2018 Beta Testing

Last Updated: 10-13-2017

Table of Contents

2018 Beta Testing - Getting Started

Welcome

Thank you for participating in the 2018 Beta testing program.

Before getting started with the testing tasks please review the following articles.

- Accessing the 2018 Beta Project
- Reporting Progress
- Trackers - Reporting Bugs
- Monitoring Via Email Notifications
- 2018 Beta Task Overview

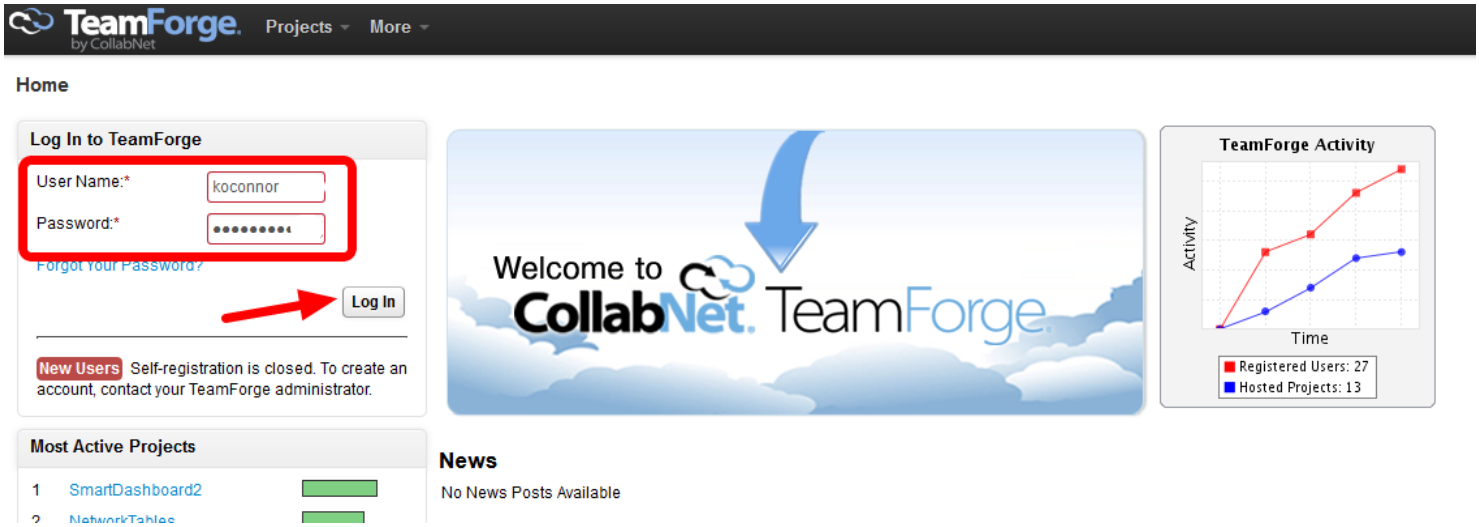
Please use the TeamForge project as your primary means of communication. Members of the control system development team at CTRE, FIRST, NI, and WPI are all monitoring the project and will strive to provide timely feedback and updates.

Accessing the 2018 Beta Project

Note: Only accepted Beta Test teams will have access to the Beta project.

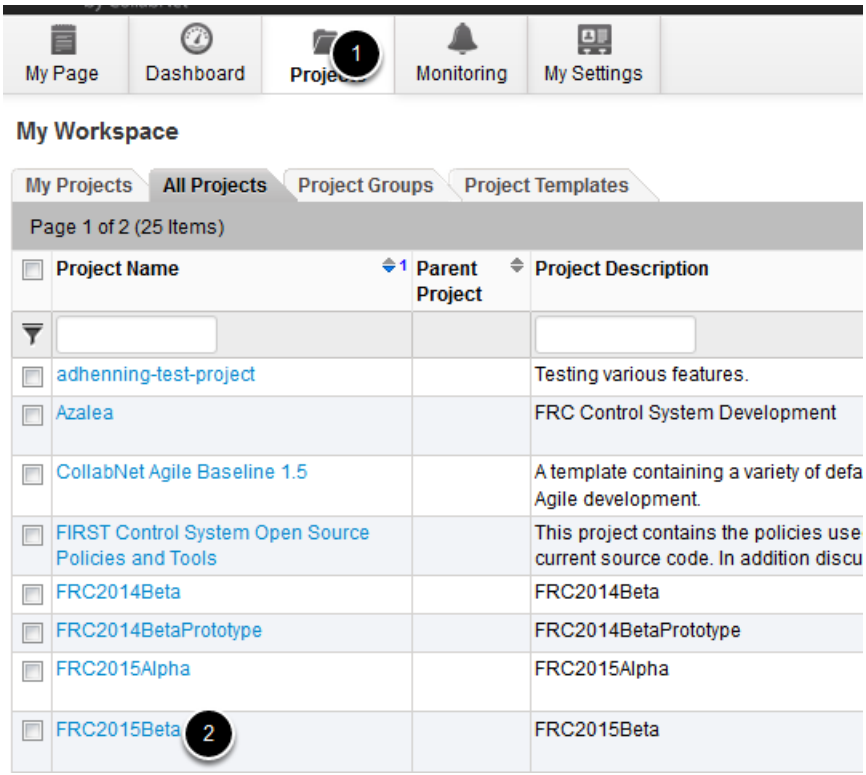
The members identified by your team to communicate with FIRST technical staff will have access to the 2018 Beta Test project on the usfirst.collab.net site.

