Control System Studio Training - BOY

Kay Kasemir

ORNL/SNS

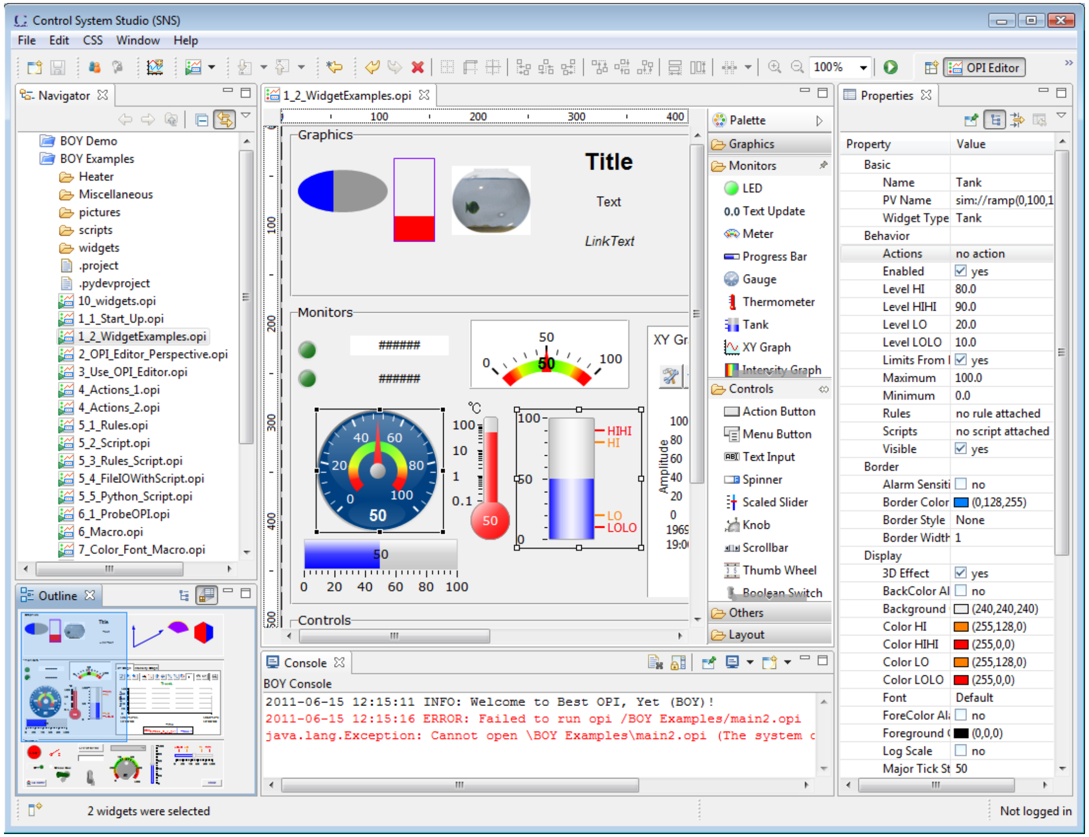
kasemirk@ornl.gov

A lot of material from Nadine Utzel, ITER and BOY online help by Xihui Chen, SNS

Jan. 2013

BOY – Best OPI, Yet

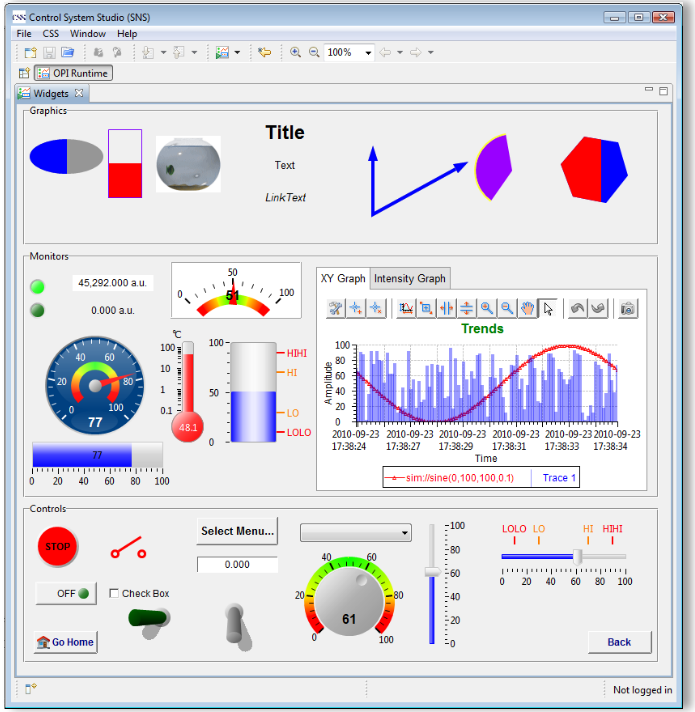
Operator Interface Editor



Property	Value
Basic	
Name	Tank
PV Name	sim://ramp(0,100,1
Widget Type	Tank
Behavior	
Actions	no action
Enabled	☑ yes
Level HI	80.0
Level HIHI	90.0
Level LO	20.0
Level LOLO	10.0
Limits From I	☑ yes
Maximum	100.0
Minimum	0.0
Rules	no rule attached
Scripts	no script attached
Visible	☑ yes
Border	
Alarm Sensiti	□ no
Border Color	(0,128,255)
Border Style	None
Border Widtł	1
Display	
3D Effect	☑ yes
BackColor Al	□ no
Background	(240,240,240)
Color HI	(255,128,0)
Color HIHI	(255,0,0)
Color LO	(255,128,0)
Color LOLO	(255,0,0)
Font	Default
ForeColor Al	□ no
Foreground (	(0,0,0)
Log Scale	□ no
Major Tick St	50

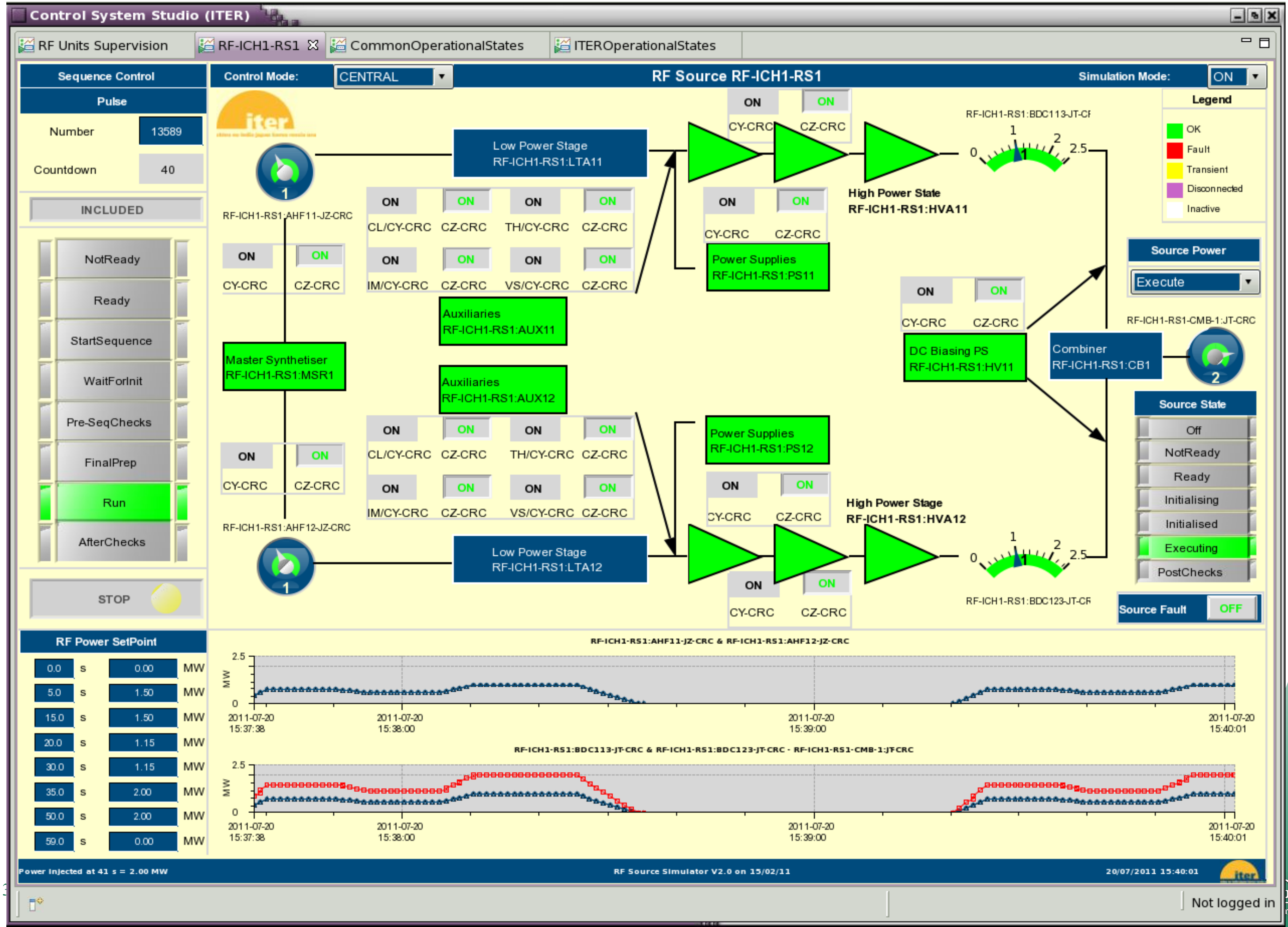
```
2011-06-15 12:15:11 INFO: Welcome to Best OPI, Yet (BOY)!
2011-06-15 12:15:16 ERROR: Failed to run opi /BOY Examples/main2.opi
java.lang.Exception: Cannot open \BOY Examples\main2.opi (The system c
```

