Environment Setup Spring Tool Suite 4

Description: “Spring Tool Suite 4 is the next generation of Spring tooling for your favorite coding environment. Largely rebuilt from scratch, it provides world-class support for developing Spring-based enterprise applications, whether you prefer Eclipse, Visual Studio Code, or Theia IDE.” – Spring.io.

Project: Download and install Spring Tool Suite 4 on local machine.

Technology / Dependencies:
This project uses the following technology:

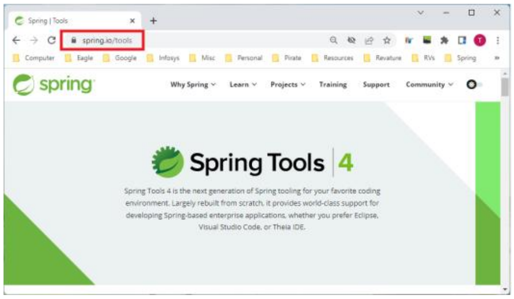
Java 1.8 (See Environment Setup 02 Java 1.8.docx)
Git Hub (See Environment Setup 01 Git Hub.docx)
Spring Tool Suite 4

Table of Contents

Install Spring Tool Suite 4

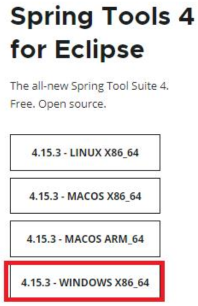
Download Spring Tool Suite 4

Using an Internet browser, browse to the Sprint Tools 4 download website (https://spring.io/tools).



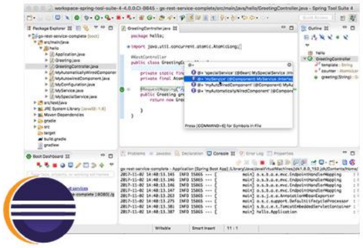
Spring Tools 4 is the next generation of Spring tooling for your favorite coding environment. Largely rebuilt from scratch, it provides world-class support for developing Spring-based enterprise applications, whether you prefer Eclipse, Visual Studio Code, or Theia IDE.

There are various download options for types of tools (Eclipse, Visual Studio, and Theia). In this document and other project documents Eclipse for Windows is used.



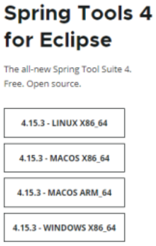
Spring Tools 4 for Eclipse

The all-new Spring Tool Suite 4. Free. Open source.



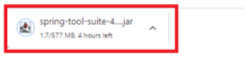
At the time this document was written Spring Tool Suite 4 version 4.15.3 was the latest. It is recommended to check back at the tools website to see if newer version of Spring Tool Suite 4 is available.

Click the button for your operating system (O/S) to begin the download. “4.15.3 – WINDOWS X86_64” is used in this document and future projects.



Spring Tools 4 for Eclipse

The all-new Spring Tool Suite 4. Free. Open source.

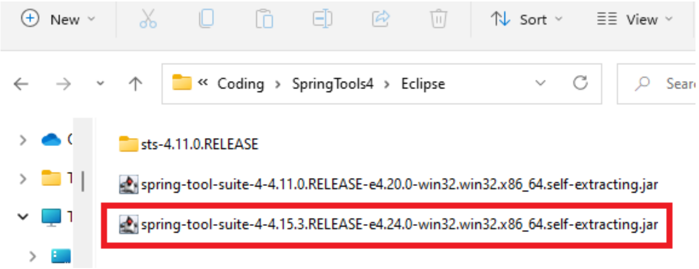


spring-tool-suite-4....jar
1.7/577 MB, 4 hours left

Depending on your browser and download settings, you might need to navigate to a save location and you should see some indication of download progress. This download might take a while to complete.

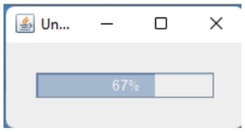
Extract Spring Tool Suite 4 Files

Once the download is complete, open the file location with your O/S file browser. Move the file to a location where you want the self-extracting jar file to extract all the files. The self-extracting jar file will create a version directory where the files will be extracted. As you can see from the screen shot, I am upgrading to a newer version of Spring Tool Suite 4. I also have a Desktop directory structure that contains subfolder for the Eclipse tool.

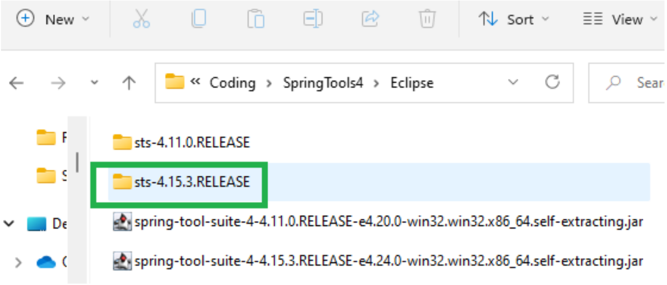


sts-4.11.0.RELEASE
spring-tool-suite-4-4.11.0.RELEASE-e4.20.0-win32.win32.x86_64.self-extracting.jar
spring-tool-suite-4-4.15.3.RELEASE-e4.24.0-win32.win32.x86_64.self-extracting.jar

Double click the self-extracting jar file to begin the extraction. You should see a progress bar.



The extraction should not take long and once completed you will see a release folder. Inside the folder will be the extracted files.



sts-4.11.0.RELEASE
sts-4.15.3.RELEASE
spring-tool-suite-4-4.11.0.RELEASE-e4.20.0-win32.win32.x86_64.self-extracting.jar
spring-tool-suite-4-4.15.3.RELEASE-e4.24.0-win32.win32.x86_64.self-extracting.jar

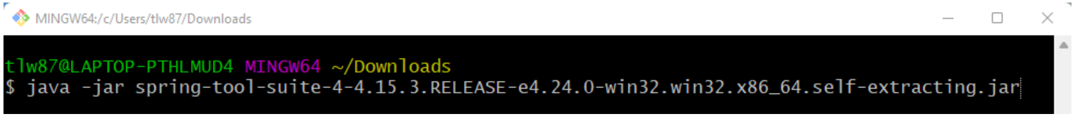
Alternate way to Extract Spring Tool Suite 4 Files

If double clicking the self-extracting jar file does not work, you do not have a zip extract program on your machine. You can go to a GIT Bash terminal and extract the files from the jar using a command like:

java -jar <filename>.jar
Where: <filename> is the name of the jar file to extract.

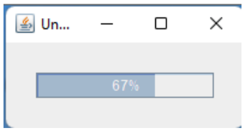
```
java -jar spring-tool-suite-4-4.15.3.RELEASE-e4.24.0-win32.win32.x86_64.self-extracting.jar
```

Run the command in Git Hub in the directory where the self-extracting jar file is located.

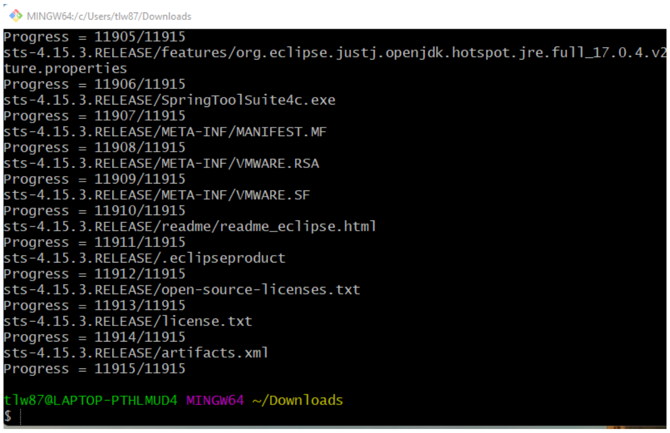


```
tlw87@LAPTOP-PTHLMUD4 MINGW64 ~/Downloads
$ java -jar spring-tool-suite-4-4.15.3.RELEASE-e4.24.0-win32.win32.x86_64.self-extracting.jar
```

A progress bar should pop up.

