Plug-in Vehicle Competitiveness (PVC) Software Analysis Tool

Quick Guide for Source Code Compilation

Karim Hamza, Jean Chu & Ken Laberteaux

March 2021

Disclaimer Notes

- ❑ URLs in this guide referring to specific repositories on GitHub may change in the future. If that happens, we anticipate that the README file of old repositories will be updated to include the new URLs

- ❑ There are several Java IDEs, each with its own specific set of steps for integrating with GitHub. This brief guide only provides guidelines if you are using **Eclipse** (https://www.eclipse.org/)

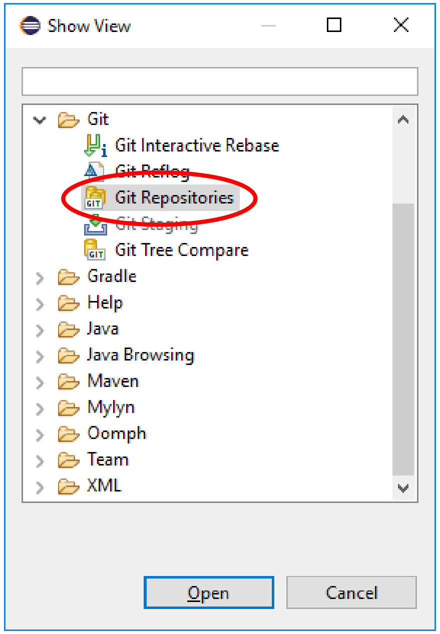
- ❑ Even within Eclipse (depending on version & updates), the location of menu commands and inputs in dialog boxes may slightly differ than current guide, but the overall steps should be similar

Contents

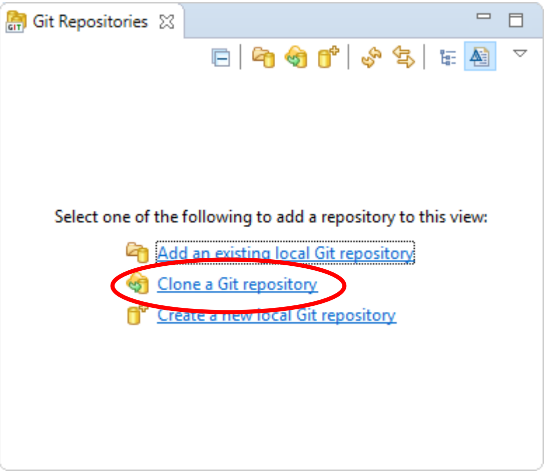
- ❑ Cloning from GitHub Repository in Eclipse
- ❑ Creating a Project in Eclipse from Cloned Repository
- ❑ Linking Projects in Eclipse
- ❑ Running PVC within Eclipse IDE
- ❑ Compiling to Runnable .jar File

Cloning from GitHub Repository

- If the “Git Repositories”-tab in Eclipse IDE isn’t already visible, it can be turned on via menu commands: Window -> Show View -> Other… ---> Git -> Git Repositories



- In Git Repositories -tab of Eclipse, select “Clone a Git Repository”



Select one of the following to add a repository to this view:

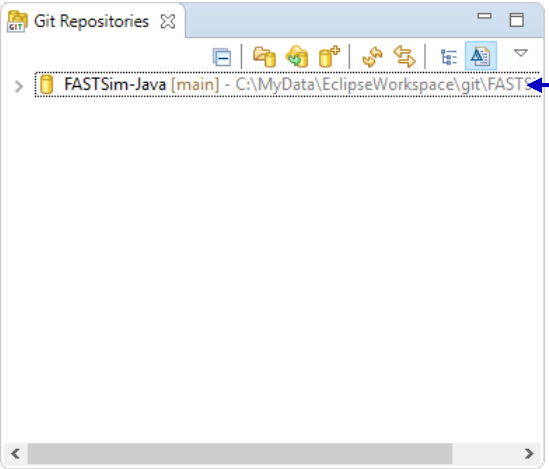
Add an existing local Git repository
Clone a Git repository
Create a new local Git repository

