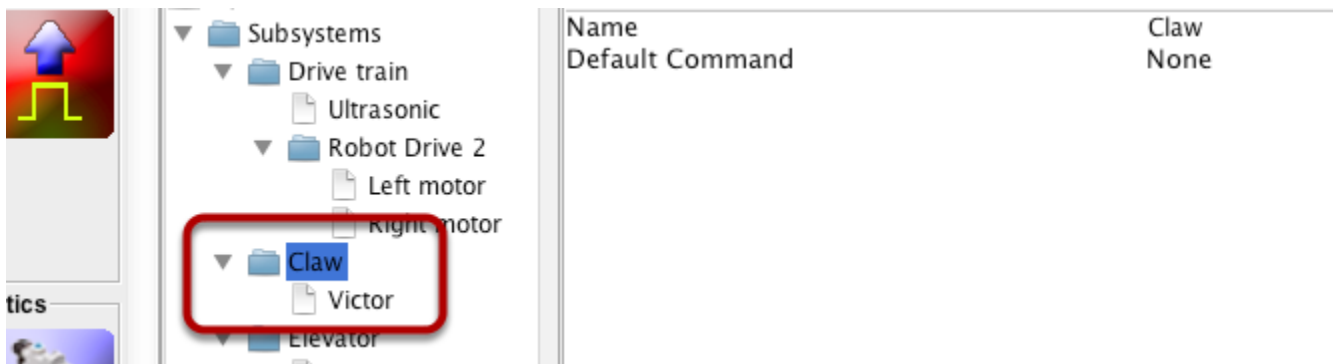


Writing the code for a subsystem in Java

Writing the code for a subsystem in Java

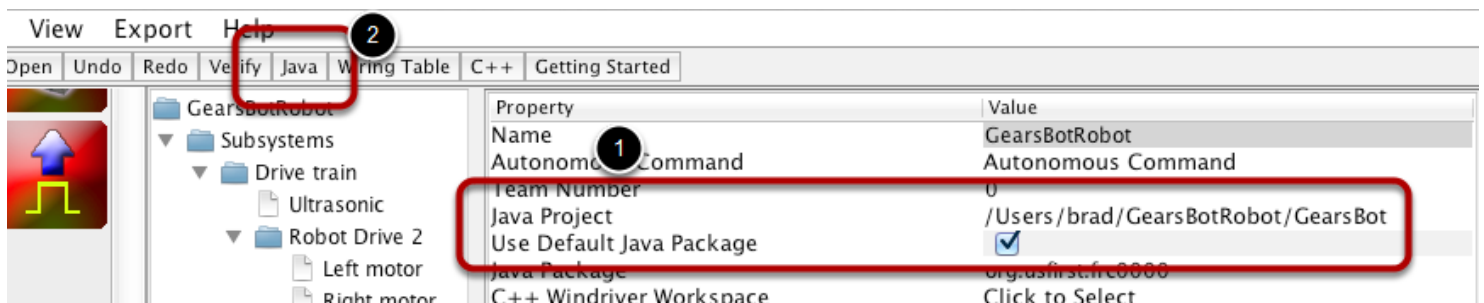
Adding code to create an actual working subsystem is very straightforward. For simple subsystems that don't use feedback it turns out to be extremely simple. In this section we will look at an example of a Claw subsystem that operates the motor for some amount of time to open or close a claw on the robot arm.

Create the subsystem



Be sure that the subsystem is defined in the RobotBuilder robot description. The Claw subsystem has a single Victor and no sensors since the claw motor operates for one second in either direction to open or close the claw.

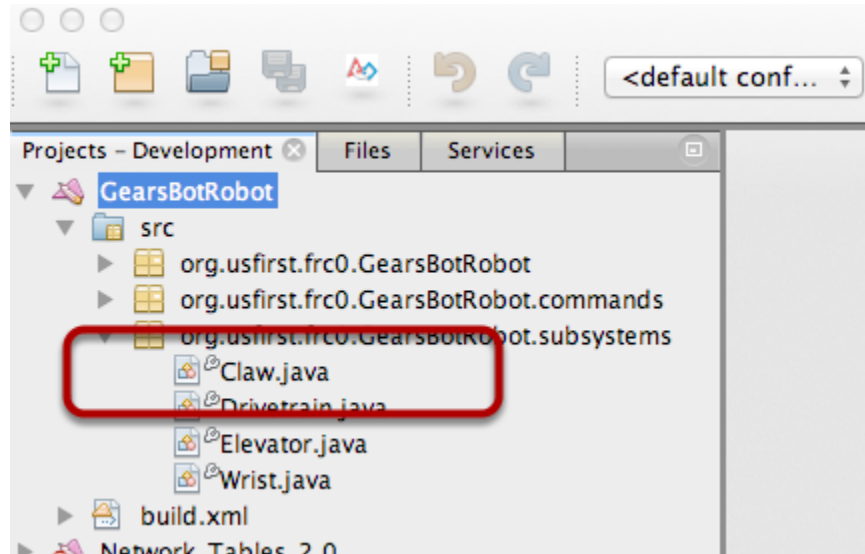
Generate code for the project



Verify that the java project location is set up (1) and generate code for the robot project (2).

Writing the code for a subsystem in Java

Open the project in Netbeans



Open the generated project in Netbeans and notice the subsystems package containing each of the subsystem files. Open the Claw.java file to add code that will open and close the claw.

Writing the code for a subsystem in Java

Add methods to open, close, and stop the claw

```
2
3 package org.usfirst.frc0.GearsBotRobot.subsystems;
4
5 import edu.wpi.first.wpilibj.*;
6 import edu.wpi.first.wpilibj.command.Subsystem;
7 import org.usfirst.frc0.GearsBotRobot.RobotMap;
8
9
10 public class Claw extends Subsystem {
11     // BEGIN AUTOGENERATED CODE, SOURCE=ROBOTBUILDER ID=DECLARATIONS
12     Victor victor = RobotMap.CLAW_VICTOR;
13     // END AUTOGENERATED CODE, SOURCE=ROBOTBUILDER ID=DECLARATIONS
14
15     public void initDefaultCommand() {
16         // BEGIN AUTOGENERATED CODE, SOURCE=ROBOTBUILDER ID=DEFAULT_COMMAND
17         // END AUTOGENERATED CODE, SOURCE=ROBOTBUILDER ID=DEFAULT_COMMAND
18     }
19
20     public void openClaw() {
21         victor.set(1.0);
22     }
23
24     public void closeClaw() {
25         victor.set(-1.0);
26     }
27
28     public void stop() {
29         victor.set(0.0);
30     }
31
32 }
33
34
35
```

Add methods to the claw.java that will open, close, and stop the claw from moving. Those will be used by commands that actually operate the claw. The comments have been removed from this file to make it easier to see the changes for this document. Notice that a member variable called "victor" is created by RobotBuilder so it can be used throughout the subsystem. Each of your dragged-in palette items will have a member variable with the name given in RobotBuilder.

See: [Writing the code for a command in Java](#) to see how to get this Claw subsystem to operate using commands. See: [Writing the code for a PIDSubsystem in Java](#) to write the code for a more complex subsystem with feedback (PIDSubsystem).