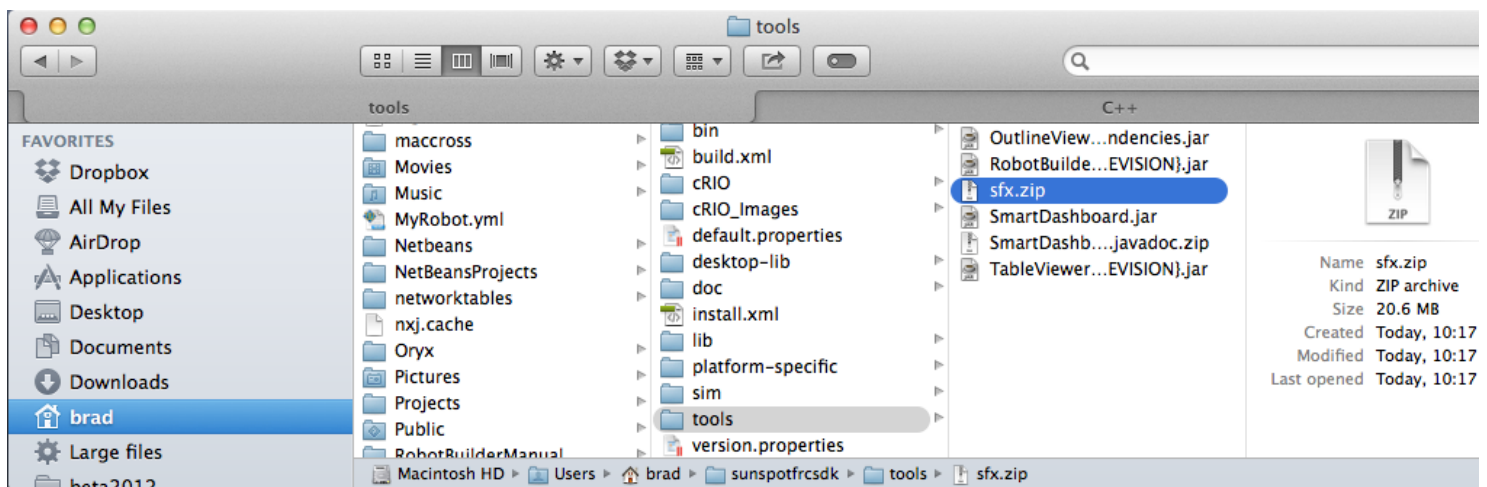


The new SmartDashboard (SFX)

The new SmartDashboard (SFX)

We have a new SmartDashboard that uses the more modern JavaFX for it's user interface. This has the promise of much richer user interfaces since it allows the widgets to have style sheets applied to them and the library is much newer. The new dashboard requires a current version of Oracle Java 7 (minimum release 7u6) to be installed on your system to get JavaFX.

