

ReasonML Install Guide

Fall 2019

Contents

1	Install Guide (Windows)	1
1.1	Installing Node	1
1.2	WSL	1
1.3	Configuring Visual Studio Code	2
2	Install Guide (Mac OSX)	2
2.1	Installing Visual Studio Code	2
2.2	Installing Various XCode Tools (NVM, Node)	2
2.3	Configuring Visual Studio Code	5
3	Using ReasonML	5
3.1	Configuring Your Project	5
3.2	Compiling ReasonML	5
3.3	Running ReasonML	5
4	Reason Tools (Browser)	6

1 Install Guide (Windows)

1.1 Installing Node

Click [here](#) to install Node.

1.2 WSL

Follow this [guide](#) to install Windows Subsystem for Linux (WSL). Select the Linux Distribution choice *Ubuntu*, which has an orange icon.

Open the "Command Prompt" on your machine and type the following command. This can take a long time to install fully. Let it sit for some time, even if it appears as if it's not doing anything. Feel free to move on to other steps while it loads.

```
npm install -g ocaml-reason-wsl
```

Inside of Visual Studio Code go to Settings > User Settings. In the upper right-hand corner, click on the icon that looks like a file with an arrow wrapped around it. A window should open up with a file in it. Copy and paste the following lines into the file in between the curly braces and save the file.

```
"reason.path.bsb": "bash -ic bsb",
"reason.path.ocamlfind": "bash -ic ocamlfind",
"reason.path.ocamlmerlin": "bash -ic ocamlmerlin",
"reason.path.opam": "bash -ic opam",
"reason.path.rebuild": "bash -ic rebuild",
"reason.path.refmt": "bash -ic refmt",
"reason.path.refmterr": "bash -ic refmterr",
"reason.path.rtop": "bash -ic rtop",
"editor.formatOnSave": true,
"reason.diagnostics.tools": ["merlin", "bsb"],
"terminal.integrated.shell.windows": "C:\\\\WINDOWS\\\\System32\\\\bash.exe"
```

After the initial install command has terminated, you can now open WSL (Ubuntu).

In a Command Prompt window, navigate to your Desktop folder by typing the following command:

```
cd /mnt/c/Users/<your username>/Desktop/
```

Create a sample project called "cs17" by typing the following lines. Hit enter between line breaks.

```
bsb -init cs17 -theme basic-reason
cd cs17
npm install
npm run build
```

After you install VS Code, you can open the VS Code application, open a terminal window within VS Code by navigating to Terminal > New Terminal. Within the VS Code terminal, open the folder you just created and run the file that you made by typing the following commands:

```
npm run build
node src/Demo.bs.js
```

You should see the message "Hello, BuckleScript and Reason!" in the VS Code terminal!

1.3 Configuring Visual Studio Code

Before VS Code is fully functional for the purposes of CS 17, we need to set up the support for our programming language, ReasonML.

1. To begin, open VS Code and select the fifth icon on the left vertical bar for "Extensions."
2. Search "reason-vscode" in the subsequent search bar.
3. Select and install the first result.
4. Restart VS Code.

5. Open a new file, and click the "Plain Text" button on the bottom purple toolbar.
6. Set "Reason" as your Language Mode.
7. Begin writing code in ReasonML!

2 Install Guide (Mac OSX)

2.1 Installing Visual Studio Code

To install Visual Studio Code (VS Code), follow this video tutorial to download the correct installation for your hardware.

2.2 Installing Various XCode Tools (NVM, Node)

Detailed instructions to install Node Version Manager (NVM) can be found [here](#), but the main steps are as follows:

1. In a terminal window, run the following command:

```
curl -o- https://raw.githubusercontent.com/nvm-sh/nvm/v0.34.0/install.sh  
| bash
```

This will install NVM and subsequently a file called 'Node.js' (js stands for JavaScript) in a manner that won't affect user permissions.

2. Open a *new* terminal window and type the following command:

```
command -v nvm
```

You should receive the following response:

```
nvm
```

If you do not receive this response one of the following might be the reason:

- Your system may not have a `.bash_profile` file where the command is set up. Create one by running the following command:

```
touch ~/.bash_profile
```

Then run the install script again.