Runtime



Similar to EDM, MEDM, SDS, DM2K, but possibly better

Example: ITER



Control System Studio (ITER)

RF Units Supervision | RF-ICH1-RS1 | CommonOperationalStates | ITEROperationalStates

Sequence Control

Pulse

Number: 13589

Countdown: 40

Control Mode: CENTRAL

RF Source RF-ICH1-RS1

Simulation Mode: ON

Source Power: Execute

Source Fault: OFF

RF Power SetPoint

0.0	s	0.00	MW
5.0	s	1.50	MW
15.0	s	1.50	MW
20.0	s	1.15	MW
30.0	s	1.15	MW
35.0	s	2.00	MW
50.0	s	2.00	MW
59.0	s	0.00	MW

Power injected at 41 s = 2.00 MW

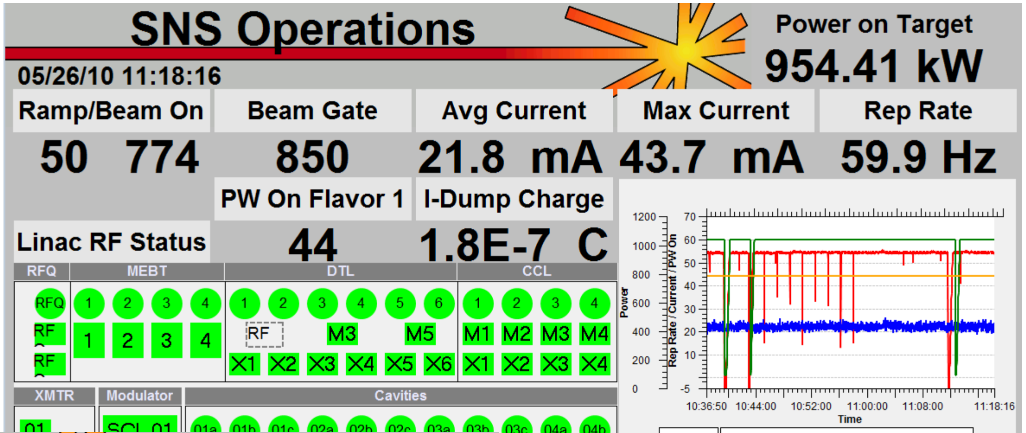
RF Source Simulator V2.0 on 15/02/11

20/07/2011 15:40:01

Not logged in

Examples: SNS

- Top-level displays created by operators

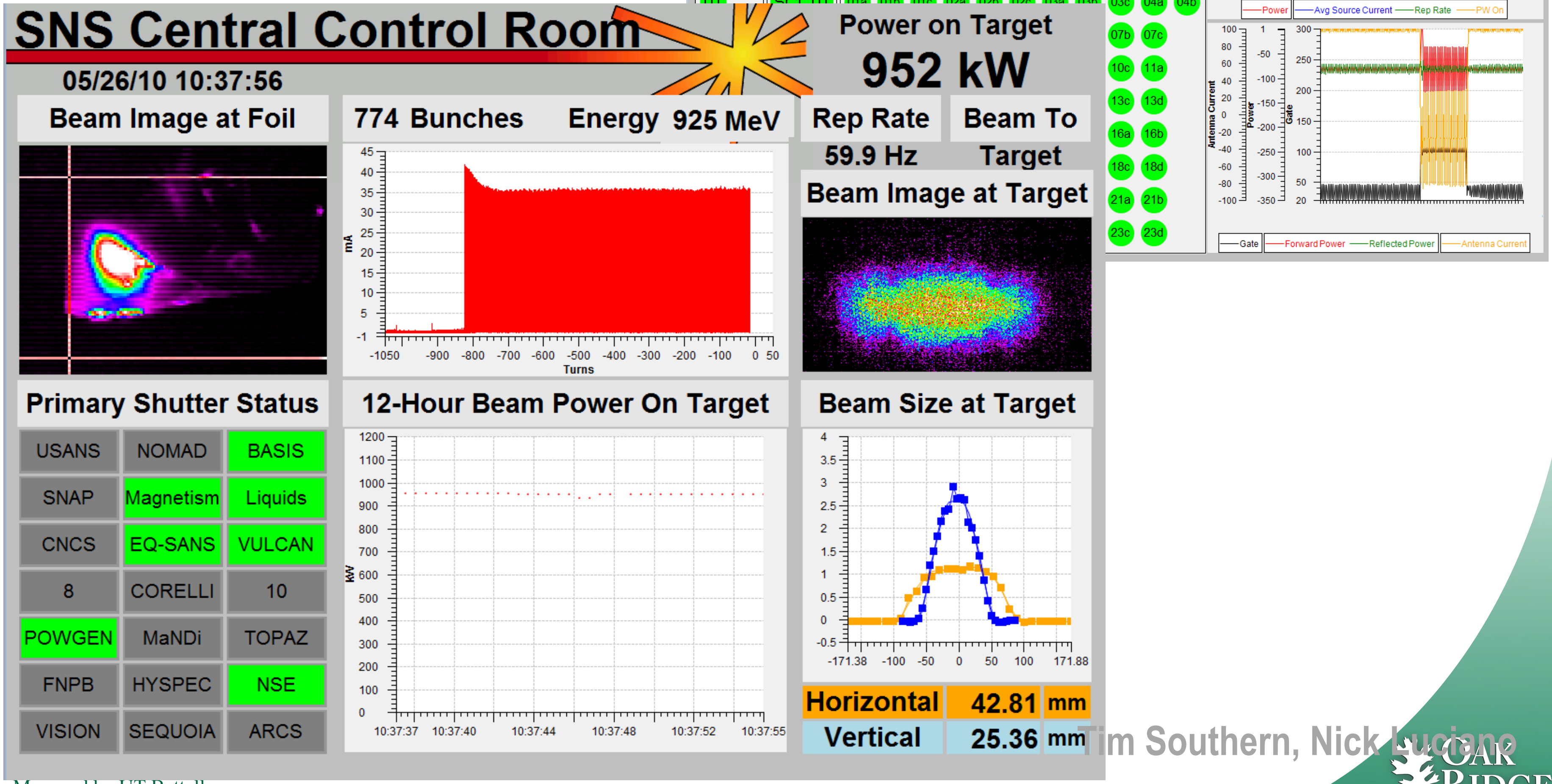


SNS Operations

05/26/10 11:18:16

Power on Target
954.41 kW

Ramp/Beam On	Beam Gate	Avg Current	Max Current	Rep Rate
50 774	850	21.8 mA	43.7 mA	59.9 Hz
	PW On Flavor 1	I-Dump Charge		
Linac RF Status	44	1.8E-7 C		



SNS Central Control Room

05/26/10 10:37:56

Power on Target
952 kW

Beam Image at Foil

774 Bunches Energy 925 MeV

Rep Rate
59.9 Hz

Beam To
Target

Beam Image at Target

Primary Shutter Status

USANS	NOMAD	BASIS
SNAP	Magnetism	Liquids
CNCS	EQ-SANS	VULCAN
8	CORELLI	10
POWGEN	MaNDi	TOPAZ
FNPB	HYSPEC	NSE
VISION	SEQUOIA	ARCS

