

Building and downloading a robot project to the roboRIO

There are two ways of running programs onto the roboRIO. You can

1. Attach to the roboRIO and run the program using the debugger from your development system OR
2. Load it onto the roboRIO flash drive so it will run on reboot.

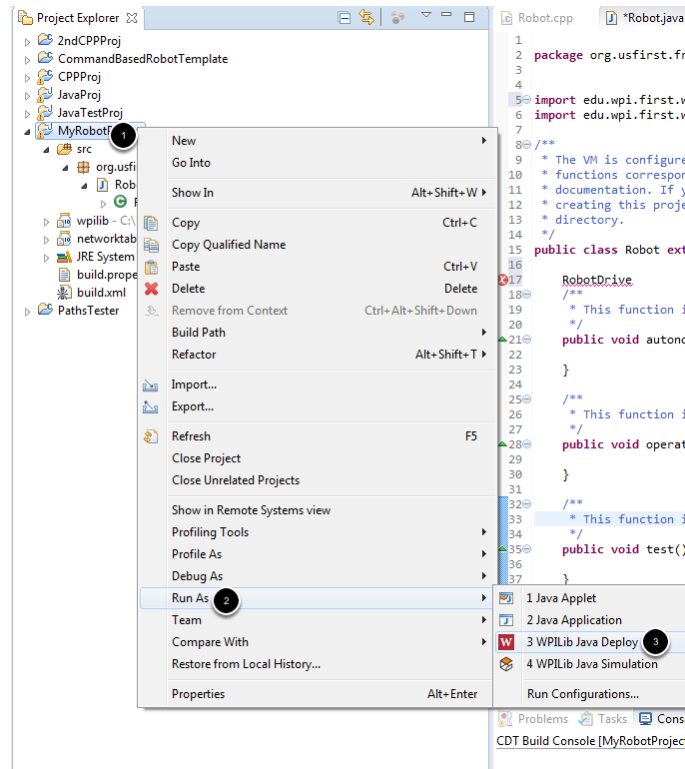
For tournaments you should always download the program so that it will be there when the robot is restarted and the match is played. This article will cover loading the program onto the roboRIO to run on reboot. The next article will cover the debugger.

Building a robot program

The first step to getting the program on the roboRIO is to build the code. This will compile and link the project files. If you have set the option to build automatically as recommended in the Eclipse installation article, this step is done each time you save a file in the project. If not, this step will be done when you start the deploy process.

Building and downloading a robot project to the roboRIO

Loading a program to run on robot startup



To have the program run automatically each time the roboRIO boots, it must be loaded in the roboRIO flash memory. For tournaments you should always download the program so that it will be there when the robot is restarted and the match is played. To do this:

1. Right click on the project in the Project Explorer
2. Select "Run As"
3. Select "WPIlib Java Deploy"

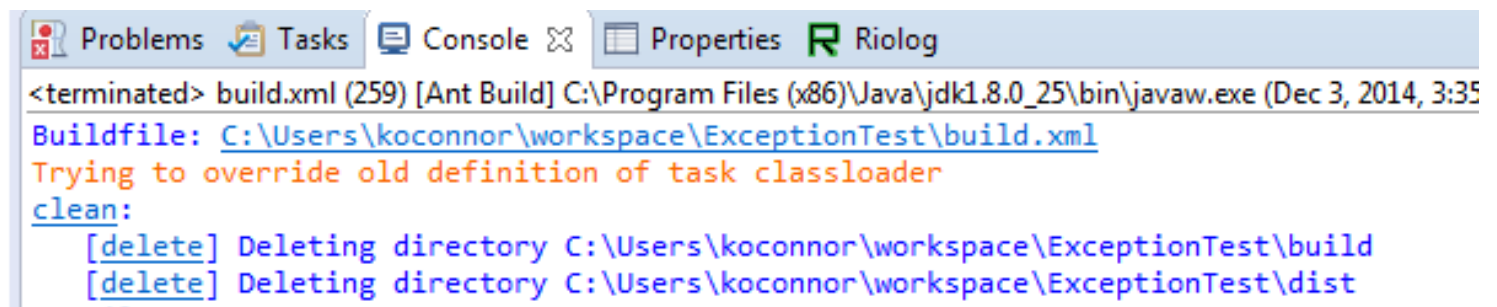
The deployment process will then begin and start outputting status information to the Console (generally located in the bottom center or bottom right of the window). After the program is downloaded it will be immediately launched, there is no need to reboot the roboRIO.