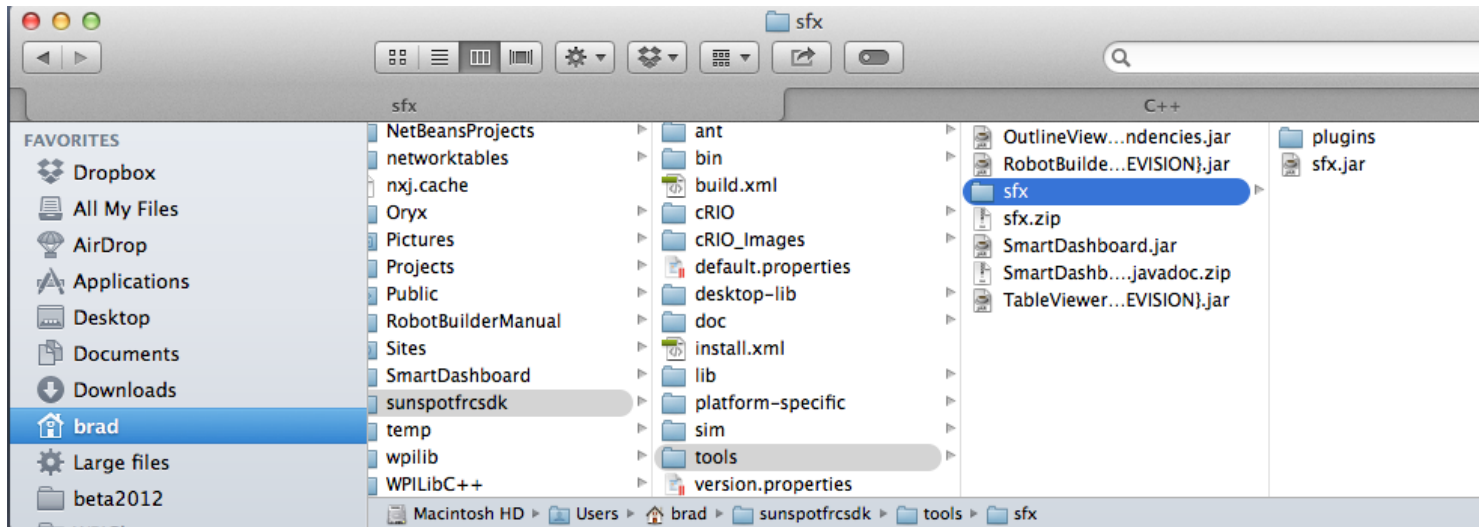
Installing SFX



The new dashboard is shipped as a zip file called sfx.zip. You must first unzip the file in <your-home-directory>/sunspotfrcsdk/tools as shown in here. If the sfx folder already exists from a previous install, delete it before unzipping the newer version.

The new SmartDashboard (SFX)

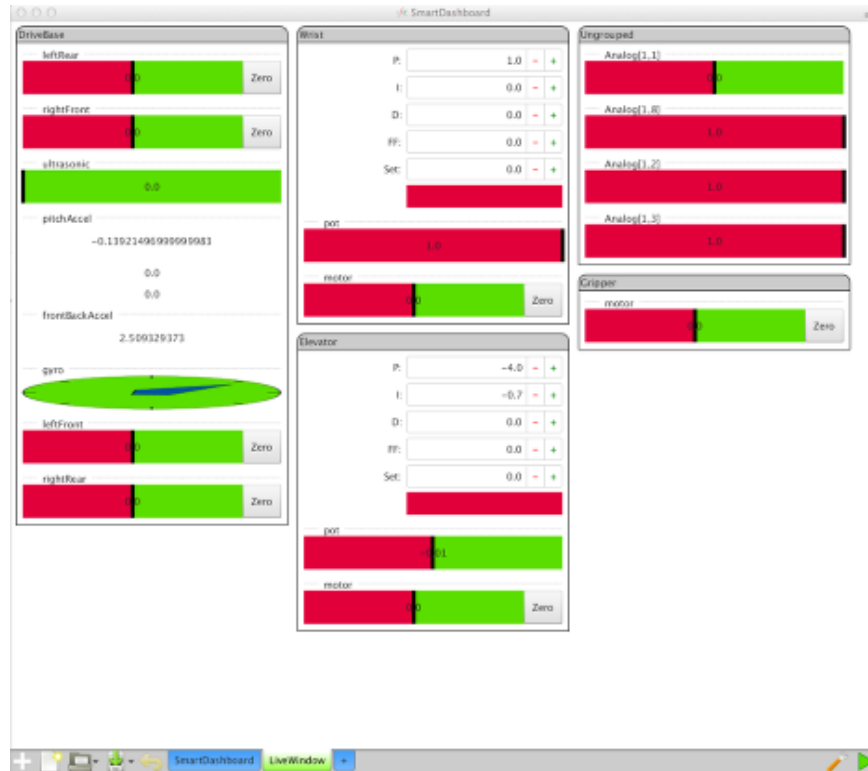
Running SFX



After unzipping the dashboard, you run it by double-clicking on sfx.jar. If the .jar file extension isn't defined on your system (depending on the platform and the java installation) it can also be run with command line, "java -jar sfx.jar".

The new SmartDashboard (SFX)

The SFX user interface



SFX will start up and automatically display any values written to the dashboard in Autonomous or Teleoperated modes and will display subsystems or individual controls in Test mode. In the above example the robot is in Test mode and there are a number of subsystems shown (DriveBase, Wrist, Elevator, and Gripper) as well as a number of Ungrouped analog inputs that are allocated but not in any subsystem.

The new SmartDashboard (SFX)