Signing in



Open your web browser and browse to [usfirst.collab.net.](usfirst.collab.net) Near the top left corner of the page enter the username and password provided in the Beta test email you received, then click "Log In".

Opening the 2018 Beta Project



Your user account should already be added to the 2018 Beta Project. To view the project:

1. Click the Projects tab on the top navigation ribbon.
2. Then click on FRC2018Beta in the projects list (image of 2015 as an example).

The Project Homepage

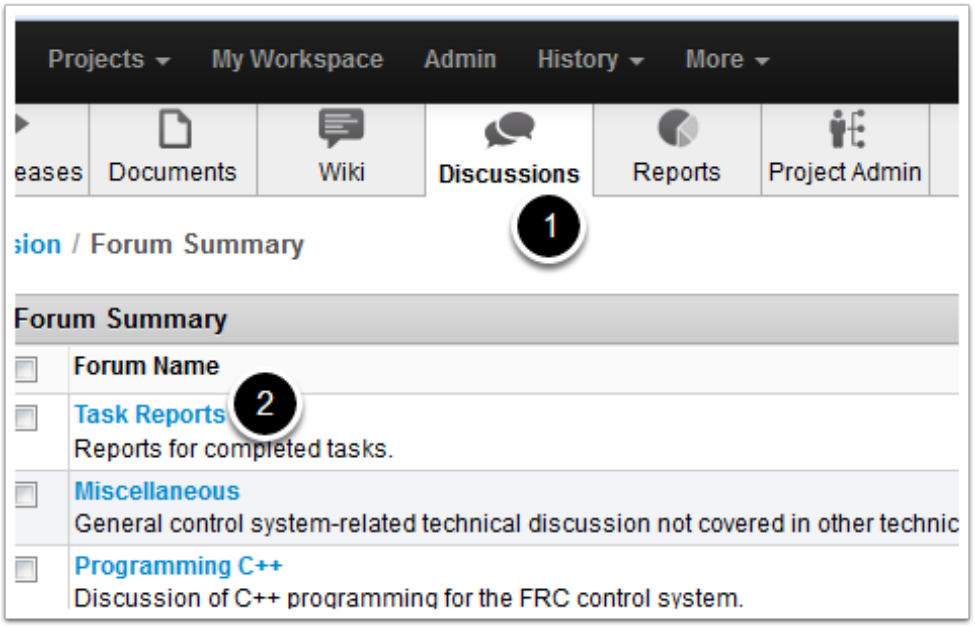
The project homepage contains some statistics and Project News. Throughout the Beta, the Project News section may be updated with the latest information from the FIRST technical staff. The top ribbon contains tabs allowing you to navigate to the different sections of the project:

1. Trackers - The Trackers tab is where the bug tracker for the project is located.
2. File Releases - The File Releases tab will host any files you will need to download as part of the Beta test process such as NI Update files.
3. Documents - the Documents tab will contain any documents to be distributed to Beta teams. Much of the documentation for the Beta test will be located on Screensteps but there may still be documents posted here as well.
4. Discussions - The Discussions tab contains a forum which will allow teams to post questions or discussions about the Beta test, software or documentation. Task reports will also be posted here.

Reporting Progress

For 2018 we are going back to posting Task reports in the Discussion section of the Teamforge project. There will be a topic for each task and reports will be posted as replies to that topic. We hope that this will make it easier for teams to see what other teams have been doing and any CS Team replies to other team's reports. Another goal is to hopefully reduce confusion about where or how to post reports and manage the other fields that were present in the old tracker system.

Accessing the Task Reports Section

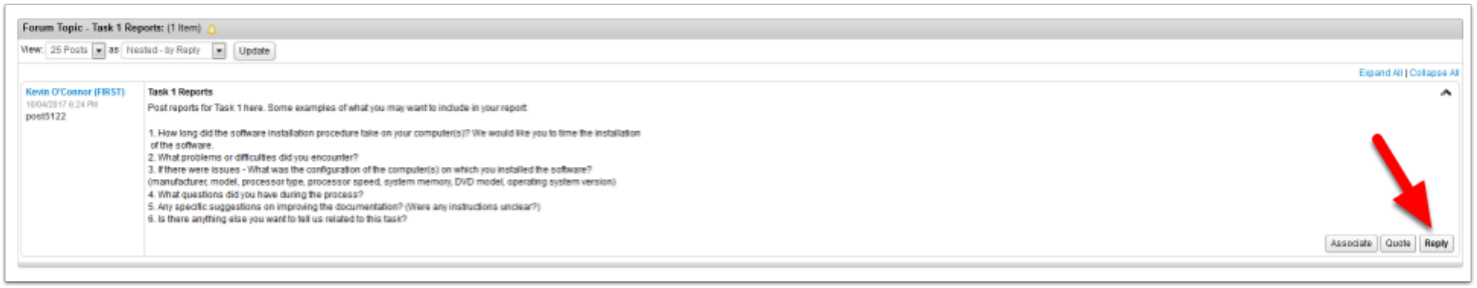


To access the Task Reports section on Teamforge, click on the **Discussions** tab, then click on the **Task Reports** section.