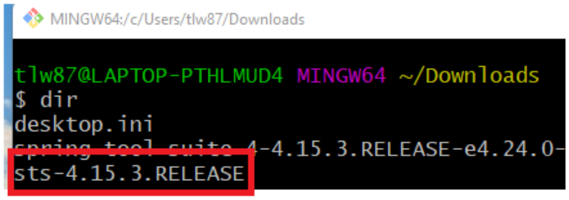


Extracted files will also scroll in the Git Hub window.



```
Progress = 11905/11915
sts-4.15.3.RELEASE/features/org.eclipse.justj.openjdk.hotspot.jre.full_17.0.4.v2
ture.properties
Progress = 11906/11915
sts-4.15.3.RELEASE/SpringToolSuite4c.exe
Progress = 11907/11915
sts-4.15.3.RELEASE/META-INF/MANIFEST.MF
Progress = 11908/11915
sts-4.15.3.RELEASE/META-INF/VMWARE.RSA
Progress = 11909/11915
sts-4.15.3.RELEASE/META-INF/VMWARE.SF
Progress = 11910/11915
sts-4.15.3.RELEASE/readme/readme_eclipse.html
Progress = 11911/11915
sts-4.15.3.RELEASE/.eclipseproduct
Progress = 11912/11915
sts-4.15.3.RELEASE/open-source-licenses.txt
Progress = 11913/11915
sts-4.15.3.RELEASE/license.txt
Progress = 11914/11915
sts-4.15.3.RELEASE/artifacts.xml
Progress = 11915/11915

tlw87@LAPTOP-PTHLMUD4 MINGW64 ~/Downloads
$
```

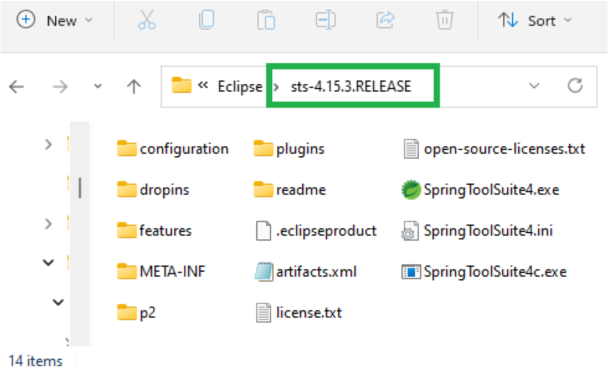
Once the extract is done, the release folder with the extracted file is created.



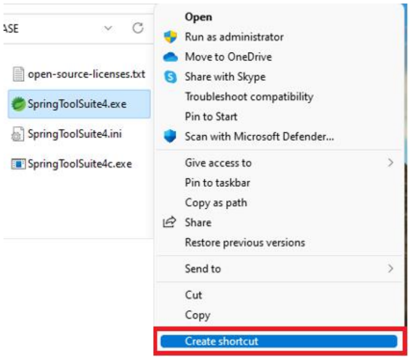
```
tlw87@LAPTOP-PTHLMUD4 MINGW64 ~/Downloads
$ dir
desktop.ini
spring-tool-suite-4-4.15.3.RELEASE-e4.24.0-
sts-4.15.3.RELEASE
```

Create Spring Tool Suite 4 Shortcut

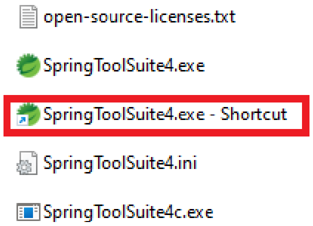
Open the release folder where the Spring Tool Suite 4 program files are located.



For convenience a shortcut can be created by right clicking on the executable file and selecting “Create shortcut”. This is for Windows O/S.



You can rename and move the shortcut to a different location if desired.



Validate Spring Tool Suite 4 Installation

Spring Tool Suite 4 is dependent on the Java 1.8 installation. Open the Spring Tool Suite 4 application by using the shortcut or the executable directly. If you encounter any errors check all the path variables configured in the Java 1.8 installation.

If STS opens without errors the tool will asked to create a workspace. For now just click "Cancel". The next section “Using Spring Tool Suite for the First Time” will setup the workspace and detail some other initial configuration setup.

Using Spring Tool Suite 4 for the First Time

When starting Spring Tool Suite 4 for the first time you should define a folder for you default workspace. This can be a new or existing folder on your desktop or in your documents folder. This workspace will contain project files that you will create going forward.

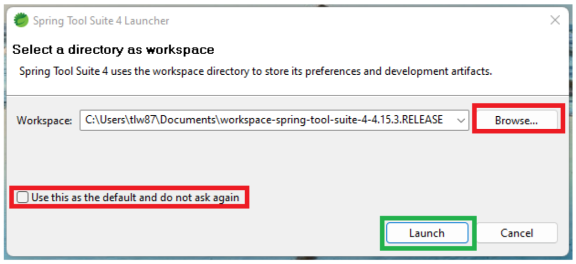
This section will also create a simple Java project to configure some configuration elements. The configuration elements will be used in future projects like selecting the current Java version we want to use and building the application.

Open the Spring Tool Suite 4 application, the splash screen will appear briefly.

Define Spring Tool Suite 4 Default Workspace

If you do not create a workspace at this time, you can create one in the future or use the default workspace.

You can use the default workspace as shown in the workspace dialog or click “Browse” and select a different folder. Click the “Use this as the default and do not ask again” checkbox if you do not want to see this dialog every time you open the program.


Spring Tool Suite 4 Launcher

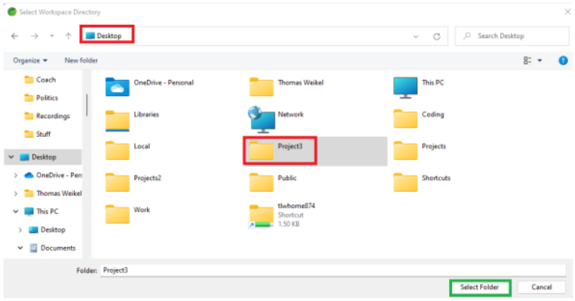
Select a directory as workspace

Spring Tool Suite 4 uses the workspace directory to store its preferences and development artifacts.

Workspace: C:\Users\tlw87\Documents\workspace-spring-tool-suite-4-4.15.3.RELEASE

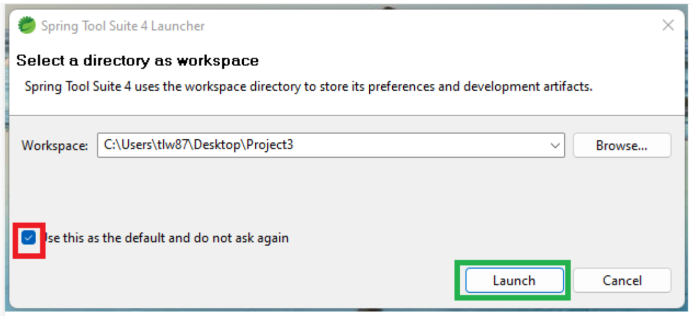
Browse...
☐ Use this as the default and do not ask again
Launch
Cancel

In this document we use “Browse…” and select a Desktop folder “Project3”



Folder: Project3

Browse and select a folder for your projects and click “Select Folder”.



Select a directory as workspace

Spring Tool Suite 4 uses the workspace directory to store its preferences and development artifacts.

Workspace: C:\Users\tlw87\Desktop\Project3

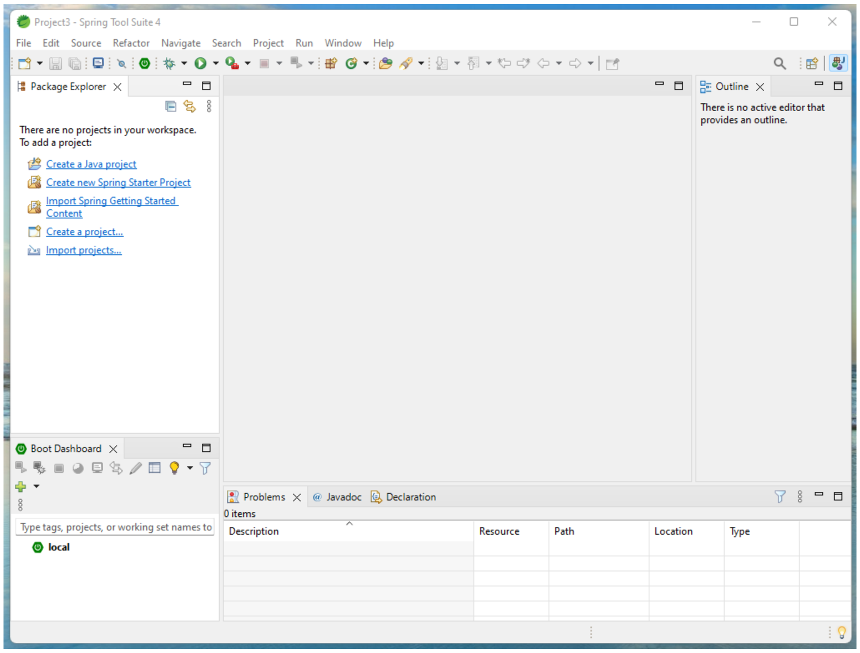
☑ se this as the default and do not ask again

Once a selection is made, click “Launch”.

The splash screen is shown again then the program opens for the first time.

Spring Tool Suite 4 Program Workspace

The screen capture below shows a typical launch of the Spring Tool Suite 4 program when it opens for the first time.



There are no projects in your workspace.
To add a project:

- Create a Java project
- Create new Spring Starter Project
- Import Spring Getting Started Content
- Create a project...
- Import projects...

