National University of Sciences and Technology (NUST)
School of Electrical Engineering and Computer Science

Department of Computing

CS312: Advanced Programming

Lab 1: Software Installation

CLO3: Installation of different software that are used to develop applications

Date: September 06, 2018

Time: Thursday (10:00 - 12:50 & 14:00 – 16:50)

Instructor: Abid Rauf
Lab Engineer: Ayesha Asif

Lab 1: Software Installation

Introduction

In this lab we will go through the steps for installation of different software/frameworks like JDK, Eclipse, Python, Kotlin, Visual Studio and some requirements for their installation, in order to develop application in these frameworks.

Objectives

Installation of different frameworks and understanding how to use these tools.

Tools/Software Requirement

None

Description

Each student must, individually install the complete application on their own systems. Students must upload their completion task report on LMS to qualify for evaluation. The following reasons can lead to losing marks

- Any exceptions or errors in software or missing files that leading to non-execution of running software / framework.
- Failure to upload the report on LMS.
- Failure to explain the installation steps or submission, during viva.

Lab Task 1:

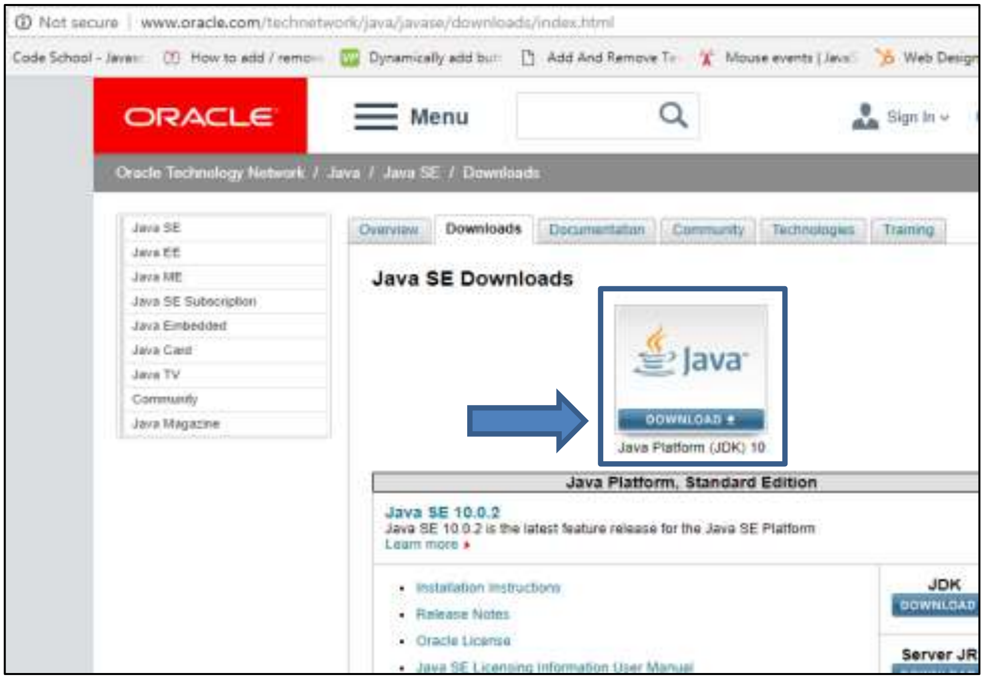
Following are the installation steps for JDK:

Step 0: Un-Install Older Version(s) of JDK/JRE

1. Recommendation that you install only the *latest* JDK. Although you can install multiple versions of JDK/JRE concurrently, it is messy.
2. If you have previously installed older version(s) of JDK/JRE, un-install ALL of them. Goto "Control Panel" ⇒ "Programs" ⇒ "Programs and Features" ⇒ Un-install ALL programs begin with "Java", such as "Java SE Development Kit ...", "Java SE Runtime ...", "Java X Update ...", etc.

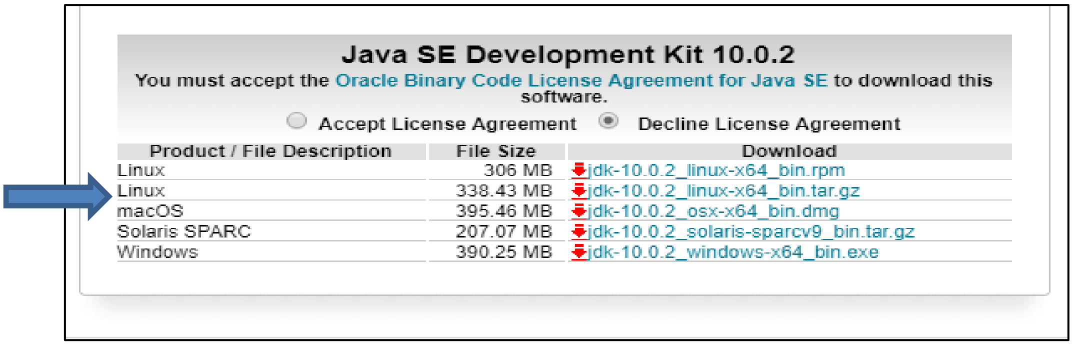
Step 1: Download JDK

1. Goto Java SE download site
   @ http://www.oracle.com/technetwork/java/javase/downloads/index.html.


Not secure | www.oracle.com/technetwork/java/javase/downloads/index.html
Code School - Javas | How to add / remo | Dynamically add but | Add And Remove Te | Mouse events | Java | Web Design
ORACLE
Menu
Sign In
Oracle Technology Network / Java / Java SE / Downloads
Java SE
Java EE
Java ME
Java SE Subscription
Java Embedded
Java Card
Java TV
Community
Java Magazine
Overview
Downloads
Documentation
Community
Technologies
Training
Java SE Downloads
Java
DOWNLOAD
Java Platform (JDK) 10
Java Platform, Standard Edition
Java SE 10.0.2
Learn more
Installation Instructions
Release Notes
Oracle License
Java SE Licensing Information User Manual
JDK
DOWNLOAD
Server JR

Java SE 10.0.2 is the latest feature release for the Java SE Platform

2. Under "Java Platform, Standard Edition" ⇒ "Java SE 10.0.{x}", where {x} denotes a fast running update number ⇒ Click the JDK's "Download" button.
3. Under "Java SE Development Kit 10.0.{x}" ⇒ Check "Accept License Agreement".
4. Choose the JDK for your operating system, i.e., "Windows" (for 64-bit Windows OS), and download the installer (e.g., "jdk-10.0.{x}_windows-x64_bin.exe" - 390MB).



Java SE Development Kit 10.0.2

You must accept the Oracle Binary Code License Agreement for Java SE to download this software.

○ Accept License Agreement ● Decline License Agreement

Product / File Description	File Size	Download
Linux	306 MB	jdk-10.0.2_linux-x64_bin.rpm
Linux	338.43 MB	jdk-10.0.2_linux-x64_bin.tar.gz
macOS	395.46 MB	jdk-10.0.2_osx-x64_bin.dmg
Solaris SPARC	207.07 MB	jdk-10.0.2_solaris-sparcv9_bin.tar.gz
Windows	390.25 MB	jdk-10.0.2_windows-x64_bin.exe

Step 2: Install JDK and JRE

Run the downloaded installer (e.g., "jdk-10.0.{x}_windows-x64_bin.exe"), which installs both the JDK and JRE.

By default:

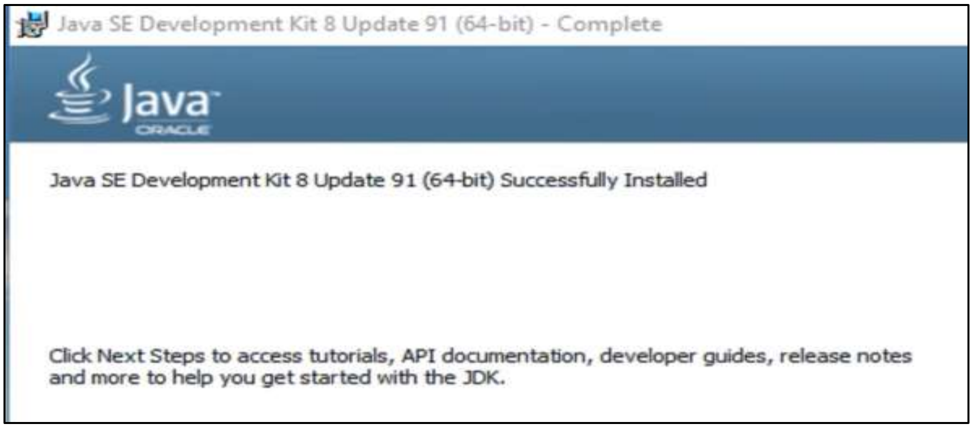
- JDK is installed in directory "C:\Program Files\Java\jdk-10.0. {x}", where {x} denotes the upgrade number; and
- JRE is installed in "C:\Program Files\Java\jre-10.0. {x}".

Notes: In 64-bit Windows, "Program Files" is meant for 64-bit programs; while "Program Files (x86)" for 32-bit programs.

Whereas you can change the default path by clicking on Change button displayed on the window.

Accept the defaults and follow the screen instructions to install JDK and JRE. First JDK is installed then in same installation JRE window will be opened, accept the defaults or change accordingly your path but don't change any other default options.

Click on "close" button in below window. JDK Installed successfully.

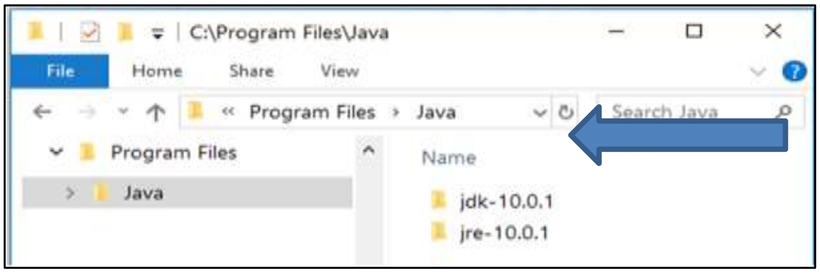


Java SE Development Kit 8 Update 91 (64-bit) - Complete

Java SE Development Kit 8 Update 91 (64-bit) Successfully Installed

Click Next Steps to access tutorials, API documentation, developer guides, release notes and more to help you get started with the JDK.

Use the "File Explorer", goto "C:\Program Files\Java" to inspect these folders. Take note of your **JDK installed directory**, the varying upgrade number, which you will need in the next step. In the following diagram, the JDK installed directory is "C:\Program Files\Java\jdk-10.0.1", where {x}=1.



C:\Program Files\Java

Your version can be different

Step 3: Include JDK's "bin" Directory in the PATH

Windows' Shell searches the current directory and the directories listed in the PATH *environment variable* (*system variable*) for executable programs. JDK's programs (such as Java compiler javac.exe and Java runtime java.exe) reside in the *sub-directory* "bin" of the JDK installed directory. You need to include "bin" in the PATH to run the JDK programs.
To edit the PATH environment variable in Windows 7/8/10:

1. Launch "Control Panel" ⇒ (Optional) System and Security ⇒ System ⇒ Click "Advanced system settings" on the left pane.
2. Switch to "Advanced" tab ⇒ Push "Environment Variables" button.
3. Under "System Variables" (the bottom pane), scroll down to select "Path" ⇒ Click "Edit...".
4. **For Windows 10 (newer releases)**:
   You shall see a **TABLE** listing all the existing PATH entries (if not, goto next step). Click "New" ⇒ Enter the JDK's "bin" directory "c:\Program Files\Java\jdk-10.0.{x}\bin" (Replace {x} with your installation number!!!) ⇒ Select "Move Up" to move this entry all the way to the TOP.