- Since PVC has dependencies on the Java implementation of FASTSim, we will start by cloning FASTSim-Java. Use the URL: https://github.com/khamza075/FASTSim-Java along with your own GitHub user name and password* or access token

* GitHub announced that only access tokens will be allowed by August 2021

Cloning from GitHub Repository (Cont’d)

- ❑ Select the “main” branch* of FASTSim-Java -- then click “Next”
- ❑ Accept default location where the local copy of the repository will be saved or click “Browse” to set a different directory/folder location -- then click “Finish”
- ❑ Git Repositories -tab of Eclipse should now include a new item for FASTSim-Java



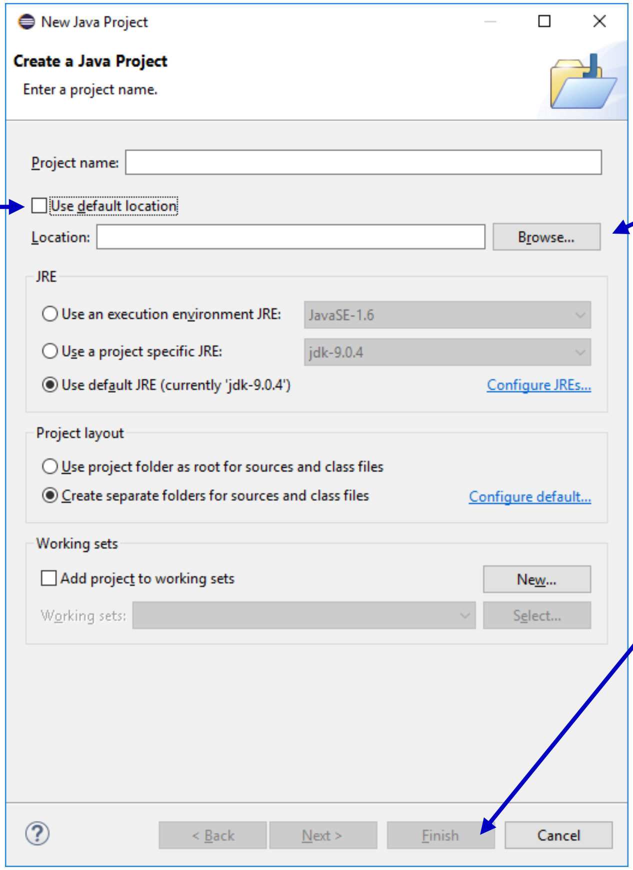
FASTSim-Java [main] - C:\MyData\EclipseWorkspace\git\FASTS

We will refer to this directory/folder location as
<~FASTSim-Java-Root>

* **<u>Important Note:</u>** if intending to contribute/suggest modifications to the open-source repository on GitHub, first go to GitHub and create “your branch” (from “main” or another branch), then clone your branch (not “main”) into Eclipse

Creating a Project from Cloned Repository

- ❑ Use the menu commands File -> New -> Java Project



New Java Project

Create a Java Project

Enter a project name.

Project name:

☐ Use default location

Location:

JRE

○ Use an execution environment JRE: JavaSE-1.6

○ Use a project specific JRE: jdk-9.0.4

● Use default JRE (currently 'jdk-9.0.4')

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:

1 **De-select “Use Default Location”**

2 **Use the “Browse…” button to navigate/select the folder**

<~FASTSim-Java-Root>\fastsimjava

This should be the folder that includes a sub-folder named ‘src’

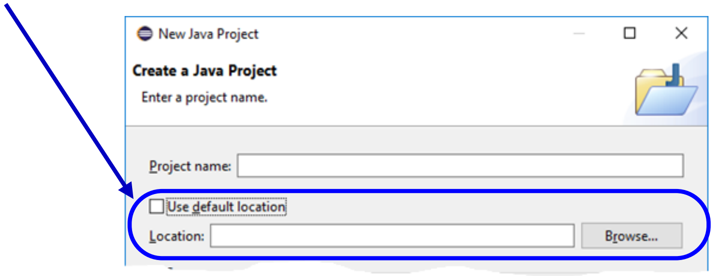
3 **Accept everything else as default, then click “Finish”**