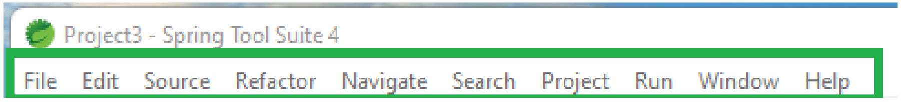
There is no active editor that provides an outline.

Type tags, projects, or working set names to

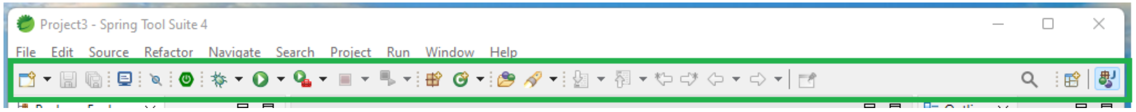
0 items

Description	Resource	Path	Location	Type

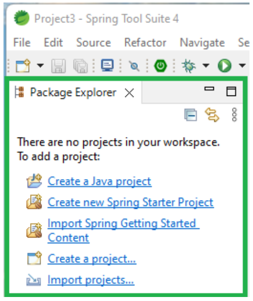
Like most Window programs, there is a main menu bar.



Below this is the program’s quick access menu bar.



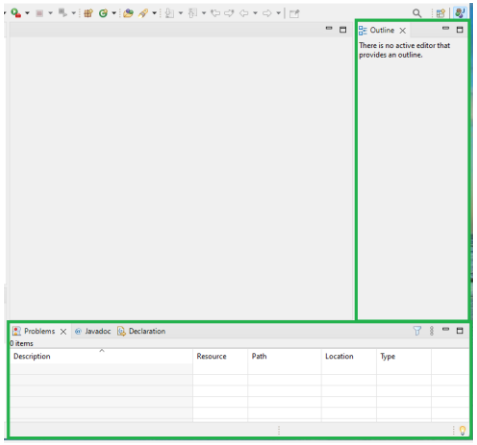
A “Package Explorer” area which will look different after creating the first project. This area will show all project loaded in the Spring Tool Suite 4 program.


Project3 - Spring Tool Suite 4
File Edit Source Refactor Navigate Se
Package Explorer

There are no projects in your workspace.
To add a project:

- Create a Java project
- Create new Spring Starter Project
- Import Spring Getting Started Content
- Create a project...
- Import projects...

There are information areas with tabs for various views. The two area can be combined to conserve space and enlarge other areas. Combining the areas is done by dragging and dropping into the other. My preference is to combine the bottom area with the “Outline” area. Additional tabs, like “Console” will be open as needed.

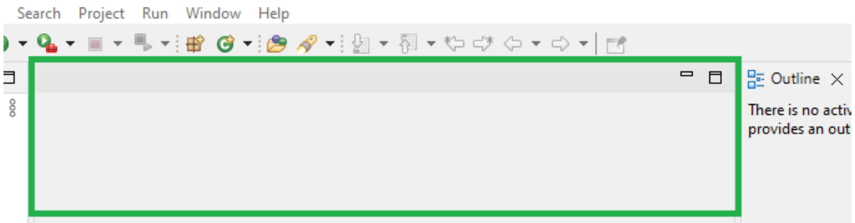


There is no active editor that provides an outline.

0 items

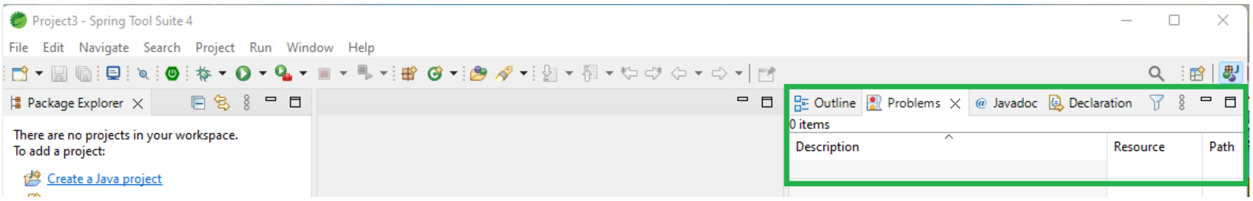
Description	Resource	Path	Location	Type

The center area is where files will be open for editing.



There is no activ
provides an out

As mentioned before, certain areas can be moved around. Here the bottom area where the “Problems” and other tabs were shown is combined with the “Outline” tab area.


Project3 - Spring Tool Suite 4
File Edit Navigate Search Project Run Window Help
Package Explorer
Outline
Problems
Javadoc
Declaration

There are no projects in your workspace.
To add a project:

- Create a Java project

0 items

Description	Resource	Path

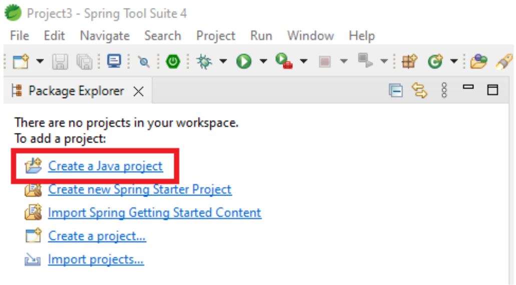
Other program area will be discussed when used.

Spring Tool Suite 4 Program First Program

We'll start with a simple classic "Hello World" Java project. Mainly to do some initial configuration to utilize the Java 8 installation libraries.

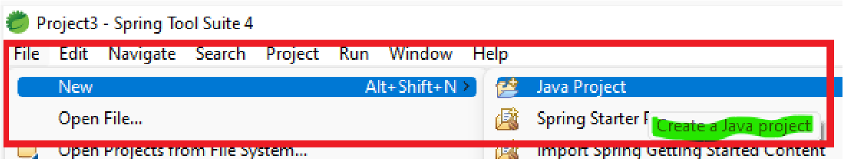
Create New Java Project Navigation Options

When using Spring Tool Suite 4 for the first time you can select the "Create a Java project" in the "Package Explorer".

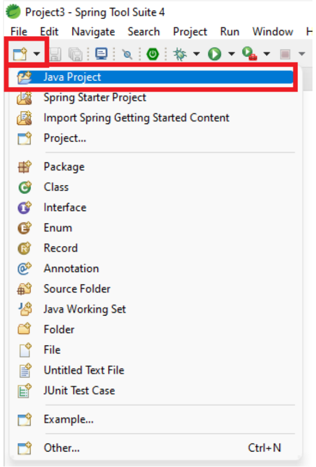


There are no projects in your workspace.
To add a project:

Alternatively, and after the first project is created, use the main menu to create Java projects.
Select File ➔New ➔ Java Project

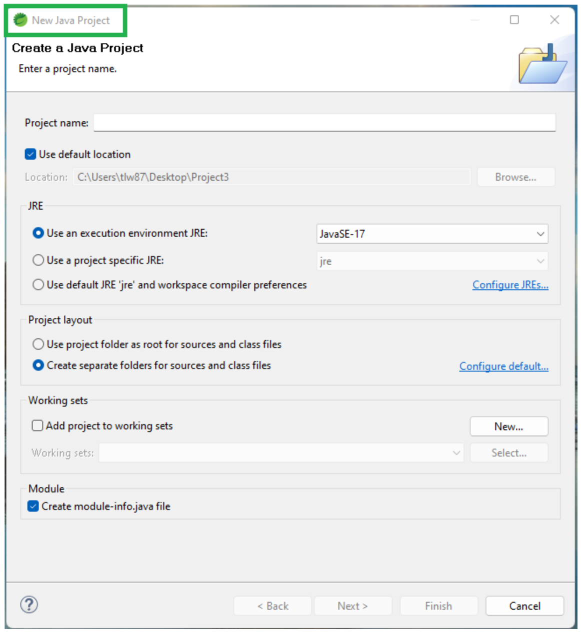


You can also use the "New" icon and select the item to be created. In this case a "Java Project".



Create “New Java Project” Dialog Window

After selecting the option to create a new Java project, a dialog window opens.



New Java Project

Create a Java Project

Enter a project name.

Project name:

☑ Use default location

Location: C:\Users\tlw87\Desktop\Project3

JRE

● Use an execution environment JRE: JavaSE-17

○ Use a project specific JRE: jre

○ Use default JRE 'jre' and workspace compiler preferences

Project layout

○ Use project folder as root for sources and class files

● Create separate folders for sources and class files

Working sets

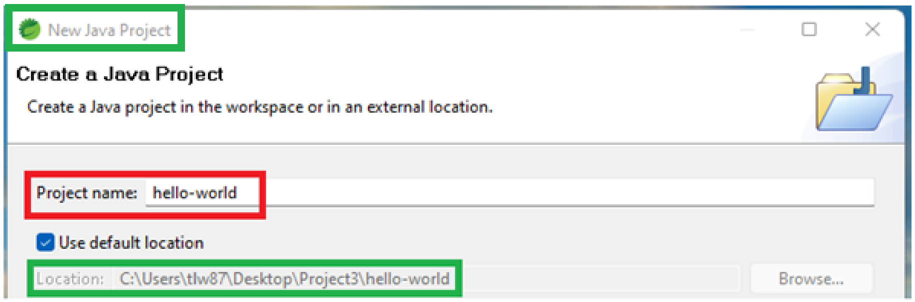
□ Add project to working sets

Working sets:

Module

☑ Create module-info.java file

Enter “Project name:” “hello-world”



New Java Project

Create a Java Project

Create a Java project in the workspace or in an external location.