Prior to newer Windows 10:

(CAUTION: Read this paragraph 3 times before doing this step! Don't push "Apply" or "OK" until you are 101% sure. There is no UNDO!!!)
(To be SAFE, copy the content of the "Variable value" to Notepad before changing it!!!)
In "Variable value" field, INSERT "c:\Program Files\Java\jdk-10.0.{x}\bin" (Replace {x} with your installation number!!!) IN FRONT of all the existing directories, followed by a semi-colon (;)which separates the JDK's bin directory from the rest of the existing directories. DO NOT DELETE any existing entries; otherwise, some existing applications may not run.

5. Variable name: PATH
   Variable name: **c:\Program Files\Java\jdk-10.0.{x}\bin;[*do not delete exiting entries...*]**

Notes: Starting from JDK 1.8, the installation created a directory "c:\ProgramData\Oracle\Java\javapath" and added to the PATH. It contains only JRE executables (java.exe, javaw.exe, and javaws.exe), but NOT the JDK executables (e.g., javac.exe).

Step 4: Verify the JDK Installation

Launch a CMD shell via one of the following means:

1. Click "Search" button ⇒ Enter "cmd" ⇒ Choose "Command Prompt", or
2. right-click "Start" button ⇒ run... ⇒ enter "cmd", or
3. (Prior to Windows 10) click "Start" button ⇒ All Programs ⇒ Accessories (or Windows System) ⇒ Command Prompt, or

4. (Windows 10) click "Start" button ⇒ Windows System ⇒ Command Prompt.

Issue the following commands to verify your JDK installation:

1. Issue "path" command to list the contents of the PATH environment variable. Check to make sure that your <JAVA_HOME>\bin is listed in the PATH. Don't type prompt>, which denotes the command prompt!!! Key in the command (highlighted) only.
2. // Display the PATH entries
3. prompt> path
   PATH=c:\Program Files\Java\jdk-10.0.{x}\bin;[other entries...]
4. Issue the following commands to verify that JDK/JRE are properly installed and display their version:
5. // Display the JRE version
6. prompt> java -version
7. Java version "10.0.{x}" 2018-04-17
8. Java(TM) SE Runtime Environment 18.3 (build 10.0.1+10)
9. Java HotSpot(TM) 64-Bit Server VM 18.3 (build 10.0.1+10, mixed mode)
10. // Display the JDK version
11. prompt> javac -version
    javac 10.0.{x}

Lab Task 2:

Following are the installation steps for Eclipse 4.8 (Photon):

Step 0: Install JDK

To use Eclipse for Java programming, you need to first install Java Development Kit (JDK). Read "How to Install JDK for Windows".(Lab Task 1)

Step 1: Download

Download Eclipse from https://www.eclipse.org/downloads. Under "Get Eclipse Photon" ⇒ Click "Download Packages". For beginners, choose the 4th entry "**Eclipse IDE for Java Developers**" and "**Windows 64-bit**" (e.g., "eclipse-java-photon-R-win32-x86_64.zip" 195MB) ⇒ Download.

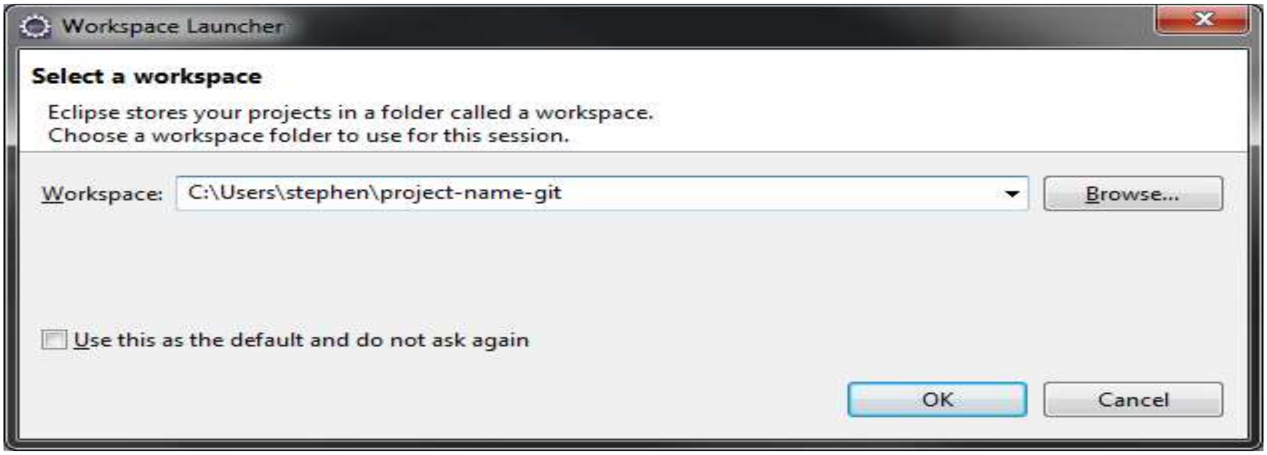
Step 2: Unzip

To install Eclipse, simply unzip the downloaded file into a directory of your choice (e.g., "c:\myproject").

There is no need to run any installer. Moreover, you can simply delete the entire Eclipse directory when it is no longer needed (without running any un-installer). You are free to move or rename the directory. You can install (unzip) multiple copies of Eclipse in the same machine.

Step 3: Create new Workspace

Created a brand-new workspace.



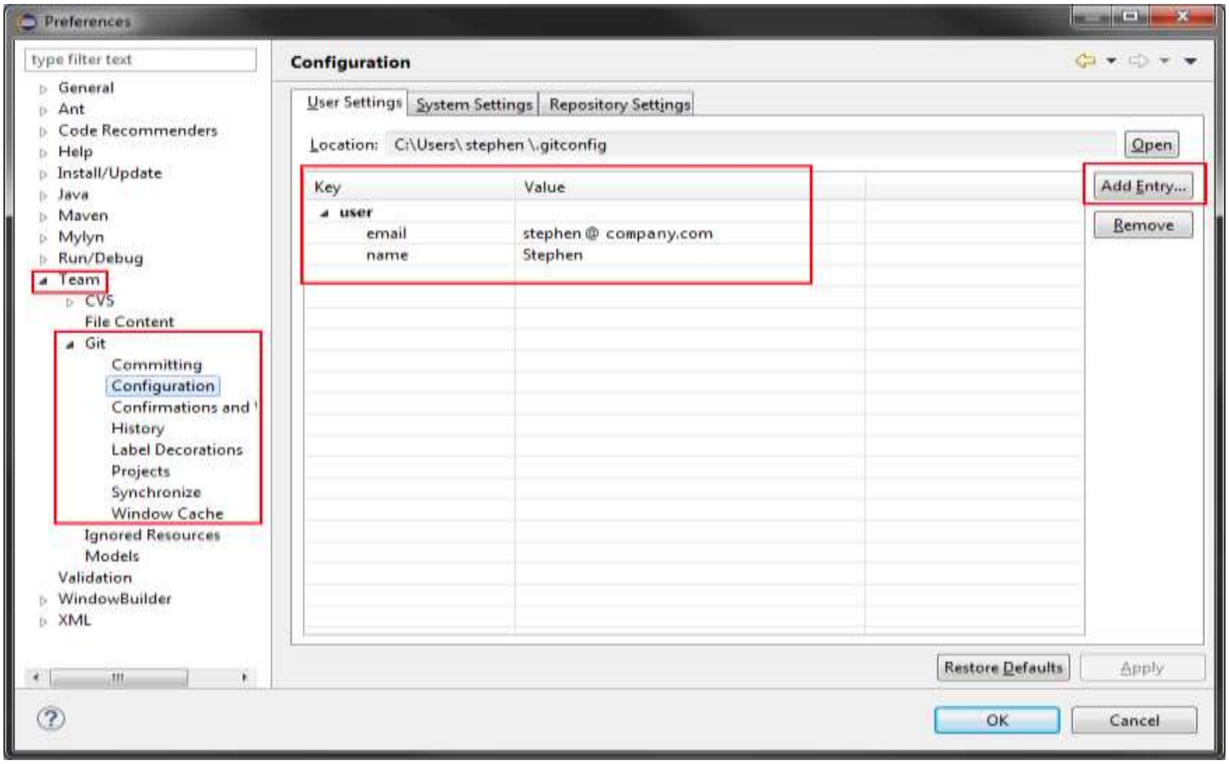
Workspace Launcher

Select a workspace

Eclipse stores your projects in a folder called a workspace.
Choose a workspace folder to use for this session.

Workspace: C:\Users\stephen\project-name-git

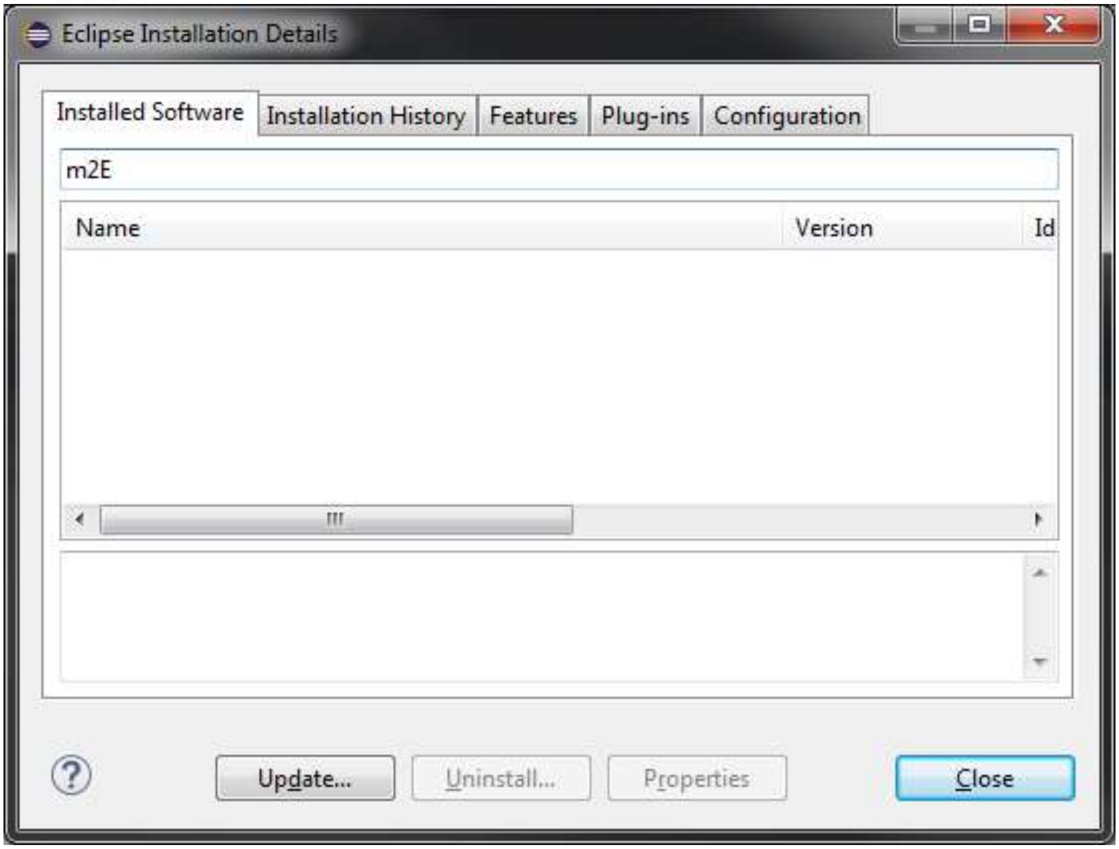
Then **(Window -> Preferences)** and saw that it recognized some settings for Git that you must have configured in a previous lifetime. If you haven't done this, just add some entries for your user settings (or System settings).