- ❑ The Package Explorer -tab of Eclipse should now show a *new project* named **fastsimjava [FASTSim-Java main]**

Linking Projects

- ❑ Repeat the Steps for Cloning from GitHub Repository
  - Use URL: https://github.com/khamza075/PVC
  - We will refer to the locally saved directory/folder on disk as **<~PVC-Root>**
- ❑ Repeat the Steps for Creating a Project from Cloned Repository
  - For “Location” use **<~PVC-Root>\source\pvc**

This should be the folder that includes a sub-folder named ’src’



New Java Project

Create a Java Project

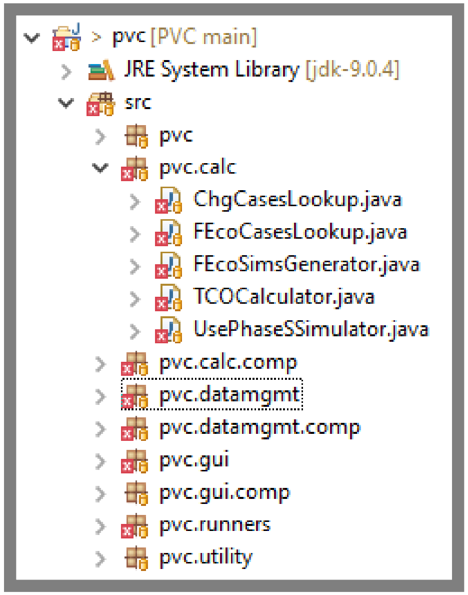
Enter a project name.

Project name:

☐ Use default location

Location:

- ❑ The Package Explorer -tab of Eclipse should now show a *new project* named **pvc [PVC main]** that includes errors -- this is simply because the dependencies of PVC on FASTSim-Java have not yet been recognized by the IDE