SFX controls

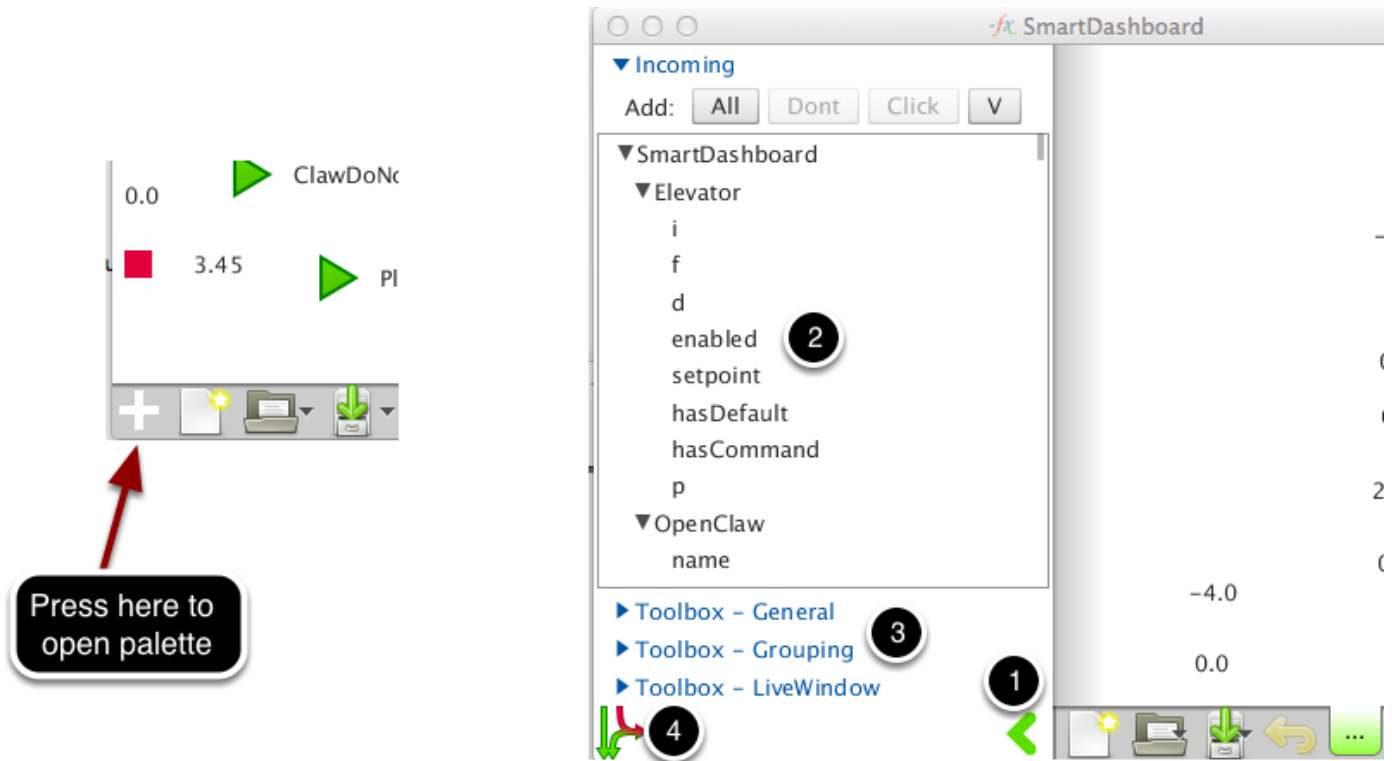


The user interface consists of a number of controls:

1. These controls are for opening the widget palette, creating a new layout, and saving the layout.
2. Tabbed windows are predefined for SmartDashboard and LiveWindow displays. You can also create your own tabs by pressing the plus (+) button. In the future you'll be able to route widgets to each of the windows through pattern matching of the name string for the corresponding values.
3. Settings and the Play button (takes the dashboard in and out of edit mode).

The new SmartDashboard (SFX)