Building and downloading a robot project to the roboRIO

Program Download Process

After you click the WPILib Java Deploy button the deploy process will proceed through a number of steps.

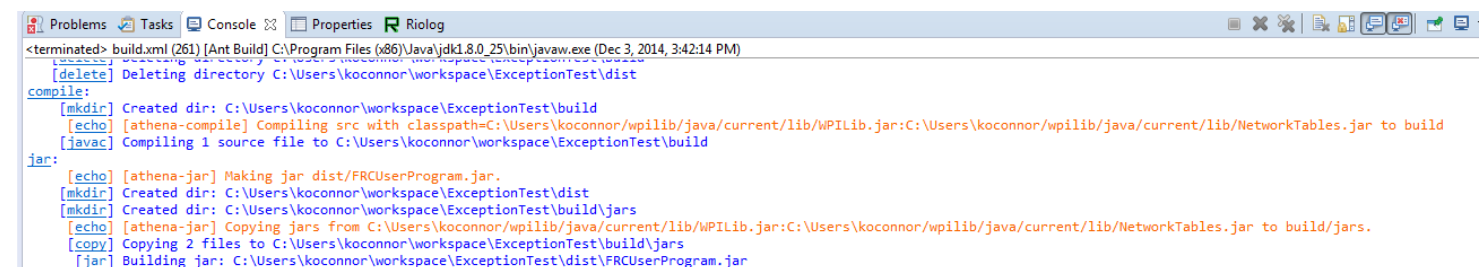
Clean



```
Problems Tasks Console Properties Riolog
<terminated> build.xml (259) [Ant Build] C:\Program Files (x86)\Java\jdk1.8.0_25\bin\javaw.exe (Dec 3, 2014, 3:35
Buildfile: C:\Users\koconnor\workspace\ExceptionTest\build.xml
Trying to override old definition of task classloader
clean:
[delete] Deleting directory C:\Users\koconnor\workspace\ExceptionTest\build
[delete] Deleting directory C:\Users\koconnor\workspace\ExceptionTest\dist
```

The deploy process runs a "clean" each time it is run. This is needed to work around a bug in Ant that causes changes to static final constants to not always be picked up by the build.

