

Tools of the Trade

This chapter will go over some of the tooling we use to make our code functional, efficient, automated, and easy to debug.

- [Table of Contents for Software Tools](#)
- [Current Solution for PhotonVision 2027](#)

Table of Contents for Software Tools

Table of Contents with links for all the software tools we use to code robots here at Yeti!

Git - A version control system widely used in the industry and on our team that helps track code changes, manage branches, and collaboration with the team.

Github - Online platform for hosting Git repositories, allows for code sharing, pull requests, and team collaboration. This is where we have all of our code stored, viewable publicly.

WPILib - The official library used by us and all other FRC Teams to code robot subsystems, motors, sensors, and much more. Should answer a good amount of questions you have, or at least lead you to somewhere where you can find an answer!

Phoenix 6 - CTRE's Motor Controller Library that allows us to configure and support devices such as TalonFX. Also really helpful and can answer a lot of the questions you have regarding Motor Control.

Phoenix Tuner - CTRE's application that allows you to configure, test, and update CTRE CAN devices like Talons. Helpful documentation regarding tuning as well.

PhotonVision - An open-source vision system that detects AprilTags and other such objects and sends pose data to the robot. Helpful documentation regarding vision related questions.

Limelight - Another vision system that detects AprilTags and other such objects, similar to PhotonVision. Helpful documentation regarding vision related questions.

AdvantageKit - A logging framework that records all robot data and actions. We plan to experiment with this as a team this season, so it has very helpful documentation!

AdvantageScope - A visual dashboard that reads logs from AdvantageKit and displays field position, sensor data, and timelines in real time or replay. This is also something we use a lot as a team!

Localization - The process used to calculate the robot's position on the field using data from encoders and vision inputs like AprilTags.

Driverstation - FIRST's official software that we use to control the robot during matches, viewing system status, and switching between robot modes.

Simulation - A WPILib feature that allows teams to test robot code on a virtual field without needing a real robot.

Path Planner - A path generation tool that lets teams design and follow autonomous paths, especially for swerve and tank drivetrains. We use this as a team to generate autonomous paths!

Choreo - Another path generation tool that lets team design and follow autonomous paths. We are looking into Choreo this off-season, and comparing it with Path Planner to see which one is better for our team to use!

Elastic and **Elastic Documentation** - A customizable FRC dashboard that connects to NetworkTables for live monitoring and debugging. This is the type of FRC Dashboard we use at YETI!

Network Tables - A public-subscribe messaging system that is used to communicate/send data between the robot, driver station, and other such things.

Current Solution for PhotonVision 2027

With SystemCore still being in development, PhotonVision does not have an official release for the vendordeps.

1. Navigate to the Actions tab on PhotonVision's Github page.
2. Click Build on the left sidebar and in the search bar on the top right type branch:2027
3. Scroll down and download "photonlib-offline"
4. Once it's downloaded extract everything inside.
5. You will have to extract twice, because the contents inside the first folder are a compressed file.
6. Once everything is extracted, first copy the .json file inside vendordeps.
7. Place that file inside the vendordeps file of wpilib (found for Windows in C:\Users\Public\wpilib\2027_alpha1)
8. Next, return to photonlib-offline and copy the photonvision folder inside the maven folder in org.
9. Place that photonvision folder inside that same org andmaven folder of wpilib (for Windows it's C:\Users\Public\wpilib\2027_alpha1\maven\org)
10. Run build and everything should work!