

Installing Software

<https://csci-1301.github.io/about#authors>

January 17, 2023 (07:48:04 PM)

Contents

1	Generalities on Installing Software	1
2	Executing Code Found on-line	2
3	Accessing an IDE	2
3.1	Installing an IDE On Your Own Computer	3
3.1.1	Installing Visual Studio On Your Own Computer	3
3.1.2	Installing MonoDevelop On Your Own Computer	4
3.1.3	Installing Rider On Your Own Computer	4
3.1.4	Installing Geany On Your Own Computer	5
3.1.5	Installing Anything Anywhere	6
3.2	Accessing One of the Computers in a Computer Lab	7
3.3	Compiling Code On-Line	7

1 Generalities on Installing Software

You probably already installed software in your life, be it VLC¹, Microsoft Teams², or Whatsapp³. However, depending on whether you installed it on a phone, a tablet, a computer, and depending on the operating systems (Android, Windows 10, iOS, Ubuntu, etc.) your experience may have varied drastically.

Between the Play store⁴, the command-line interface⁵, homebrew⁶ and the act of downloading software using your browser and then installing it using the navigator, there can be a lot of differences, but in all those circumstances you should keep security in mind. In addition to making sure that you are downloading the software from a trusted source, you should also be vigilant about the information the software will be able to access about e.g. your private life.

As data can be lost or corrupted upon downloading, many platforms now use checksums⁷ to verify the integrity of the software you downloaded before installing it. This is an excellent practice that can also be performed “by hand”, as explained for instance for the database manager MySQL⁸: the main idea is that the probability of the signature matching a tampered-with file is extremely low, and that as long as you are

¹<http://www.videolan.org/>

²<https://www.microsoft.com/en-us/microsoft-teams/download-app>

³<https://www.whatsapp.com/>

⁴https://en.wikipedia.org/wiki/Google_Play#Play_Store_on_Android

⁵https://en.wikipedia.org/wiki/Command-line_interface

⁶<http://brew.sh/>

⁷<https://en.wikipedia.org/wiki/Checksum>

⁸<https://dev.mysql.com/doc/refman/8.0/en/verifying-package-integrity.html>

downloading the signature and the software from two different sources, you are considerably reducing the attack surface⁹.

2 Executing Code Found on-line

As you progress in this class, you will be asked more and more to download and execute code hosted in our repository¹⁰. How can you tell that you can trust this code?

We have not implemented checksum-matching (yet!), but you can trust this code as it was coded by your instructors, and hosted on a platform using two-factor authentication¹¹ where every action is tracked using versioning¹². Concretely, this means that only somebody who manages to steal your instructor's credentials and their phone, and thwart all the other instructors' vigilance, would be able to host malicious code on our platform: while we certainly imagine that this is theoretically possible, we hope that you will agree that the probability is low enough for you to trust the code on this site.

As often, security is not absolute, but aims at providing reasonable confidence. Executing “blindly” code found on-line, on the other hand, gives you a good chance of facing unpleasant surprises: while there certainly is a lot of useful, good code on websites like stackoverflow¹³—your instructor probably uses such websites, by the way!—, copying-and-pasting it without understanding its purpose or general structure is almost guaranteed to, at best, not execute properly, at worst, make your system unstable or insecure.

3 Accessing an IDE

An IDE¹⁴, for “Integrated development environment”, is the software or service you will be using to write, compile, execute and debug your code. There are many available IDEs, and some can accommodate multiple different programming languages.

For C#, there are many different possibilities¹⁵: some are cross-platforms (meaning you can use them on macOS, Windows or Linux), some are provided free of charge, some have not been updated in a long time. Three natural choices are Visual Studio¹⁶, MonoDevelop¹⁷ and Rider¹⁸. While the last two are accessible on every operating systems, Visual Studio is available only for Windows, and in a slightly different version for macOS.

To access one or the other, you will need either

- a computer with the right to install software on it,
- to access one of the computers in the computer lab¹⁹, or
- a computer with internet access.