Compile & Jar

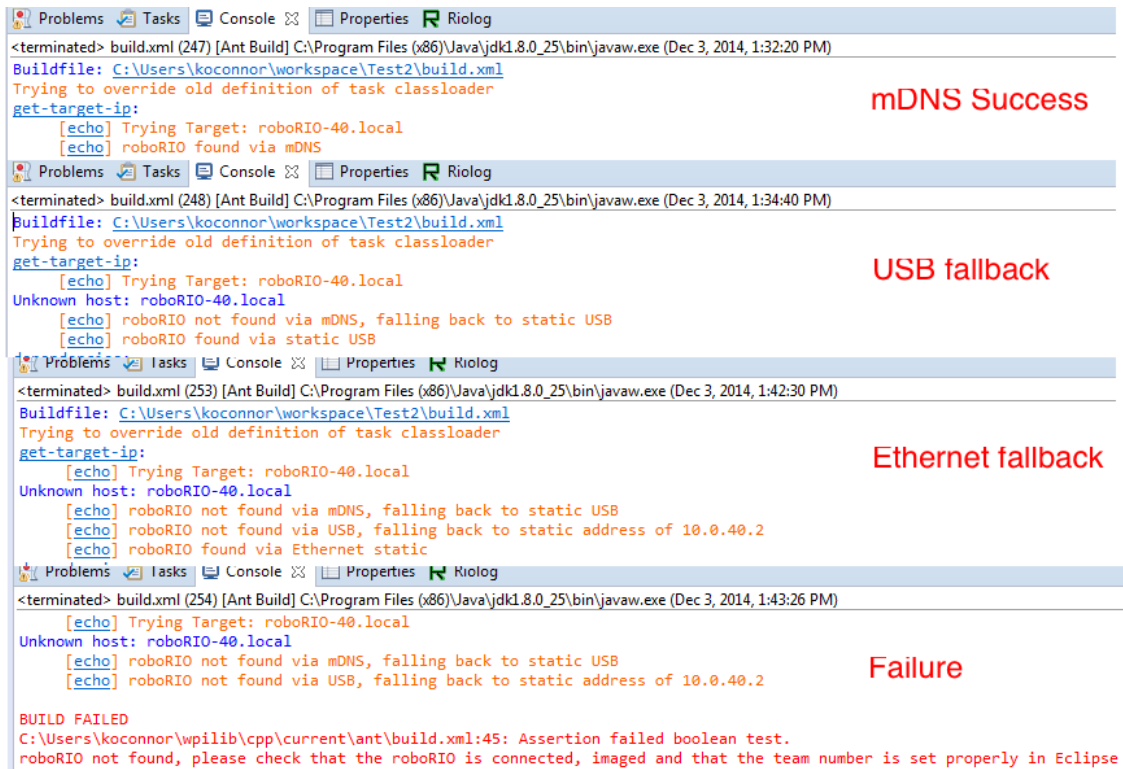


```
Problems Tasks Console Properties Riolog
<terminated> build.xml (261) [Ant Build] C:\Program Files (x86)\Java\jdk1.8.0_25\bin\javaw.exe (Dec 3, 2014, 3:42:14 PM)
[delete] Deleting directory C:\Users\koconnor\workspace\ExceptionTest\dist
compile:
[mkdir] Created dir: C:\Users\koconnor\workspace\ExceptionTest\build
[echo] [athena-compile] Compiling src with classpath=C:\Users\koconnor\wpilib\java\current\lib\WPILib.jar:C:\Users\koconnor\wpilib\java\current\lib\NetworkTables.jar to build
[javac] Compiling 1 source file to C:\Users\koconnor\workspace\ExceptionTest\build
jar:
[echo] [athena-jar] Making jar dist\FRCUserProgram.jar.
[mkdir] Created dir: C:\Users\koconnor\workspace\ExceptionTest\dist
[mkdir] Created dir: C:\Users\koconnor\workspace\ExceptionTest\build\jars
[echo] [athena-jar] Copying jars from C:\Users\koconnor\wpilib\java\current\lib\WPILib.jar:C:\Users\koconnor\wpilib\java\current\lib\NetworkTables.jar to build\jars.
[copy] Copying 2 files to C:\Users\koconnor\workspace\ExceptionTest\build\jars
[jar] Building jar: C:\Users\koconnor\workspace\ExceptionTest\dist\FRCUserProgram.jar
```

Next the code is compiled and packed up into a jar file.

Building and downloading a robot project to the roboRIO

Get Target IP



```
<terminated> build.xml (247) [Ant Build] C:\Program Files (x86)\Java\jdk1.8.0_25\bin\javaw.exe (Dec 3, 2014, 1:32:20 PM)
Buildfile: C:\Users\koconnor\workspace\Test2\build.xml
Trying to override old definition of task classloader
get-target-ip:
[echo] Trying Target: roboRIO-40.local
[echo] roboRIO found via mDNS
mDNS Success

<terminated> build.xml (248) [Ant Build] C:\Program Files (x86)\Java\jdk1.8.0_25\bin\javaw.exe (Dec 3, 2014, 1:34:40 PM)
Buildfile: C:\Users\koconnor\workspace\Test2\build.xml
Trying to override old definition of task classloader
get-target-ip:
[echo] Trying Target: roboRIO-40.local
Unknown host: roboRIO-40.local
[echo] roboRIO not found via mDNS, falling back to static USB
[echo] roboRIO found via static USB
USB fallback

<terminated> build.xml (253) [Ant Build] C:\Program Files (x86)\Java\jdk1.8.0_25\bin\javaw.exe (Dec 3, 2014, 1:42:30 PM)
Buildfile: C:\Users\koconnor\workspace\Test2\build.xml
Trying to override old definition of task classloader
get-target-ip:
[echo] Trying Target: roboRIO-40.local
Unknown host: roboRIO-40.local
[echo] roboRIO not found via mDNS, falling back to static USB
[echo] roboRIO not found via USB, falling back to static address of 10.0.40.2
[echo] roboRIO found via Ethernet static
Ethernet fallback

<terminated> build.xml (254) [Ant Build] C:\Program Files (x86)\Java\jdk1.8.0_25\bin\javaw.exe (Dec 3, 2014, 1:43:26 PM)
[echo] Trying Target: roboRIO-40.local
Unknown host: roboRIO-40.local
[echo] roboRIO not found via mDNS, falling back to static USB
[echo] roboRIO not found via USB, falling back to static address of 10.0.40.2
Failure

BUILD FAILED
C:\Users\koconnor\wpilib\cpp\current\ant\build.xml:45: Assertion failed boolean test.
roboRIO not found, please check that the roboRIO is connected, imaged and that the team number is set properly in Eclipse
```