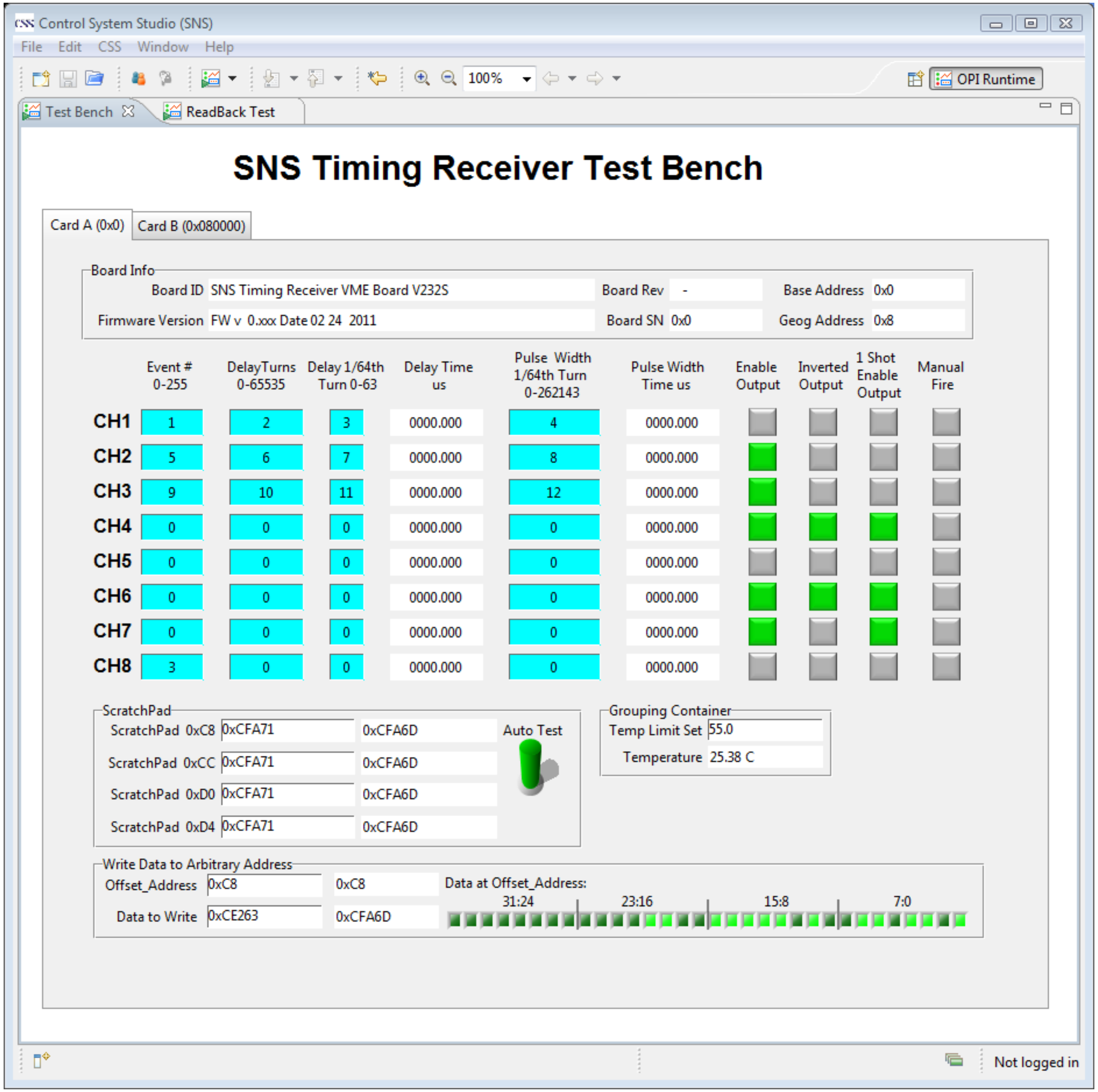
12-Hour Beam Power On Target

Beam Size at Target

Horizontal	42.81	mm
Vertical	25.36	mm

Tim Southern, Nick Luciano

Examples: SNS



SNS Timing Receiver Test Bench

Card A (0x0) | Card B (0x080000)

Board Info

Board ID	SNS Timing Receiver VME Board V232S	Board Rev	-	Base Address	0x0
Firmware Version	FW v 0.xxx Date 02 24 2011	Board SN	0x0	Geog Address	0x8

	Event # 0-255	DelayTurns 0-65535	Delay 1/64th Turn 0-63	Delay Time us	Pulse Width 1/64th Turn 0-262143	Pulse Width Time us	Enable Output	Inverted Output	1 Shot Enable Output	Manual Fire
CH1	1	2	3	0000.000	4	0000.000	⬜	⬜	⬜	⬜
CH2	5	6	7	0000.000	8	0000.000	🟩	⬜	⬜	⬜
CH3	9	10	11	0000.000	12	0000.000	🟩	⬜	⬜	⬜
CH4	0	0	0	0000.000	0	0000.000	🟩	🟩	🟩	⬜
CH5	0	0	0	0000.000	0	0000.000	⬜	⬜	⬜	⬜
CH6	0	0	0	0000.000	0	0000.000	🟩	🟩	🟩	⬜
CH7	0	0	0	0000.000	0	0000.000	🟩	⬜	🟩	⬜
CH8	3	0	0	0000.000	0	0000.000	⬜	⬜	⬜	⬜

ScratchPad

ScratchPad 0xC8	0xCFA71	0xCFA6D
ScratchPad 0xCC	0xCFA71	0xCFA6D
ScratchPad 0xD0	0xCFA71	0xCFA6D
ScratchPad 0xD4	0xCFA71	0xCFA6D

Auto Test

Grouping Container

Temp Limit Set 55.0

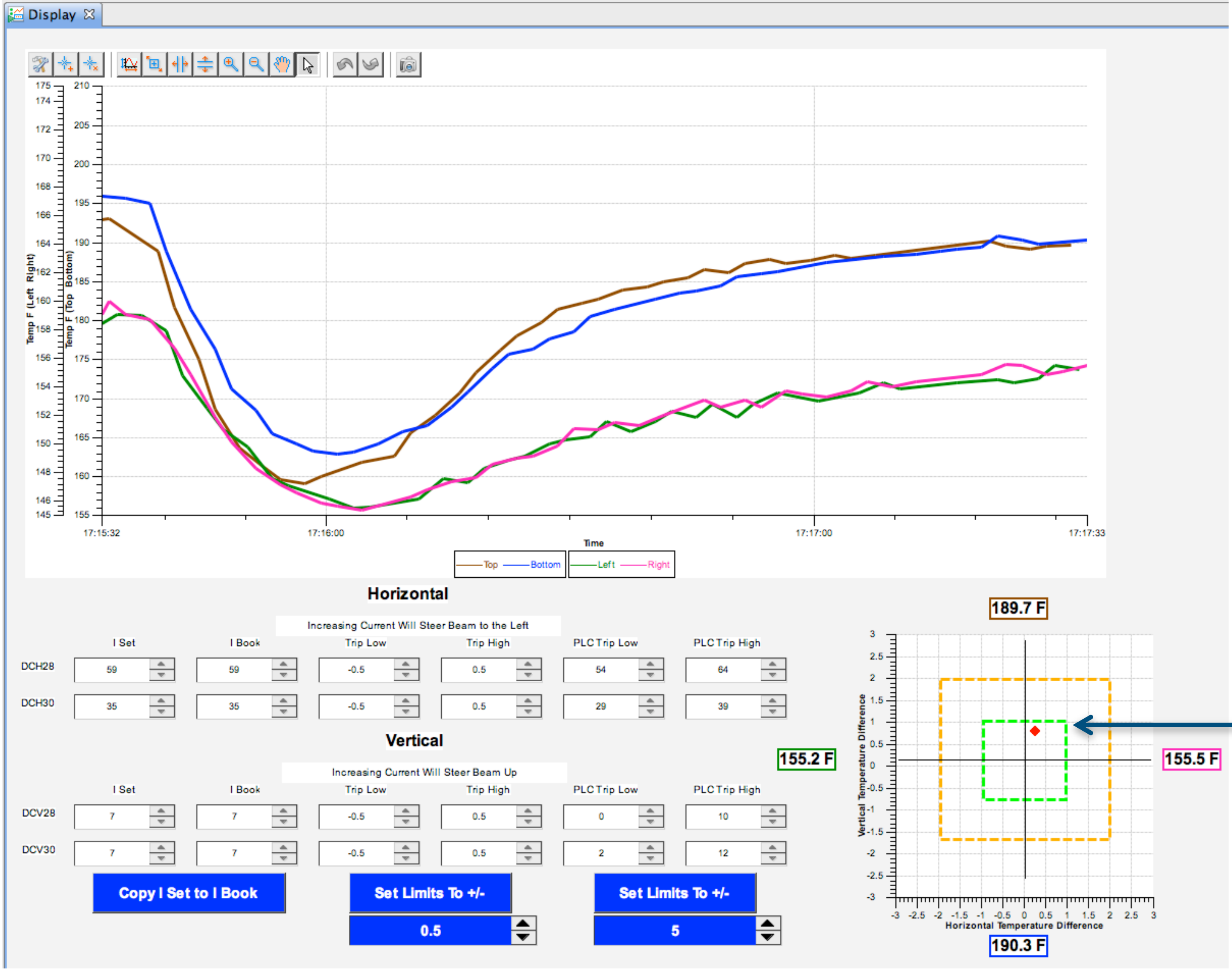
Temperature 25.38 C

Write Data to Arbitrary Address

Offset_Address	0xC8	0xC8
Data to Write	0xCE263	0xCFA6D

Data at Offset_Address:

Examples: SNS “Steering” Tool



Horizontal

Increasing Current Will Steer Beam to the Left

	I Set	I Book	Trip Low	Trip High	PLC Trip Low	PLC Trip High
DCH28	59	59	-0.5	0.5	54	64
DCH30	35	35	-0.5	0.5	29	39

Vertical

Increasing Current Will Steer Beam Up

	I Set	I Book	Trip Low	Trip High	PLC Trip Low	PLC Trip High
DCV28	7	7	-0.5	0.5	0	10
DCV30	7	7	-0.5	0.5	2	12

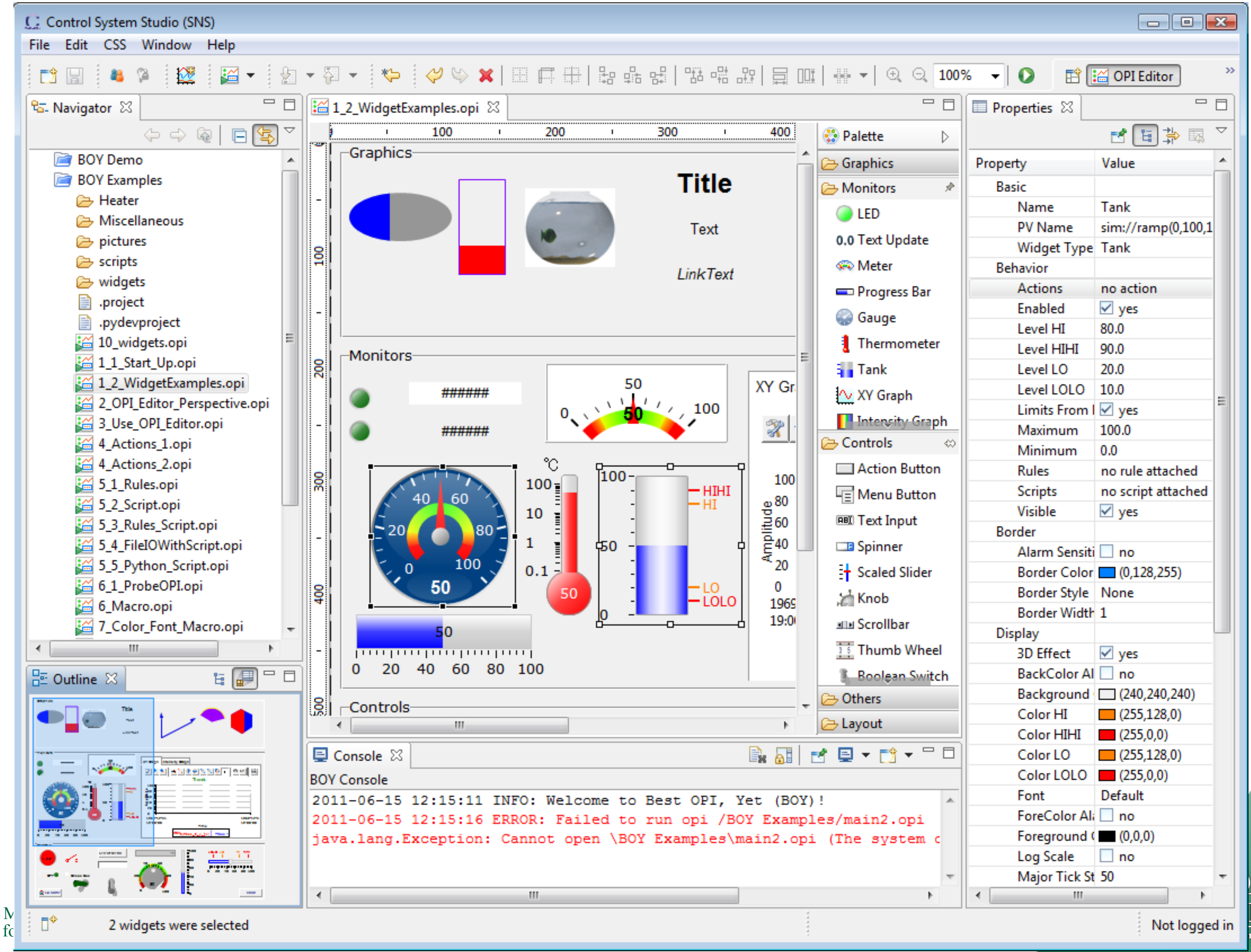
Set Limits To +/- 0.5

Set Limits To +/- 5

Try to get spot into the green, at least into orange

Tim Southern, Nick Luciano

OPI Editor



Property	Value
Basic	
Name	Tank
PV Name	sim://ramp(0,100,1
Widget Type	Tank
Behavior	
Actions	no action
Enabled	☑ yes
Level HI	80.0
Level HIHI	90.0
Level LO	20.0
Level LOLO	10.0
Limits From I	☑ yes
Maximum	100.0
Minimum	0.0
Rules	no rule attached
Scripts	no script attached
Visible	☑ yes
Border	
Alarm Sensiti	□ no
Border Color	🟦 (0,128,255)
Border Style	None
Border Width	1
Display	
3D Effect	☑ yes
BackColor Al	□ no
Background	□ (240,240,240)
Color HI	🟧 (255,128,0)
Color HIHI	🟥 (255,0,0)
Color LO	🟧 (255,128,0)
Color LOLO	🟥 (255,0,0)
Font	Default
ForeColor Al	□ no
Foreground (	⬛ (0,0,0)
Log Scale	□ no
Major Tick St	50

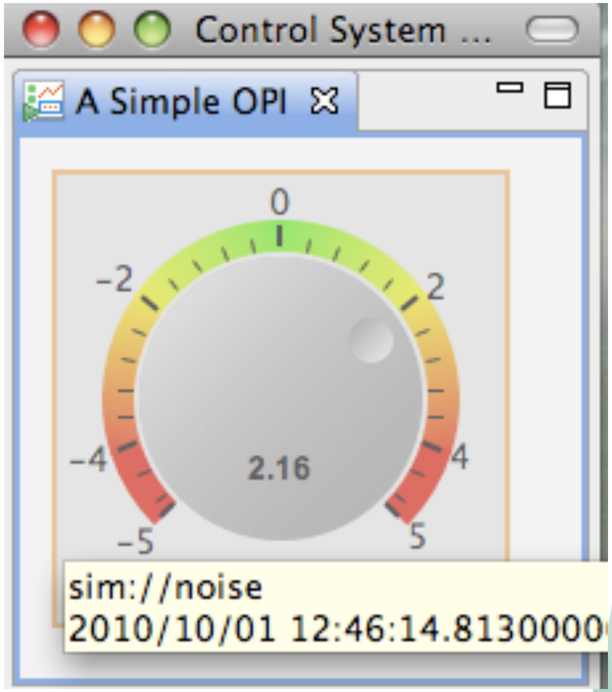
```
2011-06-15 12:15:11 INFO: Welcome to Best OPI, Yet (BOY)!
2011-06-15 12:15:16 ERROR: Failed to run opi /BOY Examples/main2.opi
java.lang.Exception: Cannot open \BOY Examples\main2.opi (The system c
```

2 widgets were selected

Not logged in

Main Idea: Simple Things are Easy

1. Drag a widget, e.g. Knob, from palette to editor
2. Enter the PV name in Properties view
3. Click the "Run" ▶ button to execute!



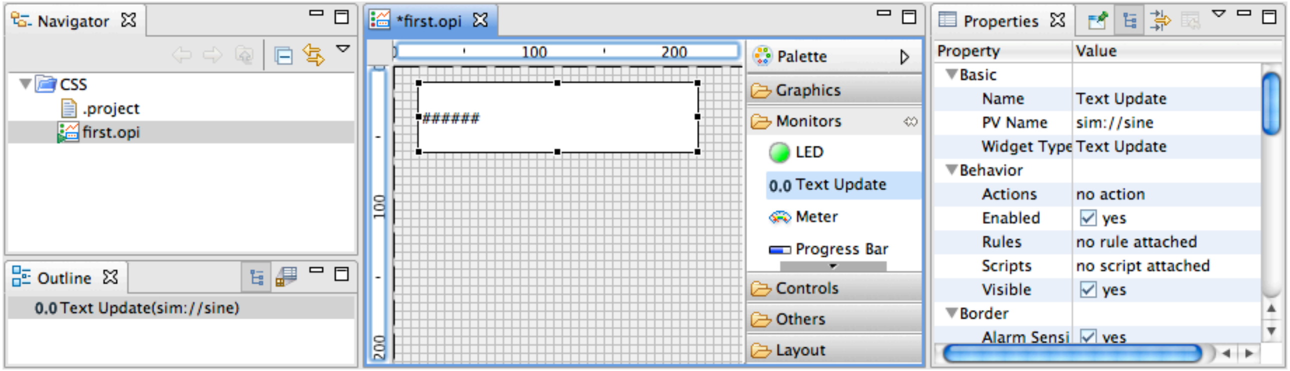
sim://noise
2010/10/01 12:46:14.8130000

What you will get

- ✓ PV *value* as text and via knob position
- ✓ PV *severity* reflected in border color
- ✓ PV *name* and *value* shown in tool-tip
- ✓ PV's *display limits* set the knob's default range
- ✓ Indicate 'disconnected' state via a pink border
- ✓ Widget will be greyed-out if read-only