Project name: hello-world

☑ Use default location

Location: C:\Users\tlw87\Desktop\Project3\hello-world

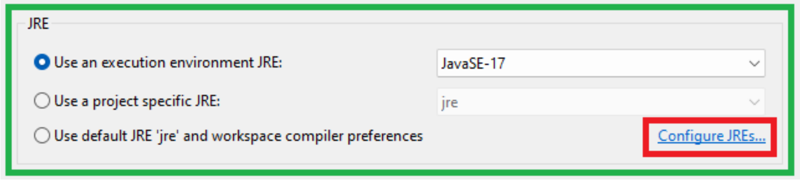
Notice the default workspace location will be used.

Set New Default Java Version for the JDK / JRE

Notice the IDE comes with a version of the JDK / JRE in this case “JavaSE-17”. This will vary depending on the version of the Spring Tool Suite 4 downloaded.

Using the installation document “Environment Setup 02 Java 1.8.docx” we installed a version of Java we want to use. This also can change over time.

Click the “Configure JREs” link to change the JDK / JRE version to use.



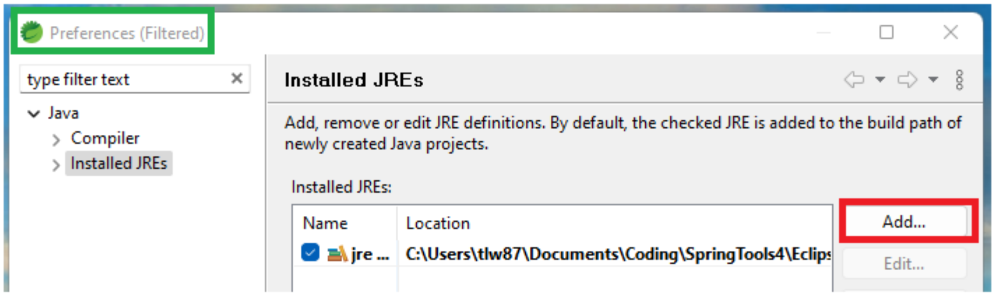
JRE

● Use an execution environment JRE: JavaSE-17

○ Use a project specific JRE: jre

○ Use default JRE 'jre' and workspace compiler preferences

In the “Preference” dialog window, click “Add…”



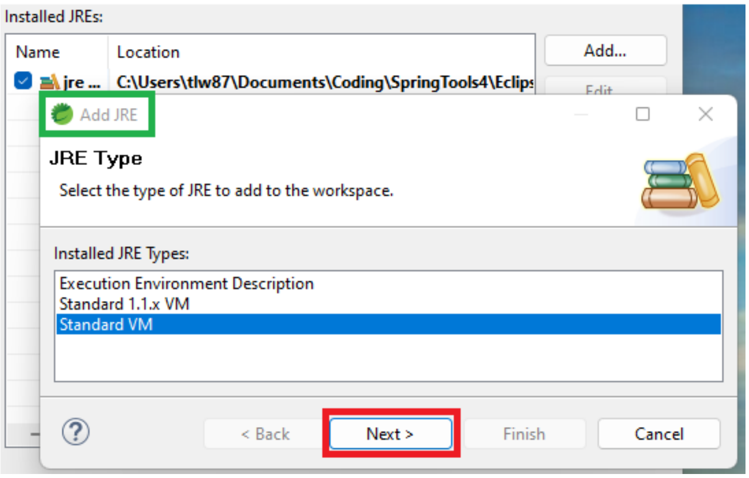
Installed JREs

Add, remove or edit JRE definitions. By default, the checked JRE is added to the build path of newly created Java projects.

Installed JREs:

Name	Location
☑ jre ...	C:\Users\tlw87\Documents\Coding\SpringTools4\Eclips

In the “Add JRE” dialog window.
Click “Next >”



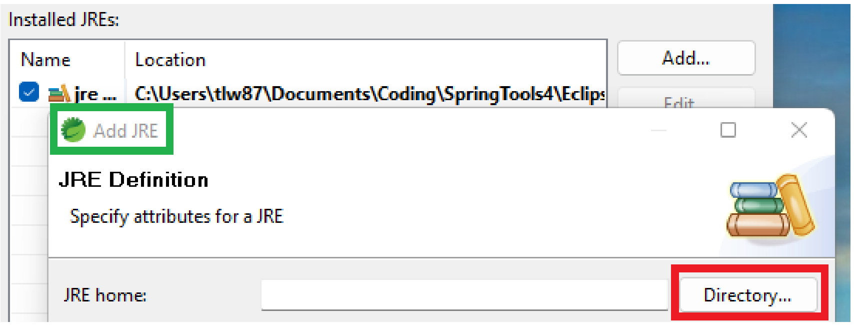
JRE Type

Select the type of JRE to add to the workspace.

Installed JRE Types:

- Execution Environment Description
- Standard 1.1.x VM
- Standard VM

Click “Directory…”



JRE Definition

Specify attributes for a JRE

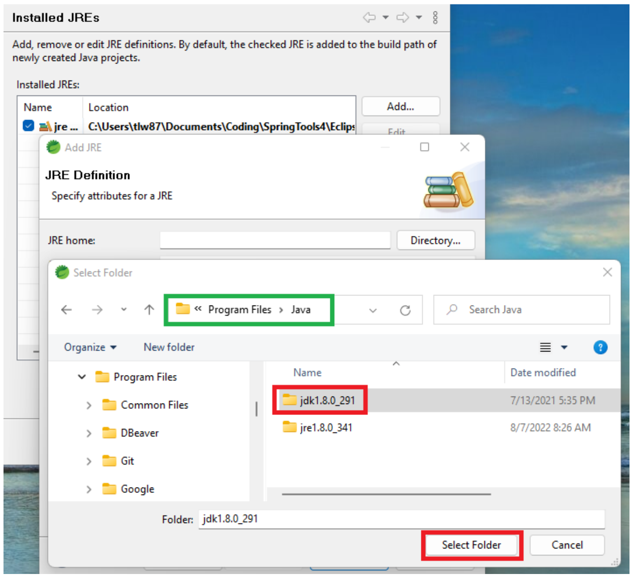
JRE home:

Navigate to where the Java 1.8 files are installed.
Typically, this should be on the C: drive “C:\Program Files\Java”
Select the jdk folder not the jre folder: jdk1.8.0_271

- A jre is included inside the jdk folder

The folder name will vary depending on which version of Java was installed.

Click “Select Folder”



Installed JREs

Add, remove or edit JRE definitions. By default, the checked JRE is added to the build path of newly created Java projects.

Installed JREs:

Name	Location
☑ jre ...	C:\Users\tlw87\Documents\Coding\SpringTools4\Eclips

Add JRE

JRE Definition

Specify attributes for a JRE

JRE home:

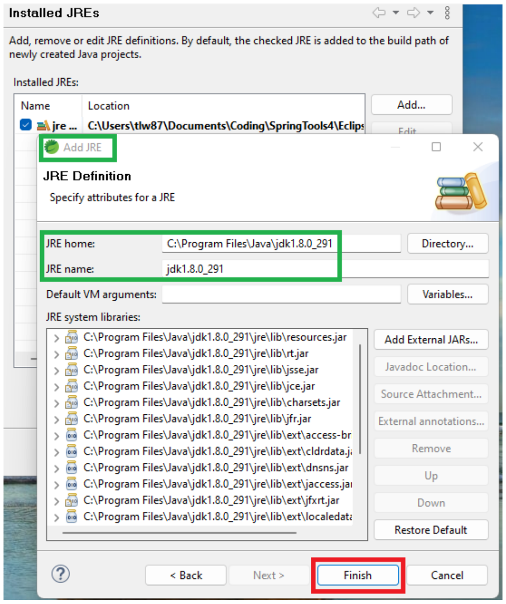
Select Folder

« Program Files › Java

Name	Date modified
jdk1.8.0_291	7/13/2021 5:35 PM
jre1.8.0_341	8/7/2022 8:26 AM

Folder: jdk1.8.0_291

Click “Finish”



Installed JREs

Add, remove or edit JRE definitions. By default, the checked JRE is added to the build path of newly created Java projects.