Location: C:\Users\stephen\.gitconfig

Key	Value
user	
email	stephen@company.com
name	Stephen

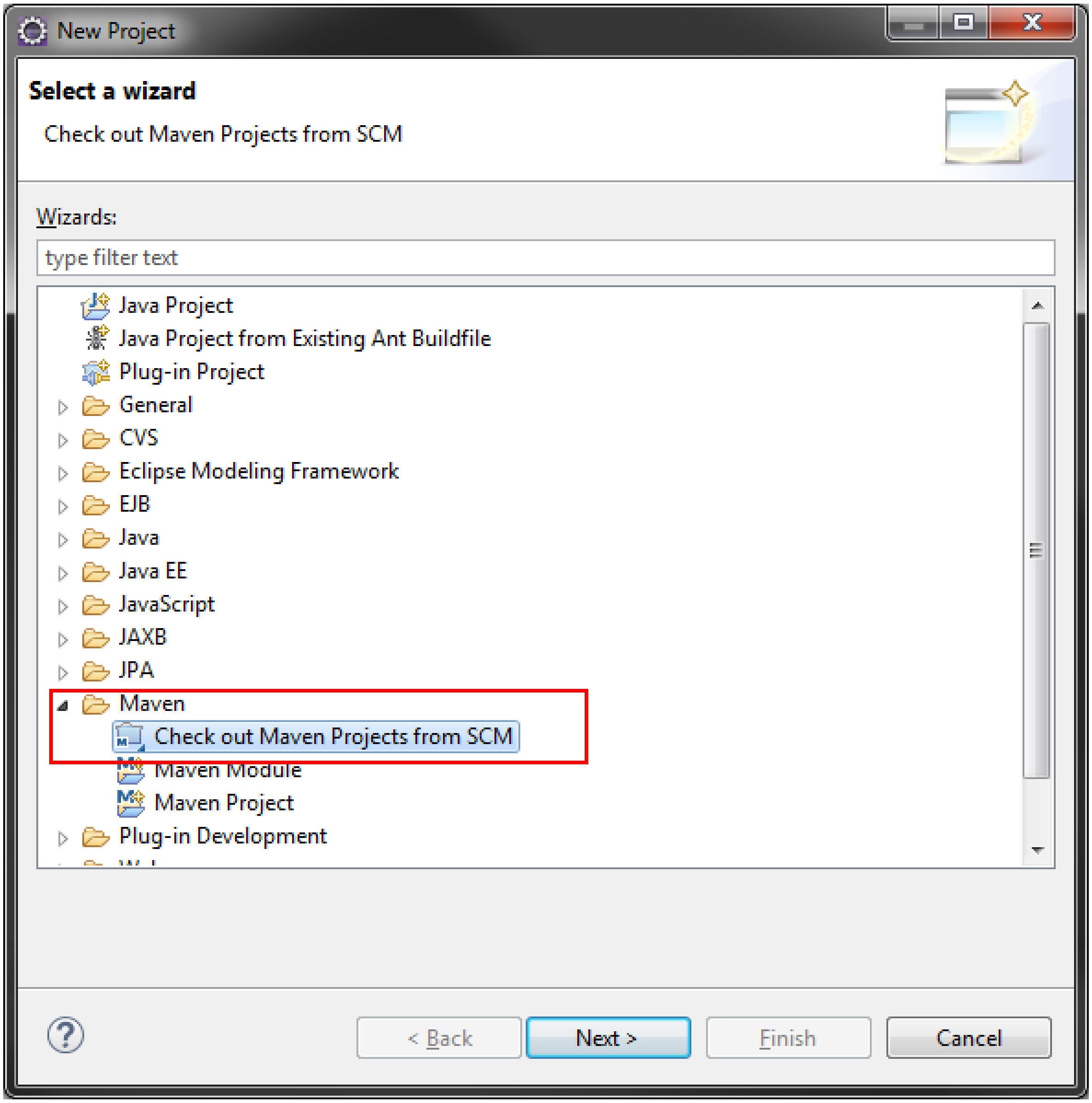
Eclipse has a handy feature called Installation Details and can be accessed by **Help -> Installation Details**



m2E

If Maven not present there you can try two options: you can install this plugin. Or you should just try a different Eclipse package, perhaps the 'Eclipse IDE for Java EE Developers' as that looks to include a bunch of Maven integration tools.

Checked the Maven settings (**Window -> Preferences** again) and saw that it picked up on my user settings.
Create a new project. Right-clicked on the empty Project Explorer view. New -> Project... -> Maven -> Check out Maven Projects from SCM.



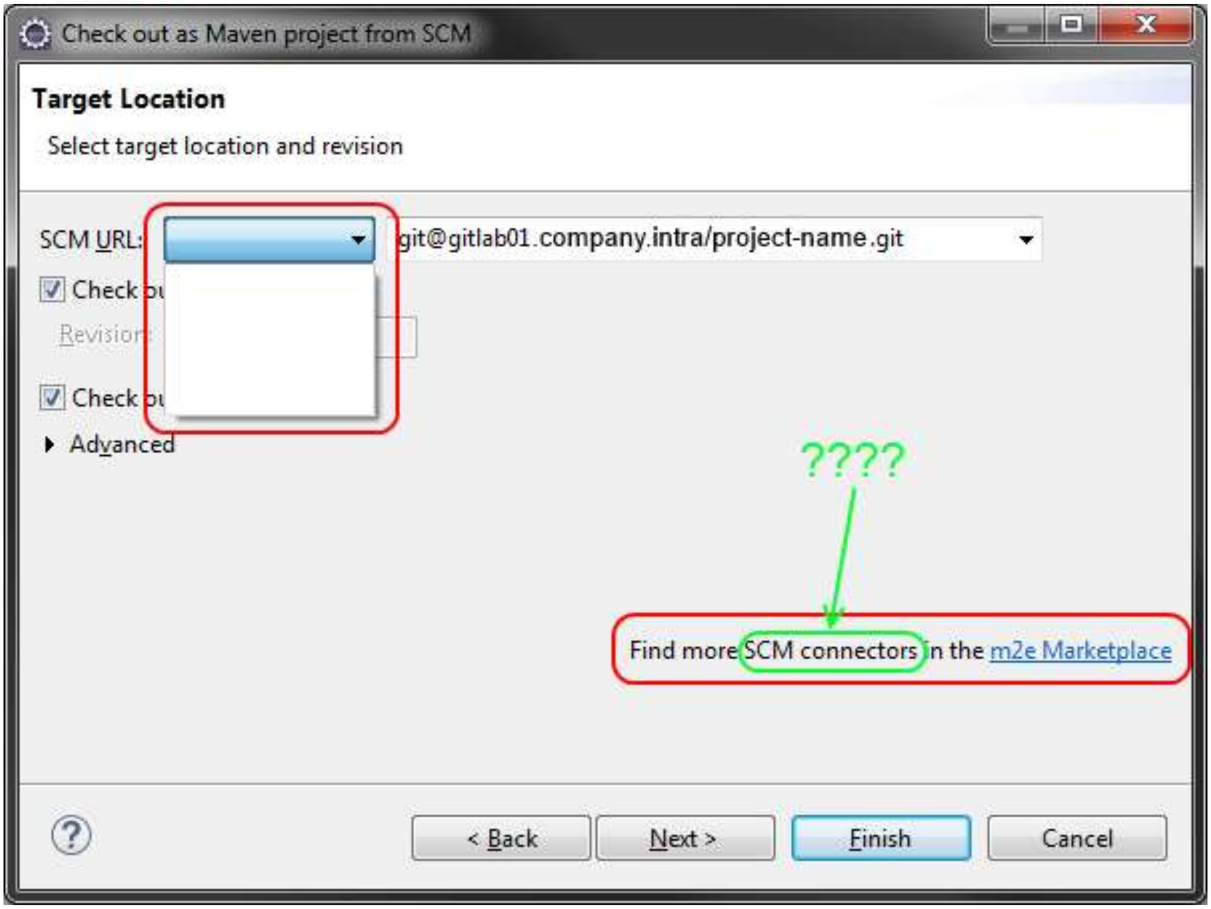
New Project

Select a wizard

Check out Maven Projects from SCM

Wizards:

Enter in the URL from gitlab.



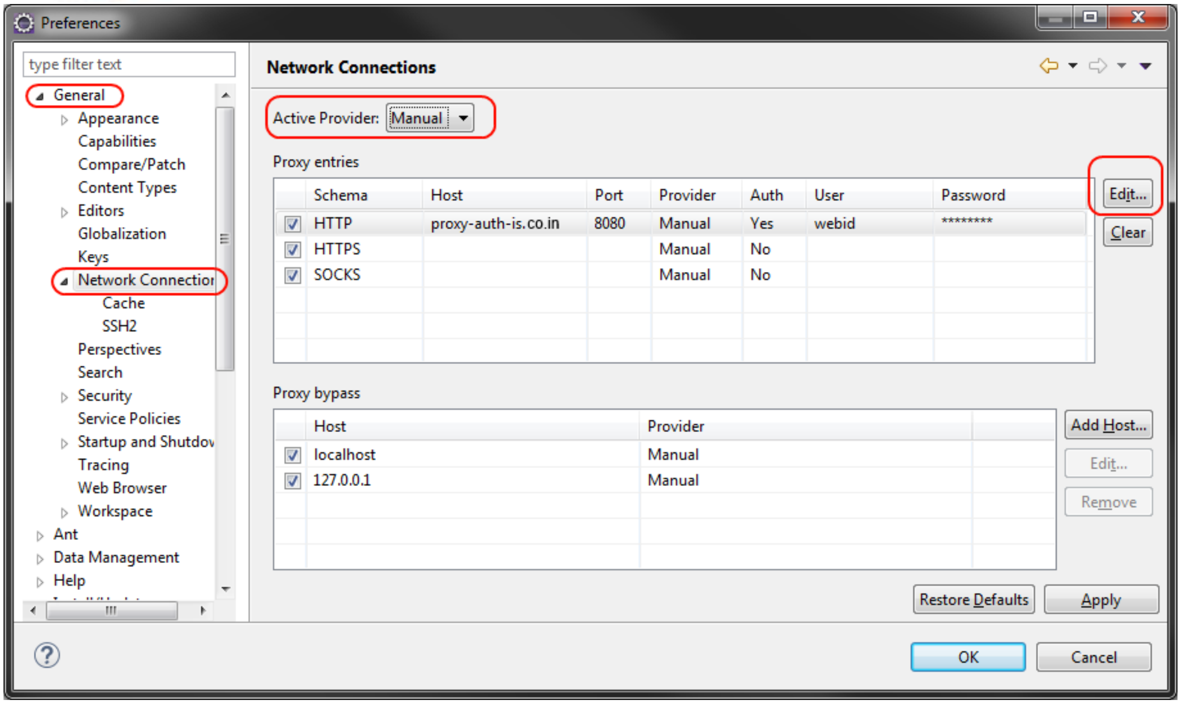
git@gitlab01.company.intra/project-name.git

Find more SCM connectors in the m2e Marketplace

Click Next. At the next screen, accepted the default settings and clicked Finish.

No projects or folders in the Project Explorer. Waite a moment. Maybe there's debugging info in the console. Open that up. Window -> Show View -> Console.