Exercise: First Display

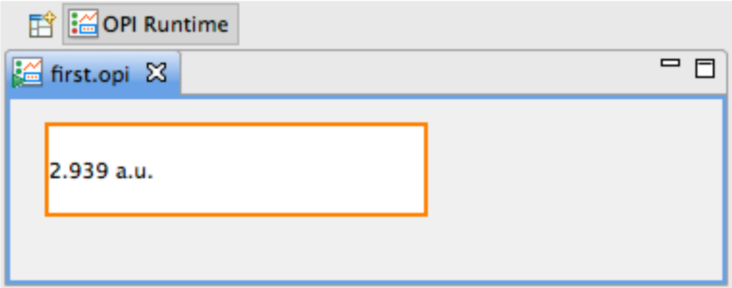
- Menu *CSS*,
  - *Display*, *OPI Editor Perspective*
  - *Display*, Install *OPI Examples*
- Navigator Context menu on CSS: *New*, *OPI File*, call it “first.opi”
  - *Or* Menu *File*, *New*, *BOY*, *OPI File*



0.0 Text Update(sim://sine)

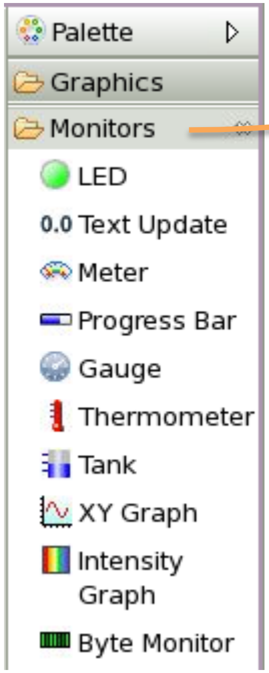
Property	Value
▼Basic	
Name	Text Update
PV Name	sim://sine
Widget Type	Text Update
▼Behavior	
Actions	no action
Enabled	☑ yes
Rules	no rule attached
Scripts	no script attached
Visible	☑ yes
▼Border	
Alarm Sensi	☑ yes

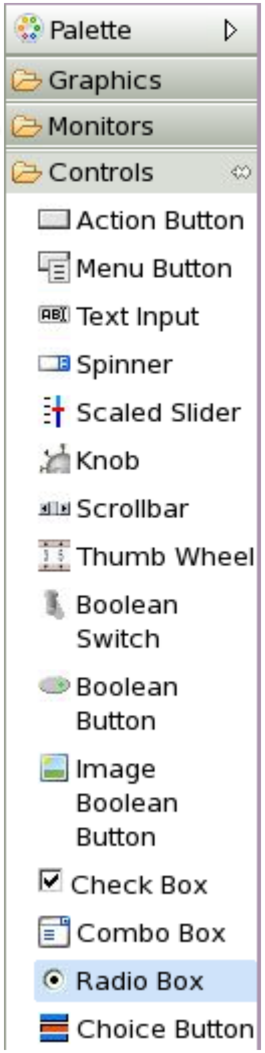
- Locate in Palette: *Monitors*, *Text Update*
  - ‘Drag’ *Text Update* onto display grid
  - Move widget around, resize
- Locate *Properties* View
  - Enter *PV Name* “sim://sine”
- Press Run ▶ button in Toolbar



2.939 a.u.

Widget Palette Hints

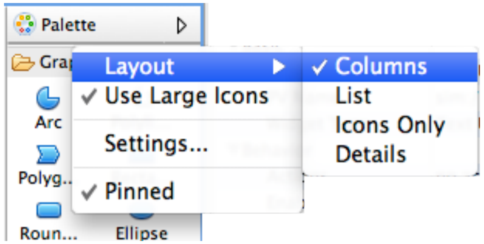




Many widgets, hard to see them all

- **Scroll**
- **Click on section header**

- **Try the ‘pins’**
- **Header Context menu offers *Columns* mode to display Widgets as small icons in columns**

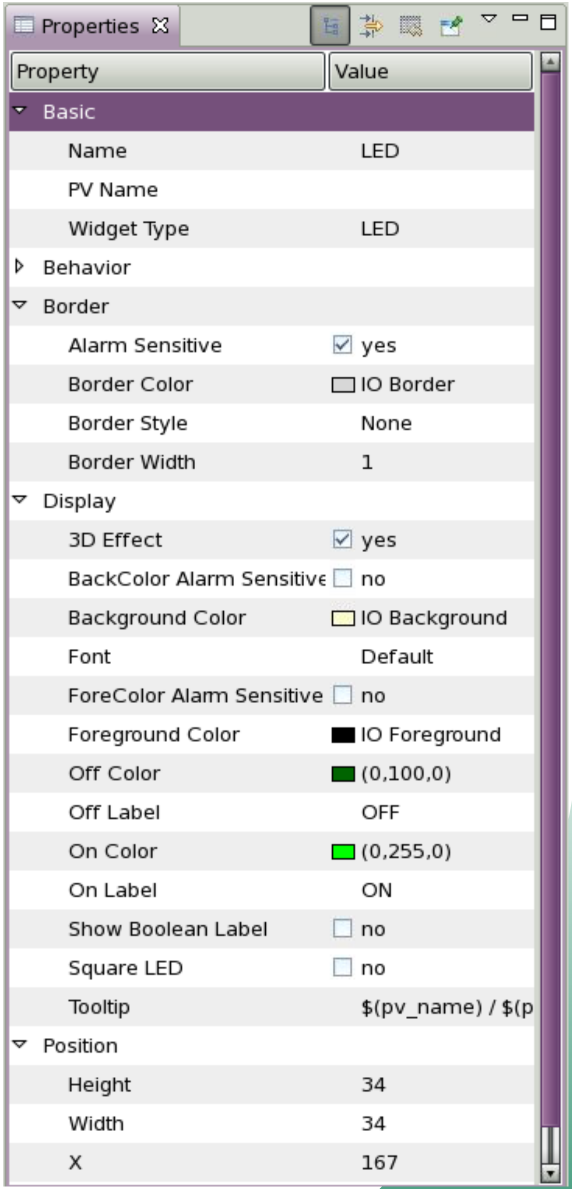


Widget Properties

- Widgets are configured by setting Properties in the *Properties* view

- Common Properties:
  - Name
  - Position*
  - Background color
  - Border

- Widgets that read/write PVs:
  - Basic: PV Name
  - Border: Alarm Sensitive
  - Behavior: Limits from PV



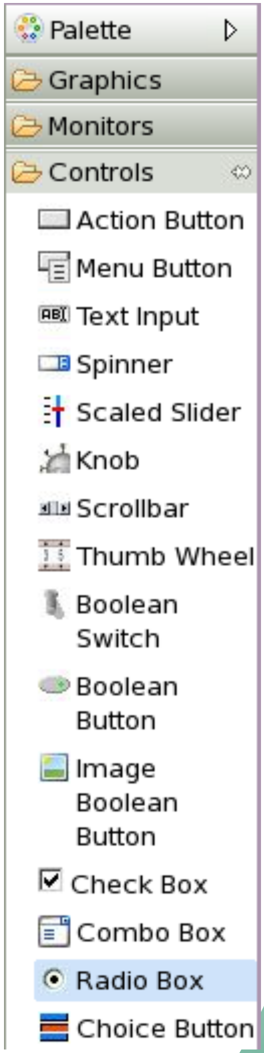
Properties

Property	Value
Basic	
Name	LED
PV Name	
Widget Type	LED
Behavior	
Border	
Alarm Sensitive	☑ yes
Border Color	□ IO Border
Border Style	None
Border Width	1
Display	
3D Effect	☑ yes
BackColor Alarm Sensitive	□ no
Background Color	□ IO Background
Font	Default
ForeColor Alarm Sensitive	□ no
Foreground Color	■ IO Foreground
Off Color	🟩 (0,100,0)
Off Label	OFF
On Color	🟩 (0,255,0)
On Label	ON
Show Boolean Label	□ no
Square LED	□ no
Tooltip	$(pv_name) / $(p
Position	
Height	34
Width	34
X	167

* Position can also be modified by moving or resizing the widget in the editor, or via Toolbar buttons to align etc.

