-get install make** if make is already installed

6. Enter the following command to initialize OPAM: **opam init**

Note: If the wrong version of OPAM was installed this command will not work. Reference the [error section](#) in order to troubleshoot this problem.

Your window should resemble this image:

```
Fri Mar 03 16:11:15 btrap@BT-PC:/mnt/c/Users/BT Rappaport$ opam init
Checking for available remotes: rsync and local, git.
- you won't be able to use mercurial repositories unless you install the hg command on your system.
- you won't be able to use darcs repositories unless you install the darcs command on your system.

[WARNING] Recommended dependencies -- most packages rely on these:
- make
- m4

==== Fetching repository information =====
```

Image of Bash Window after entering command to initialize OPAM

7. Press **'Ctrl-c'**
8. Re-enter the following command to initialize OPAM: **opam init**
9. Bash will prompt you for a yes/no response (y/n). **Type 'y' (no quotation marks)**. Wait for it to complete.
10. In order for OCaml to work properly, you have to enter a command to configure it each time you open Bash. However, this would be very tedious. Fortunately there is something called the bashrc file, a file with commands that will run automatically every time you open Bash. Therefore, all we need to do is add the command to the bashrc file.

Editing your bashrc file

1. Open the bashrc file in vim (a text editor) with the command:
vim ~/.bashrc

```
Fri Mar 03 22:20:59 btrap@BT-PC:/mnt/c/Users/BT Rappaport$ vim ~/.bashrc
```

Image of Line in Bash window.

2. **Press the letter 'i'**. This will allow you to insert text into the file. You will not be able to enter any text until you click 'i'.
3. **Press the down arrow key until you reach the end of the file.**
4. Type (or copy/ paste) the line **eval `opam config env`** into the file
Note: These are "backticks" which are obtained by pressing the key shared with tilde (~), usually located under the Escape key, without pressing the Shift button. They tilt back ` unlike single quotes which look like this `.
Note: To paste in vim you can just right click

```

if [ -f /usr/share/bash-completion/bash_completion ]; then
  . /usr/share/bash-completion/bash_completion
elif [ -f /etc/bash_completion ]; then
  . /etc/bash_completion
fi
fi

# Makes bash shell start in Windows home folder
cd /mnt/c/Users/BT\Rappaport/

# config opam each time bash is opened
eval `opam config env`

```

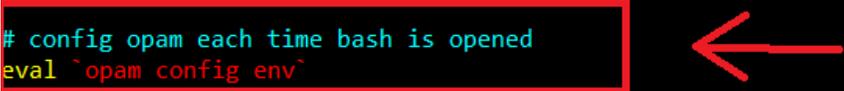


Image of bashrc file after entering the configuration command.

5. **Press the following keys in order:**

- a. **Escape**
- b. **Colon**
- c. **w**
- d. **q**
- e. **'Enter'**

Note: you need to use the Shift key to get : and not ;

This will save and exit the file. The w means write and the q means quit.

Checkpoint

1. **Close and reopen the bash shell.** This will ensure all previously made changes go into effect. To reopen your bash shell, type “bash” in the search bar and choose the first result.
2. Type the command **ocaml** and ensure it says the version is 4.04.0. If it is anything else, go to [OCaml Version Not 4.04.0](#) to fix this before continuing.

```

Thu Mar 09 19:01:57 btrap@BT-PC:/mnt/c/Users/BT Rappaport$ ocaml
OCaml version 4.04.0
#

```



If this works and it is version 4.04.0:

Ocaml and OPAM are installed and configured but they are not ready to use. Proceed to the next section to install additional packages that will make the software fully functional.

Installing Extra Packages (OCamlfind, OUnit, rlwrap)

After you finish successfully installing OPAM and OCaml you will also need some additional packages in order to have optimal functionality.