The third solution is a backup plan, as instead you will access a very minimal version of an IDE to test small snippets of code. You should not rely on it for the duration of this course.

⁹https://en.wikipedia.org/wiki/Attack_surface

¹⁰<https://github.com/csci-1301/csci-1301.github.io/>

¹¹<https://docs.github.com/en/authentication/securing-your-account-with-two-factor-authentication-2fa/about-two-factor-authentication>

¹²https://en.wikipedia.org/wiki/Software_versioning

¹³<https://stackoverflow.com/>

¹⁴https://en.wikipedia.org/wiki/Integrated_development_environment

¹⁵https://en.wikipedia.org/wiki/Comparison_of_integrated_development_environments#C%23

¹⁶<https://visualstudio.microsoft.com/>

¹⁷<https://www.monodevelop.com/>

¹⁸<https://www.jetbrains.com/rider/features/>

¹⁹<https://www.augusta.edu/its/computers-printing.php>

3.1 Installing an IDE On Your Own Computer

This part gathers some references for you to install Visual Studio²⁰, MonoDevelop²¹ and Rider²² on your own computer, regardless of your operating system. It is strongly encouraged that you do so, especially if you want to continue in a CS/IT/Cyber degree, but is not mandatory²³.

The instructions are detailed, but there are plenty of ways this can go wrong: make sure you have read and followed those instructions carefully before asking for help²⁴!

3.1.1 Installing Visual Studio On Your Own Computer

Note that we are *not* installing “Visual Studio *Code*”, but simply “Visual Studio”.

3.1.1.1 For Windows

1. Visit Azure Dev Tools for Teaching²⁵.
2. Log in using your Augusta University credentials.
3. Select “Download software”.
4. Look for Visual Studio. The path is Education → Software → Visual Studio Enterprise 2019. You can search “Services” for the “Education” group and then click “Software” if the education group is not immediately displayed. It should look like the following:



Normally, the following direct link should get you to the right page: https://portal.azure.com/?Microsoft_Azure_Education_correlationId=8ee63052-dc32-46f7-a109-e26793622dbf#. Type “Education” in the search bar to get started on the path to the program to download.

5. Download and install Visual Studio (leave all the options on their default settings).

If possible, check the box next to “Workload” → “Windows” → “.NET-Desktop Development”, or to “Workload” → “Other Toolsets” → “.NET Core cross-platform development”:

²⁰<https://visualstudio.microsoft.com/>

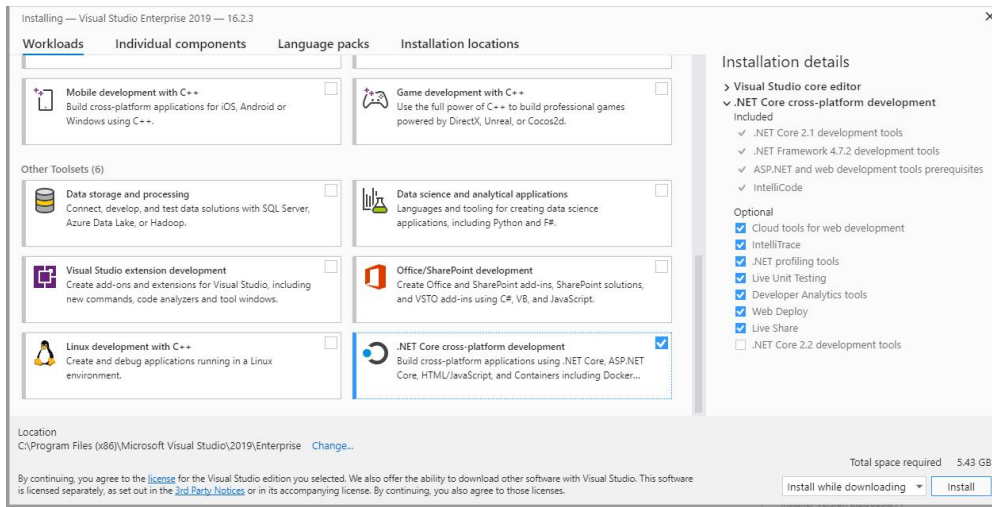
²¹<https://www.monodevelop.com/>

²²<https://www.jetbrains.com/rider/features/>

²³Unless this class is fully online, of course.