Exercise: Extend First Display

- Locate in Palette: *Controls*, *Knob*
- Drag *Knob* onto display
- Move *Knob* around, resize
- Locate Property *PV Name* for Knob
- Enter "sim://sine"
- Create another *Knob*:
  - PV Name = "loc://test",
  - "Increment" = 0.1
  - "Limits from PV" = no
- Run ▶



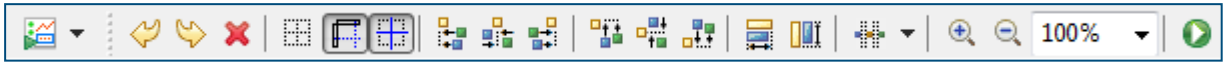
- Note how the "sim://sine" Knob is really read-only, but you can change the "loc://test" PV via the Knob

Exercise: Editing Features

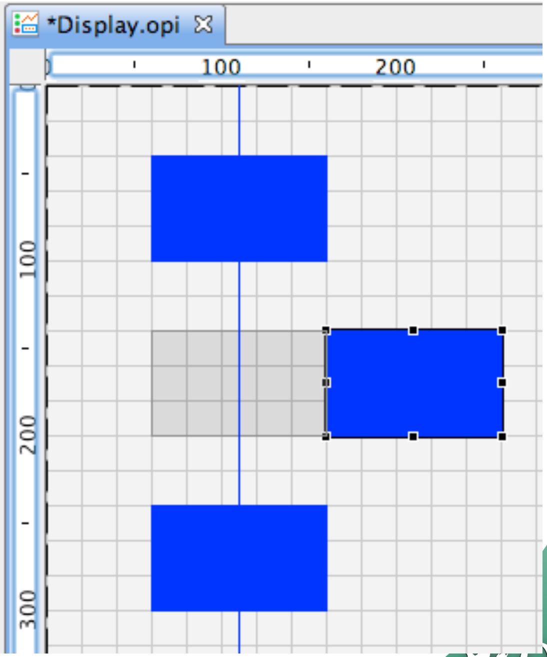
Add, duplicate Widgets in various ways

- Drag & Drop from Palette
- Copy/paste, Ctrl+Drag existing widgets to duplicate

- Arrange them on the display



- Snap to grid, guideline, other widgets
- Align, distribute

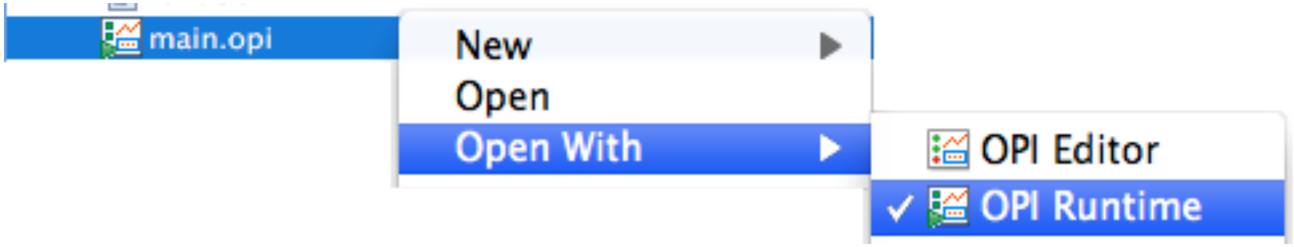


Select multiple widgets to

- Edit common properties
- Adjust size or move around

OPI Files: Run or Edit?

- Default: Double-click on *.opi in Navigator opens in “OPI Runtime”, i.e. executes the display
- Context menu allows to select
  a) Editor to edit?
  b) Runtime to execute?



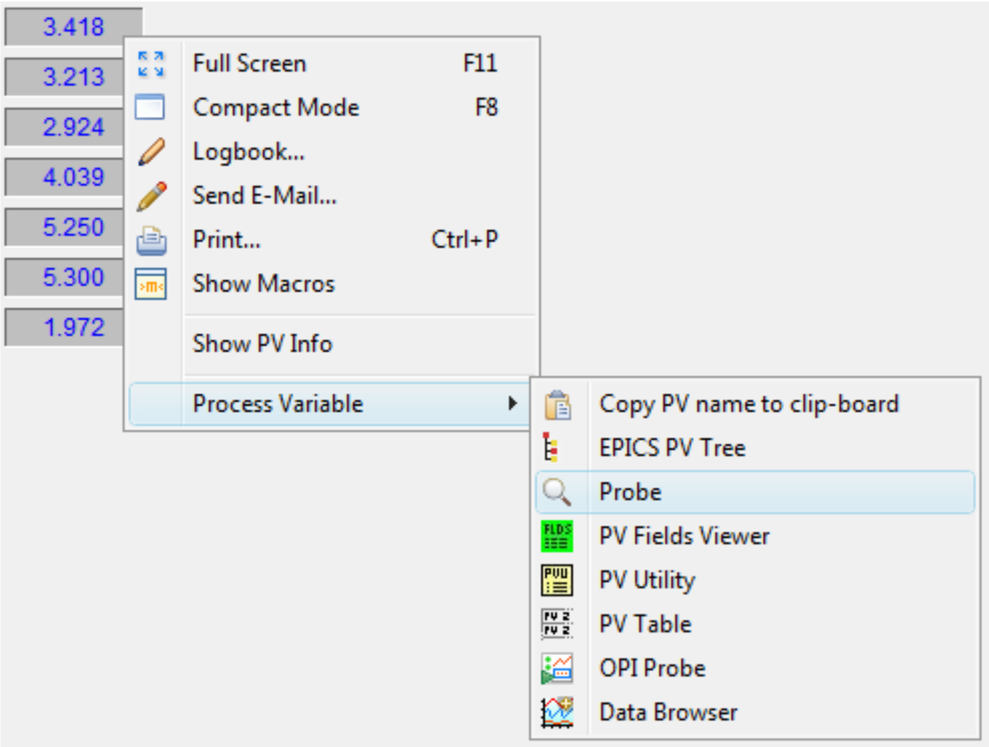
- Once you select “Editor”, that will become the double-click default
  – Select “Runtime” once to restore previous default

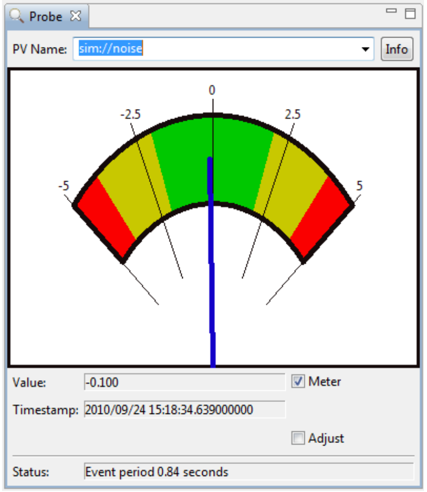
Exercise: Edit vs. Runtime Mode

- Close all CSS Editors (Menu *File*, *Close All*)
- In the Navigator, double-click on the first.opi that you created before
  - Does it open in the Editor or Runtime?
- In the Navigator, open the Context Menu on first.opi and select Open With, OPI Editor.
  - Close first.opi, now double-click the file in the Navigator. Does it open in the Editor?
- In the Navigator, open the Context Menu on first.opi and select Open With, OPI Runtime.
  - Close first.opi, now double-click the file in the Navigator. Does it open in the Runtime?

Exercise: Send PV to other CSS tools

- Run the OPI that you created
- Use CSS Process Variable context menu on a widget that displays a PV to open Probe





PV Name: sim://noise

Value: -0.100 ☑ Meter

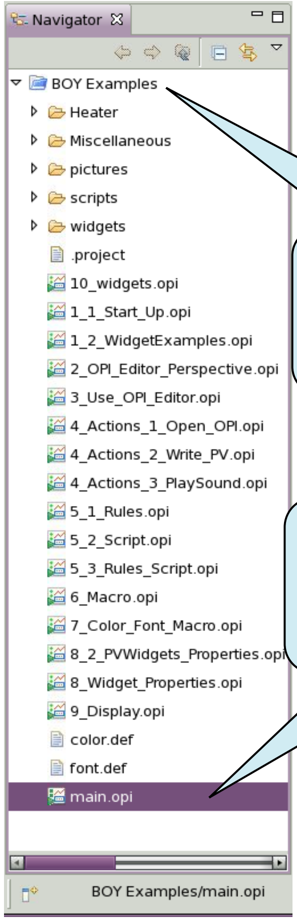
Timestamp: 2010/09/24 15:18:34.639000000