- Another reason is that you might need to restart your terminal instance. Try opening a new tab/window in your terminal and retry.

If the above doesn't fix the problem, open your `.bash_profile` by running the following commands:

```
cd ~  
vim .bash_profile
```

Click the letter i, this will go into insert mode in vim, a text editor. Type out the following lines into the top of this file:

```
export NVM_DIR=~/.nvm  
source ~/.nvm/nvm.sh
```

Click ‘esc’ to get out of insert mode, and then type :wq to save the file and quit vim. Close terminal and open a new one and try to run command -v nvm again.

3. To install the file ‘Node.js’ via NVM, type the following command in your terminal window:

```
nvm install node
```

4. Open the VS Code application, click the ‘Terminal’ button on the top toolbar, select ‘New Terminal’ and type the following command (note that copy/pasting the following command might give a ^ character of the incorrect form, so you may need to delete that character and re-type before hitting enter):

```
perl -e'exit(!(grep(m{^/usr/local/bin$},split(";", $ENV{PATH}))) > 0)'&& echo "found it"
```

Note: The above line is not in a code block because it would mess up the formatting of these special characters. If you have issues with this command, please post on Piazza.

You should receive the following response:

```
found it
```

If you do not receive this as a response, there is something wrong with your startup files. Please post on Piazza for help with this.

5. Now, type the following command:

```
npm install -g bs-platform
```

6. Close *both* terminal windows and open a new terminal window.
7. Type the following series of commands:

```
cd ~  
mkdir cs17  
cd cs17  
bsb -init test-app -theme basic-reason  
cd test-app  
ls -aR
```

8. Examine the tree of files shown and locate in the ‘src’ directory a file called ‘Demo.re’
9. Navigate to the ‘src’ directory and rename the file to ‘Foo.re’

```
cd src
mv Demo.re Foo.re
```

10. To compile the file, type the following command. This will create a file called 'Foo.bs.js', a JavaScript file. JavaScript is the language that backs ReasonML. There is also a language called BuckleScript ('.bs') that compiles ReasonML code into JavaScript, which the computer can then evaluate.

```
npm run build
```

11. You can run this file and get results by typing the following command:

```
node Foo.bs.js
```

12. Want to add another Reason file? Create an empty file called 'Bar.re' and run the following command to compile BOTH files.

```
touch Bar.re
npm run build
```

2.3 Configuring Visual Studio Code

Before VS Code is fully functional for the purposes of CS 17, we need to set up the support for our programming language, ReasonML.

1. To begin, open VS Code and select the fifth icon on the left vertical bar for "Extensions."
2. Search "reason-vscode" in the subsequent search bar.
3. Select and install the first result.
4. Restart VS Code.
5. Open a new file, and click the "Plain Text" button on the bottom purple toolbar.
6. Set "Reason" as your Language Mode.
7. Begin writing code in ReasonML!

3 Using ReasonML

3.1 Configuring Your Project

When using VS Code, in order to run your programs, all of your files must live in their appropriate directories. Whenever you want to create a new project, you must run the following command to set up the appropriate language configuration BuckleScript files that will convert ReasonML to JavaScript, which is how the computer will interpret the program.

Note: VS Code requires all project folders to be named with **lowercase** letters.

```
bsb -init <project_name> -theme basic-reason
```

This command will automatically create a number of important configuration files, as well as an 'src' directory. Within the 'src' directory, there will be an auto-generated 'Demo.re' file, which you may delete. All of your Reason ('.re') files must be located in this 'src' directory.

3.2 Compiling ReasonML

From your main project directory, you will run the command:

```
npm run build
```

3.3 Running ReasonML

Within your 'src' directory of your project directory, you will run the command to run the JavaScript file created after compiling your ReasonML code:

```
node <file_name>.bs.js
```

You can also run your files directly from your project directory by including the 'src' directory name in the file path:

```
node src/<file_name>.bs.js
```

4 Reason Tools (Browser)

One awesome feature of ReasonML, because it is backed by JavaScript, is that it can be run in your Internet browser. Follow the instructions here to obtain the Chrome or Firefox extensions.

Please let us know if you find any mistakes, inconsistencies, or confusing language in this or any other CS 17 document by filling out the anonymous feedback form: <http://cs.brown.edu/courses/csci0170/feedback>.