The palette

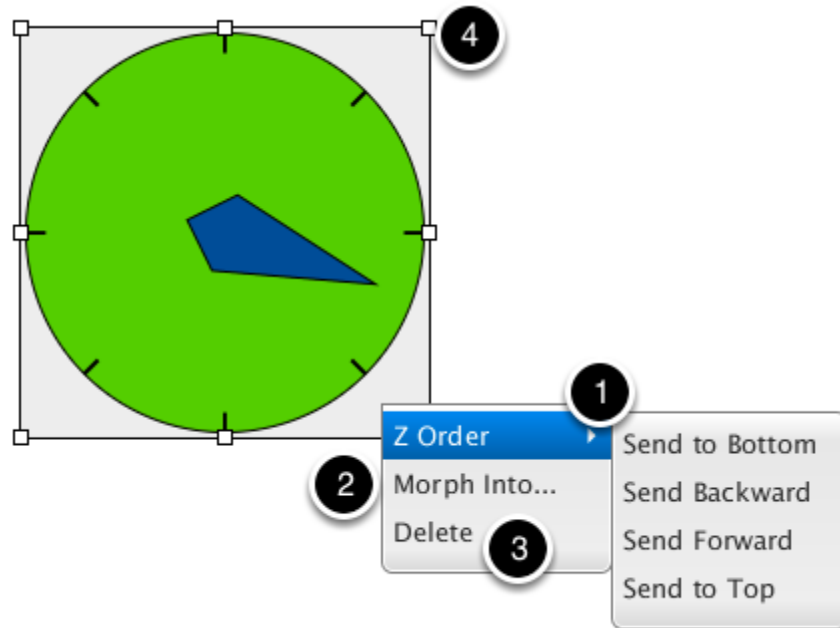


The palette area is accessed by pressing the plus (+) button and closed with the close button (1). There is a hierarchical list of values (2) that come from the data source (the running program SmartDashboard class methods) that can be dragged into onto the screen. Doing this will create a widget for that value. In addition, you can place widgets on the dashboard and associate them with values later by selecting from the toolbox sections (3).

The data source button (4) opens a dialog that allows you set the IP address for the robot as well as map data sources to the dashboard. This is currently unsupported and will be more completely implemented in a future release.

The new SmartDashboard (SFX)

Editing currently displayed values in the dashboard



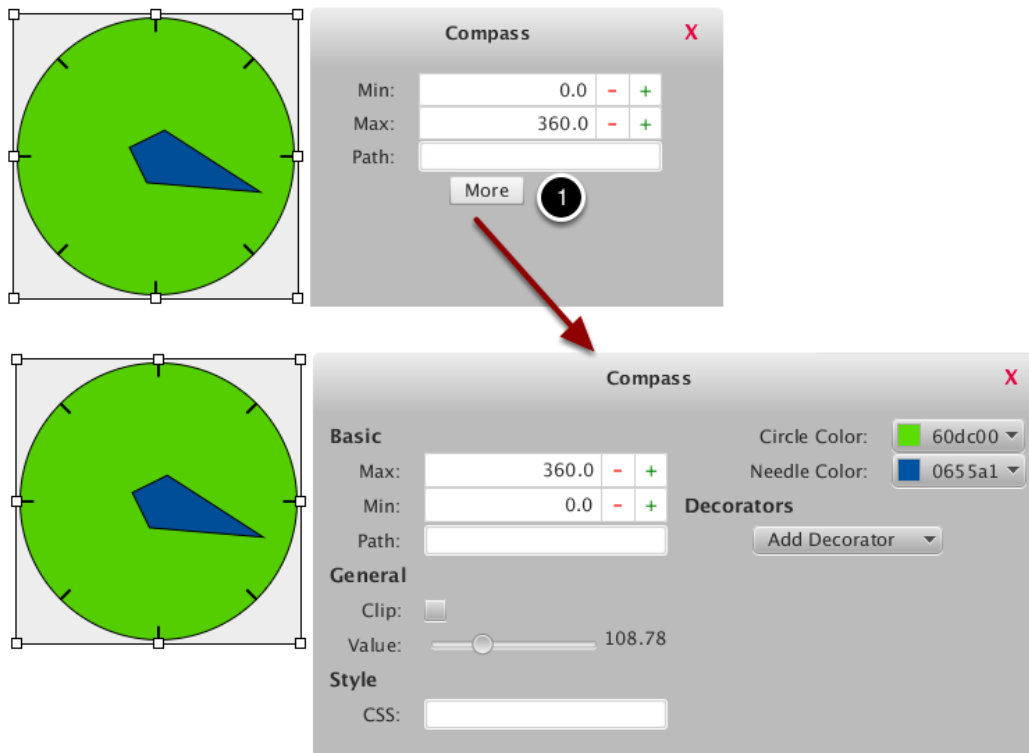
Right-clicking on a value brings up a context menu with a number of options for the value being displayed. You can:

1. Change the stacking order of the widget with respect to other widgets above or below this one.
2. Change this widget into another type of widget (morph) that can display the same value with a different graphical representation.
3. Delete the widget from the layout. Widgets deleted can be added back in later by selecting them from the palette.

The widget may also be resized by dragging the resize handles (4) to change the width or height of the value.