Installed JREs:

Name	Location
☑ jre ...	C:\Users\tlw87\Documents\Coding\SpringTools4\Eclips

Add JRE

JRE Definition

Specify attributes for a JRE

JRE home: C:\Program Files\Java\jdk1.8.0_291

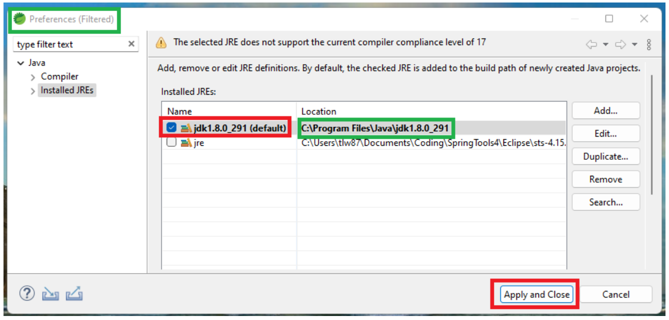
JRE name: jdk1.8.0_291

Default VM arguments:

JRE system libraries:

```
C:\Program Files\Java\jdk1.8.0_291\jre\lib\resources.jar
C:\Program Files\Java\jdk1.8.0_291\jre\lib\rt.jar
C:\Program Files\Java\jdk1.8.0_291\jre\lib\jsse.jar
C:\Program Files\Java\jdk1.8.0_291\jre\lib\jce.jar
C:\Program Files\Java\jdk1.8.0_291\jre\lib\charsets.jar
C:\Program Files\Java\jdk1.8.0_291\jre\lib\jfr.jar
C:\Program Files\Java\jdk1.8.0_291\jre\lib\ext\access-br
C:\Program Files\Java\jdk1.8.0_291\jre\lib\ext\cldrdata.j
C:\Program Files\Java\jdk1.8.0_291\jre\lib\ext\dnsns.jar
C:\Program Files\Java\jdk1.8.0_291\jre\lib\ext\jaccess.ja
C:\Program Files\Java\jdk1.8.0_291\jre\lib\ext\jfxrt.jar
C:\Program Files\Java\jdk1.8.0_291\jre\lib\ext\localedat
```

Select the checkbox for the JDK from the previous steps.
Click “Apply and Close”



Preferences (Filtered)

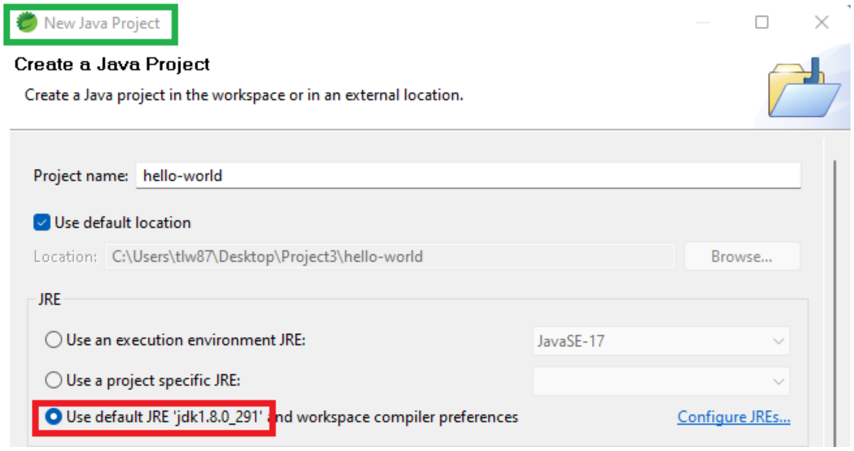
The selected JRE does not support the current compiler compliance level of 17

Add, remove or edit JRE definitions. By default, the checked JRE is added to the build path of newly created Java projects.

Installed JREs:

Name	Location
☑ jdk1.8.0_291 (default)	C:\Program Files\Java\jdk1.8.0_291
□ jre	C:\Users\tlw87\Documents\Coding\SpringTools4\Eclipse\sts-4.15.

Select the radio button for the jdk 1.8 version from the previous steps.



New Java Project

Create a Java Project

Create a Java project in the workspace or in an external location.

Project name: hello-world

☑ Use default location

Location: C:\Users\tlw87\Desktop\Project3\hello-world Browse...

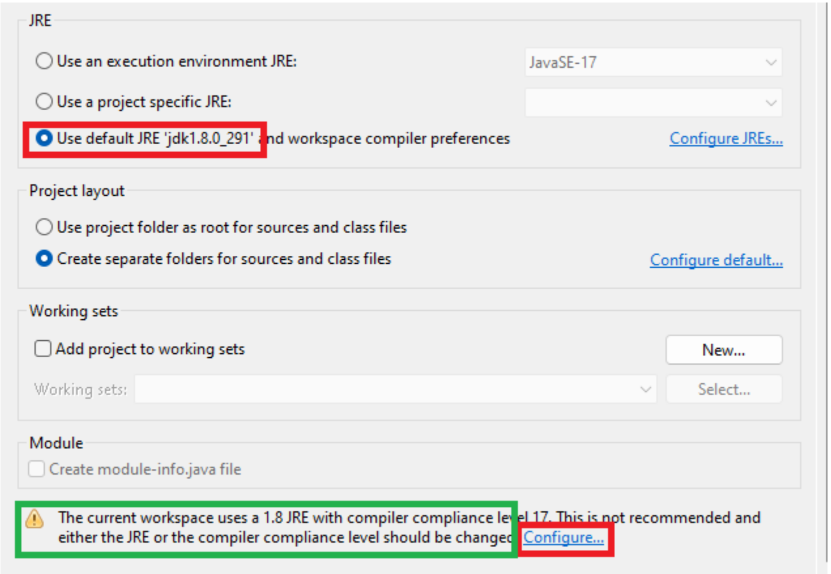
JRE

○ Use an execution environment JRE: JavaSE-17

○ Use a project specific JRE:

● Use default JRE 'jdk1.8.0_291' and workspace compiler preferences Configure JREs...

When selecting the radio button to use the new default JDK a warning is shown. Click "Configure..." link.



JRE

○ Use an execution environment JRE: JavaSE-17

○ Use a project specific JRE:

● Use default JRE 'jdk1.8.0_291' and workspace compiler preferences Configure JREs...

Project layout

○ Use project folder as root for sources and class files

● Create separate folders for sources and class files Configure default...

Working sets

☐ Add project to working sets New...

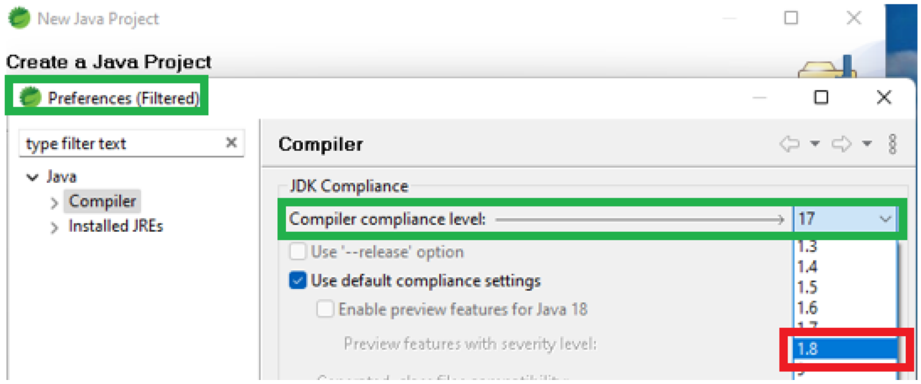
Working sets: Select...

Module

☐ Create module-info.java file

⚠ The current workspace uses a 1.8 JRE with compiler compliance level 17. This is not recommended and either the JRE or the compiler compliance level should be changed. Configure...

Change the "Compiler compliance level:" to 1.8



New Java Project

Create a Java Project

Preferences (Filtered)

type filter text

- Java
  - Compiler
  - Installed JREs

Compiler

JDK Compliance

Compiler compliance level: 17

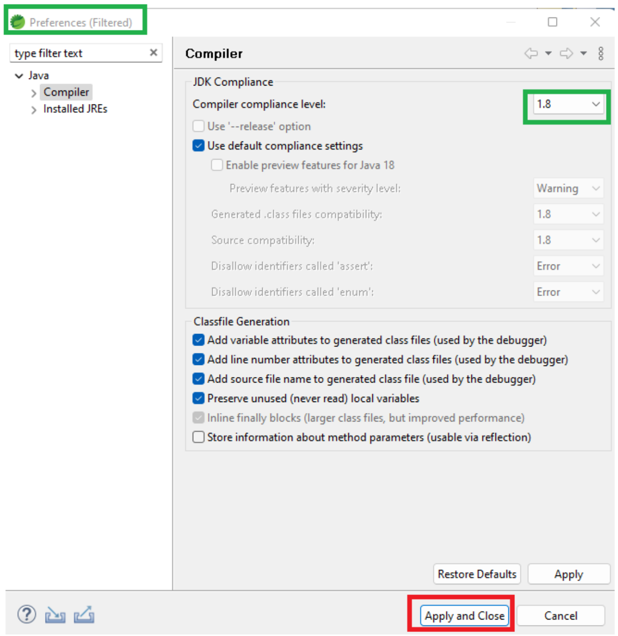
☐ Use '--release' option

☑ Use default compliance settings

☐ Enable preview features for Java 18

Preview features with severity level:

Click “Apply and Close”



Preferences (Filtered)

type filter text

- Java
  - Compiler
  - Installed JREs

Compiler

JDK Compliance

Compiler compliance level: 1.8

□ Use '--release' option

☑ Use default compliance settings

□ Enable preview features for Java 18

Preview features with severity level: Warning

Generated .class files compatibility: 1.8

Source compatibility: 1.8

Disallow identifiers called 'assert': Error

Disallow identifiers called 'enum': Error

Classfile Generation

☑ Add variable attributes to generated class files (used by the debugger)

☑ Add line number attributes to generated class files (used by the debugger)

☑ Add source file name to generated class file (used by the debugger)

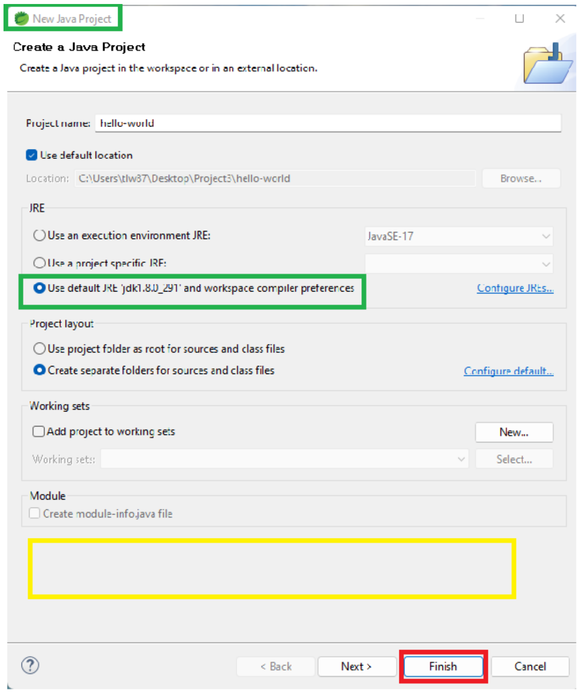
☑ Preserve unused (never read) local variables

☑ Inline finally blocks (larger class files, but improved performance)

□ Store information about method parameters (usable via reflection)

The warning message is resolved.

Click “Finish”



New Java Project

Create a Java Project

Create a Java project in the workspace or in an external location.

Project name: hello-world

☑ Use default location

Location: C:\Users\tlw37\Desktop\Project3\hello-world

JRE

○ Use an execution environment JRE: JavaSE-17

○ Use a project specific JRE:

● Use default JRE 'jdk1.8.0_291' and workspace compiler preferences

Project layout

○ Use project folder as root for sources and class files

● Create separate folders for sources and class files

Working sets

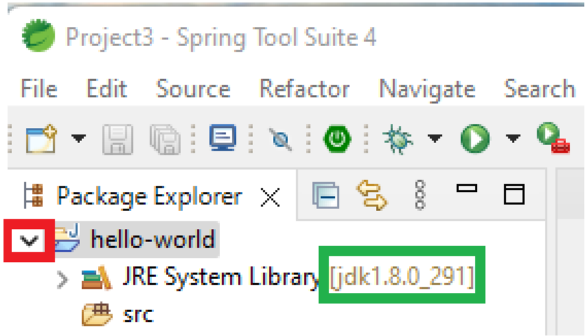
□ Add project to working sets

Working sets:

Module

□ Create module-info.java file

Expanding the "hello-world" project you can see the new JDK version



Project3 - Spring Tool Suite 4

hello-world

JRE System Library [jdk1.8.0_291]

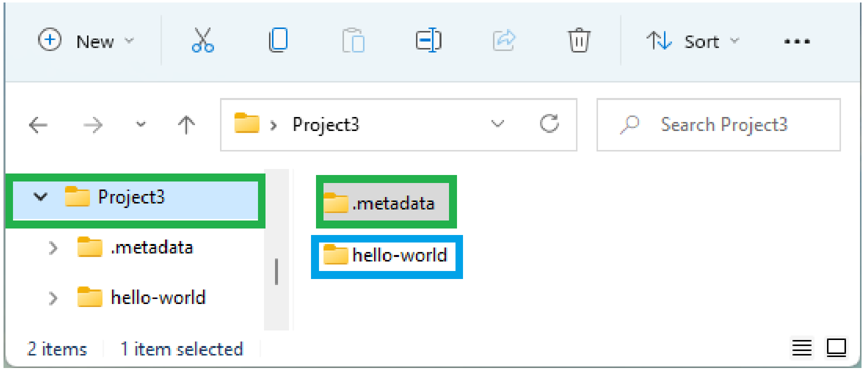
src

The steps performed in this section "Set New Default Java Version for the JDK / JRE" sets the Java 1.8 JDK / JRE as the default. This means that these steps are not performed again, and any new project will use the default Java 1.8 JDK / JRE.

Continue Spring Tool Suite 4 Program First Program

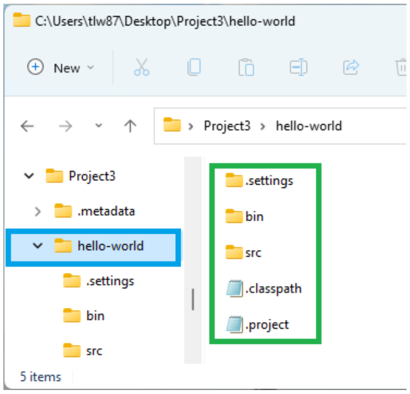
Workspace Folder Description

In the workspace folder, there is a “.metadata” containing files used by the Spring Tool Suite 4 program.



As you create new Java project, they will appear as folders in the workspace.
The “hello-world” project create in the previous step has a folder in the workspace.
The folders will contain all relevant files necessary for your project.
Each project create will have a different project folder and different project files.

The folder for a simple Java project, such as “hello-world” contains Spring Tool Suite 4 items used by the program (.settings folder, .classpath and .project files).



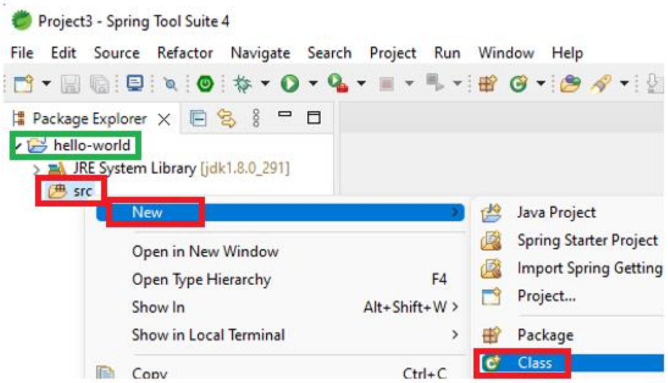
The .classpath file contains information to build the project.
The .project file is Spring Tool Suite 4 specific information concerning the project.
The bin folder will contain the .class files containing the byte code to run the project in the JVM.
The src folder will contain the .java files containing the source code to create the .class files when compiled.

Finish the Java “hello-world” Project

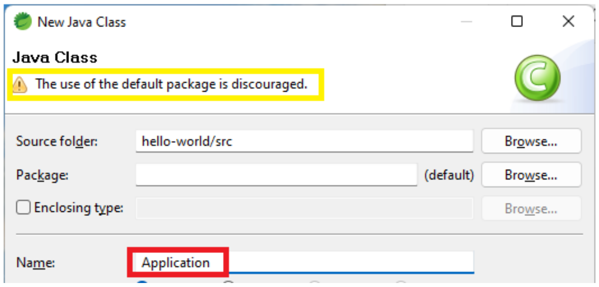
Create a class file

Right click on src folder

Select ➔New ➔Class



Once an “Name” is entered in the “New Java Class” dialog window a warning message is displayed. Best practices require creating packages for storing the source code files. For this simple “hello-world” project it would not really make a difference. But as project created becomes more complex, packages help to organize our source code files with like processing.



New Java Class

Java Class

⚠ The use of the default package is discouraged.

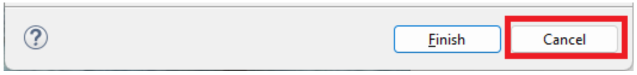
Source folder: hello-world/src

Package: (default)

□ Enclosing type:

Name: Application

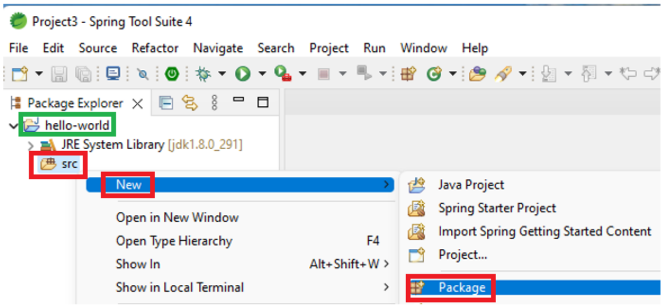
Click “Cancel” at the bottom of the dialogue window and create a package.