OCamlfind

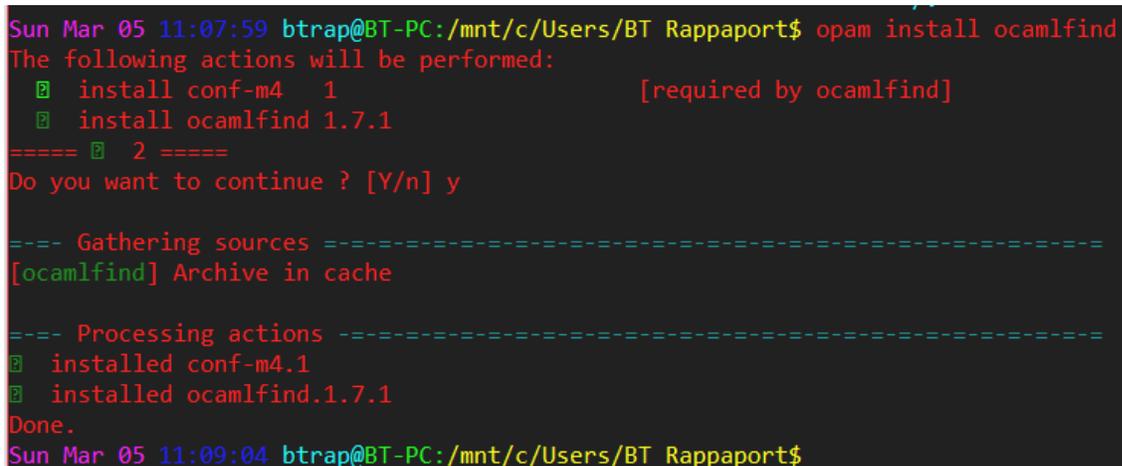
OCamlfind is a package manager that allows users to install, modify, or remove OCaml derived programs. It is extremely useful for compiling OCaml programs and/or libraries.

1. Enter the following command to install OCamlfind: **opam install ocamlfind**
2. After entering the command above, Bash will prompt you with a yes/no response (y/n).

Type the letter ‘y’ when prompted.

After, these steps OCamlfind should be installed successfully.

The output of the steps above should look like the image below:



```
Sun Mar 05 11:07:59 btrap@BT-PC:/mnt/c/Users/BT Rappaport$ opam install ocamlfind
The following actions will be performed:
  [?] install conf-m4 1 [required by ocamlfind]
  [?] install ocamlfind 1.7.1
===== [?] 2 =====
Do you want to continue ? [Y/n] y

--- Gathering sources -----
[ocamlfind] Archive in cache

--- Processing actions -----
[?] installed conf-m4.1
[?] installed ocamlfind.1.7.1
Done.
Sun Mar 05 11:09:04 btrap@BT-PC:/mnt/c/Users/BT Rappaport$
```

Image of Bash window installing OCamlfind.

If your output does not look like this, go to the error section. Links to common Ocamlfnd errors:

- [No package found](#)
- [Install conf m4.1 failed](#)

OUnit

OUnit is a software that allows users to test the code they wrote in OCaml to ensure it works properly. With OUnit, you can make a variety of test cases of expected behavior and then run the tests on the code you have written. When you run the tests, OUnit will print out whether you passed or failed your tests. If the output of the OCaml code corresponds with what the tests expected, then OUnit will indicate that all tests have passed. Therefore, OUnit is very useful for testing OCaml code.

In order to install OUnit follow these steps:

1. Enter the following command to install OUnit: **opam install ounit**
2. After entering the command above and wait for the y/n response as before when installing Ocamlfind

After these steps, OUnit should be installed successfully.

The output of the steps above should look like the image below:

```
Sun Mar 05 11:09:04 btrap@BT-PC:/mnt/c/Users/BT Rappaport$ opam install ounit
The following actions will be performed:
  [x] install ocamlbuild 0.9.3          [required by ounit]
  [x] install ounit      2.0.0
===== [x] 2 =====
Do you want to continue ? [Y/n] y

==-- Gathering sources ==-----
[ounit] Archive in cache
[default] https://opam.ocaml.org/archives/ocamlbuild.0.9.3+opam.tar.gz downloaded

==-- Processing actions ==-----
[x] installed ocamlbuild.0.9.3
[x] installed ounit.2.0.0
Done.
Sun Mar 05 11:11:12 btrap@BT-PC:/mnt/c/Users/BT Rappaport$
```

Image of Bash window installing OUnit.

If your output does not look like this, go to the [error section](#).