Find the proxy settings:



Network Connections

Active Provider: Manual

Proxy entries

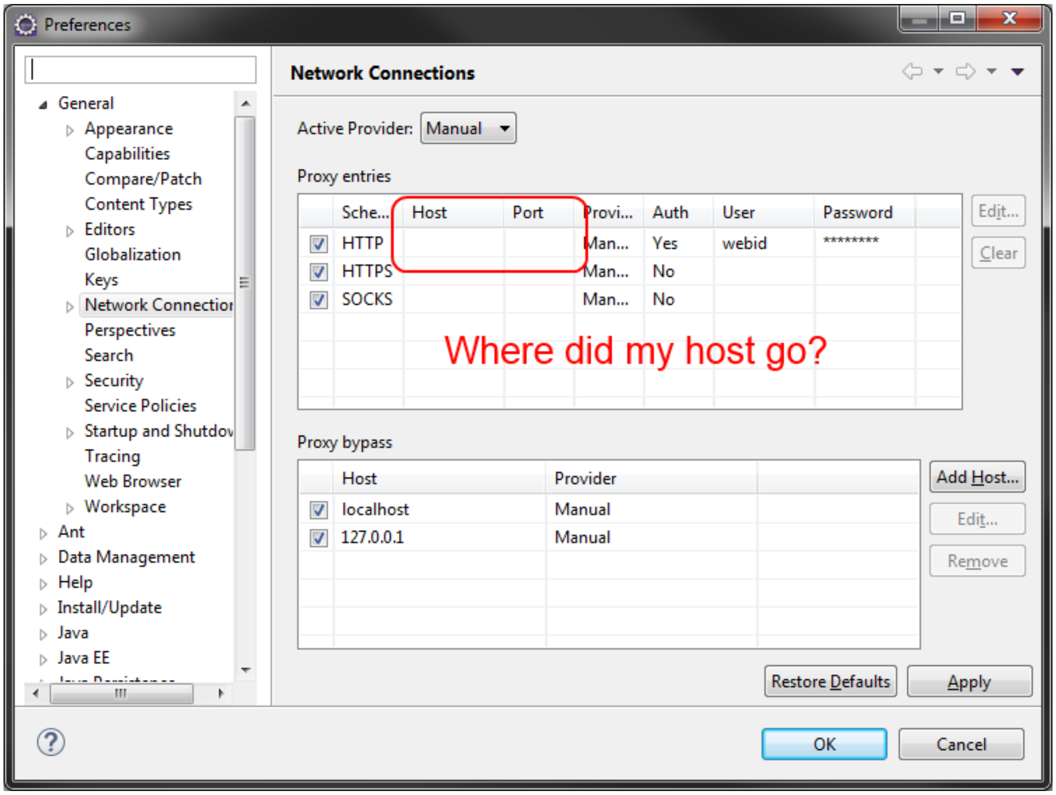
	Schema	Host	Port	Provider	Auth	User	Password
☑	HTTP	proxy-auth-is.co.in	8080	Manual	Yes	webid	********
☑	HTTPS			Manual	No		
☑	SOCKS			Manual	No		

Proxy bypass

	Host	Provider
☑	localhost	Manual
☑	127.0.0.1	Manual

Hit apply and close it. Right-clicked on the empty Project Explorer view again. New -> Project... -> Maven -> Check out Maven Projects from SCM. Click on the m2e Marketplace link.

If Nothing happens. Go back to the Network Connections preference. The Host and Port are empty, but the User and Password are filled.



Network Connections

Active Provider: Manual

Proxy entries

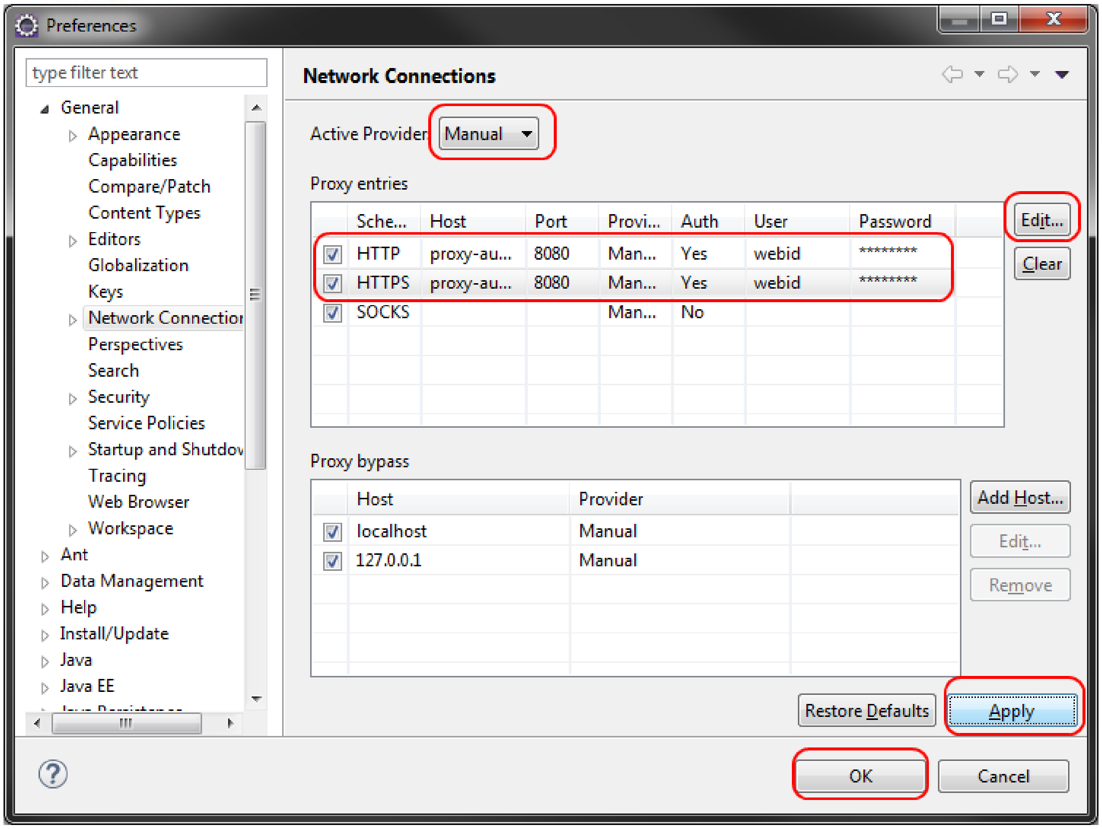
	Sche...	Host	Port	Provi...	Auth	User	Password
☑	HTTP			Man...	Yes	webid	********
☑	HTTPS			Man...	No		
☑	SOCKS			Man...	No		

Where did my host go?

Proxy bypass

	Host	Provider
☑	localhost	Manual
☑	127.0.0.1	Manual

Try setting BOTH the HTTP and HTTPS proxy to use the same thing. Click Apply and OK again.



Network Connections

Active Provider: Manual

Proxy entries

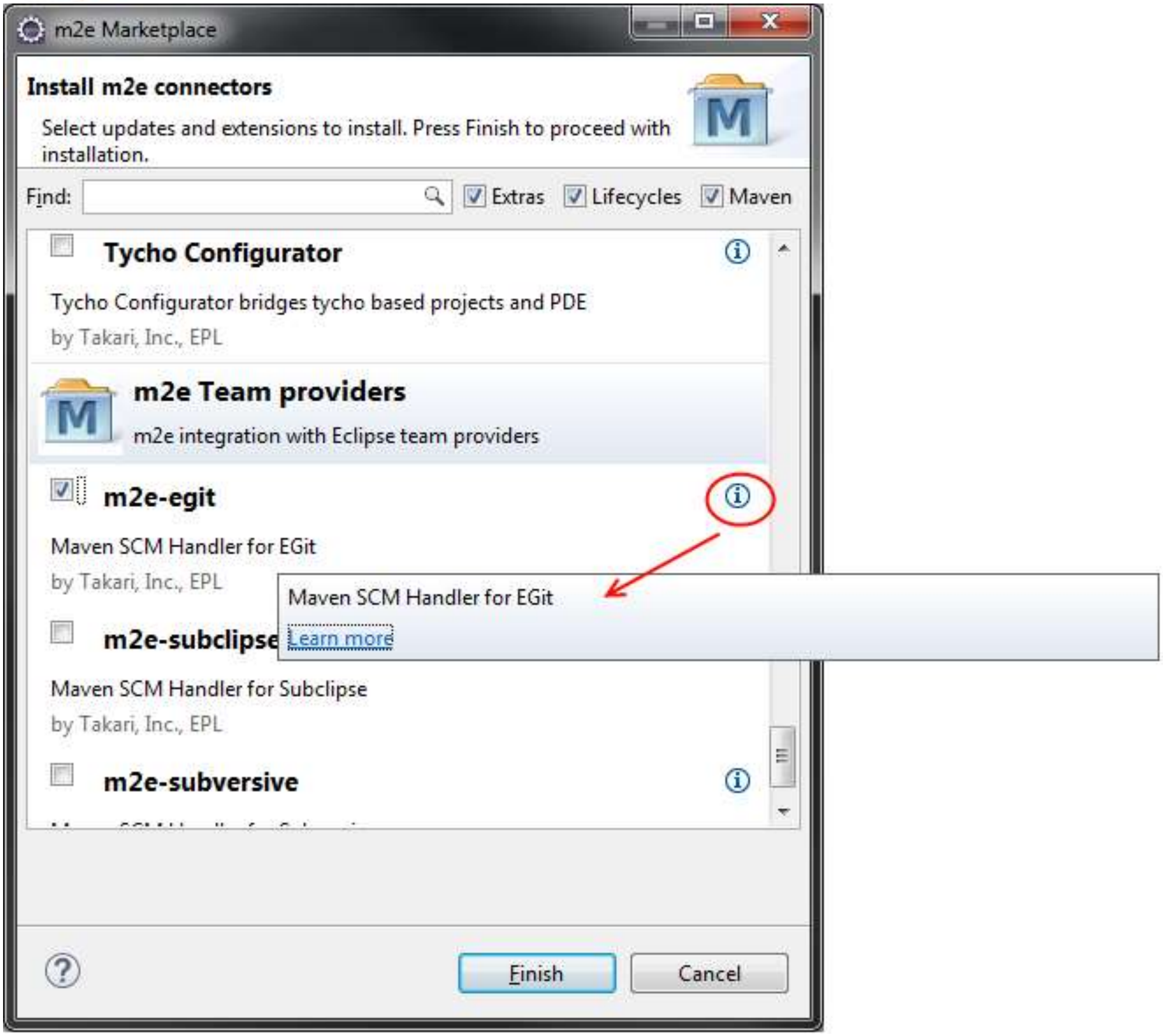
	Sche...	Host	Port	Provi...	Auth	User	Password
☑	HTTP	proxy-au...	8080	Man...	Yes	webid	********
☑	HTTPS	proxy-au...	8080	Man...	Yes	webid	********
☑	SOCKS			Man...	No		

Proxy bypass

	Host	Provider
☑	localhost	Manual
☑	127.0.0.1	Manual

Right-clicked on the empty Project Explorer view again. New -> Project... -> Maven -> Check out Maven Projects from SCM. Click on the m2e Marketplace link.

EGit is the name of the Git plugin for Eclipse.



Install m2e connectors

Select updates and extensions to install. Press Finish to proceed with installation.