The new SmartDashboard (SFX)

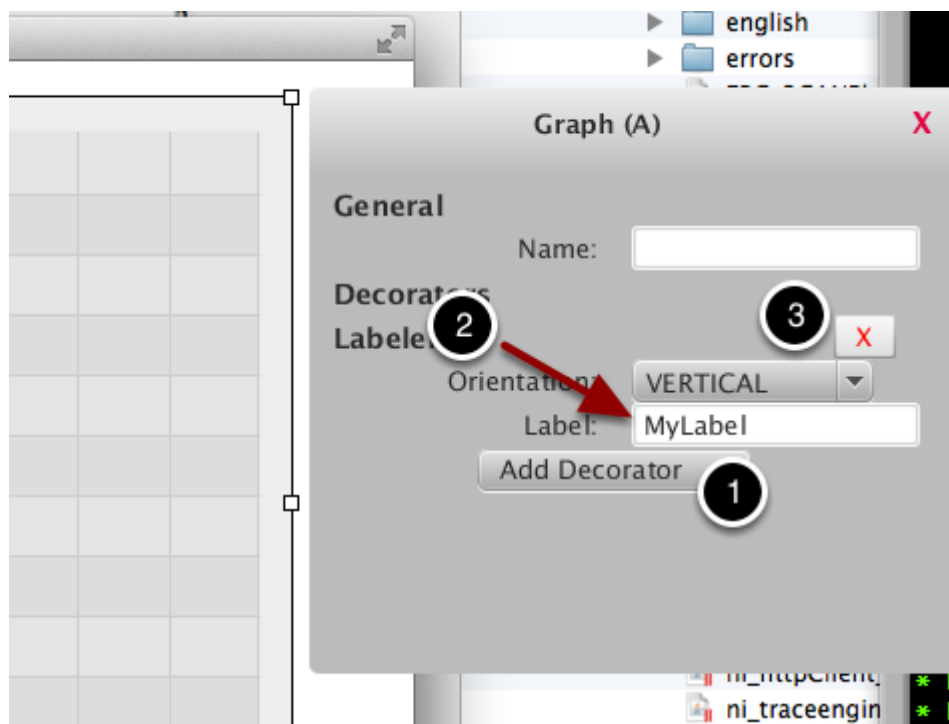
Edit Properties



Clicking on the control selects it and brings up a properties list. Most controls only show a few basic properties by default, but can show all properties by clicking the "More" button (1) This shows all properties grouped by type, as well as the decorators section, which is explained next. If you don't know what a property is, hover over its name for a short description of what it does. Properties are updated instantly, no need to save.

The new SmartDashboard (SFX)