Rlwrap

When running the OCaml toplevel, a system which allows interactive use of OCaml, you cannot use the arrows to go back and change mistakes you have made. This can be very aggravating because if you make a mistake you will have to re-type everything over again. Using Rlwrap for the OCaml toplevel gives the ability to use arrows to fix mistakes before entering a command, making using the OCaml toplevel a much more friendly experience.

```
Mon Mar 13 21:08:53 btrap@BT-PC:/mnt/c/Users/BT Rappaport$ ocaml
OCaml version 4.04.0

# let thing - a + b^[D^[D^[D^[D^[D^[D^[D^[D^[D
```

Image of what happens when you try to use the left arrow to go back and change something without Rlwrap.

In order to install Rlwrap follow these steps:

1. Enter the following command to install rlwrap: **sudo apt-get install rlwrap**
2. Wait for the installation of rlwrap to end

After these steps, rlwrap should be installed successfully.

This process should look like this in your Bash shell:

```
Thu Mar 09 19:25:59 btrap@BT-PC:/mnt/c/Users/BT Rappaport$ sudo apt-get install rlwrap
sudo: unable to resolve host BT-PC
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following package was automatically installed and is no longer required:
  os-prober
Use 'apt-get autoremove' to remove it.
The following NEW packages will be installed:
  rlwrap
0 upgraded, 1 newly installed, 0 to remove and 36 not upgraded.
Need to get 0 B/73.4 kB of archives.
After this operation, 229 kB of additional disk space will be used.
Selecting previously unselected package rlwrap.
(Reading database ... 50495 files and directories currently installed.)
Preparing to unpack .../rlwrap_0.37-5_amd64.deb ...
Unpacking rlwrap (0.37-5) ...
Processing triggers for man-db (2.6.7.1-1ubuntu1) ...
Setting up rlwrap (0.37-5) ...
update-alternatives: using /usr/bin/rlwrap to provide /usr/bin/readline-editor (readline-editor) in auto mode
Thu Mar 09 19:26:44 btrap@BT-PC:/mnt/c/Users/BT Rappaport$
```

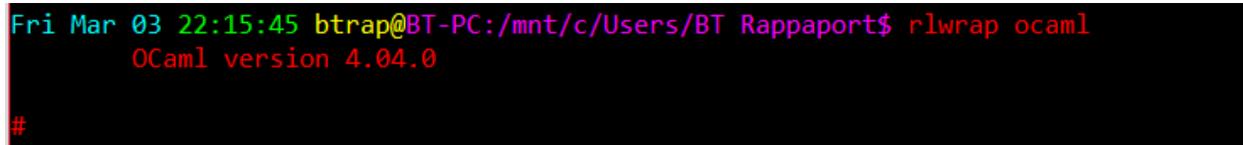
Successful installation of rlwrap

Ensuring Correct Installation and Configuration

After the installation of “rlwrap” is complete, you are finished installing and configuring everything. However, it is considered best practice to manually verify that the installation of all previous software packages was successful.

In order to test if the previous installations and configurations were successful follow these instructions:

1. **Close and reopen the bash shell.** This will ensure all previously made changes go into effect. This is analogous to pressing a reset button. To reopen your bash shell after closing, click on the Start Windows icon in the lower left corner of the screen. In the search bar that is revealed type “bash” and double click the first result.
2. Run the command **rlwrap ocaml** and verify that you get an OCaml top level of version 4.04.0 by checking if the subsequent output prompt looks like the one pictured below.



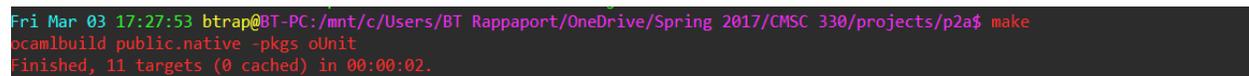
```
Fri Mar 03 22:15:45 btrap@BT-PC:/mnt/c/Users/BT Rappaport$ rlwrap ocaml
OCaml version 4.04.0
#
```

Image of output of **rlwrap ocaml**

If version is not 4.04.0, refer to [“Ocaml Version Not 4.04.0”](#) for further support.

3. As a final check, verify that an OCaml project can be compiled by the computer. If you have an OCaml project with a Makefile, just go to the directory the project is in and type the command **make**.

The output prompt pictured below verifies that the computer is able to compile an OCaml program. If you get an error instead, refer to the Troubleshooting Errors section:



```
Fri Mar 03 17:27:53 btrap@BT-PC:/mnt/c/Users/BT Rappaport/OneDrive/Spring 2017/CMSC 330/projects/p2a$ make
ocamlbuild public.native -pkgs oUnit
Finished, 11 targets (0 cached) in 00:00:02.
```

Image of a successful **make**

This final verification indicates a successful installation and configuration of OCaml for Windows 10.