The Task Reports Section

In the Task Reports section, you will see a Topic for each task. To see the reports and discussion for this task, click on the Topic Name. (Note: this example image only has a single Task).

Reporting Progress



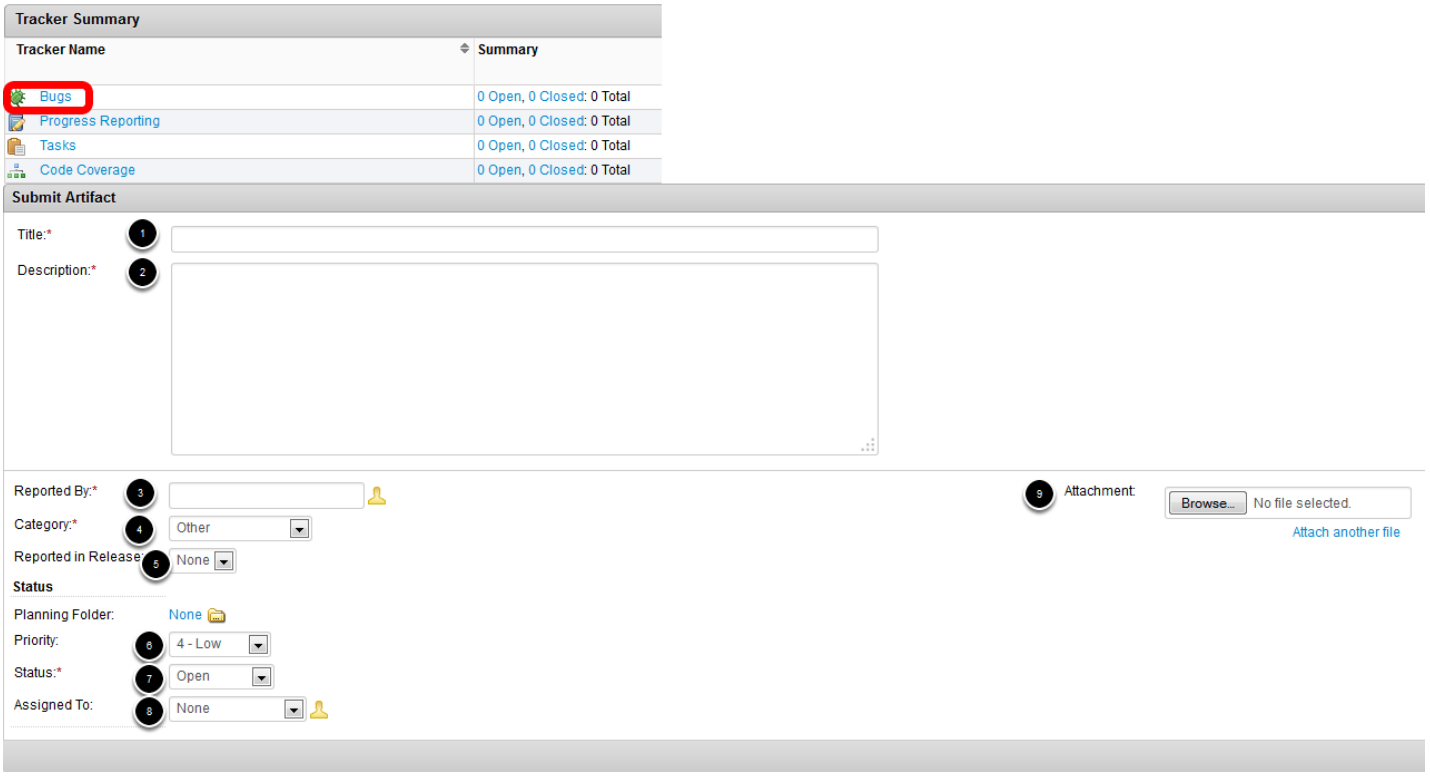
Each Topic will have a summary post at the top with suggested feedback. When you have completed the task, click **Reply** on this top summary post to post your report. Then compose or paste your report into the editor and click **Save**.

We strongly prefer reports be done as text posts as opposed to attachments.

Trackers - Reporting Bugs

For the 2018 Beta, the Trackers tab will be used only for filing bugs (different than the past few seasons). For reporting progress, see Reporting Progress.

Bug Tracker



The bug tracker is used for reporting bugs you discover in WPILib or other FRC software. If you are not sure if particular behavior is a bug or not, err on the side of reporting the bug (or at least posting in the Discussion section).