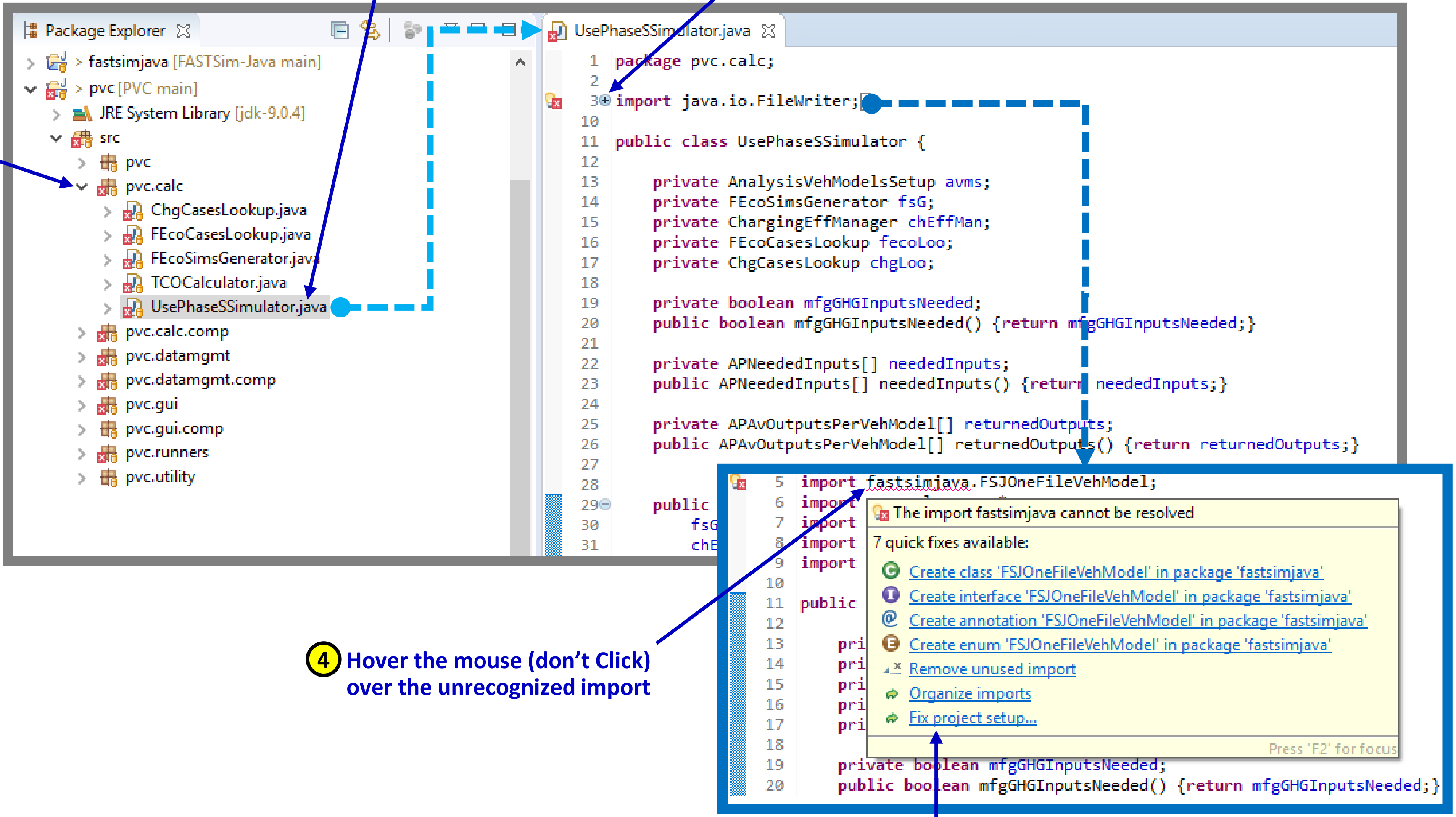
```
> pvc [PVC main]
  JRE System Library [jdk-9.0.4]
  src
    pvc
    pvc.calc
      ChgCasesLookup.java
      FEcoCasesLookup.java
      FEcoSimsGenerator.java
      TCOCalculator.java
      UsePhaseSSimulator.java
    pvc.calc.comp
    pvc.datamgmt
    pvc.datamgmt.comp
    pvc.gui
    pvc.gui.comp
    pvc.runners
    pvc.utility
```

Linking Projects (Cont’d)

1 Click to collapse any of the sub-packages that include errors

2 Double-Click any of the Classes that include errors to open them

3 Click the ‘+’ sign in import statement that includes error to show all import statements



```
package pvc.calc;

import java.io.FileWriter;

public class UsePhaseSSimulator {

    private AnalysisVehModelsSetup avms;
    private FEcoSimsGenerator fsG;
    private ChargingEffManager chEffMan;
    private FEcoCasesLookup fecoLoo;
    private ChgCasesLookup chgLoo;

    private boolean mfgGHGInputsNeeded;
    public boolean mfgGHGInputsNeeded() {return mfgGHGInputsNeeded;}

    private APNeededInputs[] neededInputs;
    public APNeededInputs[] neededInputs() {return neededInputs;}

    private APAvOutputsPerVehModel[] returnedOutputs;
    public APAvOutputsPerVehModel[] returnedOutputs() {return returnedOutputs;}


    public
        fsG
        chE
```

```
import fastsimjava.FSJOneFileVehModel;
import
import
import
import

public

    pri
    pri
    pri
    pri
    pri

    private boolean mfgGHGInputsNeeded;
    public boolean mfgGHGInputsNeeded() {return mfgGHGInputsNeeded;}
```

The import fastsimjava cannot be resolved

7 quick fixes available:

- Create class 'FSJOneFileVehModel' in package 'fastsimjava'
- Create interface 'FSJOneFileVehModel' in package 'fastsimjava'
- Create annotation 'FSJOneFileVehModel' in package 'fastsimjava'
- Create enum 'FSJOneFileVehModel' in package 'fastsimjava'
- Remove unused import
- Organize imports
- Fix project setup...