Adding labels to dashboard widgets

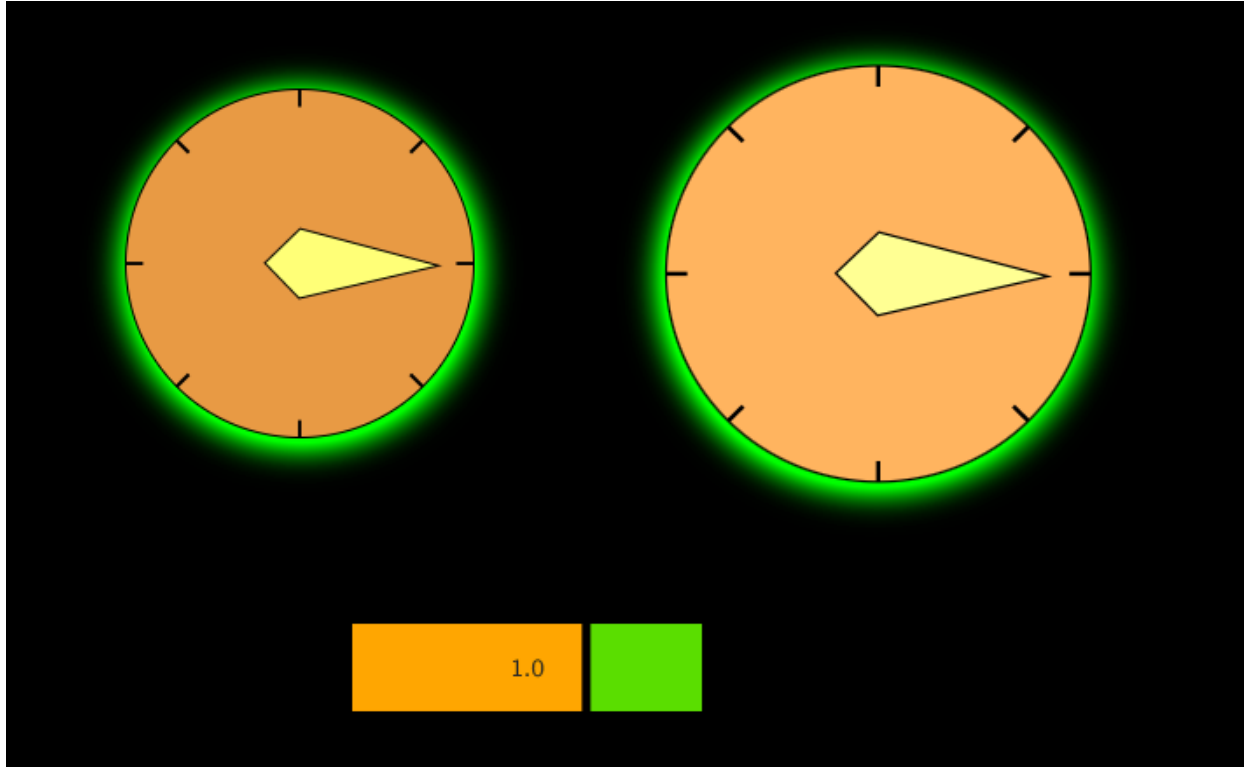


You can "decorate" a widget with a label. To add a label, click on the widget then click on the "More" button to view the extended properties. From there, click the "Add Decorator" button (1) and select "labeler" to add a label to the widget. You can then type the label text into the label field (2) in the property panel for the widget. Select Horizontal to place the label before the control and vertical to place the label underneath the control.

To later delete the label from the widget, press the red X button to the right of the decorator to be removed (in this case, the labeler) (3)

The new SmartDashboard (SFX)

Using css styles to modify the look of widgets



Most controls support CSS styling as they use JavaFX primitives. At the present time, you must enter css rules in a text box, but this will change with a full designer in a future release. All CSS is JavaFX CSS and is NOT the same as browser CSS. See the [JavaFX CSS reference page](#) for commands.

The above layout was made with a canvas with

```
-fx-background-color: black;
```

a ValueMeter with

```
-fx-background-color: orange;
```

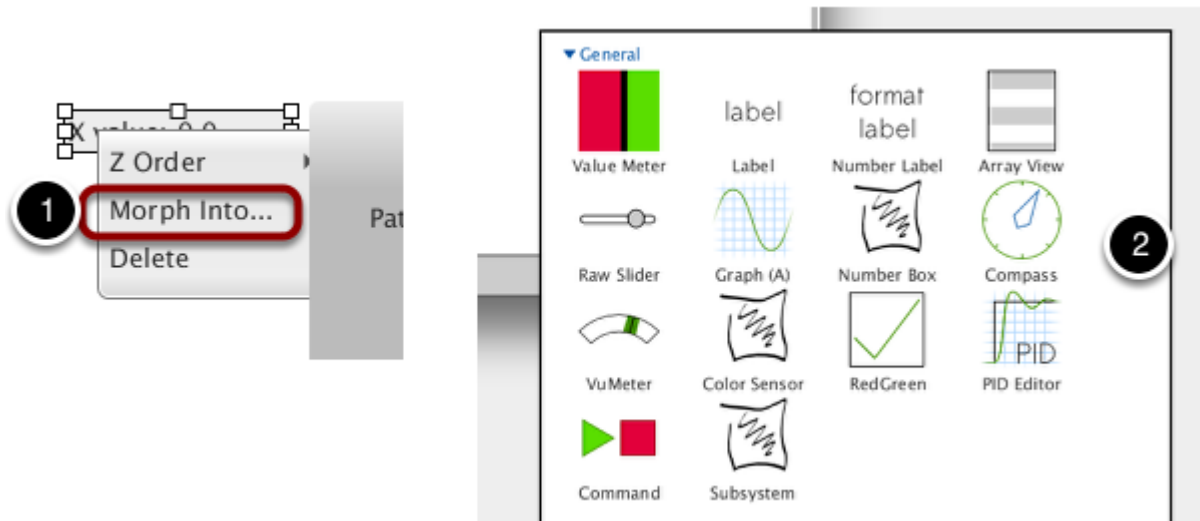
and two Compasses with

```
-fx-effect: dropshadow(gaussian, lime, 30, 0.5, -1, 3);
```

The new SmartDashboard (SFX)