On the Trackers tab, click on Bugs to reach the Bug Tracker. To Submit a new bug, click **Submit New Artifact**. The fields of the artifact are:

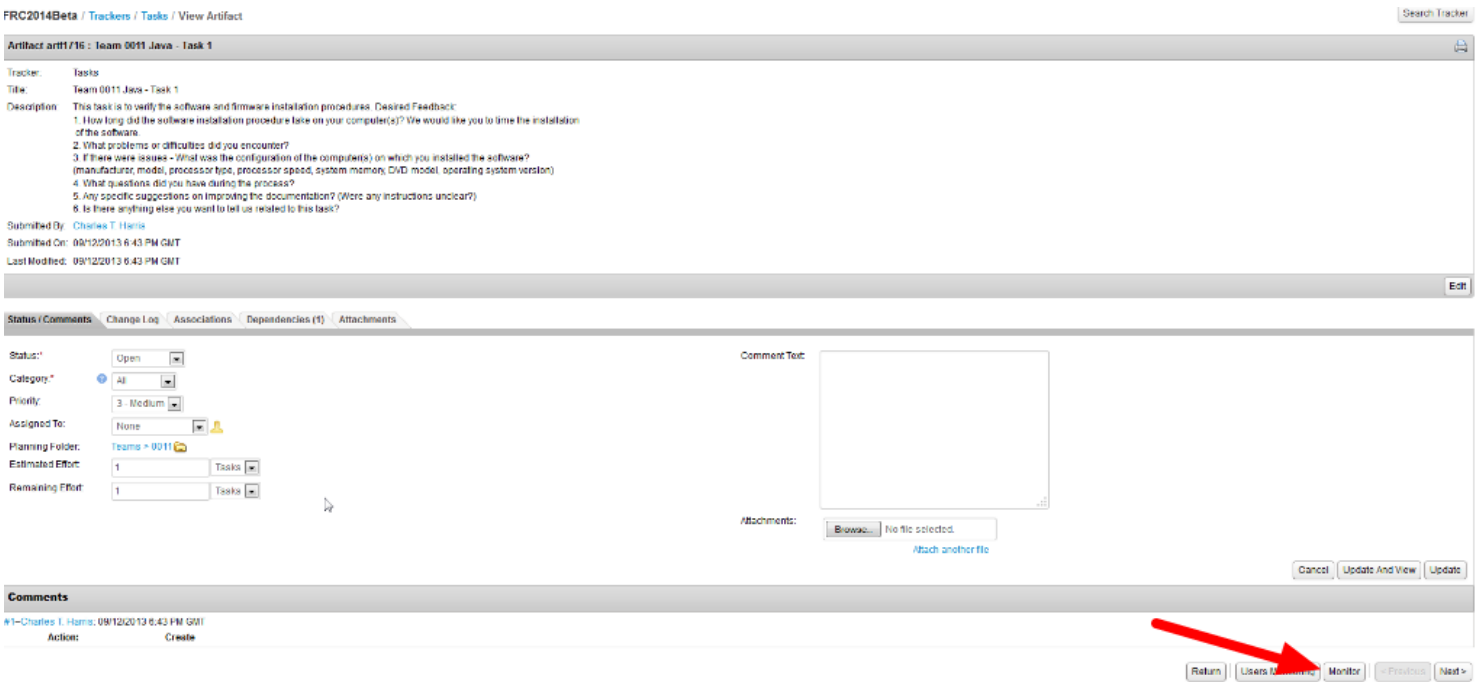
1. Title - Give your bug a meaningful title which describes the issue

2. Description - Describe the bug including observed behavior, expected behavior and steps to reproduce
3. Reported By - Enter your TeamForge user name or click the silhouette to open the selector where you can select your username.
4. Category - Select the appropriate affected software from the dropdown list (LabVIEW WPILib, Java WPILib, Driver Station, etc.)
5. Reported in Release - Select which version of the file you were using, if available.
6. Priority - If desired you may estimate a priority based on bug severity. Leaving at the default priority does not mean the bug will not be addressed. All new bugs are analyzed by the Control System team and assigned an appropriate priority.
7. Status - When reporting a new bug this should be left as Open.
8. Assigned To - When reporting a bug this should be left blank. The Control System team uses this field to indicate who is working on a given artifact.
9. Attachment - Attach any additional file that may be appropriate, examples of appropriate files would be a patch for the bug, code to reproduce the bug (single file or zip) or an image of the bug (if appropriate/necessary).

Monitoring Via Email Notifications

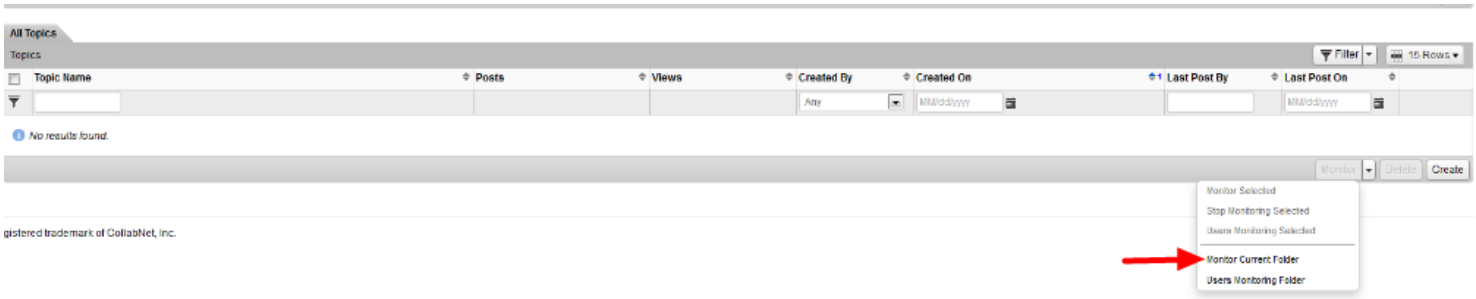
You may wish to monitor specific items or even whole sections of the Beta project via email. TeamForge has a large degree of flexibility in setting up e-mail monitoring.