Create a package

Right click on src folder

Select ➔New ➔Package

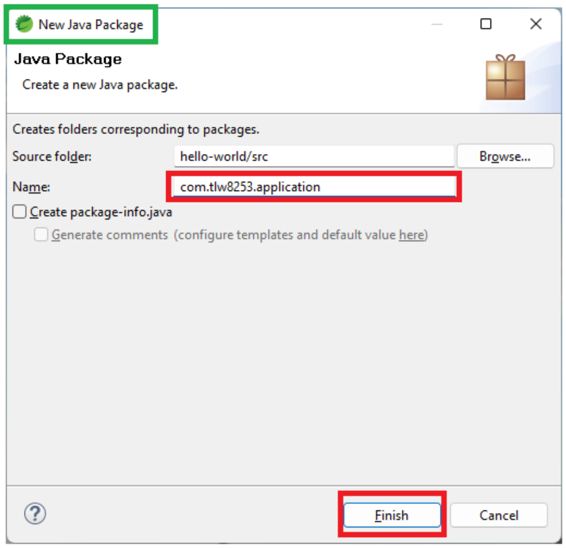


In the “New Java Package” dialogue window enter a package “Name:”
Generally a package name starts with com followed by a company identifier and the package name in the format: com.companyName.packageName.

You can find detail explanation of naming convention at the Oracle site https://docs.oracle.com/javase/tutorial/java/package/namingpkgs.html and other web searches.

> Naming Conventions
> Package names are written in all lower case to avoid conflict with the names of classes or interfaces.
>
> Companies use their reversed Internet domain name to begin their package names—for example, com.example.mypackage for a package named mypackage created by a programmer at example.com.
>
> Name collisions that occur within a single company need to be handled by convention within that company, perhaps by including the region or the project name after the company name (for example, com.example.region.mypackage). – Oracle site referenced above.

For this example “Name:” com.tlw8253.application
Click “Finish”



New Java Package

Java Package
Create a new Java package.

Creates folders corresponding to packages.

Source folder: hello-world/src

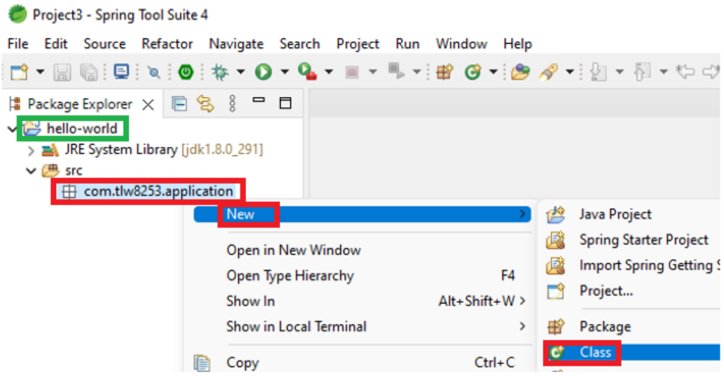
Name: com.tlw8253.application

☐ Create package-info.java

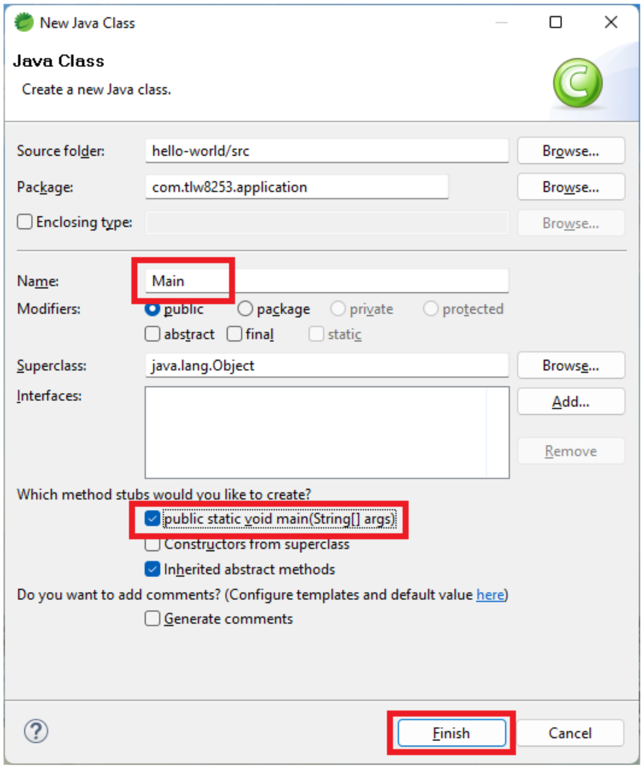
☐ Generate comments (configure templates and default value here)

Create a class file in the package

Right click on package created above
Select ➔New ➔Class



In the “New Class” dialogue window.
Enter a “Name:” “Main”
Select checkbox “public static_void …”
Click “Finish”



Java Class

Create a new Java class.

Source folder: hello-world/src

Package: com.tlw8253.application

□ Enclosing type:

Name: Main

Modifiers: ● public ○ package ○ private ○ protected

□ abstract □ final □ static

Superclass: java.lang.Object

Interfaces:

Which method stubs would you like to create?

☑ public static void main(String[] args)

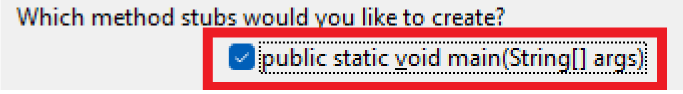
□ Constructors from superclass

☑ Inherited abstract methods

Do you want to add comments? (Configure templates and default value here)

□ Generate comments

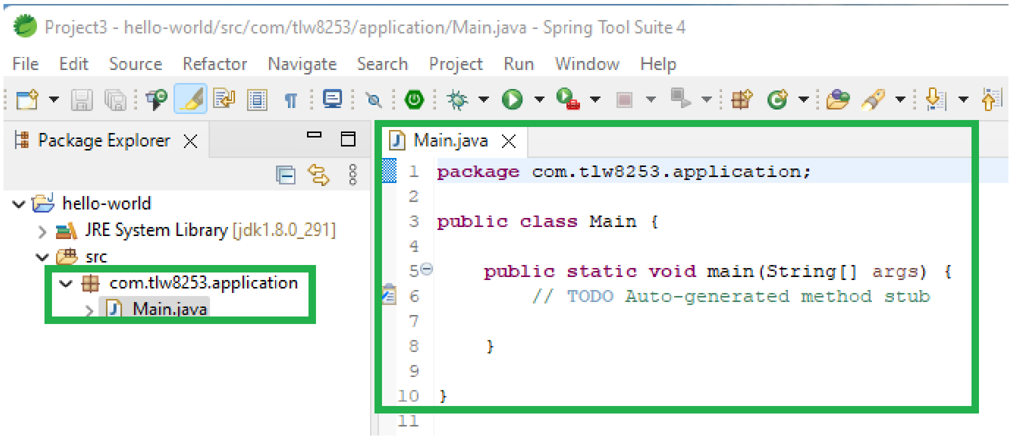
By selecting this checkbox, the IDE will stub out an runnable version of a Java program.



Which method stubs would you like to create?

☑ public static void main(String[] args)

The class “Main.java” is created under the package and is automatically open for editing.



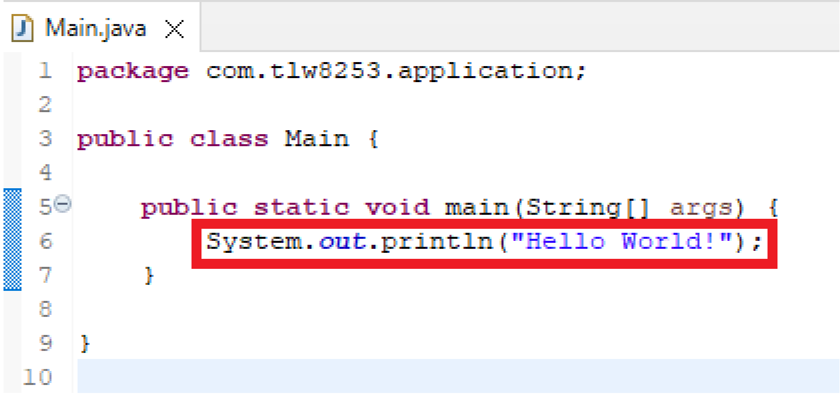
Project3 - hello-world/src/com/tlw8253/application/Main.java - Spring Tool Suite 4

- hello-world
  - JRE System Library [jdk1.8.0_291]
  - src
    - com.tlw8253.application
      - Main.java

```java
package com.tlw8253.application;

public class Main {

    public static void main(String[] args) {
        // TODO Auto-generated method stub

    }

}
```