Find: ☑ Extras ☑ Lifecycles ☑ Maven

□ **Tycho Configurator**

Tycho Configurator bridges tycho based projects and PDE

by Takari, Inc., EPL

m2e Team providers

m2e integration with Eclipse team providers

☑ **m2e-egit**

Maven SCM Handler for EGit

by Takari, Inc., EPL

Maven SCM Handler for EGit

Learn more

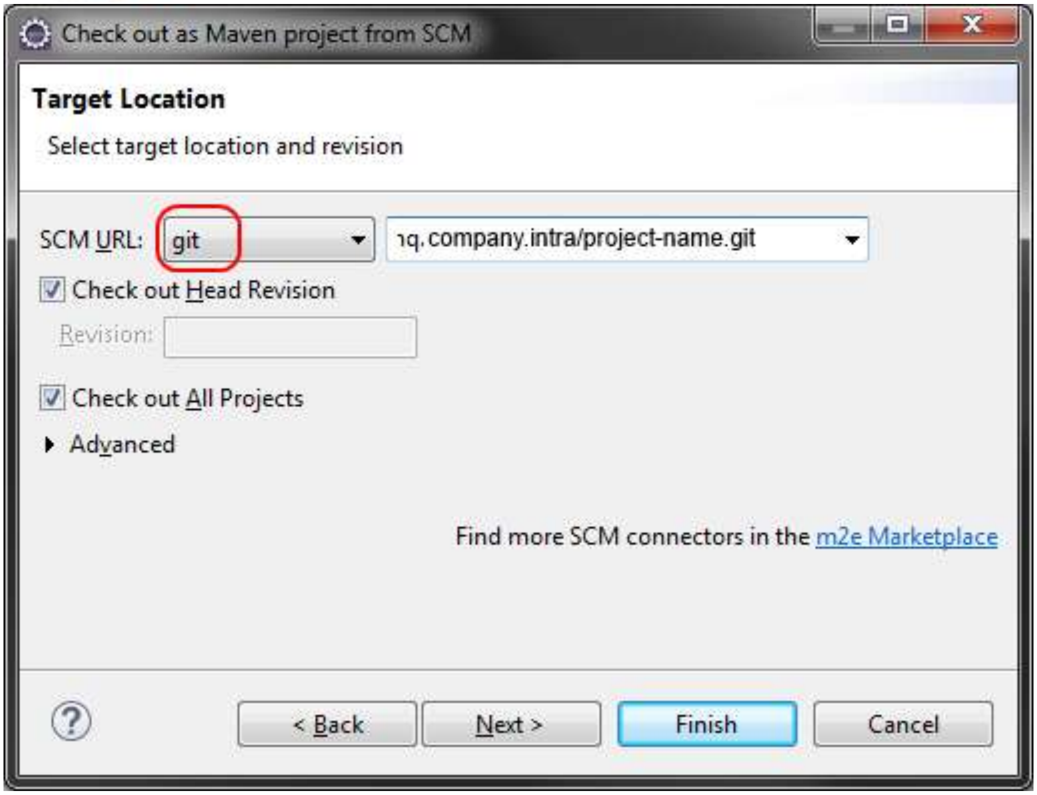
□ **m2e-subclipse**

Maven SCM Handler for Subclipse

by Takari, Inc., EPL

□ **m2e-subversive**

Right-clicked on the empty Project Explorer view again. New -> Project... -> Maven -> Check out Maven Projects from SCM.



Target Location

Select target location and revision

SCM URL: git ▾ ٦q.company.intra/project-name.git ▾

☑ Check out Head Revision

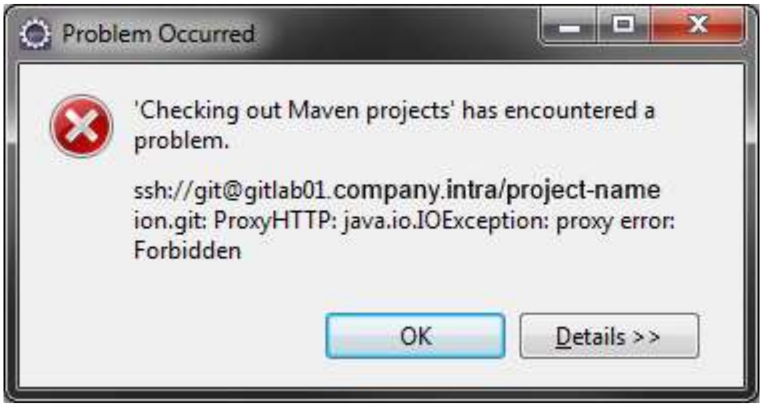
Revision:

☑ Check out All Projects

▸ Advanced

Find more SCM connectors in the m2e Marketplace

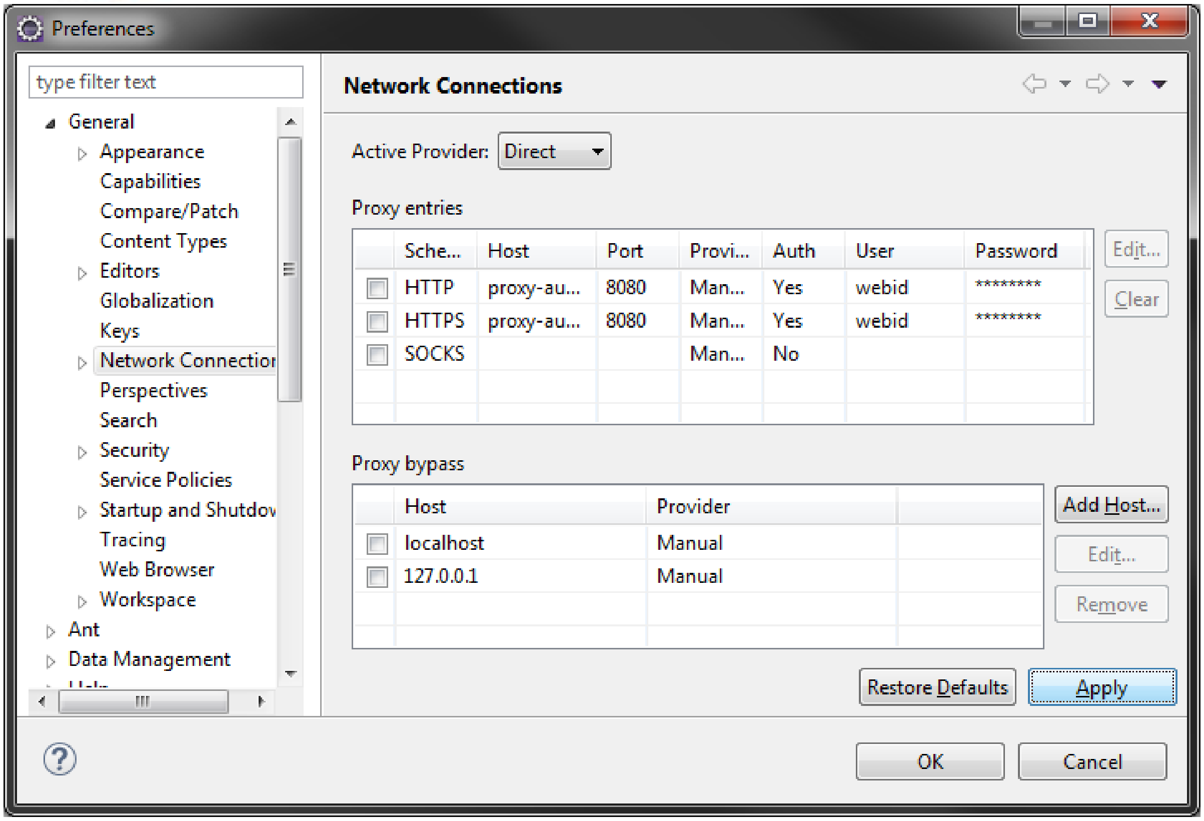
Turns out that blank field. Click through these screens.



'Checking out Maven projects' has encountered a problem.

ssh://git@gitlab01.company.intra/project-name ion.git: ProxyHTTP: java.io.IOException: proxy error: Forbidden

Looks like a proxy error, probably related to not having turned off the proxy, disabled the proxy for internal addresses (and this host is clearly an internal one). Both SSH/Git protocols should not be operating over HTTP. Proxy was only for HTTP and HTTPS. But change those settings again.



Network Connections

Active Provider: Direct

Proxy entries

	Sche...	Host	Port	Provi...	Auth	User	Password
□	HTTP	proxy-au...	8080	Man...	Yes	webid	********
□	HTTPS	proxy-au...	8080	Man...	Yes	webid	********
□	SOCKS			Man...	No		

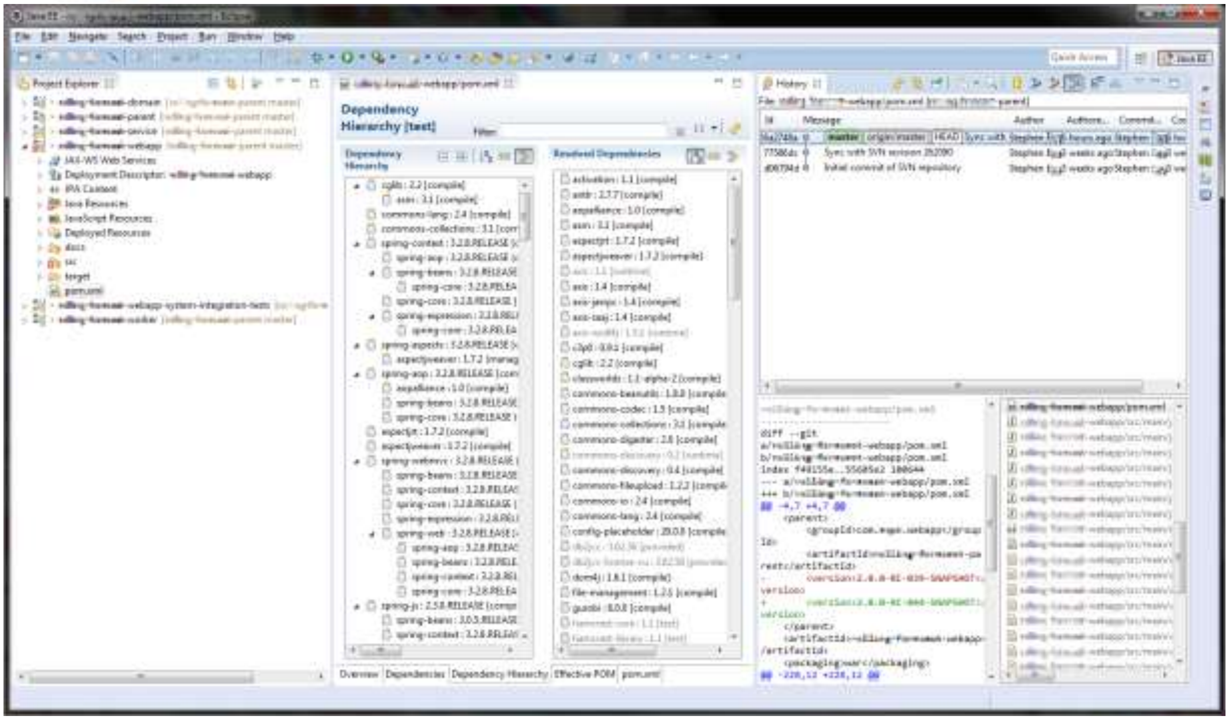
Proxy bypass

	Host	Provider
□	localhost	Manual
□	127.0.0.1	Manual

The proxy settings ignoring internal hosts correctly. Just clicked the proxy settings to say Direct and hit Apply then OK.

Right-clicked on the empty Project Explorer view again. New -> Project... -> Maven -> Check out Maven Projects from SCM. I click through these screens.

After a few minutes of copying the repository and building the project, everything is there. Maven recognizes it and the Git plugin recognizes it!


File Edit Navigate Search Project Run Window Help
Project Explorer
Dependency Hierarchy [test]
Filter:
Dependency Hierarchy
Resolved Dependencies
History
Overview Dependencies Dependency Hierarchy Effective POM pom.xml

Creating new workspaces. Use the second, not accepted answer here to transfer your settings over quickly. Using a package manager / build tool to be able to checkout a project from scratch, resolve dependencies, and get it running in 5 minutes or less.

Lab Task 3:

Following are the installation steps for Kotlin:

Step 1: Make sure your computer is ready and Android studio is installed on your computer.