Monitoring Individual Items



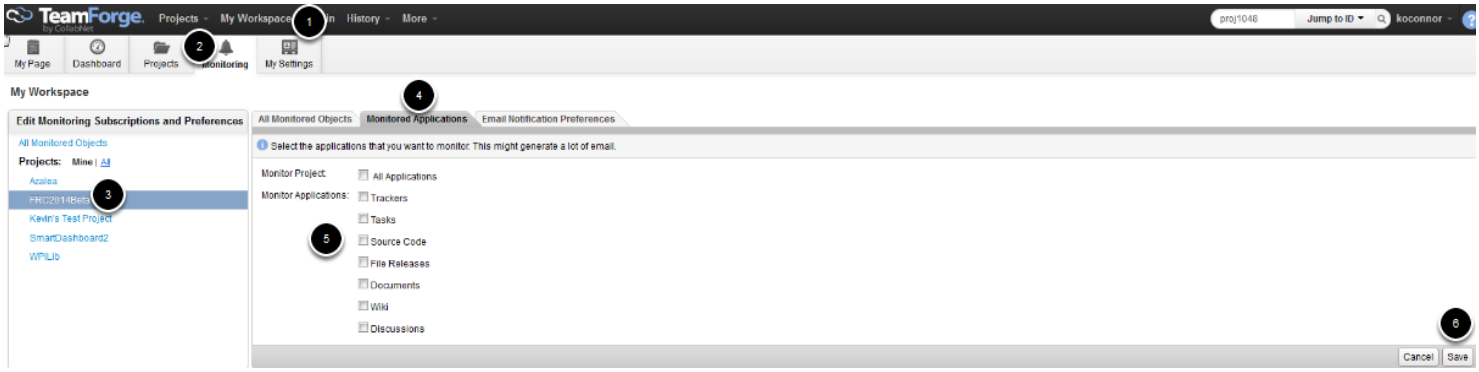
You can choose to monitor a number of different individual items (forum threads, tracker artifacts, file releases, etc.) via e-mail. Items that can be subscribed to will have a Monitor button located at the bottom of the screen. Clicking this button will subscribe you to the item and notify you of changes via e-mail. To unsubscribe, click the Stop Monitoring button (which will replace the Monitor button when you are subscribed).

Monitoring folders



"Folders" (discussion sections, trackers, etc.) can also be monitored via e-mail. To monitor a folder click the dropdown arrow next to the Monitor button and select **Monitor Current Folder**.

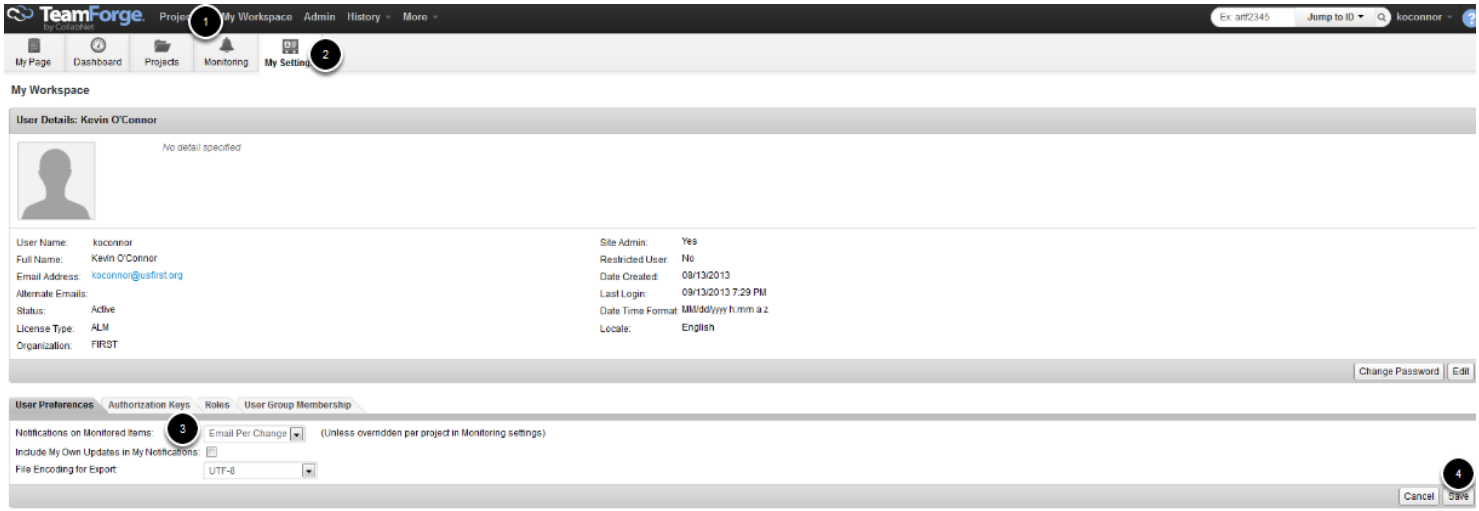
Monitoring Project Sections



It is also possible to set up for entire sections of the site. To set up this monitoring:

1. Click My Workspace
2. Click on the Monitoring Tab
3. Select the project from the list on the left
4. Click on the Monitored Applications tab
5. Check the box for any section you wish to subscribe to
6. Click Save