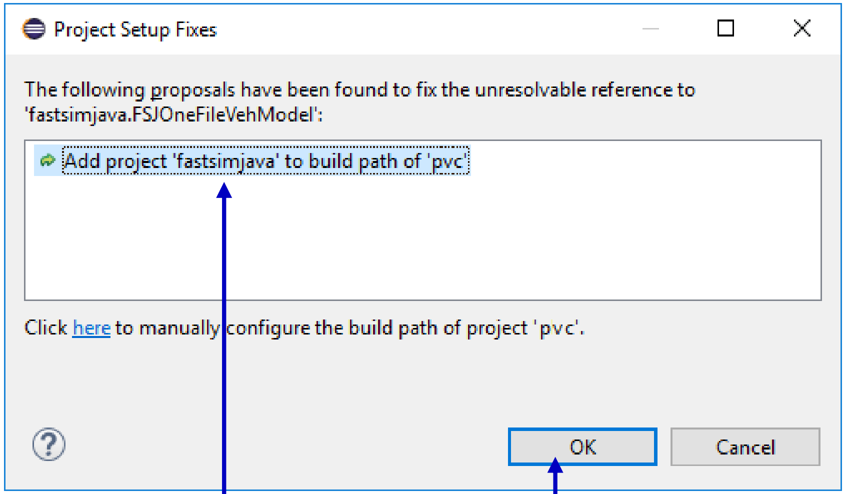
Press 'F2' for focus

4 Hover the mouse (don’t Click) over the unrecognized import

5 Select “Fix project setup” emissions

Linking Projects (Cont'd)

- ❑ Eclipse IDE *should* automatically be suggesting the correct project to link, if not, use manual configuration to add it



Project Setup Fixes

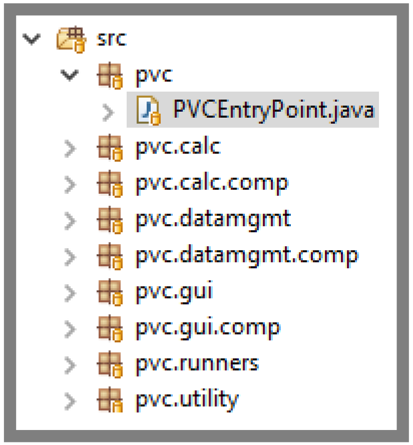
The following proposals have been found to fix the unresolvable reference to 'fastsimjava.FSJOneFileVehModel':

Add project 'fastsimjava' to build path of 'pvc'

Click here to manually configure the build path of project 'pvc'.

6 Accept the suggestion to add 'fastsimjava'

- ❑ This should resolve all the previously shown errors in the Project Explorer -tab