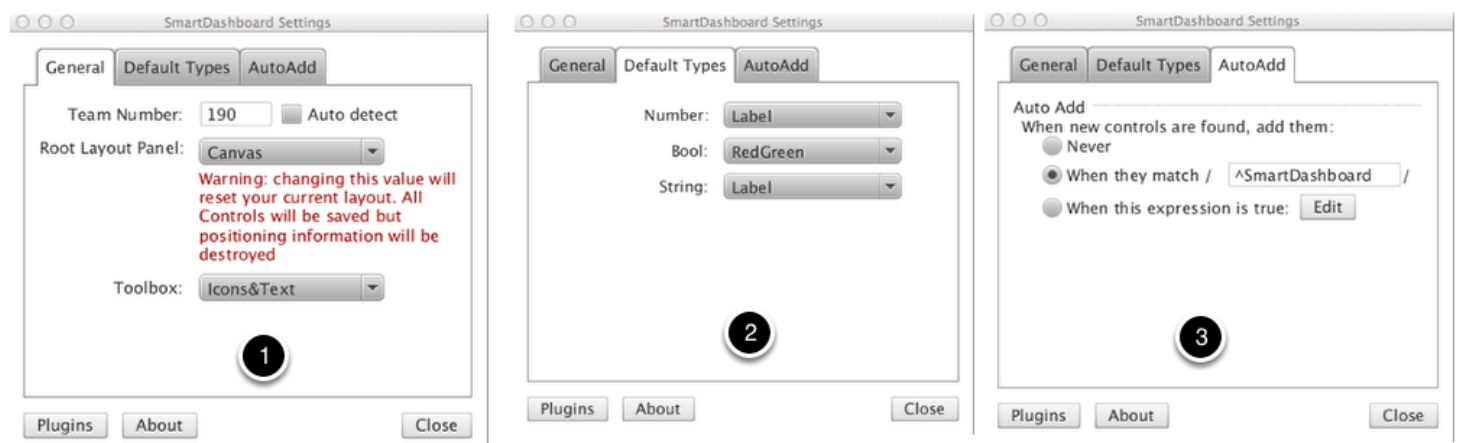
and different fill and needle colors.

Morphing widgets from one type to another



You can change the type of a widget to another widget of a compatible type. For example, if the default type for a numeric value is a text field, you can change it to a graph. Right-click on the widget and select "Morph..." (1) and choose one of the new widget types in the popup window (2). Only widgets with compatible types will be shown. Note that the conversion attempts to save properties, but is lossy if the properties have different names or do not exist.

Settings in SFX

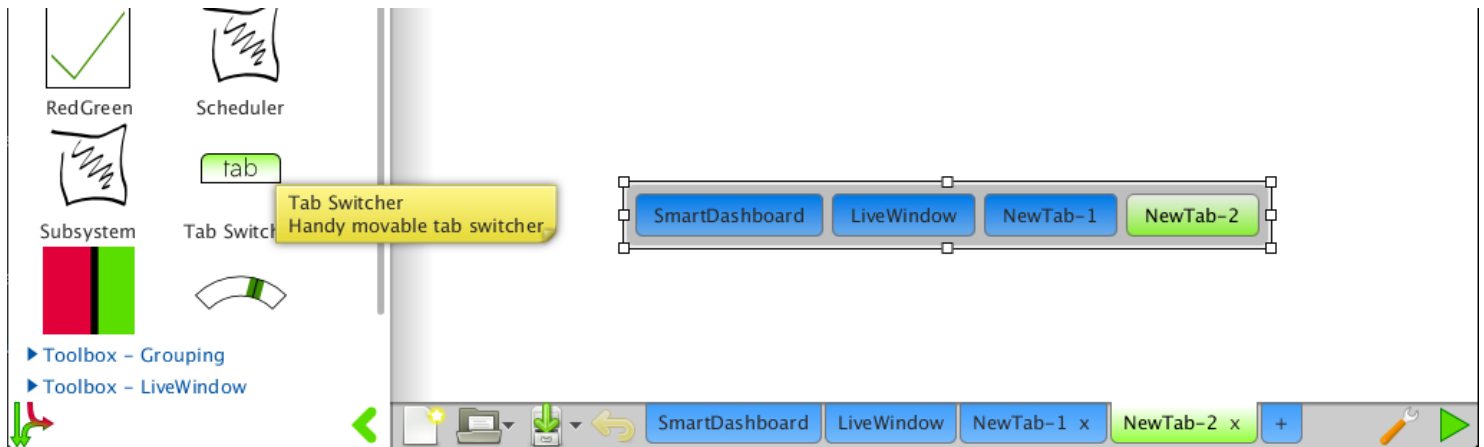


The new SmartDashboard (SFX)

There are a number of settings for SFX to set the behavior to match your requirements. The settings are in 3 panels as shown above. Described below are the options that can be set:

1. **General settings** - This sets the team number which is used for communications with the robot. The root panel layout is the default JavaFX layout type that is used for laying out the root panel in a tab. Changing this will change the way the widgets are placed on the panel when automatically added as well as your ability to manipulate the widgets once they are placed. The toolbox option sets how toolbox items will be displayed in the palette (when clicking on the plus (+) in the lower corner of the screen).
2. **Default types** - these are the types of widgets that will be automatically created when the SmartDashboard values are loaded in the "SmartDashboard" tab in sfx. For each type of data, numeric, boolean, or string select the widget type that should be created.
3. **AutoAdd settings** - widgets are automatically added to the "SmartDashboard" panel when they meet the criteria shown in this editor. By default, you can see that names that start with SmartDashboard in the name are automatically placed.