Troubleshooting Errors

There are many possible issues that can come up while installing OPAM and OCaml. The majority of these issues have to do with your computer not having something necessary installed beforehand or will something not being up to date enough. This section will go through the most common errors and explain what to do in order to fix them.

OCaml Version Not 4.04.0

If you run the command `ocaml -version` and it says any other version besides 4.04.0, then OPAM did not install the newest version of OCaml on your computer.

```
Fri Mar 03 16:45:52 btrap@BT-PC:/mnt/c/Users/BT Rappaport$ ocaml -version
The OCaml toplevel, version 4.02.3
```

Result of the command `ocaml -version` not being 4.04.0

In order to obtain the newest version of OCaml, do the following:

1. Run the command `opam switch 4.04.0`. This will take a while. Wait for it to complete. The result should look like this:

```
Fri Mar 03 17:38:28 btrap@BT-PC:/mnt/c/Users/BT Rappaport$ opam switch 4.04.0
==== Installing compiler 4.04.0 =====
[default.comp] https://github.com/ocaml/ocaml/archive/4.04.0.tar.gz downloaded
Now compiling OCaml. This may take a while, please bear with us...
Done.

==== Gathering sources =====

==== Processing actions =====
[ ] installed base-bigarray.base
[ ] installed base-threads.base
[ ] installed base-unix.base
Done.
# To setup the new switch in the current shell, you need to run:
eval `opam config env`
Fri Mar 03 17:58:22 btrap@BT-PC:/mnt/c/Users/BT Rappaport$ eval `opam config env`
Fri Mar 03 22:11:47 btrap@BT-PC:/mnt/c/Users/BT Rappaport$
```

Result of running `opam switch 4.04.0`

2. Run the command `eval `opam config env``. This will configure the new version of OCaml to work properly within the bash window you currently have open.

Failed 'opam init' command

If the **opam init** command fails with

```
[ERROR] Your version of OPAM (1.1.1) is not recent enough to read  
the version of OPAM installed is too old to properly work.
```

In order to fix this, do the following:

1. Run the command **opam update** - This will update OPAM so that you can upgrade to the newest version. This may take a while. Wait for it to complete.
2. Run the command **opam upgrade** - This will upgrade OPAM to the newest version based on the updates you downloaded. This may take a while. Wait for it to complete.
3. Go back to the **opam init** step in the instructions and continue again from there.

No package found for Ocamlfind or Ounit

If your Ocamlfind or Ounit installations result in

```
[ERROR] No package named ocamlfind found.
```

OR

```
[ERROR] No package named ounit found.
```

then your version of OPAM may be outdated.

In order to fix this, do the following:

1. Run the command **opam update** - This will update OPAM so that you can upgrade to the newest version. This may take a while. Wait for it to complete.
2. Run the command **opam upgrade** - This will upgrade OPAM to the newest version based on the updates you downloaded. This may take a while. Wait for it to complete.
3. Go back to the step you were up to in the instructions and continue again from there (either installing Ocamlfind or OUnit).

Ocamlfind - install conf m4.1 failed

If you get the following error while trying to install Ocamlfind, your computer is missing m4, something which ocamlfind depends on.

```
[ERROR] The compilation of conf-m4 failed at "sh -exc echo | m4".

#=== ERROR while installing conf-m4.1 =====#
# opam-version 1.2.2
# os linux
# command sh -exc echo | m4
# path /home/btrap/.opam/system/build/conf-m4.1
# compiler system (4.02.3)
# exit-code 127
# env-file /home/btrap/.opam/system/build/conf-m4.1/conf-m4-710-1872ec.env
# stdout-file /home/btrap/.opam/system/build/conf-m4.1/conf-m4-710-1872ec.out
# stderr-file /home/btrap/.opam/system/build/conf-m4.1/conf-m4-710-1872ec.err
### stderr ###
# + echo
# + m4
# sh: 1: m4: not found

--- Error report -----
The following actions were aborted
  [x] install ocamlfind 1.7.1
The following actions failed
  [x] install conf-m4 1
No changes have been performed

--- conf-m4.1 troubleshooting -----
=> This package relies on external (system) dependencies that may be missing. `opam depext conf-m4.1' may help you find the
correct installation for your system.
```

