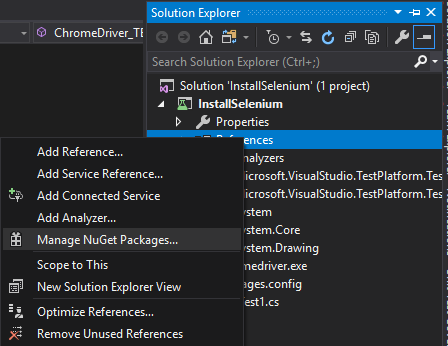
**SELENIUM NOTES**

**REQUIREMENTS**

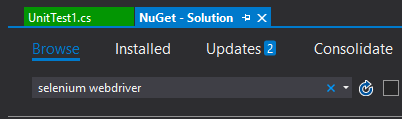
1. Selenium Web Driver
   1. Instructions below
2. Chrome Driver
   1. found [HERE](https://sites.google.com/a/chromium.org/chromedriver/downloads)

**Install Selenium Web Driver**

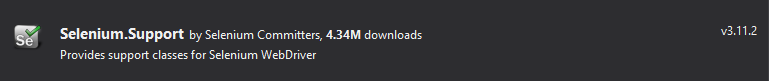
1. In Visual Studio, right click References > Manage NuGet Packages…



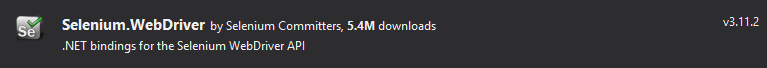
1. Switch to the “Browse” tab
2. Search: ‘selenium webdriver’



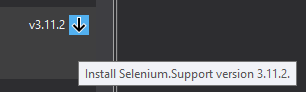
1. Install [Selenium.Support](https://www.nuget.org/packages/Selenium.Support/)



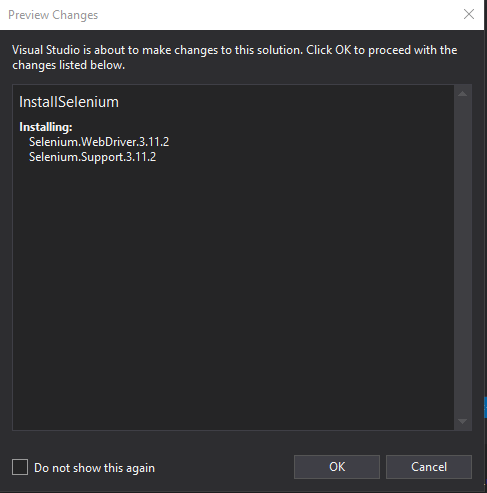
1. Install [Selenium.WebDriver](https://www.nuget.org/packages/Selenium.WebDriver/)



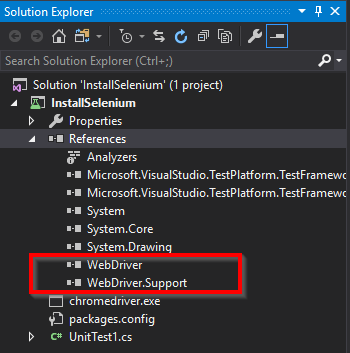
1. *Note:* If you click this icon for Selenium Support, it will download Selenium.WebDriver because Selenium.Support depends on Selenium.WebDriver:



1. In the Changes dialog, click OK:



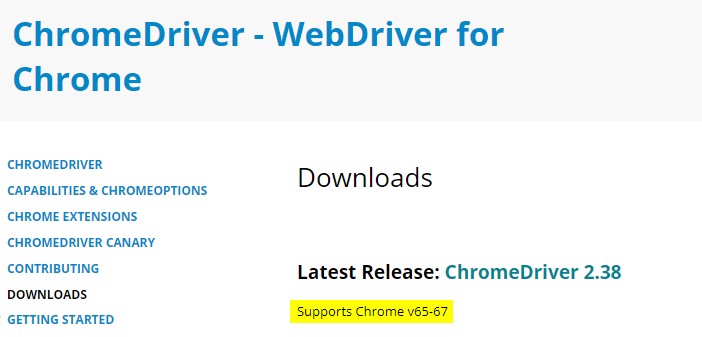
1. Verify that the install succeeded:



1. Build Solution

**Install Chrome Driver**

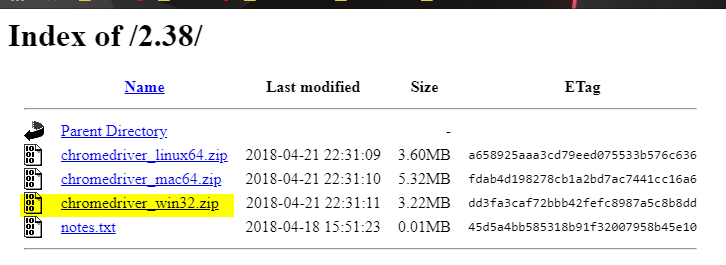
1. Navigate [HERE](https://sites.google.com/a/chromium.org/chromedriver/downloads)
2. Download the Latest Release



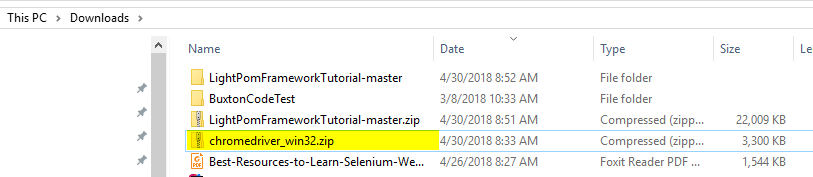
*Note:* Version number may be different than the screenshot above

*Note:* The highlighted text displays which version of Chrome the Driver supports

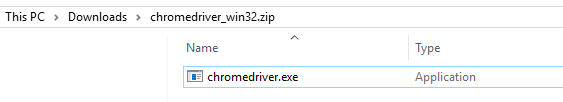
1. Choose your OS:



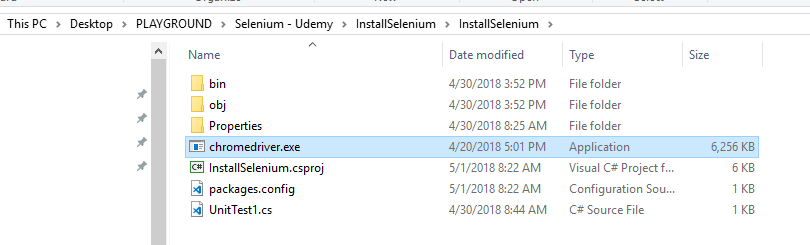
The download will be in a .zip format:



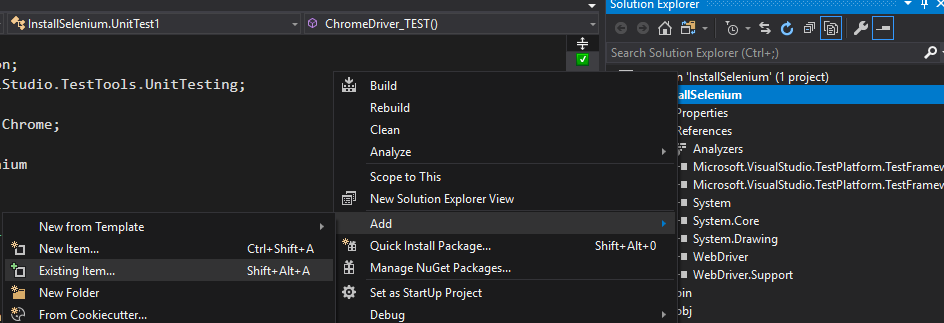
1. Open the .zip folder, and copy the .exe:



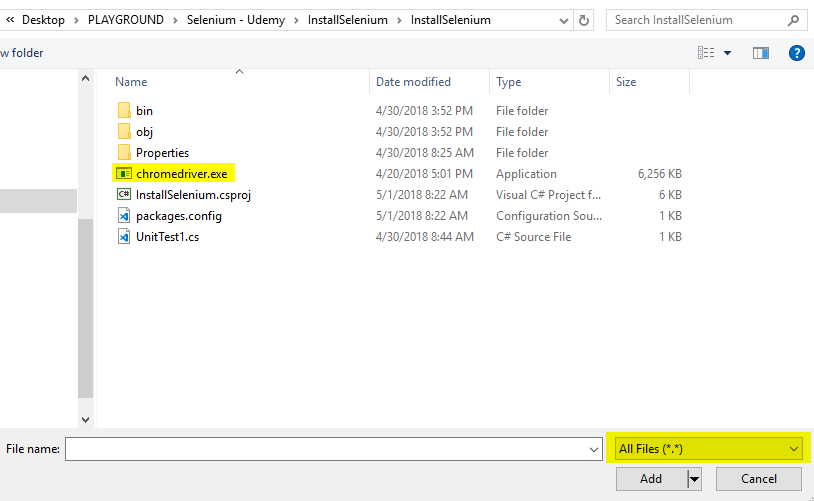
1. Drop the .exe file in the desired solution, so that it appears in the same level as the bin and obj folders:



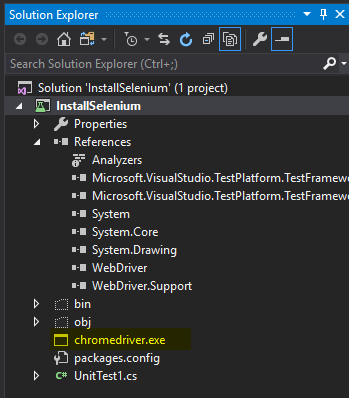
1. To add the driver to the project, right click the project > add > existing item:



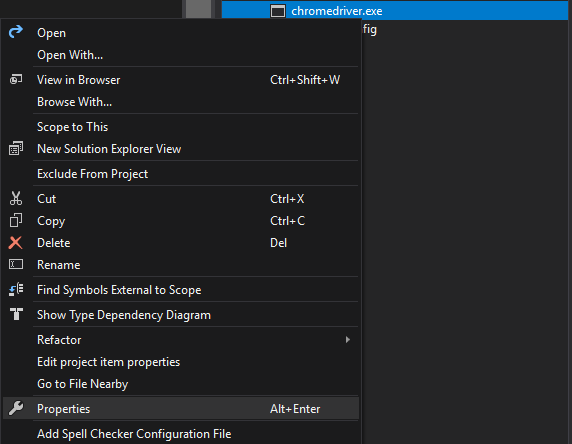
1. Navigate to the directory with the Chrome Driver, change the file type dropdown to “All Files”, select chromedriver.exe and click add:



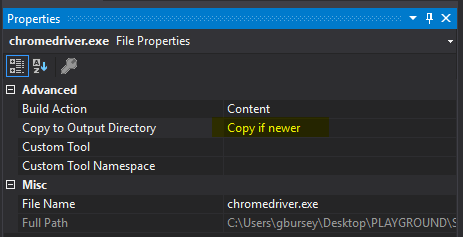
1. Verify that the chromedriver.exe has been added to the project:



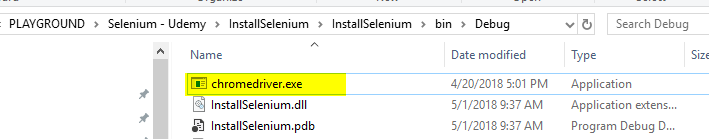
1. Right click the chromedriver.exe > properties:



1. Change Copy to Output Directory to “Copy if newer” (upon build - this will only copy the .exe if there is a newer version available):

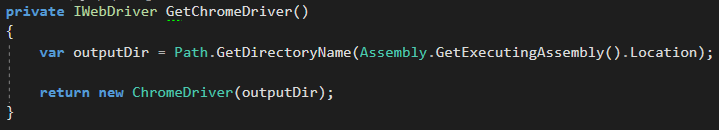


Upon a project build, the dependencies are built in the solution’s /bin/Debug folder:

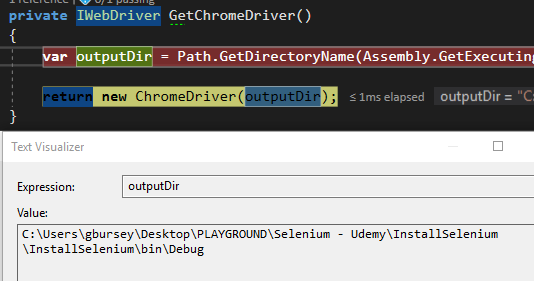


**Dynamically get ChromeDriver path**

Setup a method as follows:



It will find the path to the chromedriver.exe like this:



**Using Statements**

The following using statements are necessary:

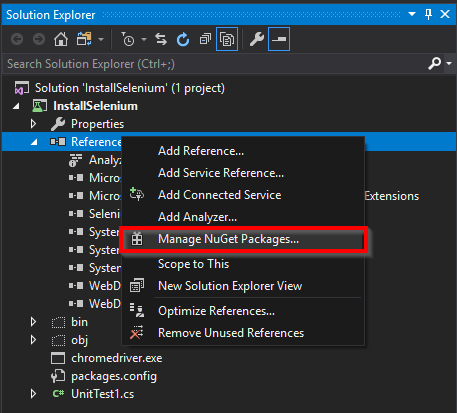
* using System.IO;
* using System.Reflection;
* using Microsoft.VisualStudio.TestTools.UnitTesting;
* using OpenQA.Selenium;
* using OpenQA.Selenium.Chrome;

**Selenium Waits**

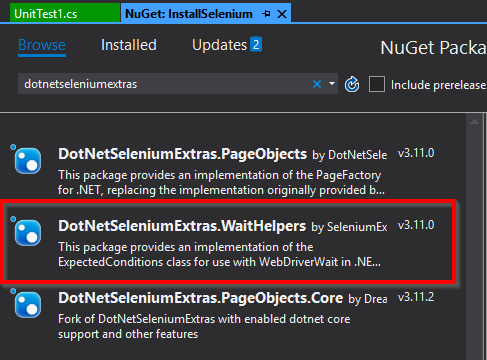
Because Selenium is [choosing to deprecate](https://www.ultimateqa.com/selenium-3-11-released/) it’s Explicit Wait package, ExpectedConditions.cs – if we want to use Selenium’s ExpectedConditions, we will need the following.

***\*NOTE\**** - there is no active maintainer of this repository. So we should not heavily rely on this and really only use it when necessary.

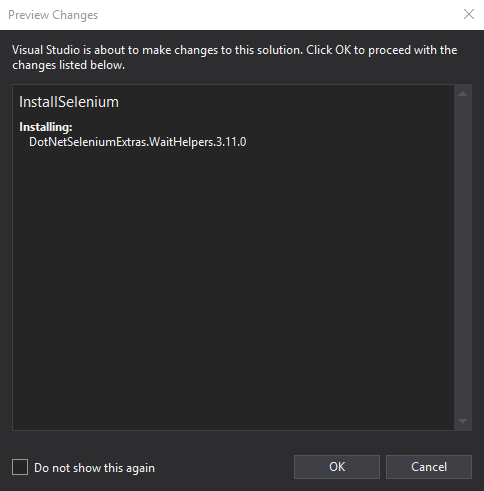
In the desired project, right click References > Manage NuGet packages:



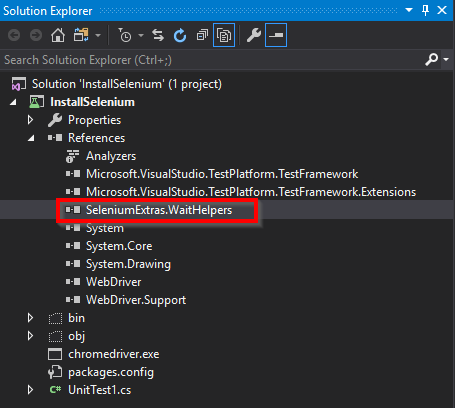
Go to the Browse tab and search for “*dotnetseleniumextras*”:



Install the “*DotNetSeleniumExtras.WaitHelpers*”:



Verify the package has been installed:



The following using statement must be imported:

using SeleniumExtras.WaitHelpers;

using ExpectedConditions = SeleniumExtras.WaitHelpers.ExpectedConditions;

asdf