Step 2 - Download latest version @ https://developer.android.com/studio/preview/

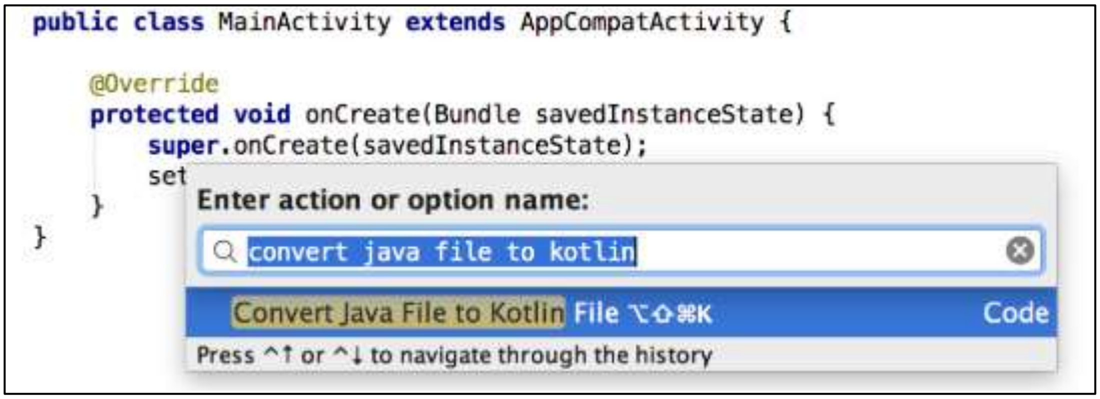
The Kotlin plugin is bundled with Android Studio starting from version 3.0. If you use an earlier version, you'll need to install the Kotlin plugin.

Step 3 - If you are looking at the "Welcome to Android Studio" screen, choose Configure | Plugins | Install JetBrains plugin…

Step 4 - Converting Java code to Kotlin

Open MainActivity.java file. Then invoke action **Convert Java File to Kotlin File**. You can do it by several ways. The easiest one is to invoke Find Action and start typing an action name (like in a screencast below).

Alternatively, you can call this option via the *Code* | *Convert Java File to Kotlin File* menu entry or by using the corresponding shortcut (you can find it at the menu entry).



```
public class MainActivity extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        set
    }
}
```

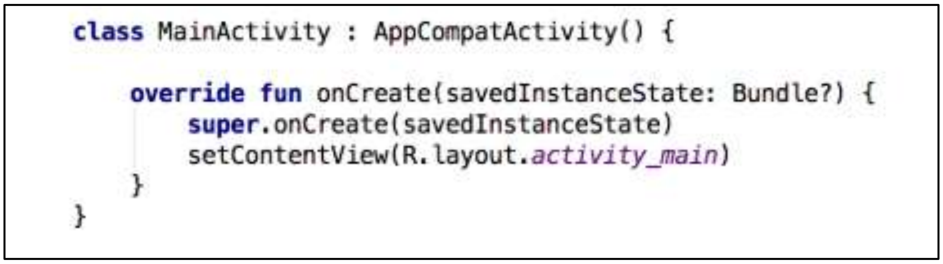
Enter action or option name:

convert java file to kotlin

Convert Java File to Kotlin File ⌥⇧⌘K Code

Press ^↑ or ^↓ to navigate through the history

Step 5 - After the conversion you should have an activity written in Kotlin.

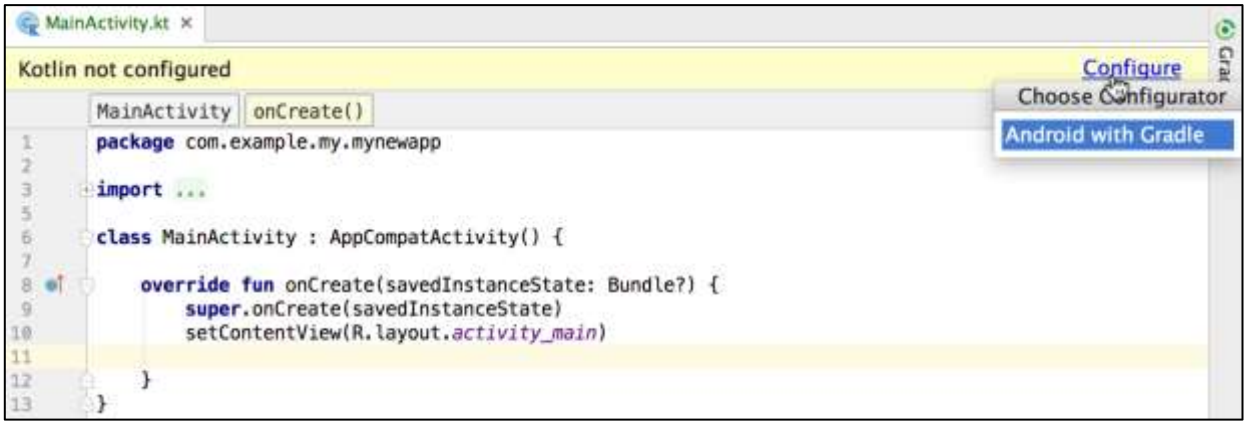


```
class MainActivity : AppCompatActivity() {

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)
    }
}
```

Step 6 - Configuring Kotlin in the project

If you start editing this file, Android Studio shows you a prompt that Kotlin is not configured, so you can configure it. Alternatively, you can invoke the configuration by selecting Tools | Kotlin | Configure Kotlin in Project from the main menu.

Alternatively, you can invoke the configuration by selecting Tools | Kotlin | Configure Kotlin in Project from the main menu.

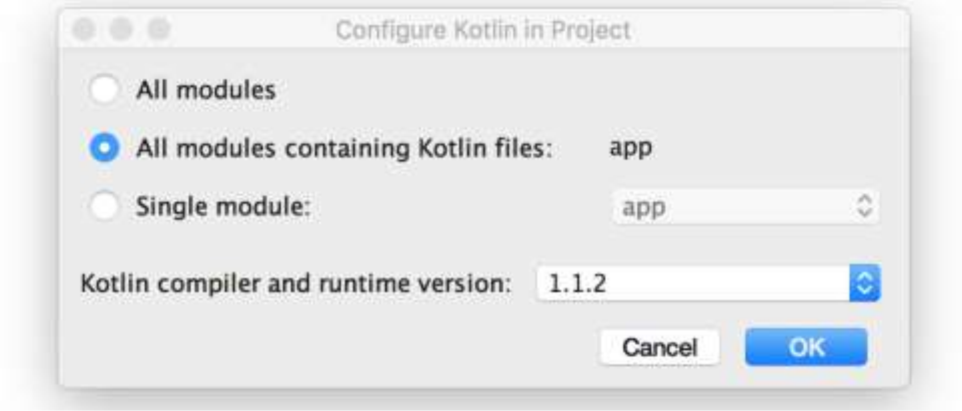


Kotlin not configured

```
package com.example.my.mynewapp

import ...

class MainActivity : AppCompatActivity() {

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)

    }
}
```

You are then prompted for the version of Kotlin.

Step 7- Choose the latest available from the list of installed versions.



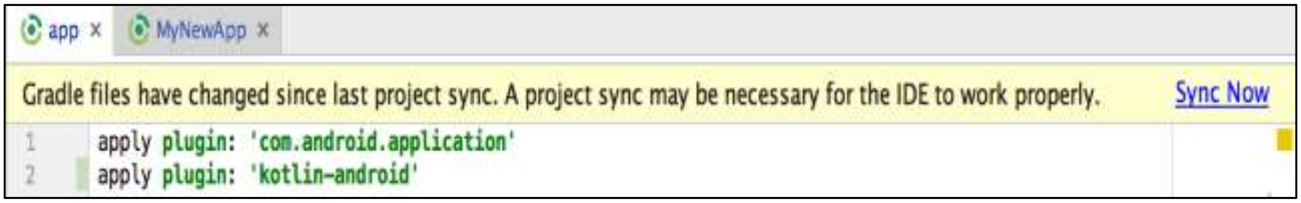
○ All modules

● All modules containing Kotlin files: app

○ Single module: app

Kotlin compiler and runtime version: 1.1.2

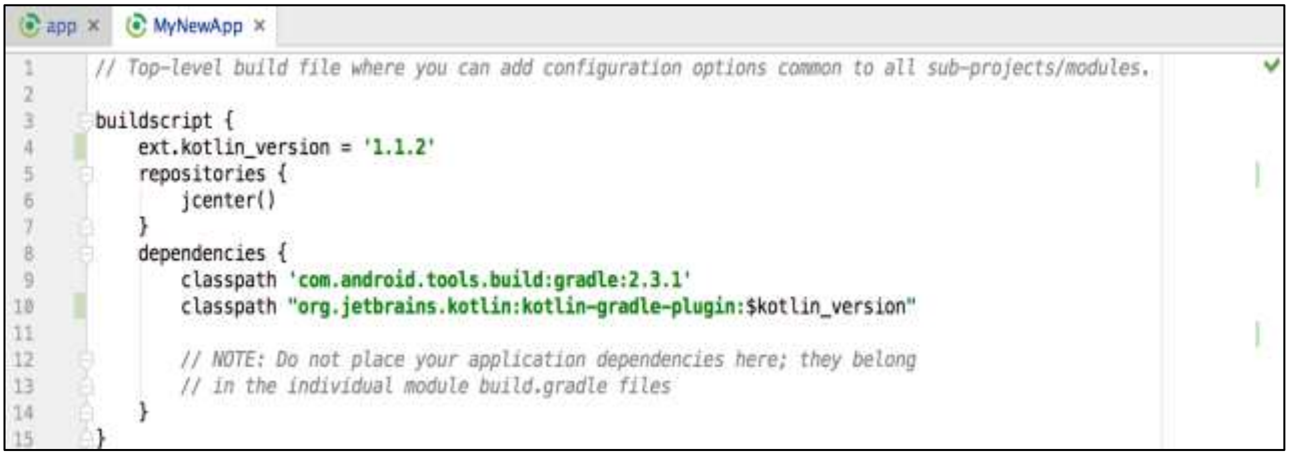
Step 8- After you configure Kotlin, build.gradle file for the application should be updated. Now you can see that *apply plugin: 'kotlin-android'* and the kotlin-stdlib dependency were added.



Gradle files have changed since last project sync. A project sync may be necessary for the IDE to work properly.

```
apply plugin: 'com.android.application'
apply plugin: 'kotlin-android'
```

Step 9- The last thing to do is to sync the project. You can press 'Sync Now' in a prompt or invoke an action **Sync Project with Gradle Files**.



```
// Top-level build file where you can add configuration options common to all sub-projects/modules.

buildscript {
    ext.kotlin_version = '1.1.2'
    repositories {
        jcenter()
    }
    dependencies {
        classpath 'com.android.tools.build:gradle:2.3.1'
        classpath "org.jetbrains.kotlin:kotlin-gradle-plugin:$kotlin_version"

        // NOTE: Do not place your application dependencies here; they belong
        // in the individual module build.gradle files
    }
}
```

Step 10- Building and publishing the Kotlin application for Android

Lab Task 4:

Following are the installation steps for Visual Studio:

Step 1 - Make sure your computer is ready for Visual Studio

Before you begin installing Visual Studio:

1. Check the system requirements. These requirements help you know whether your computer supports Visual Studio 2017.
2. Apply the latest Windows updates. These updates ensure that your computer has both the latest security updates and the required system components for Visual Studio.
3. Reboot. The reboot ensures that any pending installs or updates don't hinder the Visual Studio install.
4. Free up space. Remove unneeded files and applications from your %SystemDrive% by, for example, running the Disk Cleanup app.

Step 2 - Download Visual Studio

Next, download the Visual Studio bootstrapper file. To do so, click the following button, select the edition of Visual Studio 2017 that you want, click **Save**, and then click **Open folder**.

Download Visual Studio 2017

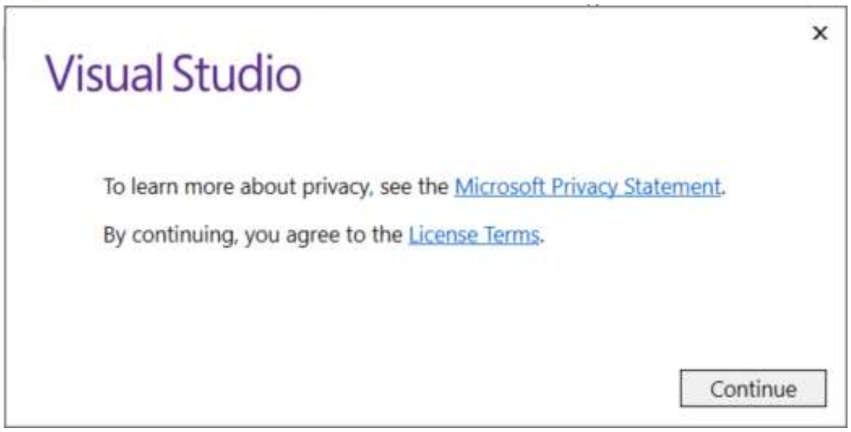
Step 3 - Install the Visual Studio installer

Then, run the bootstrapper file to install the Visual Studio Installer. This new lightweight installer includes everything you need to both install and customize Visual Studio 2017.