Result of `install conf-m4 1` failing

To fix this, do the following (If you have any trouble see the picture below the instructions):

1. Run the command `opam depext conf-m4.1`. This tells OPAM to search for and install the missing dependencies.
2. During the running of this command **Bash will prompt you twice for a yes/no response (y/n). Respond 'y' to each of these.**
3. After some processing, it will say

```
# The following system packages are needed:
# - m4
The following command needs to be run through "sudo":
  apt-get install m4
Allow ? [Y/n] y
```

4. **Respond 'y'**
5. Bash will then prompt you for your password. **Type your password and click 'Enter'.**
6. Wait for it to finish and for your prompt to show up again. Once it does so, continue following the instructions again from `opam install ocamlfind`

```

Fri Mar 03 16:35:50 btrap@BT-PC:/mnt/c/Users/BT_Rappaport$ opam depext conf-m4.1
OPAM plugin depext is not installed. Install it on the current switch? [Y/n] y

The following actions will be performed:
  [ ] install base-ocamlbuild base           [required by ocamlbuild]
  [ ] install ocamlbuild 0                   [required by cmdliner]
  [ ] install cmdliner 0.9.8                 [required by depext]
  [ ] install depext 1.0.2
===== [ ] 4 =====
Do you want to continue ? [Y/n] y

--- Gathering sources -----
[default] https://opam.ocaml.org/archives/depext.1.0.2+opam.tar.gz downloaded
[default] https://opam.ocaml.org/archives/cmdliner.0.9.8+opam.tar.gz downloaded

--- Processing actions -----
  [ ] installed base-ocamlbuild.base
  [ ] installed ocamlbuild.0
  [ ] installed cmdliner.0.9.8
  [ ] installed depext.1.0.2
Done.

--- Carrying on to "opam depext conf-m4.1" -----

# Detecting depexts using flags: x86_64 linux ubuntu
# The following system packages are needed:
# - m4
The following command needs to be run through "sudo":
  apt-get install m4
Allow ? [Y/n] y
sudo: unable to resolve host BT-PC
[sudo] password for btrap:
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following package was automatically installed and is no longer required:
  os-prober
Use 'apt-get autoremove' to remove it.
The following NEW packages will be installed:
  m4
0 upgraded, 1 newly installed, 0 to remove and 36 not upgraded.
Need to get 195 kB of archives.
After this operation, 342 kB of additional disk space will be used.
Get:1 http://archive.ubuntu.com/ubuntu/ trusty/main m4 amd64 1.4.17-2ubuntu1 [195 kB]
Fetched 195 kB in 0s (301 kB/s)
Selecting previously unselected package m4.
(Reading database ... 33986 files and directories currently installed.)
Preparing to unpack .../m4_1.4.17-2ubuntu1_amd64.deb

```

Picture of the process to fix **install conf-m4** failing

Glossary

apt-get	A command line package management tool that facilitates package installation, updating, and upgrading.
add-apt-repository	A command line tool used to add a Personal Package Archive (PPA) to your sources list so that you are able to download the contents from that repository.
Bash	Bash, or the Bourne Again shell (see Shell definition), is the standard shell for common users. Bash is a command processor that usually runs in a text window, which allows the user to type commands that cause actions. An example is the command “ls” which lists all the files in the folder (directory) you are currently in.
bashrc file	A shell (see Shell definition) script that runs every time a new bash shell is opened by the user. This script (file with commands) allows users to have commands automatically run every time a bash shell is open. This is very useful because it means you do not have to configure things manually every time.
make	Make is a tool that automatically figures out and then recompiles source files of a program which need to be recompiled and/or linked. Make uses a file called the makefile (or Makefile) in order to know of how to build your program.
Repository	A centralized place where data, multiple databases or files, is stored and maintained for distribution. In our case, the avsm repository is where all the files to install OCaml and OPAM are located.
sudo	The sudo command stands for "super user do". It enables you to execute a command with elevated privilege. When you run sudo, it prompts you for your personal password and if you correctly enter that password it runs the command. Sudo is used to run commands that would give you a permission denied error without having administrative access.
Shell	Shell is the UNIX word for the interactive user interface with the operating system. The shell understands and executes the commands a user enters, as seen when you enter commands like “opam init”.