

SmartDashboard Quick Start

Paul Malmsten

pmalmsten@wpi.edu

Introduction

For the 2011 season, WPI has included a new software library and remote client to simplify sending data to a laptop dashboard display. When a log command is run on the robot, the remote dashboard will automatically display its name and value on the screen.

Robot Side

The following examples show Java syntax. However, the smart dashboard robot side library is also provided for C++ with a very similar naming scheme.

It is only necessary to run this once. Once initialized, one may send data to the dashboard like so:

```
SmartDashboard.log(left_joystick_value, "Left Joystick");
```

The first argument is the value to log; the second is the name of the field as it should appear on the dashboard. This method supports all primitive data types as well as strings. For more information about the methods available to you, see the WPILib documentation or source code.

Dashboard (laptop) Side

In order to use the SmartDashboard client, your Driver Station must be configured to use a remote dashboard. To enable this feature, see the 'Setup' tab of the FRC Driver Station, click the appropriate toggle switch, and ensure that the specified IP address is correct.

The SmartDashboard client currently consists of a single jar file. On most computers, double-clicking this file will run the dashboard.

If for any reason this does not work, one may also run the SmartDashboard by invoking Java from an appropriate command terminal:

```
> java -jar path\to\SmartDashboard.jar
```

In order for this command to work, Java must be on the system's execution PATH. For more information about how to do this, see the Java online tutorial here:

<http://download.oracle.com/javase/tutorial/essential/environment/paths.html>

Once the dashboard is running, an information bar and colored status indicator at the bottom of the window will inform you if anything is wrong. Robot code which makes log() calls must be running on your robot before the client will indicate that it is receiving data.

Dashboard Customization

Once the dashboard client is initialized and the robot code is sending data, default widgets will be automatically placed on the screen. These may be relocated to any unoccupied place on the screen by dragging them.

Many data types can be displayed by more than one type of widget. To change the current type of widget being displayed for a particular data field, right-click on a widget and select a replacement from any of the choices which are listed next to the “Change to...” menu item.

Preferences

A variety of widgets offer various preferences which may be changed at any time. To view or modify these preferences, right click on the widget to modify and select the “Preferences” menu option. In the resulting popup window, double click on the value textbox to enter a new preference value. Preference changes will be applied immediately.

Saving/Loading a Configuration Profile

Once customized to your liking, a profile may be saved by clicking “Save” from the file menu. At this time, only one profile is supported, and it is saved to a user’s home directory (e.g. C:\Users\John on Windows, /home/John on other platforms). This profile is called SmartDashboard.serialized and may be copied to any number of client computers as long as it is in a user’s home directory.

To load a profile, select “Load” from the File menu.

Note: widget preferences are currently stored separately from profile information and may not follow a copied profile. Widget types, locations, and names will always follow a profile.

Compatibility Notes

This class is not compatible with the existing Dashboard classes in the Java and C++ library code or the existing FRC Dashboard provided by National Instruments. Calls to these libraries in robot code must not be interspersed!

If you choose to use the SmartDashboard library at *any* location within your code, you must *never* call the old Dashboard code anywhere else or you will risk corrupting the data stream sent to the client.

Enjoy!

~Paul Malmsten and the Smart Dashboard Team