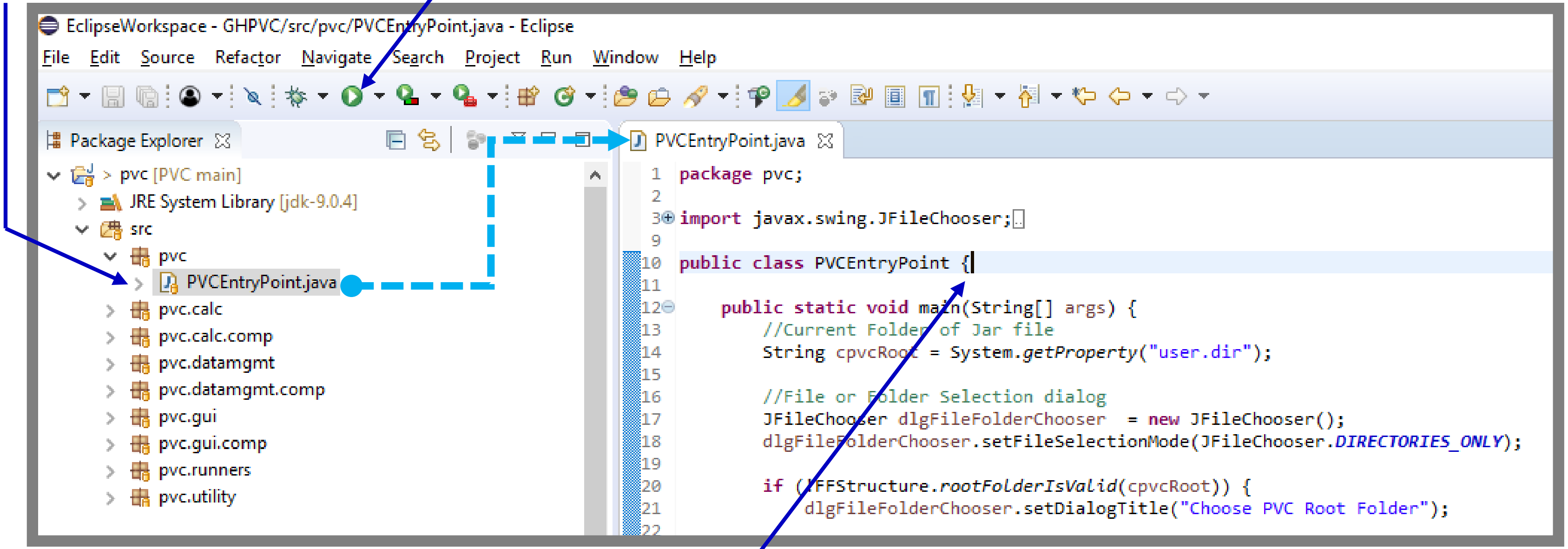
src
- pvc
  - PVCEntryPoint.java
- pvc.calc
- pvc.calc.comp
- pvc.datamgmt
- pvc.datamgmt.comp
- pvc.gui
- pvc.gui.comp
- pvc.runners
- pvc.utility

Running PVC within Eclipse IDE

- ❑ <u>**Important Note**</u>: For PVC to correctly run, it requires a certain minimum amount of data to be arranged in a certain directory/folder structure
  - To obtain a sample and prepare the directory structure, Locate the file named "**PVC-Public.zip**" (in /data/ part of PVC repository) and unzip it to any location of choice on local disk
  - We will refer to the unzip directory/folder location on disk (which includes a file named "_lastAnalyzed.csv") as **<~PVC-Data-Root>**
- ❑ Using the Package Explorer -tab of Eclipse

1 Locate the class "PVCEntryPoint" & double-click to open it

3 Click the "run" icon

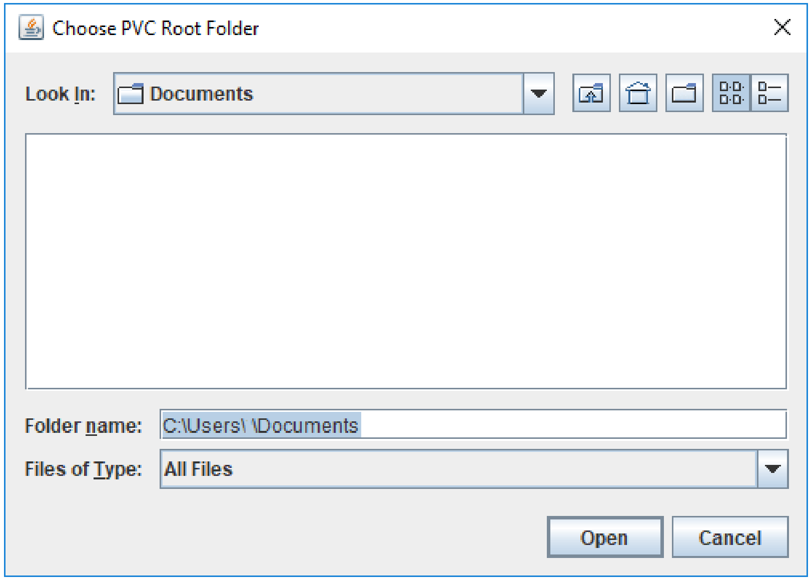


```java
package pvc;

import javax.swing.JFileChooser;

public class PVCEntryPoint {

    public static void main(String[] args) {
        //Current Folder of Jar file
        String cpvcRoot = System.getProperty("user.dir");

        //File or Folder Selection dialog
        JFileChooser dlgFileFolderChooser  = new JFileChooser();
        dlgFileFolderChooser.setFileSelectionMode(JFileChooser.DIRECTORIES_ONLY);

        if (!FFStructure.rootFolderIsValid(cpvcRoot)) {
            dlgFileFolderChooser.setDialogTitle("Choose PVC Root Folder");
```