²⁴[labs/Introduction/#how-to-ask-a-question](https://aka.ms/devtoolsforteaching)

²⁵<https://aka.ms/devtoolsforteaching>



6. Enter the product key you obtained previously, following the instructions in the documentation²⁶. Normally, clicking on “View key” on the screen pictured in the fourth step above should give you access to a key, that you simply need to copy-and-paste in the menu you can access on Visual Studio by clicking on “Select File” → “Account Settings” → “License with a Product Key”.

3.1.1.2 For Mac Download a version of Visual Studio at <https://visualstudio.microsoft.com/vs/mac/>. It differs a bit from the Windows version, but that should not impact your experience in this class. The only Visual Studio feature we rely on is the ability to create “Console Apps with C#”, which is equally available in both the Windows and Mac versions.

3.1.2 Installing MonoDevelop On Your Own Computer

Unfortunately, MonoDevelop offers pre-packaged release only for linux distributions

- If you are using linux (e.g. Ubuntu, Debian, etc.), then please head out to MonoDevelop’s download page²⁷.
- If you are using MacOS, you can have a look at the compilation instructions²⁸, but it is very likely that you will find them impossible to understand.
- If you are using Windows, you can have a look at the compilation instructions²⁹, but it is very likely that you will find them impossible to understand.

3.1.3 Installing Rider On Your Own Computer

You can download Rider from their website³⁰, for any operating system. Note that, as a student, you can obtain a licence for free³¹: simply fill out this form³², making sure you use your @augusta.edu email account, and you should receive a free licence instantaneously!

Note that JetBrains offers to use a SHA-256 checksum (for instance, for the linux version³³) for you to check that your download has not been tampered with. In any case, you can consult their detailed instructions³⁴

²⁶<https://docs.microsoft.com/en-us/visualstudio/ide/how-to-unlock-visual-studio?view=vs-2019>

²⁷<https://www.monodevelop.com/download/#fndtn-download-lin>

²⁸<https://www.monodevelop.com/developers/building-monodevelop/#macos>

²⁹<https://www.monodevelop.com/developers/building-monodevelop/#windows>

³⁰<https://www.jetbrains.com/rider/download/>

³¹<https://www.jetbrains.com/community/education/#students>

³²<https://www.jetbrains.com/shop/eform/students>

³³<https://download.jetbrains.com/rider/JetBrains.Rider-2022.2.2.tar.gz.sha256>

³⁴https://www.jetbrains.com/help/rider/Installation_guide.html#standalone

to install and execute Rider on any operating system.

3.1.4 Installing Geany On Your Own Computer

Note: *This method will only allow you edit and compile individual .cs files, and will not compile C# Solution Projects (which will be required by the fifth week of lab). We would recommend learning another one of the IDEs listed above. If you still seek to use Geany throughout this course, you will be on your own to find a way to configure Geany to work with C# Solution Projects, and could start by reading this exchange³⁵ (which is about projects in Linux, but applies equally well to projects in C#) or this one³⁶.*

You can download Geany from their website³⁷, for any operating system. Unfortunately, Geany does not natively build and compile C# code. To use Geany as a text editor for C#, we must download the Mono C# compiler from their website³⁸. Make sure to download the most recent version to assure your compiler has the most up-to-date version of “.NET”.

Once you installed Mono, locate the “csc.bat”, “csc.exe” or “csc” file in Mono’s “bin” folder and copy the file path. This path can be of the form

```
C:\Program Files (x86)\Mono\bin\csc.bat
```

on windows, or

```
/usr/bin/csc
```

on Unix systems.

Now open a .cs file using Geany. Click the arrow next to the “Build” Button and click “Set Build Commands” from the dropdown menu.