1. From your **Downloads** folder, double-click the bootstrapper that matches or is similar to one of the following files:
   - **vs_enterprise.exe** for Visual Studio Enterprise
   - **vs_professional.exe** for Visual Studio Professional
   - **vs_community.exe** for Visual Studio Community

   If you receive a User Account Control notice, click **Yes**.

2. We'll ask you to acknowledge the Microsoft License Terms and the Microsoft Privacy Statement. Click **Continue**.



Visual Studio

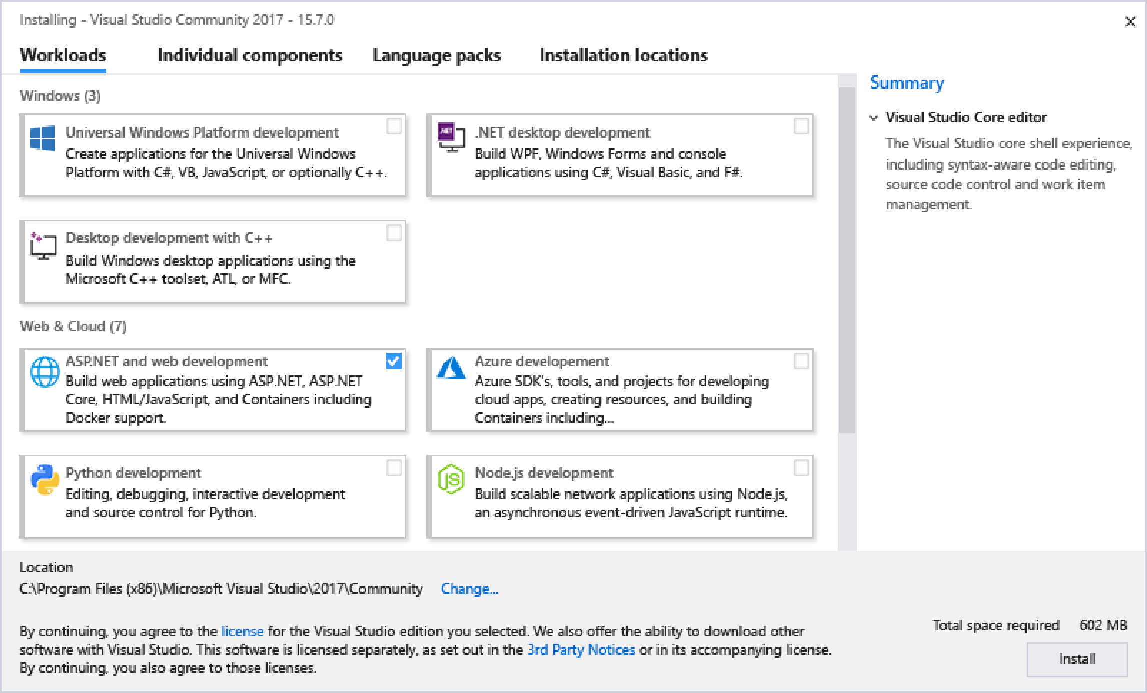
To learn more about privacy, see the Microsoft Privacy Statement.

By continuing, you agree to the License Terms.

Step 4 - Select workloads

After the installer is installed, you can use it to customize your installation by selecting the feature sets—or workloads—that you want. Here's how.

1. Find the workload you want in the **Installing Visual Studio** screen.



Installing - Visual Studio Community 2017 - 15.7.0

Windows (3)

□ Universal Windows Platform development
Create applications for the Universal Windows Platform with C#, VB, JavaScript, or optionally C++.

□ .NET desktop development
Build WPF, Windows Forms and console applications using C#, Visual Basic, and F#.

□ Desktop development with C++
Build Windows desktop applications using the Microsoft C++ toolset, ATL, or MFC.

Web & Cloud (7)

☑ ASP.NET and web development
Build web applications using ASP.NET, ASP.NET Core, HTML/JavaScript, and Containers including Docker support.

□ Azure developement
Azure SDK's, tools, and projects for developing cloud apps, creating resources, and building Containers including...

□ Python development
Editing, debugging, interactive development and source control for Python.

□ Node.js development
Build scalable network applications using Node.js, an asynchronous event-driven JavaScript runtime.

Summary

Visual Studio Core editor
The Visual Studio core shell experience, including syntax-aware code editing, source code control and work item management.

Location
C:\Program Files (x86)\Microsoft Visual Studio\2017\Community Change...

By continuing, you agree to the license for the Visual Studio edition you selected. We also offer the ability to download other software with Visual Studio. This software is licensed separately, as set out in the 3rd Party Notices or in its accompanying license. By continuing, you also agree to those licenses.

Total space required 602 MB

For example, choose the ".NET desktop development" workload. It comes with the default core editor, which includes basic code editing support for over 20 languages, the ability to open and edit code from any folder without requiring a project, and integrated source code control.

2. After you select the workload(s) you want, click **Install**.

   Next, status screens appear that show the progress of your Visual Studio installation.

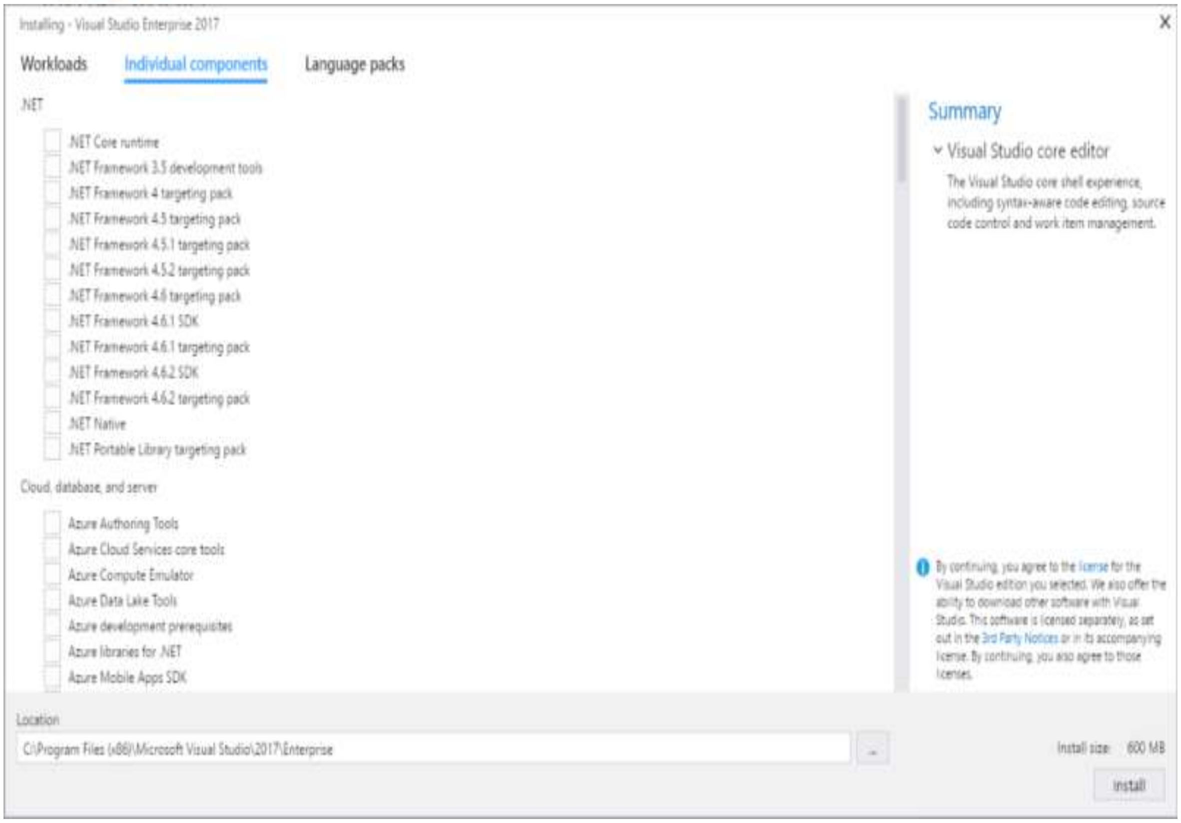
3. After the new workloads and components are installed, click **Launch**.

Note:

At any time after installation, you can install workloads or components that you didn't install initially. If you have Visual Studio open, go to **Tools** > **Get Tools and Features...** which opens the Visual Studio Installer. Or, open **Visual Studio Installer** from the Start menu. From there, you can select the workloads or components that you wish to install, then click **Modify**.

Step 5 - Select individual components (Optional)

If you don't want to use the Workloads feature to customize your Visual Studio installation, you can do so by installing individual components instead. To select individual components, click the **Individual components** option from the Visual Studio Installer, select what you want, and then follow the prompts.



.NET

- ☐ .NET Core runtime
- ☐ .NET Framework 3.5 development tools
- ☐ .NET Framework 4 targeting pack
- ☐ .NET Framework 4.5 targeting pack
- ☐ .NET Framework 4.5.1 targeting pack
- ☐ .NET Framework 4.5.2 targeting pack
- ☐ .NET Framework 4.6 targeting pack
- ☐ .NET Framework 4.6.1 SDK
- ☐ .NET Framework 4.6.1 targeting pack
- ☐ .NET Framework 4.6.2 SDK
- ☐ .NET Framework 4.6.2 targeting pack
- ☐ .NET Native
- ☐ .NET Portable Library targeting pack

Cloud, database, and server

- ☐ Azure Authoring Tools
- ☐ Azure Cloud Services core tools
- ☐ Azure Compute Emulator
- ☐ Azure Data Lake Tools
- ☐ Azure development prerequisites
- ☐ Azure libraries for .NET
- ☐ Azure Mobile Apps SDK

Summary

˅ Visual Studio core editor

The Visual Studio core shell experience, including syntax-aware code editing, source code control and work item management.

By continuing, you agree to the license for the Visual Studio edition you selected. We also offer the ability to download other software with Visual Studio. This software is licensed separately, as set out in the 3rd Party Notices or in its accompanying license. By continuing, you also agree to those licenses.

C:\Program Files (x86)\Microsoft Visual Studio\2017\Enterprise

Step 6 - Install language packs (Optional)

Step 7 - Change the installation location (Optional)

Step 8 - Start developing

After Visual Studio installation is complete, click the **Launch** button to get started developing with Visual Studio.

1. Click **File**, and then click **New Project**.
2. Select a project type.

   For example, to build a C++ app, click **Installed**, expand **Visual C++**, and then select the C++ project type that you want to build.

   To build a C# app, click **Installed**, expand **Visual C#**, and then select the C# project type that you want to build.

Lab Task 5:

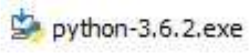
Following are the installation steps for Python:

You should download and install the latest version of Python. The current latest (as of Fall 2018) is Python 3.6.4. or Python 3.7.0.

The Python download requires about 30 Mb of disk space; keep it on your machine, in case you need to re-install Python. When installed, Python requires about an additional 90 Mb of disk space.