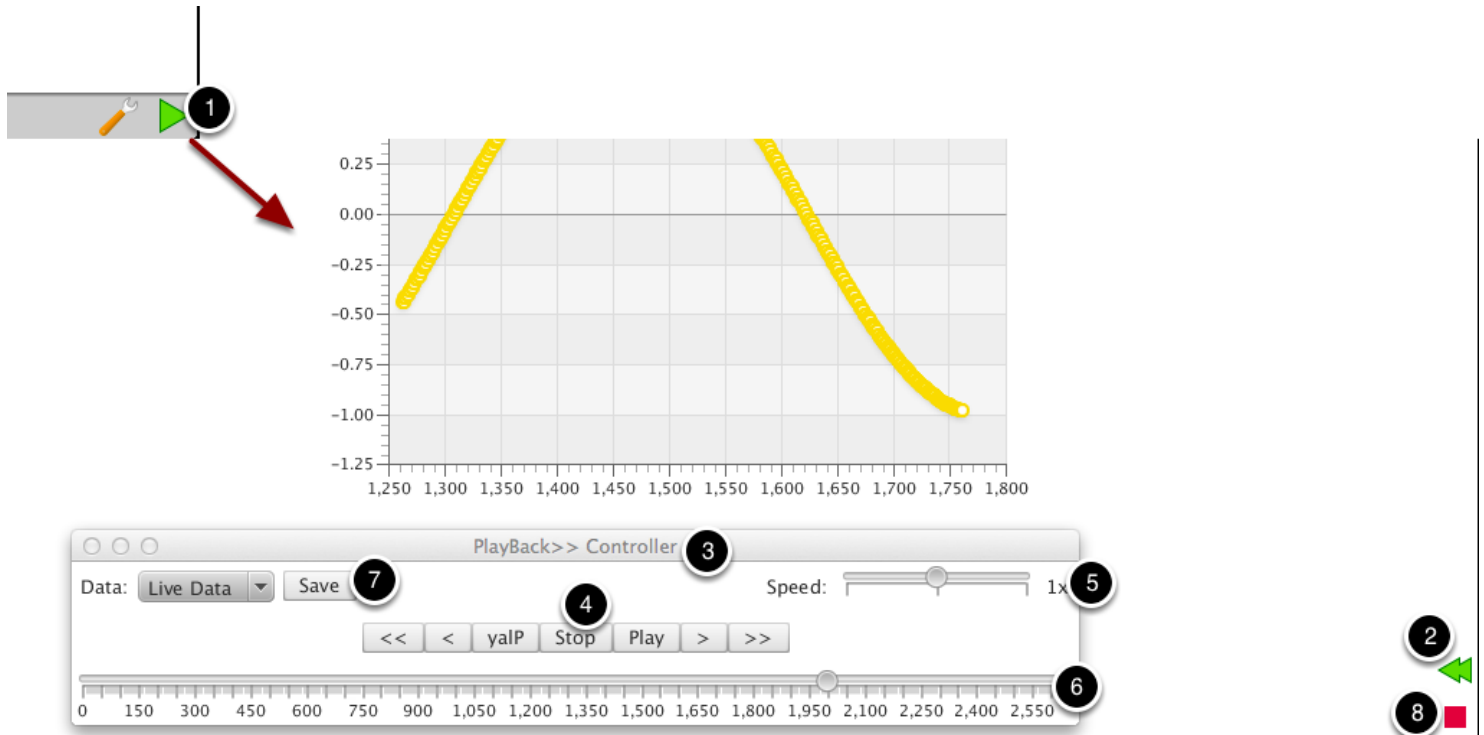
Tab Switcher



When in run mode, tabs are hidden. LiveWindow will automatically show and hide, however it is difficult to move between custom tabs. The tab switcher control enables you to switch tabs even when running. Simply drag it onto a tab from the General toolbox and run. Note you should probably add it to multiple tabs.

The new SmartDashboard (SFX)

Running & Playback



All controls display data in any mode, but you can only interact with controls in run mode. Hit the green run button (1) in the bottom right to enter run mode.

Once in run mode, you can either use the dashboard, exit (8), or open playback (2). Playback continually records changes in data over time and allows you to play these changes back at any time. When you open playback (3), it pauses all data (though it continues to record in the background). You can use the media controls (4) to step around the data, and play at speed (which can be controlled by 5), or drag the slider (6) manually to scrub around. Data can also be saved using (7) and loaded at another time to save matches, for example. To exit playback, just close the window (3)