2 Place the mouse cursor somewhere within the code section of the class

Running PVC within Eclipse IDE (Cont’d)

- ❑ As PVC code runs within the IDE it will not automatically locate the unzipped data, and will prompt the user for its location



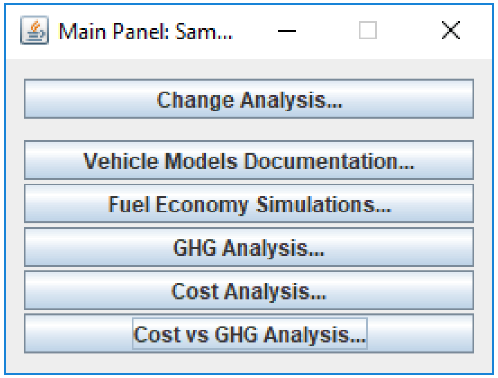
Choose PVC Root Folder

Look In: Documents

Folder name: C:\Users\ \Documents

Files of Type: All Files

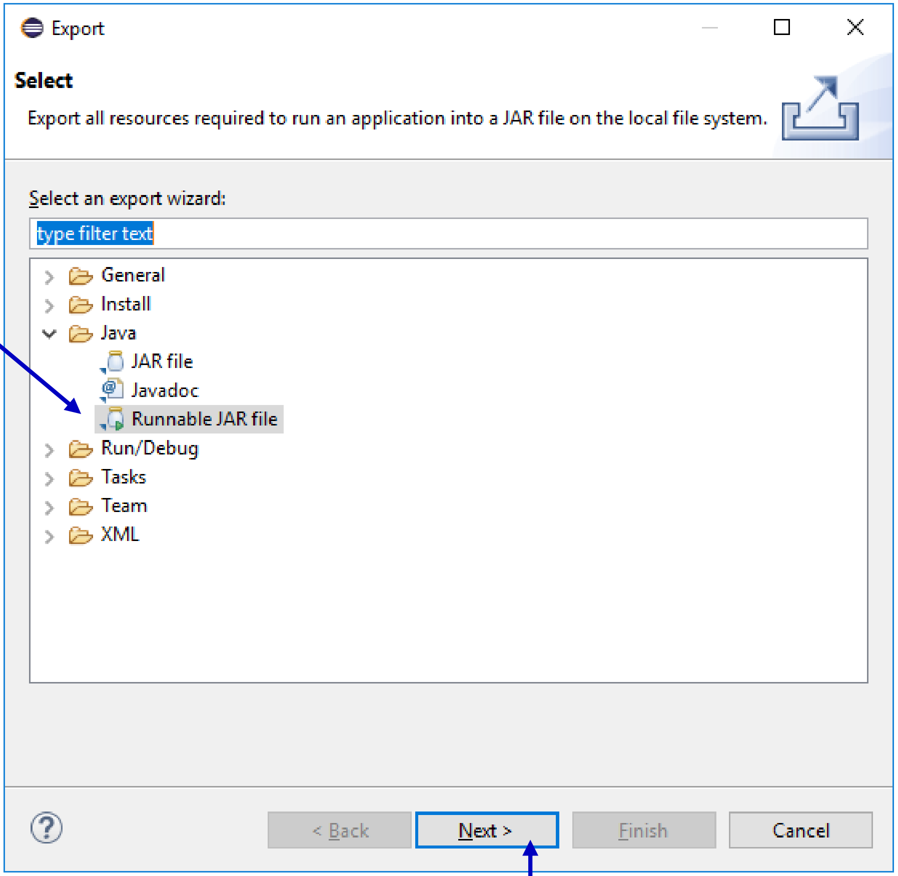
- ❑ Navigate to **<~PVC-Data-Root>** then click “Open”, which will then display the main panel of PVC software*



* Refer to PVC User Guide for more details on how to use the software

Compiling to Runnable .jar File

- ❑ Within the main menus of Eclipse IDE, select **File -> Export…**



Export all resources required to run an application into a JAR file on the local file system.