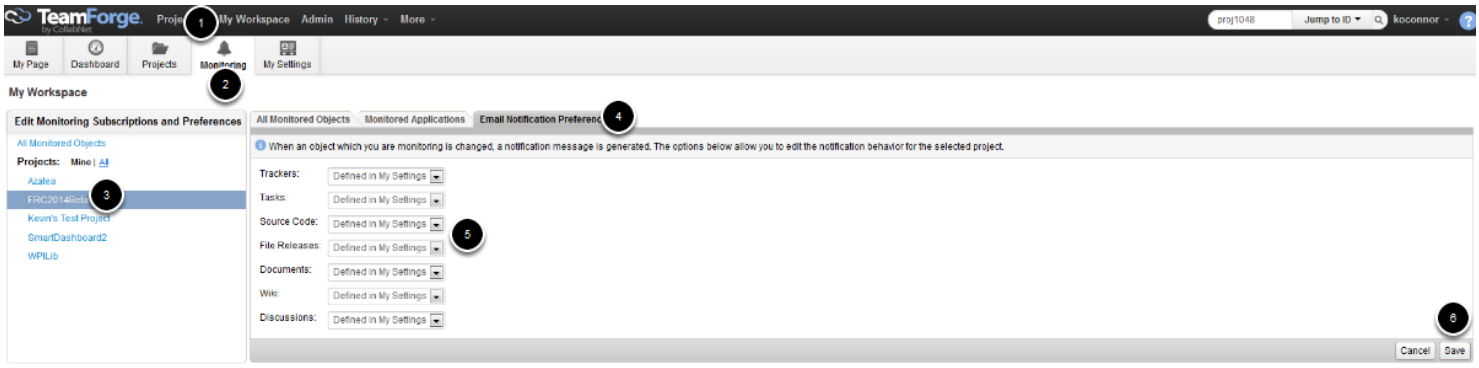
Monitoring Settings



You can set your overall notification settings to e-mail you per change or to send you a daily digest email with all changes. To do this:

1. Click My Workspace
2. Click My Settings
3. Select the desired setting
4. Click Save

Per Section Notification Settings



You can also customize notifications on a per section basis. To do this:

1. Click My Workspace
2. Click Monitoring
3. Click on the FRC2018Beta project on the left side
4. Select the Email Notification Preferences tab

5. Change the settings for any section of the project you wish to have override your base settings.
6. Click Save

New for 2018

C++/Java WPILib

- Added new base class, TimedRobot. TimedRobot is similar to IterativeRobot but loops on a timer instead of syncing to the arrival of DS packets. This provides for more consistent timing of robot actions and controls, at the expense of potentially processing some inputs for multiple loops, or skipping some inputs (depending on loop time and jitter).
- The RobotDrive class has been split into seperate 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
- TalonSRX renamed to PWMTalonSRX
    - CTRE's new software also renames CANTalonSRX to TalonSRX
- A periodic method has been added inside subsystems

Java

- New JRE (zulu) from Azul Systems. This allows us to package the JRE with the Eclipse plugins and deploy it automatically, removing the need for the manual Java Installer tool.
- Sendable Chooser now keeps entries ordered

LabVIEW

- Increase robustness of USB Camera connections and streaming
- New simulation tools. The FRC roboRIO Project wizard has new "Learn LabVIEW" simulation choices which are a maze and a ball shooter. These include fully functional simulation robots. Documentation for these is not yet complete.
- Updated to LabVIEW 2017

FRC Driver Station powered by NI LabVIEW

- Improved connectivity diagnostics (firewall info, etc.)
- Removed protocol selector (to connect to cRIO systems, use the FRC 2014 DS)

Network Tables

- Improved Connection Synchronization Behavior

    - This should reduce or eliminate the occurrences of "phantom keys" on the Dashboard when changes are made to code to rename or remove keys without restarting the Dashboard
    - The original synchronization behavior was troublesome for two reasons:
        1. It had unpredictable behavior for updated values
        2. It brought back to life deleted values
    - Instead of relying on the server to inform the client regarding reconnections, the client keeps track of what values have been modified by user code on the client. When the client connects to the server, the following occurs:
        - For entries that have been modified by user code on the client:
            - If the entry is not persistent, the server value is overwritten with the client value
            - If the entry does not exist on the server, the client sends an assignment to the server to recreate it on the server
        - For entries that have not been modified by user code on the client:
            - The client value is overwritten with the server value
            - If the entry does not exist on the server, the client deletes the entry

- JSON/CBOR/MessagePack library is now included in C++ for easier mapping of structured data into raw NetworkTables entries. This is a lightweight version of https://github.com/nlohmann/json, imported into the wpi namespace.
- API updated to provide handle-based access for improved performance. This also allows multiple simultaneous independent instances to be created (e.g. for unit testing purposes).
    - Added new classes and interfaces (NetworkTableInstance, NetworkTableEntry, NetworkTableValue, and others). In Java, these, along with a new version of the NetworkTable class, are now located in a different package (edu.wpi.first.networktables, rather than edu.wpi.first.wpilibj.networktables)
    - The NetworkTable class is still available, but many functions are now deprecated in favor of NetworkTableInstance and NetworkTableEntry functions.
    - In Java, many JNI functions changed; it's recommended users of the old JNI functions change to using NetworkTableInstance instead of JNI directly.
    - There is a "default" instance that is configured by the WPILib robot code during startup; this can be retrieved using NetworkTableInstance.getDefault(). You can then get NetworkTable's by calling NetworkTableInstance.getTable(String). This replaces the old NetworkTable.getTable() static function.
    - NetworkTableEntry's are returned by NetworkTable.getEntry(String). NetworkTableEntry provides functions for getting and setting values similar to the old NetworkTable string-based getters and setters, but is faster because it uses handles rather than string lookups.