In this step, the plugin determines the IP of the roboRIO. The plugin will try to reach the roboRIO via the following methods, in order:

1. mDNS (roboRIO-TEAM-FRC.local where TEAM is your team number)
2. DNS (roboRIO-TEAM-FRC.lan where TEAM is your team number)
3. USB static IP (172.22.11.2)
4. Ethernet static IP (10.TE.AM.2)

If the roboRIO is accessible via any of these methods, the deploy will proceed using that address as the target. If the roboRIO cannot be found using any of these methods, the deploy will abort. See [Connection Failure in the Troubleshooting](#) section below.

Building and downloading a robot project to the roboRIO

Dependencies

```
buildfile: C:\Users\koconnor\workspace\testz\build.xml
Trying to override old definition of task classloader
get-target-ip:
[echo] Trying Target: roboRIO-40.local
[echo] roboRIO found via mDNS
dependencies:
[echo] roboRIO image version validated
deploy:
get-target-ip:
[echo] Trying Target: roboRIO-40.local
[echo] roboRIO found via mDNS
dependencies:
BUILD FAILED
C:\Users\koconnor\wpilib\cpp\current\ant\build.xml:119: Assertion failed boolean test.
roboRIO Image does not match plugin, allowed image version: 19
```

Image OK

Image mismatch

```
[echo] roboRIO image version validated
[echo] Checking for JRE. If this fails install the JRE using these instructions: https://wpilib
[sshexec] Connecting to roboRIO-40.local:22
[sshexec] cmd : test -d /usr/local/frc/JRE
deploy:
```

JRE OK

```
dependencies:
[echo] roboRIO image version validated
[echo] Checking for JRE. If this fails install the JRE using these instructions: https://wpilib
[sshexec] Connecting to roboRIO-40.local:22
[sshexec] cmd : test -d /usr/local/frc/JRE
BUILD FAILED
C:\Users\koconnor\wpilib\java\current\ant\build.xml:230: Remote command failed with exit status 1
```

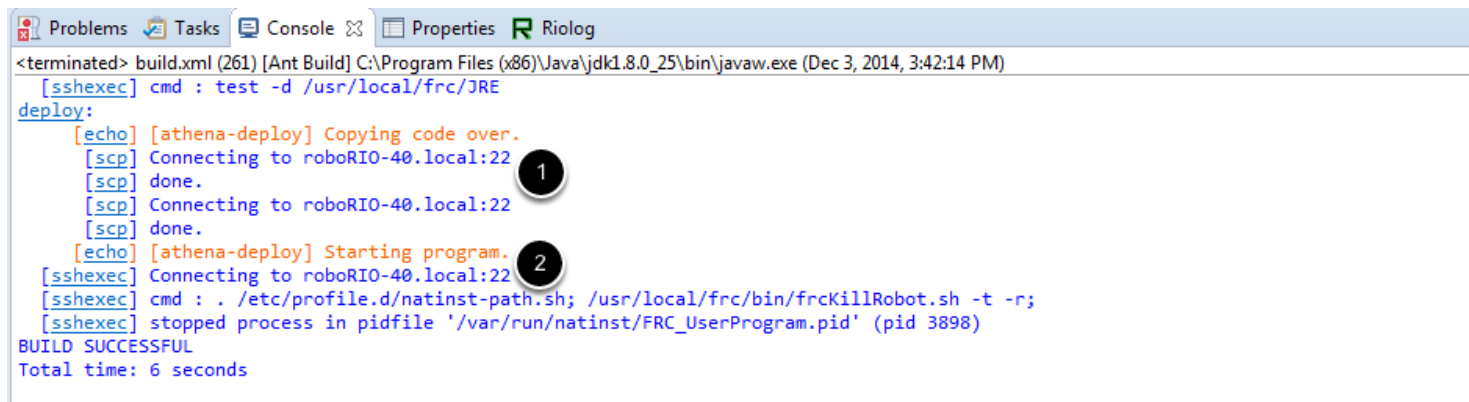
JRE Missing

The dependencies section checks that the roboRIO contains any dependencies necessary for operation. For Java at this time, this consists of two components:

1. Check of the roboRIO image version. If the validation succeeds you will get a message that the version was validated and the deploy will proceed. If the check fails, you will see a message indicating the allowed image version for this version of the plugins, see [Image Check Failure under Troubleshooting](#) below for more details.
2. Check that the JRE is installed on the roboRIO. If not, it is deployed automatically by the plugins.