Step 1: Downloading:

1. Goto @ https://www.python.org/downloads/
2. Click the Download Python 3.6.2 button (or the latest version available).
3. The file named python-3.6.2.exe should start downloading into your standard download folder. This file is about 30 Mb so it might take a while to download fully if you are on a slow internet connection (it took me about 10 seconds over a cable modem).
4. The file should appear as



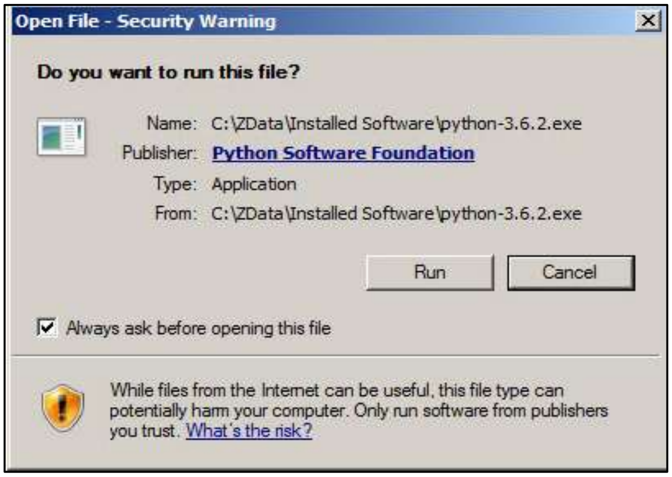
python-3.6.2.exe

5. Move this file to a more permanent location, so that you can install Python (and reinstall it easily later, if necessary).
6. Feel free to explore this webpage further; if you want to just continue the installation, you can terminate the tab browsing this webpage.
7. Start the **Installing** instructions directly below.

Step 2: Installing:

1. Double-click the icon labeling the file python-3.6.2.exe.

An Open File - Security Warning pop-up window will appear.



Open File - Security Warning

Do you want to run this file?

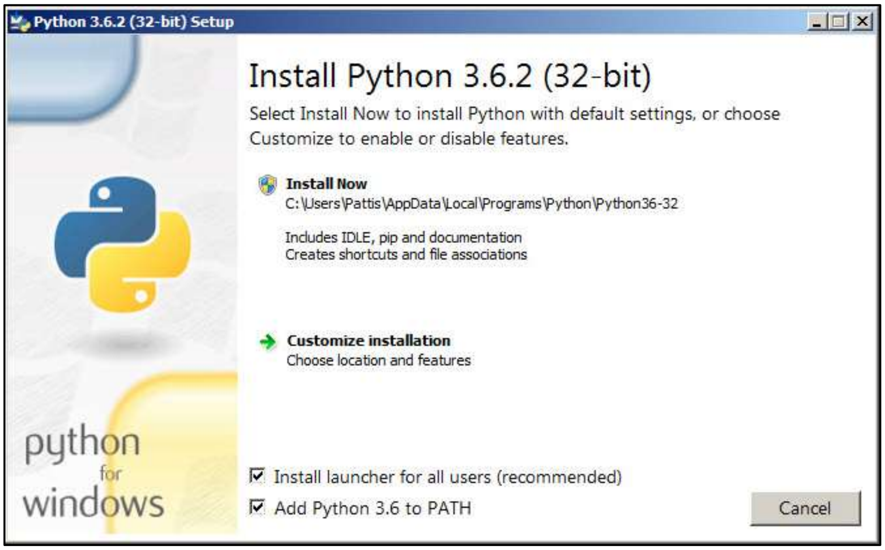
Name: C:\ZData\Installed Software\python-3.6.2.exe
Publisher: Python Software Foundation
Type: Application
From: C:\ZData\Installed Software\python-3.6.2.exe

☑ Always ask before opening this file

While files from the Internet can be useful, this file type can potentially harm your computer. Only run software from publishers you trust. What's the risk?

2. Click **Run**.

A **Python 3.6.2 (32-bit) Setup** pop-up window will appear.



Python 3.6.2 (32-bit) Setup

Install Python 3.6.2 (32-bit)

Select Install Now to install Python with default settings, or choose Customize to enable or disable features.

Install Now
C:\Users\Pattis\AppData\Local\Programs\Python\Python36-32

Includes IDLE, pip and documentation
Creates shortcuts and file associations

Customize installation
Choose location and features

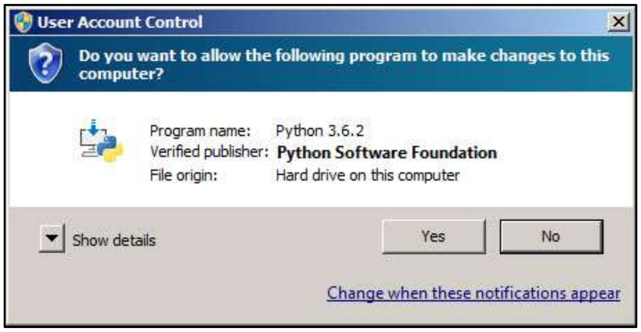
☑ Install launcher for all users (recommended)
☑ Add Python 3.6 to PATH

3. Ensure that the **Install launcher for all users (recommended)** and the **Add Python 3.6 to PATH** checkboxes at the bottom are checked.

   If the Python Installer finds an earlier version of Python installed on your computer, the **Install Now** message will instead appear as **Upgrade Now** (and the checkboxes will not appear).

4. Highlight the **Install Now** (or **Upgrade Now**) message, and then click it.

   A **User Account Conrol** pop-up window will appear, posing the question **Do you want the allow the following program to make changes to this computer?**



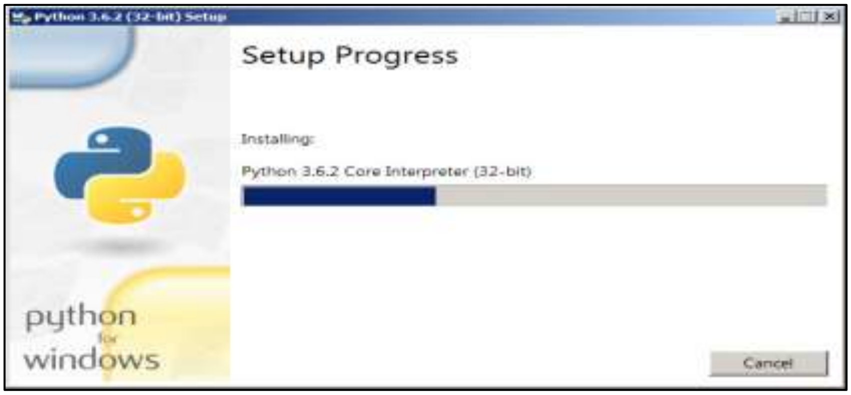
User Account Control

Do you want to allow the following program to make changes to this computer?

Program name: Python 3.6.2
Verified publisher: **Python Software Foundation**
File origin: Hard drive on this computer

5. Click the **Yes** button.

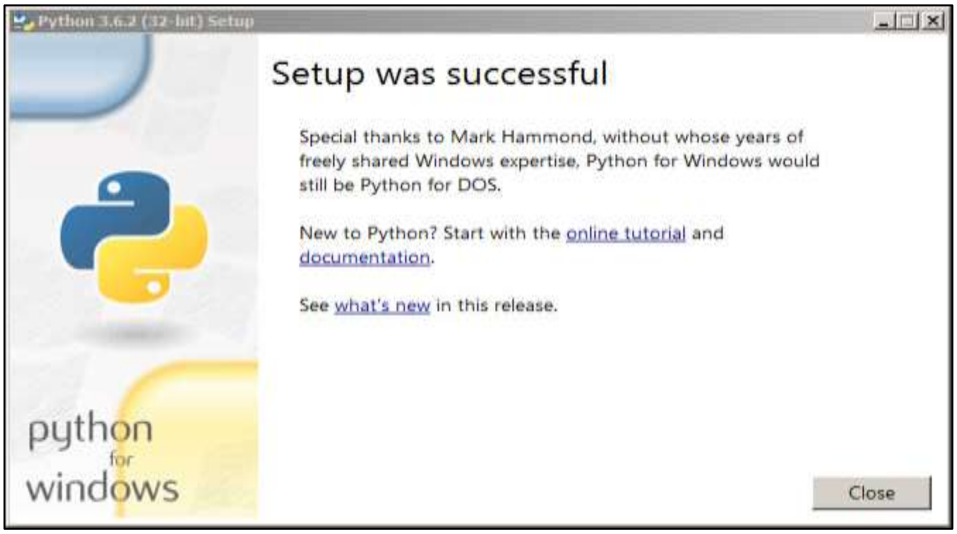
A new **Python 3.6.2 (32-bit) Setup** pop-up window will appear with a **Setup Progress** message and a progress bar.



Setup Progress

Installing:

Python 3.6.2 Core Interpreter (32-bit)



Setup was successful

Special thanks to Mark Hammond, without whose years of freely shared Windows expertise, Python for Windows would still be Python for DOS.

New to Python? Start with the online tutorial and documentation.

See what's new in this release.

During installation, it will show the various components it is installing and move the progress bar towards completion. Soon, a new **Python 3.6.2 (32-bit) Setup** pop-up window will appear with a **Setup was successfuly** message.

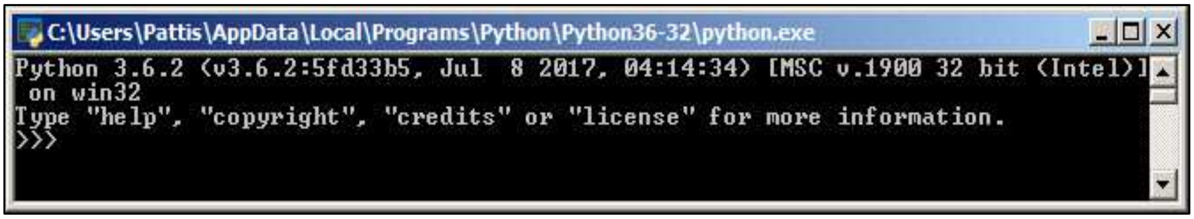
6. Click the **Close** button.

Python should now be installed.

Step 3: Verifying

To try to verify installation,

1. Navigate to the



```
C:\Users\Pattis\AppData\Local\Programs\Python\Python36-32\python.exe
Python 3.6.2 (v3.6.2:5fd33b5, Jul  8 2017, 04:14:34) [MSC v.1900 32 bit (Intel)]
 on win32
Type "help", "copyright", "credits" or "license" for more information.
>>>
```

   directory **C:\Users\Pattis\AppData\Local\Programs\Python\Python36-32** (or to whatever directory Python was installed: see the pop-up window for Installing step 3).
2. Double-click the icon/file **python.exe**.

   The following pop-up window will appear.

A pop-up window with the title **C:\Users\Pattis\AppData\Local\Programs\Python\Python36-32** appears, and inside the window; on the first line is the text **Python 3.6.2 ...** (notice that it should also say 32 bit). Inside the window, at the bottom left, is the prompt >>>: type **exit()** to this prompt and press **enter** to terminate Python.

Lab Task

1. Install JDK if already installed, uninstall the older and install new version.
2. Install Eclipse, after JDK installation.
3. Install Kotlin, in Android Studio as a plugin.
4. Install latest version of Visual Studio. (You can install an earlier version as well because this is a heavy software that slow down your system, replacement can be 2012 or 2015 version)
5. Install Python latest version.
6. [Optional] Run simple programs to check all the software are working properly.

For help use

- https://www3.ntu.edu.sg/home/ehchua/programming/howto/JDK_Howto.html
- http://www.oracle.com/technetwork/java/javase/documentation/index.html

Deliverables

- Each submission is individual with the following composition:
  - Documentation (Comments)

- Convert your submission file into a zip folder and name it as given below, finally upload the zip folder to LMS.
    - Name – Registration No. – Section

Grade Criteria

This lab is graded. Min marks: 0. Max marks: 10.

Activity	Minimum	Maximum
Task 1	0	2
Task 2	0	2
Task 3	0	2
Task 4	0	2
Task 5	0	2
Task 6 (Bonus)	0	2
Total	**0**	**10**