- In Java, callbacks are now executed by a Java thread rather than a native thread.

SmartDashboard

- Brand new Shuffleboard dashboard added.
- SFX dashboard removed (will be provided as a separate download, but is no longer being actively maintained)

OutlineViewer

- Rewritten in JavaFX
- Enter team number along with server name
- Support for non-standard port numbers
- Checkboxes for boolean values

Eclipse Plugins

- Much faster deploy
    - Checks multiple network addresses in parallel
    - Uses WebDAV instead of SSH for transferring files

- Improved roboRio image version detection
- Automatically deploys Java JRE if not already installed
    - Java installer is no longer necessary and has been removed from the plugins

- Coming Soon - New TCP based console plugin (replacing existing NetConsole)

2018 Beta Testing - Tasks

2018 Beta Task Overview

This document provides an overview of the Tasks assigned for the 2017 Control System Beta Test. Further details on each Task can be found in the subsequent pages of this section.

Task 0 - Background and Preparation

The purpose of Task 0 is to complete some housekeeping tasks and familiarize yourself with the tools and documentation for the Beta Test before beginning. This task consists of 1 Objective:

1. Beta Test Tools and Documentation Familiarization

Task 1 - Install Software (Targeted Completion 10/27/17)

The purpose of Task 1 is to complete installation of the 2018 Control System software. This task consists of 1 Objective

1. Install the necessary software for your assigned Beta Test language (LabVIEW or Eclipse + plugins, NI FRC Update Suite, CTRE installer if using CANTalon SRX)

Task 2 - Basic Benchtop Test (Targeted Completion 11/3/17)

The purpose of Task 2 is to get the base 2018 Control System software set up and running. This helps minimize complexity to ease debugging of any issues that do occur. In this task you will set up the development environment for your programming language, configure the roboRIO, load and debug a simple program. This Task consists of 2 Objectives:

1. Configure the roboRIO
2. Create, Build and Load Default Program

Task 3 - Port Previous Robot Code (Targeted Completion 11/10/17)

The purpose of Task 3 is to port a previous year's robot code (2017 preferred) with the 2018 Control System software and bring it to full functionality. This Task consists of 2 Objectives and 1 Optional Objective:

1. Port Existing Robot Code to 2018 System
2. Wireless Operation of FRC Robot
3. (Optional) Offseason Competition with Existing FRC Robot

Task 4 - New Code and Advanced Feature Testing (Targeted Completion 11/24/17)

Note: Objectives in Task 4 may be completed in any order.

The purpose of Task 4 is to write new code in order to test features and functionality of the new Control System software. While we have written up a number of ideas, please feel free to come up with your own and report on what you tried/are trying and how it went.

1. Network Tables Stress Testing (increase update rate in your code, test reconnect scenarios)
2. "Shuffleboard" (new JavaFX Dashboard) testing
3. Camera streaming robustness, especially in LabVIEW (multiple cameras, disconnect/reconnect, etc.)

International Team Note

A number of our international teams may have access to computers with foreign language Operating Systems installed (Spanish, French, Hebrew, etc.) This note describes some additional feedback that would be helpful to us if you have access to such a machine to use as part of your Beta test effort.

Tasks

If you have access to a computer with a foreign language version of Windows, it would be helpful to our Beta Test effort to utilize that machine as part of your Beta testing. Specifically, try testing each of the unique software aspects (installation, imaging the roboRIO, running the DS, creating projects, building and downloading code, debugging code, programming a radio, etc.).

Feedback

In your task reports for the appropriate tasks please indicate that you completed the task with a machine with a foreign language OS.

Task 0: Beta Test Tools and Documentation Familiarization

Task: Learn about the tools and documentation sources for the 2018 FRC Beta

Beta Test Tools

The 2018 FRC Control System Beta Test will primarily be administered through a TeamForge project. If you have not been part of an Alpha or Beta test effort in the last few years, please take some time to read through the following articles to learn more about the tool being used:

Accessing the 2018 Beta Project

Trackers - Reporting Bugs

Monitoring Via Email Notifications

Beta Test Documentation

There will be 2 primary sources of Beta Test specific documentation. The first source is this ScreenSteps manual which will primarily contain things like task instructions. The Beta Test ScreenSteps page will be publicly accessible without a username or password.

The other source of documentation will be the Documents section of the TeamForge project. For more information on how to access that, see Accessing the 2018 Beta Project.

Please read through the documentation in this ScreenSteps manual to understand the tasks associated with the Beta Test.

Desired Feedback

No report is necessary for this task.

Task 1 - Objective 1: 2018 Software Installation

Task: Choose, acquire, and install the software required for developing FRC software in the language you have been assigned (C++, Java or LabVIEW).