Building and downloading a robot project to the roboRIO

Copy and Start code



```
<terminated> build.xml (261) [Ant Build] C:\Program Files (x86)\Java\jdk1.8.0_25\bin\javaw.exe (Dec 3, 2014, 3:42:14 PM)
[sshexec] cmd : test -d /usr/local/frc/JRE
deploy:
[echo] [athena-deploy] Copying code over.
[scp] Connecting to roboRIO-40.local:22 1
[scp] done.
[scp] Connecting to roboRIO-40.local:22
[scp] done.
[echo] [athena-deploy] Starting program. 2
[sshexec] Connecting to roboRIO-40.local:22
[sshexec] cmd : . /etc/profile.d/natinst-path.sh; /usr/local/frc/bin/frcKillRobot.sh -t -r;
[sshexec] stopped process in pidfile '/var/run/natinst/FRC_UserProgram.pid' (pid 3898)
BUILD SUCCESSFUL
Total time: 6 seconds
```

The remaining parts of the deploy then proceed as follows:

1. Copy user program and robotCommand file. The robotCommand file tells the framework that launches the code what to launch (C++ vs. Java and Debug vs deploy).
2. Start code. The plugin calls a script file which kills the existing user program, then restarts the LabVIEW RT process. This starts the code in the same way it will be launched when the roboRIO is booted.

Deployed Code

There is no need to reboot the roboRIO to run the new code, it will begin running automatically. After the deploy completes, each time the roboRIO is rebooted it will run the deployed code until new code (from any FRC language) is downloaded.

Viewing program output

The console output from the program does not appear in the regular Eclipse console. To view the console output (System.out statements as well as WPILib output) see the next article, [Using Riolog to view console output](#).

Building and downloading a robot project to the roboRIO

Next Steps

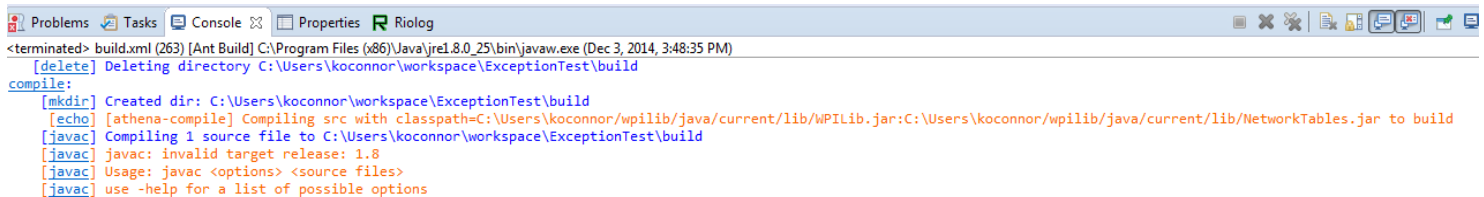
To continue Getting Started with the Control System, see [Programming your radio for home use](#)

To continue reading about Java, scroll down and click Next to read about [Using Riolog to view console output](#).

Troubleshooting

This section contains troubleshooting steps for commonly encountered issues.