□ Adjust

Status: Event period 0.84 seconds

Example Displays

- **Installed via Menu *CSS*, *Display*, *Install OPI Examples***



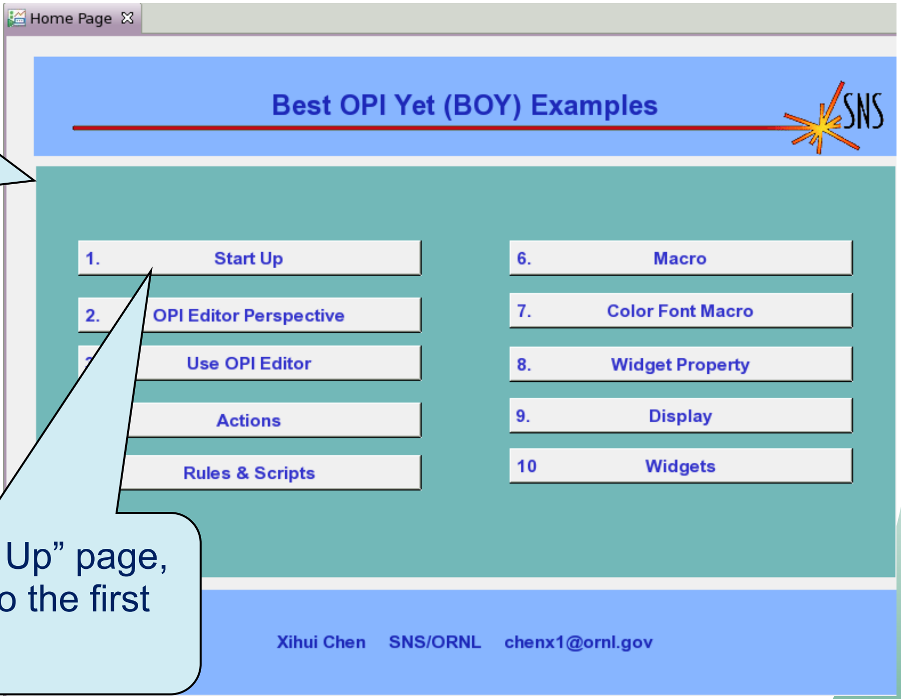
BOY Examples
- Heater
- Miscellaneous
- pictures
- scripts
- widgets
- .project
- 10_widgets.opi
- 1_1_Start_Up.opi
- 1_2_WidgetExamples.opi
- 2_OPI_Editor_Perspective.opi
- 3_Use_OPI_Editor.opi
- 4_Actions_1_Open_OPI.opi
- 4_Actions_2_Write_PV.opi
- 4_Actions_3_PlaySound.opi
- 5_1_Rules.opi
- 5_2_Script.opi
- 5_3_Rules_Script.opi
- 6_Macro.opi
- 7_Color_Font_Macro.opi
- 8_2_PVWidgets_Properties.opi
- 8_Widget_Properties.opi
- 9_Display.opi
- color.def
- font.def
- main.opi

BOY Examples/main.opi

Note new project named BOY Examples

Double-click on main.opi file to open

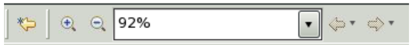
Explore the examples

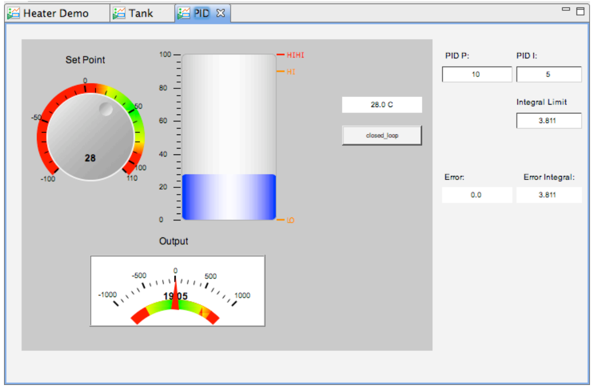


Check the “Start Up” page, which is similar to the first two exercises

Remember: You can *Open With*, .. *Editor* to see implementation

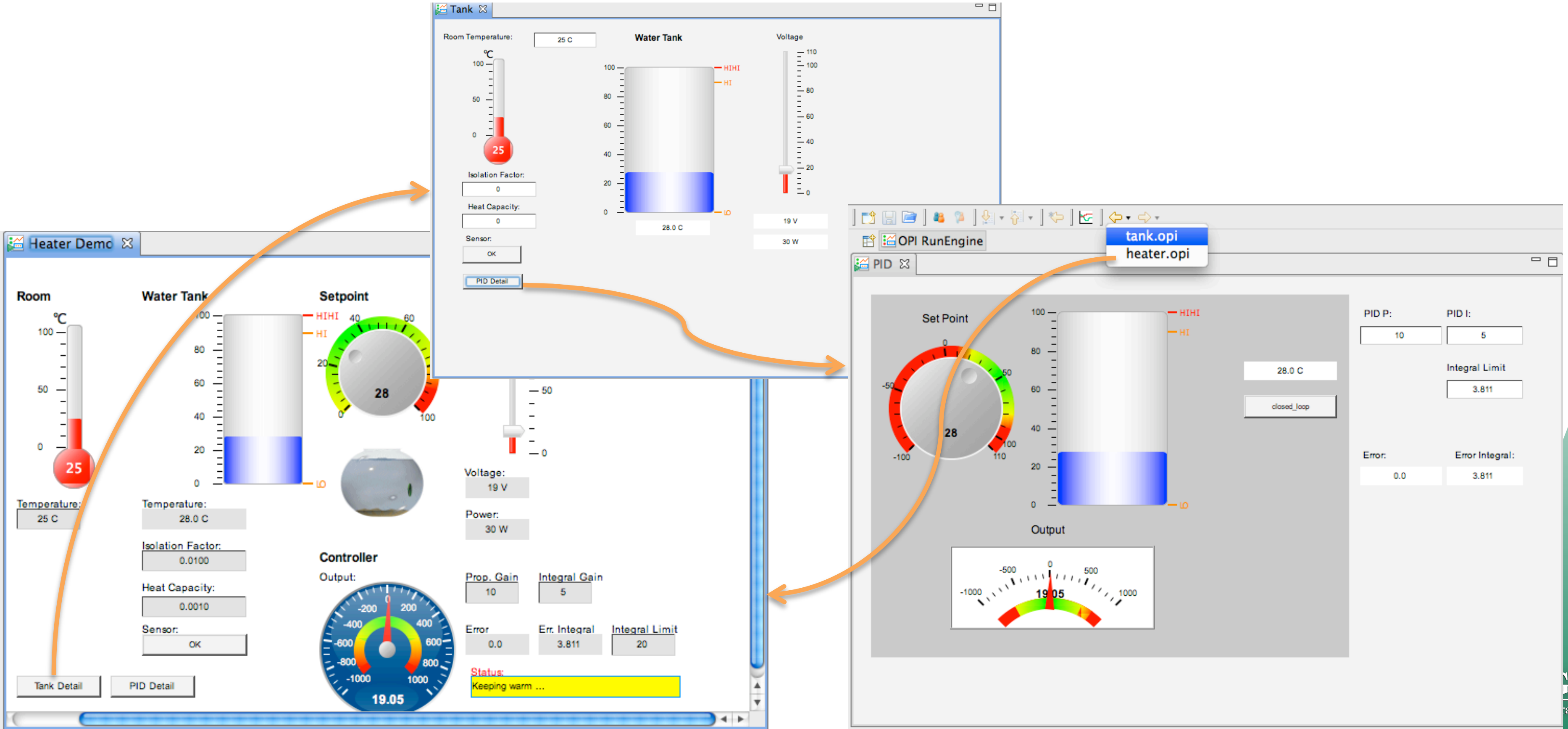
Exercise: Screen Navigation

- Similar to hyperlinks in a Web Browser:
  - Default: Linked display replaces the current display.
  - Zoom in/out, go "back" via toolbar: 
  - Use context menu to open in 'tabs' or new Window



OPIs in 'Tabs'