Overview

The 2018 FRC Control System can be programmed in LabVIEW, Java or C/C++. Teams should use the language they have been assigned for the Beta test.

Note: For this task, there is supplemental documentation provided in TeamForge for all languages. See the Documents and File Releases sections on TeamForge.

Desired Feedback

This Objective is part of Task 1. Please keep the following questions in mind as you complete the Objective and include this information, as appropriate, in your Task 1 report.

1. How long did the software installation procedure take on your computer(s)? We would like you to time the installation of the software.

2. What problems or difficulties did you encounter?

3. If there were issues - What was the configuration of the computer(s) on which you installed the software? (manufacturer, model, processor type, processor speed, system memory, DVD model, operating system version)

4. What questions did you have during the process?

5. Any specific suggestions on improving the documentation? (Were any instructions unclear?)

6. Is there anything else you want to tell us related to this task?

Task 2 - Objective 1: Configure the roboRio

Task: Image and configure your roboRIO controller.

Overview

Before using the roboRIO controller with 2018 software, it must be imaged with the 2018 image and configured with your FRC team number. As part of the purpose of this Beta test is to test the documentation and its ease of use, no direct links to setup instructions will be provided. Instructions for configuring the roboRIO are available on the Getting Started with the 2018 Control System page.

Note, the image version in the documentation may not be updated yet. You should use the latest 2018 image available after installing the latest 2018 Beta Update.

Desired Feedback

This Objective is part of Task 2. Please keep the following questions in mind as you complete the Objective and include this information, as appropriate, in your Task 2 report.

1. What problems or difficulties did you encounter?

2. What questions did you have during the process?

3. Any specific suggestions on improving the documentation? (Were any instructions unclear?)

4. Is there anything else you want to tell us related to this task?

Task 2 - Objective 2: Create, Build, and Load Program for Benchtop Testing

Task: To create, build, load and run a basic robot program (template of your choosing) on the roboRIO benchtop setup.

Overview

The purpose of this task is to verify the ability to create and download a basic robot program from the chosen template in the assigned programming language. This will verify that the development environment are set up correctly and working properly before moving on to a more complex system. For this task you should create, build and download a basic program to the roboRIO to run at startup, then verify that you are able to communicate with and enable the program using the Driver Station.

As part of the purpose of this Beta test is to test the documentation and its ease of use, no direct links to setup instructions will be provided. Instructions for all languages are available on the Getting Started with the 2018 Control System page.

Desired Feedback

This Objective is part of Task 2. Please keep the following questions in mind as you complete the Objective and include this information, as appropriate, in your Task 2 report.

1. What problems or difficulties did you encounter?

2. What questions did you have during the process?

3. Any specific suggestions on improving the documentation? (Were any instructions unclear?)

4. Is there anything else you want to tell us related to this task?

Task 3 - Objective 1: Port Existing Robot Code to 2018 System

Task: Port 2017 Robot code to 2018 system.

Overview

The purpose of this task is to port your 2017 robot code to the 2018 development environment and bring it to a fully functional state. Make sure to post in the discussions or file a bug in the tracker for any items that do not seem to be working properly.

Desired Feedback

This Objective is part of Task 3. Please keep the following questions in mind as you complete the Objective and include this information, as appropriate, in your Task 3 report.

1. Note any required changes to your robot program not detailed in the documentation for porting robot code.
2. Using the Charts tab of the Driver Station, report the resource usage of your robot code (roboRIO CPU usage, free RAM)
3. What problems or difficulties did you encounter?
4. What questions did you have during the process?
5. Any specific suggestions on improving the documentation? (Were any instructions unclear?)
6. Is there anything else you want to tell us related to this task?

(Optional) Task 3 - Objective 3: Offseason Competition with Existing FRC Robot

This Objective is **optional** and does not need to completed in order to continue with the other Tasks and Objectives.

Task: Participate in an offseason event with a 2017 robot which has been retrofit with the 2018 FRC Control System software.

Overview

The purpose of this task is to test the 2018 Control System in a competition environment.

Desired Feedback

This Objective is part of Task 3. Please keep the following questions in mind as you complete the Objective and include this information, as appropriate, in your Task 3 report.

1. What problems or difficulties did you encounter?
2. Is there anything else you want to tell us related to this task?

Task 4 - Objective 1: Testing New Features

Task: Test any new features or heavily modified code

Overview

The purpose of this task is to test any newly developed or heavily modified features for the 2018 season. This will help the Control System team find any bugs or issues with the new code. Below is a list of some of the new or modified features:

LabVIEW

- Camera Streaming robustness (camera connect/disconnect, network connect/disconnect)
- Better multiple camera support (especially for MS Lifecam)
- Network tables (dashboard) connect/reconnect behavior

C++\Java

- Network tables connect/reconnect behavior (value overwrites, "ghost" keys, etc.)
- "Shuffleboard" - New JavaFX dashboard

Desired Feedback

This Objective is part of Task 4. Please keep the following questions in mind as you complete the Objective and include this information, as appropriate, in your Task 4 report.

1. Describe what features you tested and how you tested them.
2. Describe any bugs you encountered.
3. What problems or difficulties did you encounter?
4. What questions did you have during the process?
5. Any specific suggestions on improving the documentation? (Were any instructions unclear?)
6. Is there anything else you want to tell us related to this task?