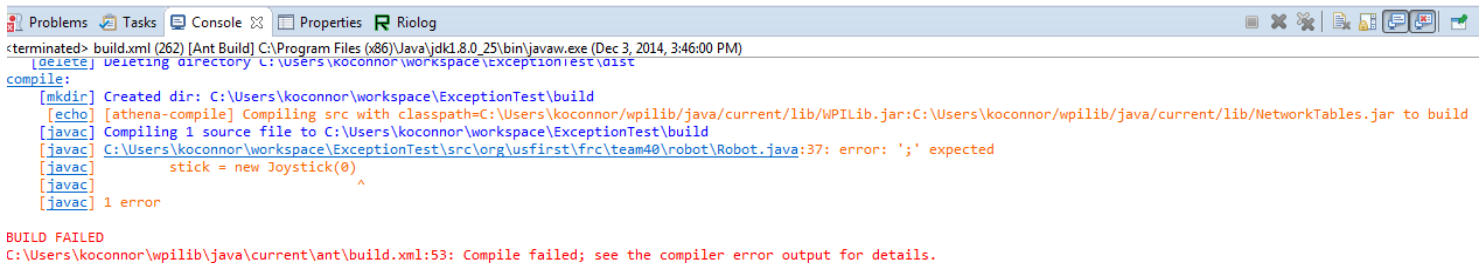
Javac invalid target or Javac not found



```
<terminated> build.xml (263) [Ant Build] C:\Program Files (x86)\Java\jre1.8.0_25\bin\javaw.exe (Dec 3, 2014, 3:48:35 PM)
[delete] Deleting directory C:\Users\koconnor\workspace\ExceptionTest\build
compile:
[mkdir] Created dir: C:\Users\koconnor\workspace\ExceptionTest\build
[echo] [athena-compile] Compiling src with classpath=C:\Users\koconnor\wpilib\java\current\lib\WPILib.jar:C:\Users\koconnor\wpilib\java\current\lib\NetworkTables.jar to build
[javac] Compiling 1 source file to C:\Users\koconnor\workspace\ExceptionTest\build
[javac] javac: invalid target release: 1.8
[javac] Usage: javac <options> <source files>
[javac] use -help for a list of possible options
```

If you see an error that says "javac: invalid target release" or "unable to find a javac compiler" your JDK is not configured properly in Eclipse. Make sure you have [installed a JDK](#) and [configured Eclipse to use that JDK](#) as described in the Installing Eclipse document.

Build Failure



```
<terminated> build.xml (262) [Ant Build] C:\Program Files (x86)\Java\jdk1.8.0_25\bin\javaw.exe (Dec 3, 2014, 3:46:00 PM)
[delete] Deleting directory C:\Users\koconnor\workspace\ExceptionTest\dist
compile:
[mkdir] Created dir: C:\Users\koconnor\workspace\ExceptionTest\build
[echo] [athena-compile] Compiling src with classpath=C:\Users\koconnor\wpilib\java\current\lib\WPILib.jar:C:\Users\koconnor\wpilib\java\current\lib\NetworkTables.jar to build
[javac] Compiling 1 source file to C:\Users\koconnor\workspace\ExceptionTest\build
[javac] C:\Users\koconnor\workspace\ExceptionTest\src\org\usfirst\frs\team40\robot\Robot.java:37: error: ';' expected
[javac]         stick = new Joystick(0)
[javac]                               ^
[javac] 1 error

BUILD FAILED
C:\Users\koconnor\wpilib\java\current\ant\build.xml:53: Compile failed; see the compiler error output for details.
```

Building and downloading a robot project to the roboRIO

A "Compile failed" error as shown above indicates a failure building the code. The lines that start with "javac" will indicate the source of the error (a missing semicolon on line 37 of Robot.java in the example above).

Connection Failure

If the deploy process fails to connect to the roboRIO:

- First verify that the roboRIO is powered on, connected to the PC and that the roboRIO is imaged and that your team number is set properly in Eclipse (Window->Preferences->WPILib Preferences).
- The next step is to verify that your computer networking is configured in a way it can communicate with the roboRIO. Visit the roboRIO Network Troubleshooting document in the Troubleshooting section for more details.
- Check that your firewall is disabled or that an exception has been added for Eclipse. For the Windows Firewall, see Configuring Windows Firewall in the Troubleshooting section for more details.

Image Check Failure

If the Image Check fails that means that the Image version on your roboRIO did not match the image version that the plugins were built against. To check for updates to the Eclipse plugins, make sure you installed the plugins from the web repository and that you have an active internet connection, then click Help->Check For Updates. To check for updates to the NI FRC Update Suite, which may contain a new roboRIO image, visit the URL from the [Installing the FRC Update Suite \(All Languages\)](#) document. If an updated version is available, after installing it as described in that document, re-image your roboRIO using the instructions in the [Imaging your roboRIO](#) document.

It's also possible that the plugins are failing to update installed files. Close Eclipse, locate the WPILib Java folder (C:\Users\USERNAME\wpilib\java on Windows) and delete the folder. Restart Eclipse and try deploying again.