Update the code with a print statement.

```
System.out.println("Hello World!");
```

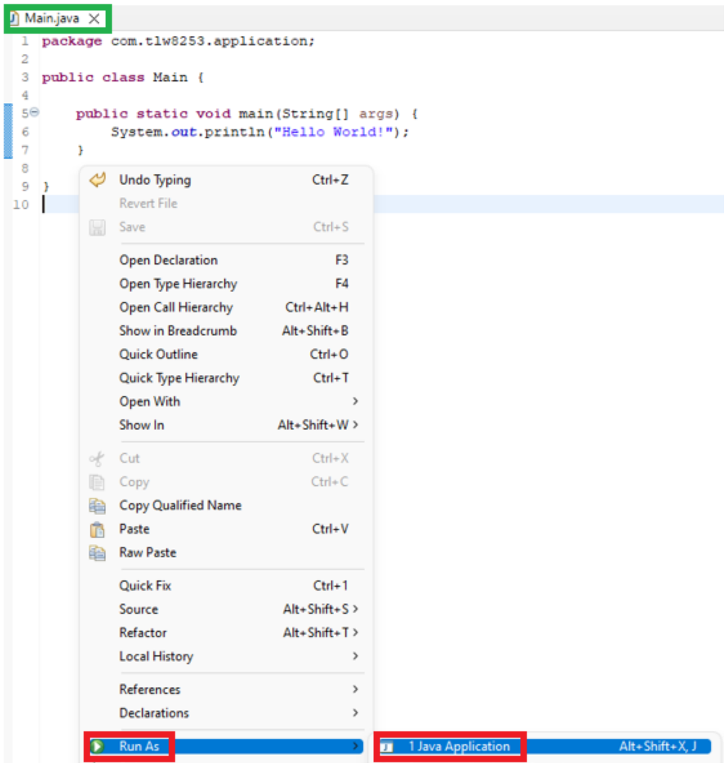


```
package com.tlw8253.application;

public class Main {

    public static void main(String[] args) {
        System.out.println("Hello World!");
    }

}

```

Save the file.

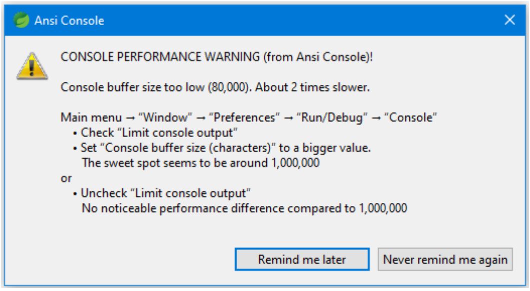
Run the Java “hello-world” application

There are multiple ways to run the application. The simplest is to right-click inside the source code area. Select “Run As” ➔ “1. Java Application”



```
package com.tlw8253.application;

public class Main {

    public static void main(String[] args) {
        System.out.println("Hello World!");
    }

}
```

If a warning message concerning buffer size appears, ignore for now and click “Remind me Later”. The last section in this document will address fixing the issue.



CONSOLE PERFORMANCE WARNING (from Ansi Console)!

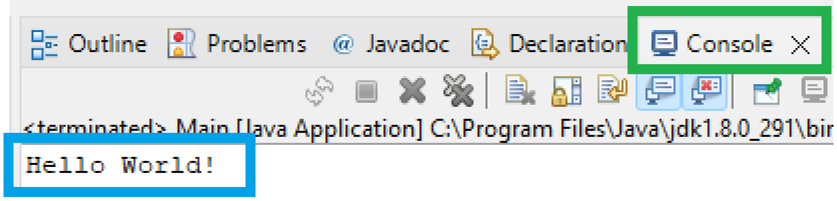
Console buffer size too low (80,000). About 2 times slower.

Main menu → “Window” → “Preferences” → “Run/Debug” → “Console”
- Check “Limit console output”
- Set “Console buffer size (characters)” to a bigger value.
  The sweet spot seems to be around 1,000,000

or
- Uncheck “Limit console output”
  No noticeable performance difference compared to 1,000,000

A new tab “Console” appears in the information area next to the other tabs. The location of the information area will vary, but if following the instructions in this document, the areas were combined to the right of the center section.

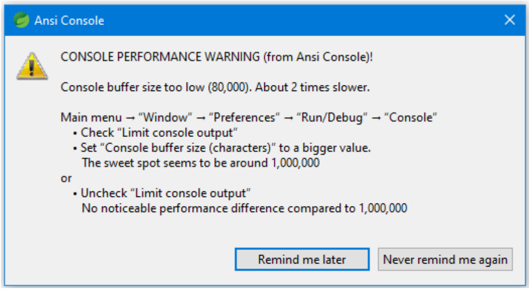
The console shows the output of the simple “hello-world” application.
In other applications the console will show more information and errors.



```
<terminated> Main [Java Application] C:\Program Files\Java\jdk1.8.0_291\bir
Hello World!
```

Fix Console Performance Warning

If at any time the following message is seen it can be resolved by following the instructions in the warning. This section also details what the warning message steps to perform.



Ansi Console

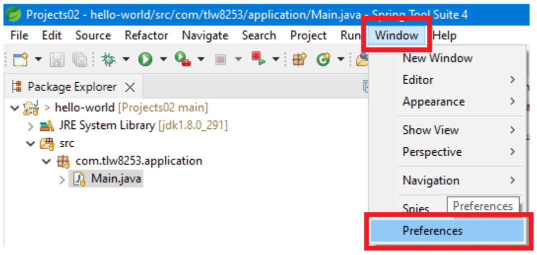
CONSOLE PERFORMANCE WARNING (from Ansi Console)!

Console buffer size too low (80,000). About 2 times slower.

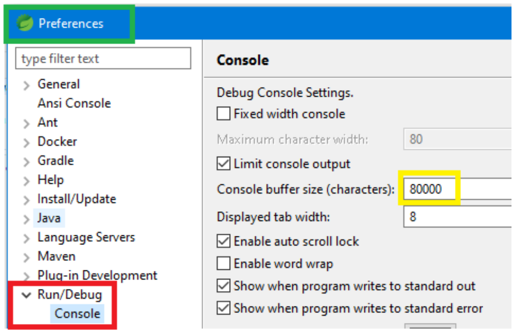
Main menu → "Window" → "Preferences" → "Run/Debug" → "Console"
- Check "Limit console output"
- Set "Console buffer size (characters)" to a bigger value.
  The sweet spot seems to be around 1,000,000

or
- Uncheck "Limit console output"
  No noticeable performance difference compared to 1,000,000

On the Main menu tool bar select "Window ➔ Preferences".



In the "Preferences" dialogue window.
Select "Run/Debug ➔ Console"



Console

Debug Console Settings.

☐ Fixed width console

Maximum character width: 80

☑ Limit console output

Console buffer size (characters): 80000

Displayed tab width: 8

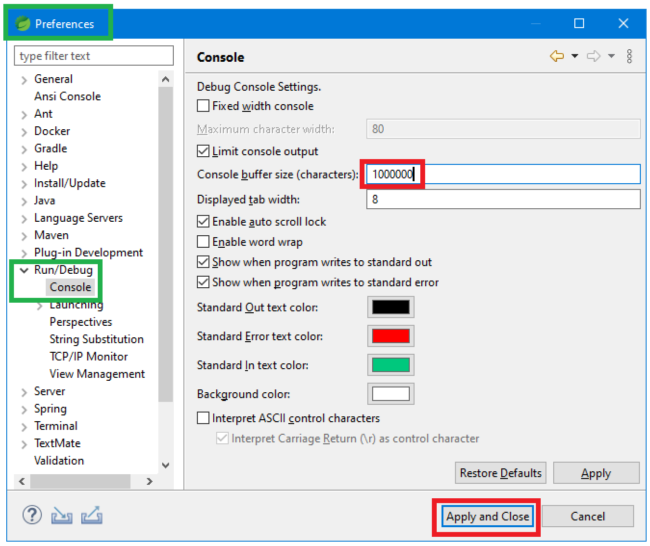
☑ Enable auto scroll lock

☐ Enable word wrap

☑ Show when program writes to standard out

☑ Show when program writes to standard error

Update the current “Console buffer size (characters):” to the size recommended in the warning message: “1000000”.
Click “Apply and Close”.



Console

Debug Console Settings.

☐ Fixed width console

Maximum character width: 80

☑ Limit console output

Console buffer size (characters): 1000000

Displayed tab width: 8

☑ Enable auto scroll lock

☐ Enable word wrap

☑ Show when program writes to standard out

☑ Show when program writes to standard error

Standard Out text color: ⬛

Standard Error text color: 🟥

Standard In text color: 🟩

Background color: ⬜

☐ Interpret ASCII control characters

☑ Interpret Carriage Return (\r) as control character

If the warning is seen in the future, increase the buffer size again.

This concludes the environment setup for the Spring Tool Suite 4.