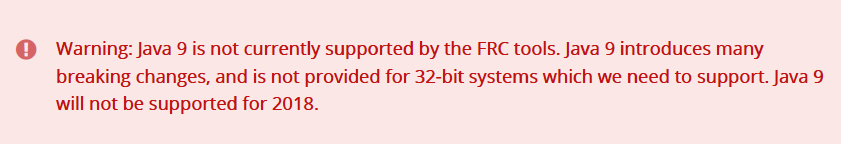
# Java Porting Guide - 2017 to 2018

When Java teams look at the WPILib APIs for 2018 they should see something that looks very familiar. However, working with third party CTRE Phoenix Framework v5.x.x.x from the previous CTRE Toolsuite 4.4.1.14 we will see noticeable changes. Many changes have occurred with interfaces with speed controllers and robot drive classes.

CANTalon has been removed from WPILib. See this [link](https://github.com/CrossTheRoadElec/Phoenix-Documentation/blob/master/Migration%20Guide.md) for more info and find the [CTRE Toolsuite installer here](http://www.ctr-electronics.com/control-system/motor-control/talon-srx.html#product_tabs_technical_resources).

The Eclipse plugins have been tested with Eclipse Luna, Eclipse Mars, Eclipse Neon, and Eclipse Oxygen. Teams with existing installs from 2017 can update their installations to 2018 ensuring you have the current setup in Eclipse.



The RobotDrive class has been split into separate classes for different drive base platform types. These classes currently include Differential Drive (common 4wd/6wd/8wd/tank/etc. platforms), Killough Drive (3 omni's) and Mecanum.

Creating a RobotDrive object with CANTalonSXR speed controllers

**Java - 2017**

*Name Space*

**import** com.ctre.CANTalon;

**import** edu.wpi.first.wpilibj.RobotDrive;

*Constructor*

//Drive Train Declares

**public** **static** CANTalon *leftFrontTalonSRX*;

**public** **static** CANTalon *leftRearTalonSRX*;

**public** **static** CANTalon *rightFrontTalonSRX*;

**public** **static** CANTalon *rightRearTalonSRX*;

**public** **static** RobotDrive *driveTrainRobotDrive*;

//Drive motor declares (Drive #1-4)

*leftFrontTalonSRX*  = new CANTalon(1);

*leftRearTalonSRX*  = new CANTalon(2);

*rightFrontTalonSRX*  = new CANTalon(3);

*rightRearTalonSRX* = new CANTalon(4);

//Creates the new robot drive to pass to subsystem

*driveTrainRobotDrive* = new RobotDrive(*leftFrontTalonSRX*, *leftRearTalonSRX*, *rightFrontTalonSRX*,

*rightRearTalonSRX*);

*Parameters*

*Joystick inputs from stickY, stickX*

robotDrive*.arcadeDrive(stickY, stickX,* **false***);*

**Java – 2018**

*Name Space*

**import** com.ctre.phoenix.motorcontrol.can.WPI\_TalonSRX;

**import** edu.wpi.first.wpilibj.SpeedControllerGroup;

**import** edu.wpi.first.wpilibj.drive.DifferentialDrive;

*Constructor*

//Declare Drive Train

**public** **static** WPI\_TalonSRX *leftFrontTalonSRX*;

**public** **static** WPI\_TalonSRX *leftRearTalonSRX*;

**public** **static** WPI\_TalonSRX *rightFrontTalonSRX*;

**public** **static** WPI\_TalonSRX *rightRearTalonSRX*;

**public** **static** DifferentialDrive *drivetrainRobotDrive41*;

**public** **static** SpeedControllerGroup *leftDrive*;

**public** **static** SpeedControllerGroup *rightDrive*;

//Declare each speed controller used

*leftFrontTalonSRX* = **new** WPI\_TalonSRX (1);

*leftRearTalonSRX* = **new** WPI\_TalonSRX (2);

*rightFrontTalonSRX* = **new** WPI\_TalonSRX (3);

*rightRearTalonSRX* = **new** WPI\_TalonSRX (4);

//set each speed controller group

*leftDrive* = **new** SpeedControllerGroup(*leftFrontTalonSRX*.getWPILIB\_SpeedController(),

*leftRearTalonSRX*.getWPILIB\_SpeedController());

*rightDrive* = **new** SpeedControllerGroup(*rightFrontTalonSRX*.getWPILIB\_SpeedController(),

*rightRearTalonSRX*.getWPILIB\_SpeedController());

//set differential drive to each speed controller group

*drivetrainRobotDrive41* = **new** DifferentialDrive(*leftDrive*, *rightDrive*);

*Parameters*

robotDrive41.arcadeDrive(stickX, stickY, **false**);

Creating a Single Motor object with CANTalonSXR speed controllers

**Java – 2017**

*Name Space*

**import** com.ctre.CANTalon;

*Constructor*

//Single Motor declare

**public** **static** CANTalon *singleMotor1*;

*singleMotor1*  = new CANTalon(1);

*Parameters*

//Sets motor output for full speed

singleMotor1.set(1.0);

**Java – 2018**

*Name Space*

**import** com.ctre.phoenix.motorcontrol.can.WPI\_TalonSRX;

**import** com.ctre.phoenix.motorcontrol.ControlMode;

*Constructor*

**public** **static** WPI\_TalonSRX *singleMotor1*;

*singleMotor1* = **new** TalonSRX(1);

*Parameters*

singleMotor1.set(ControlMode.***PercentOutput***,1.0);

***\*Refer to the CTRE documentation for further information on control modes.***