1 Select "Runnable JAR file"

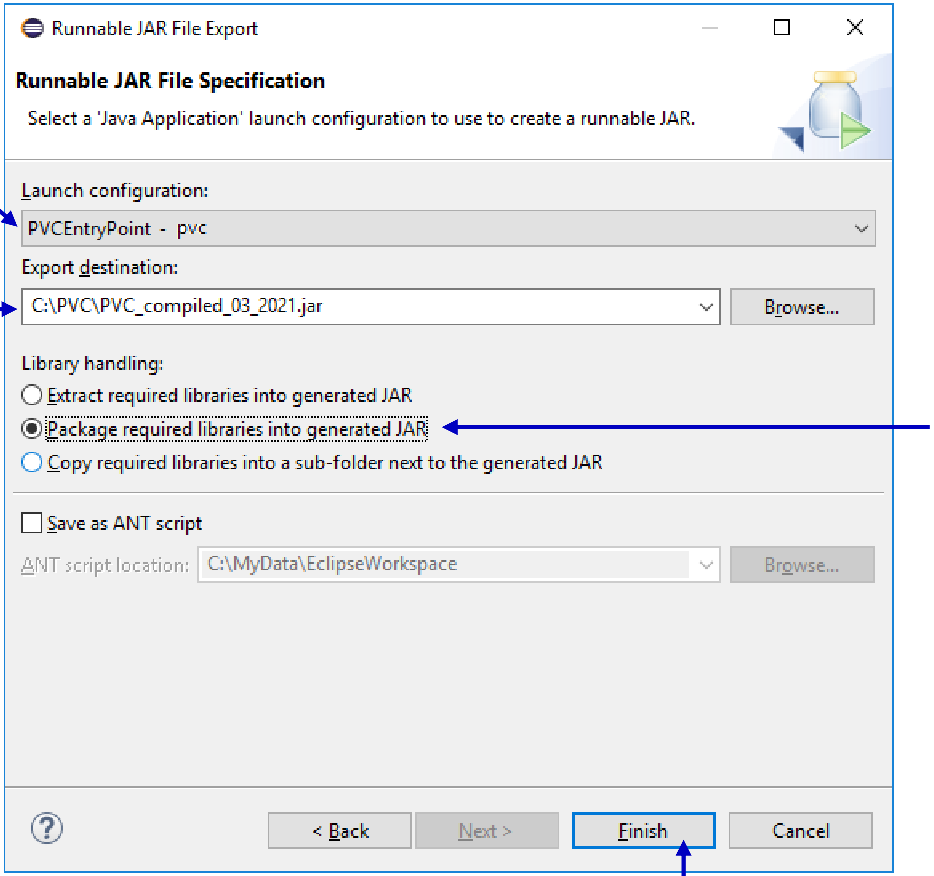
2 Click "Next"

Compiling to Runnable .jar File (Cont'd)

3 **Ensure that PVCEntryPoint (from the correct project in Eclipse) is the Launch Configuration**

4 **Set the destination output folder and name of the runnable .jar file**

It is recommend that the output folder should be **<~PVC-Data-Root>** so that double-clicking the .jar file (to run), lets it to automatically recognize the location of data files (instead of prompting the user)



Runnable JAR File Export

Runnable JAR File Specification

Select a 'Java Application' launch configuration to use to create a runnable JAR.

Launch configuration: PVCEntryPoint - pvc

Export destination: C:\PVC\PVC_compiled_03_2021.jar

Library handling:

○ Extract required libraries into generated JAR

● Package required libraries into generated JAR

○ Copy required libraries into a sub-folder next to the generated JAR

□ Save as ANT script

ANT script location: C:\MyData\EclipseWorkspace

5 **Select "Package required libraries into generated JAR"**

6 **Click "Finish"**