Figure 1: Accessing the menu to set build commands

In the “Set Build Commands” window, erase the entry next to the “Compile” button and paste the file path to the “csc.bat” in quotation marks. After the file path, create a single space followed by “%f” with the quotation marks. All in all, you should have something of the form

³⁵<https://stackoverflow.com/q/54041013>

³⁶<https://stackoverflow.com/q/8264323>

³⁷<https://www.geany.org/>

³⁸<https://www.mono-project.com/download/stable/>

```
"C:\Program Files (x86)\Mono\bin\csc.bat" "%f"
```

in the “Command” field of the “Compile” line.

Confirm the change by clicking OK and now you will be able to compile, build, and execute **standalone** .cs files.

Set Build Commands

#	Label	Command	Working directory	Reset
C# commands				
1.	Compile	C:\Program Files (x86)\Mono\bin\csc.bat "%f"		
2.				
3.				
Error regular expression:				
Independent commands				
1.	Make	make		
2.	Make Custom Target...	make		
3.	Make Object	make %e.o		
4.				
Error regular expression:				
<i>Note: Item 2 opens a dialog and appends the response to the command.</i>				
Execute commands				
1.	Execute	%e.exe		
2.				
<i>%d, %e, %f, %p, %l are substituted in command and directory fields, see manual for details.</i>				

Figure 2: Setting the build commands

3.1.5 Installing Anything Anywhere

If the IDE you would like to adopt is not available for your operating system, you can use a Virtual Machine³⁹ manager to run a linux-based distribution or a Windows image on top of your operating system.

For this, and regardless of your current operating system, you will need a Virtual Machine⁴⁰ manager.

³⁹https://en.wikipedia.org/wiki/Virtual_machine

⁴⁰https://en.wikipedia.org/wiki/Virtual_machine

1. You can use
 - a) “VMware Fusion 11.x Pro” (only for MacOS, available on OnTheHub⁴¹),
 - b) Virtual Box⁴² (for Linux and Mac),
 - c) Virtual Machine Manager⁴³ (for Linux).
2. Download a version of “Microsoft Operating Systems” from Azure Dev Tools for Teaching⁴⁴, or a linux-based distribution (typically, ubuntu⁴⁵ has a good reputation of being accessible and user-friendly).
3. Install and run your version of Windows or Linux from your virtual machine, and follow the corresponding instructions to install the IDE you are interested in.

Note that it is illegal to run macOS in a virtual environment that is not hosted on a mac computer⁴⁶, which drastically reduces the interest for you to consider this option.

3.2 Accessing One of the Computers in a Computer Lab

Please refer to this page from AU’s Information Technology⁴⁷ to know where the computer labs are located. Visual Studio should be pre-installed on every computer.

3.3 Compiling Code On-Line

As a backup or only to test snippets of code, you can compile C# code online. Multiple online platforms exist, such as:

- https://www.tutorialspoint.com/compile_csharp_online.php
- https://www.onlinegdb.com/online_csharp_compiler
- <https://www.jdoodle.com/compile-c-sharp-online/>
- <https://dotnetfiddle.net/>
- https://www.w3schools.com/cs/trycs.asp?filename=demo_helloworld

Note that none of them are endorsed by the school and that they can pose security and privacy challenges: never enter any sensitive information and do not rely on them too heavily. However, they can be a good support if you would like to test a short snippet of code but do not have access at the moment to a computer with MonoDevelop or Visual Studio installed.

⁴¹<https://e5.onthehub.com/WebStore/OfferingDetails.aspx?o=cff1cf98-42ce-e811-810b-000d3af41938&pmv=00000000-0000-0000-000000000000&ws=2020165a-723a-de11-b696-0030485a8df0&vsro=8>

⁴²<https://www.virtualbox.org/>

⁴³<https://virt-manager.org/>

⁴⁴<https://aka.ms/devtoolsforteaching>

⁴⁵<https://ubuntu.com/appliance/vm>

⁴⁶<https://law.stackexchange.com/q/18282>

⁴⁷<https://www.augusta.edu/its/computers-printing.php>