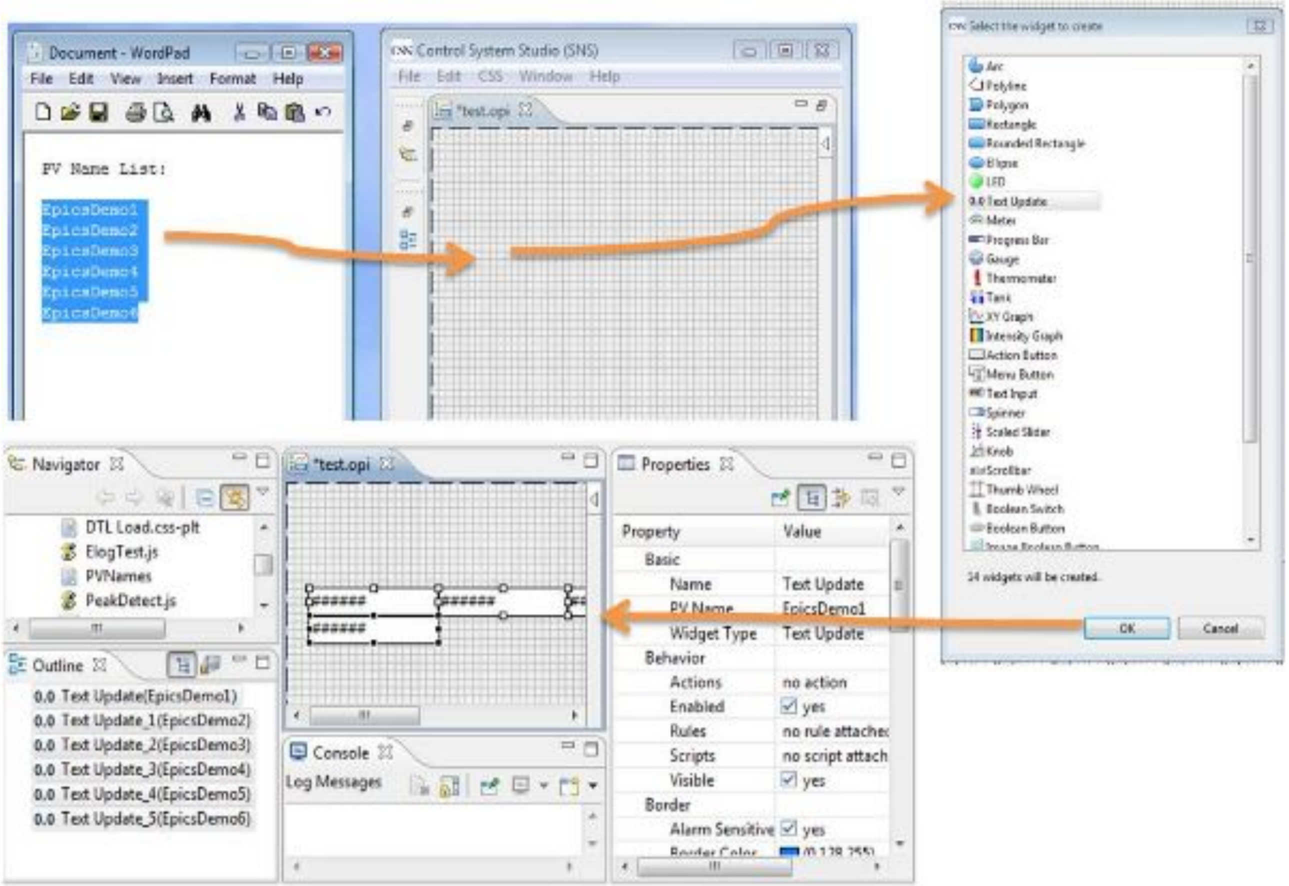
Try with OPI Examples: Open in tab, … Window



Hint: Drop PV Names

- Assume you have some text document with a list of PVs

- How to quickly create a display with Text Update widgets for these PVs?
  - Just drag the names into the display
  - Will be prompted for the type of widget



PV Name List:

EpicsDemo1
EpicsDemo2
EpicsDemo3
EpicsDemo4
EpicsDemo5
EpicsDemo6

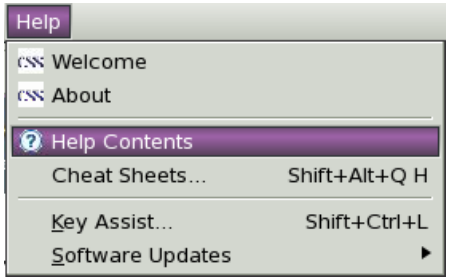
0.0 Text Update(EpicsDemo1)
0.0 Text Update_1(EpicsDemo2)
0.0 Text Update_2(EpicsDemo3)
0.0 Text Update_3(EpicsDemo4)
0.0 Text Update_4(EpicsDemo5)
0.0 Text Update_5(EpicsDemo6)

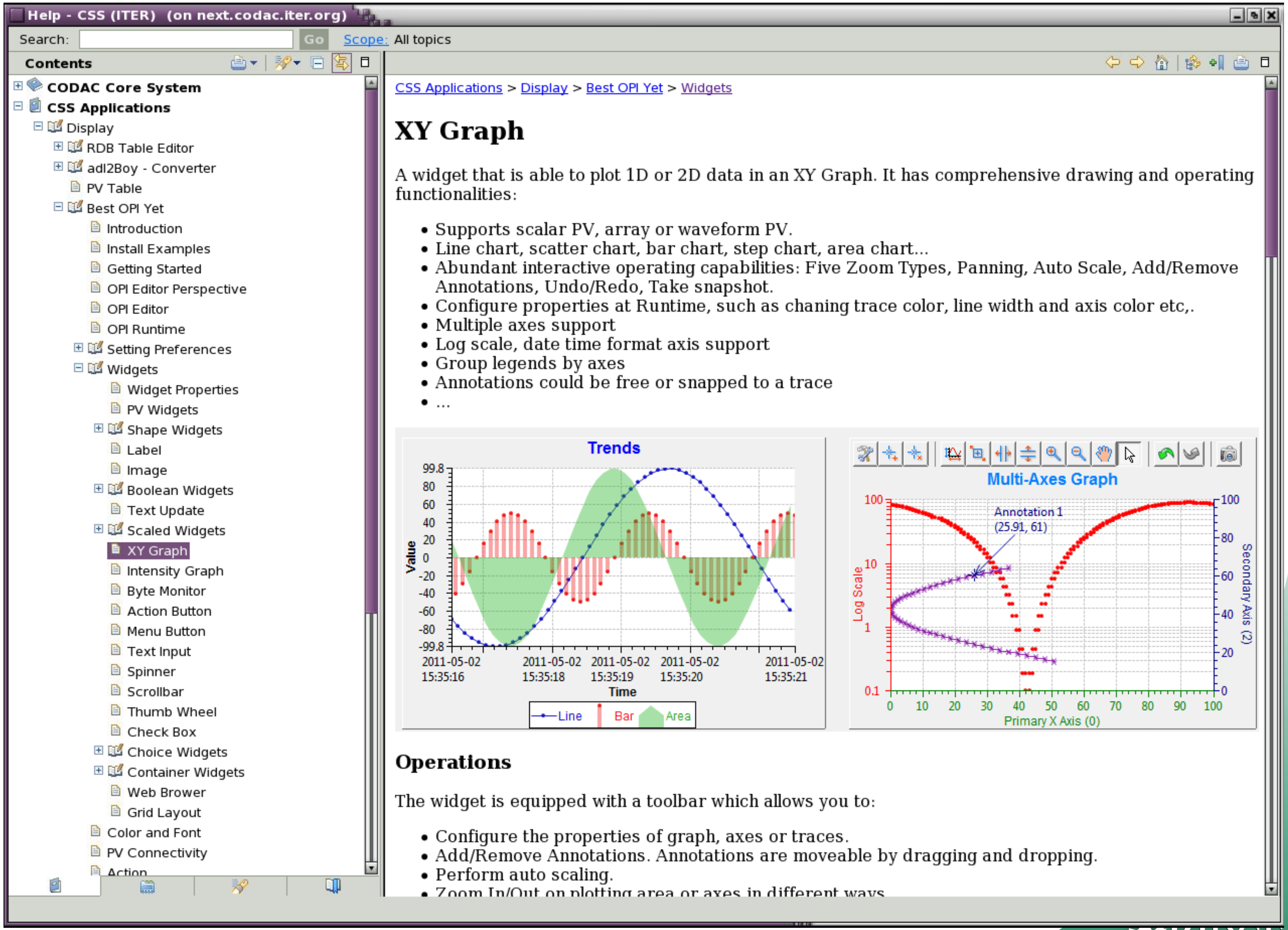
Property	Value
Basic	
Name	Text Update
PV Name	EpicsDemo1
Widget Type	Text Update
Behavior	
Actions	no action
Enabled	☑ yes
Rules	no rule attached
Scripts	no script attach
Visible	☑ yes
Border	
Alarm Sensitive	☑ yes
Border Color	(0,128,255)

14 widgets will be created.

Exercise: View Online Help

- Find the “Widgets” section





Help - CSS (ITER) (on next.codac.iter.org)

Search: Go Scope: All topics

Contents

- CODAC Core System
- CSS Applications
  - Display
    - RDB Table Editor
    - adl2Boy - Converter
    - PV Table
    - Best OPI Yet
      - Introduction
      - Install Examples
      - Getting Started
      - OPI Editor Perspective
      - OPI Editor
      - OPI Runtime
      - Setting Preferences
      - Widgets
        - Widget Properties
        - PV Widgets
        - Shape Widgets
        - Label
        - Image
        - Boolean Widgets
        - Text Update
        - Scaled Widgets
        - XY Graph
        - Intensity Graph
        - Byte Monitor
        - Action Button
        - Menu Button
        - Text Input
        - Spinner
        - Scrollbar
        - Thumb Wheel
        - Check Box
        - Choice Widgets
        - Container Widgets
        - Web Brower
        - Grid Layout
      - Color and Font
      - PV Connectivity
      - Action

CSS Applications > Display > Best OPI Yet > Widgets

XY Graph

A widget that is able to plot 1D or 2D data in an XY Graph. It has comprehensive drawing and operating functionalities:

- Supports scalar PV, array or waveform PV.
- Line chart, scatter chart, bar chart, step chart, area chart...
- Abundant interactive operating capabilities: Five Zoom Types, Panning, Auto Scale, Add/Remove Annotations, Undo/Redo, Take snapshot.
- Configure properties at Runtime, such as chaning trace color, line width and axis color etc,.
- Multiple axes support
- Log scale, date time format axis support
- Group legends by axes
- Annotations could be free or snapped to a trace
- ...

Operations

The widget is equipped with a toolbar which allows you to:

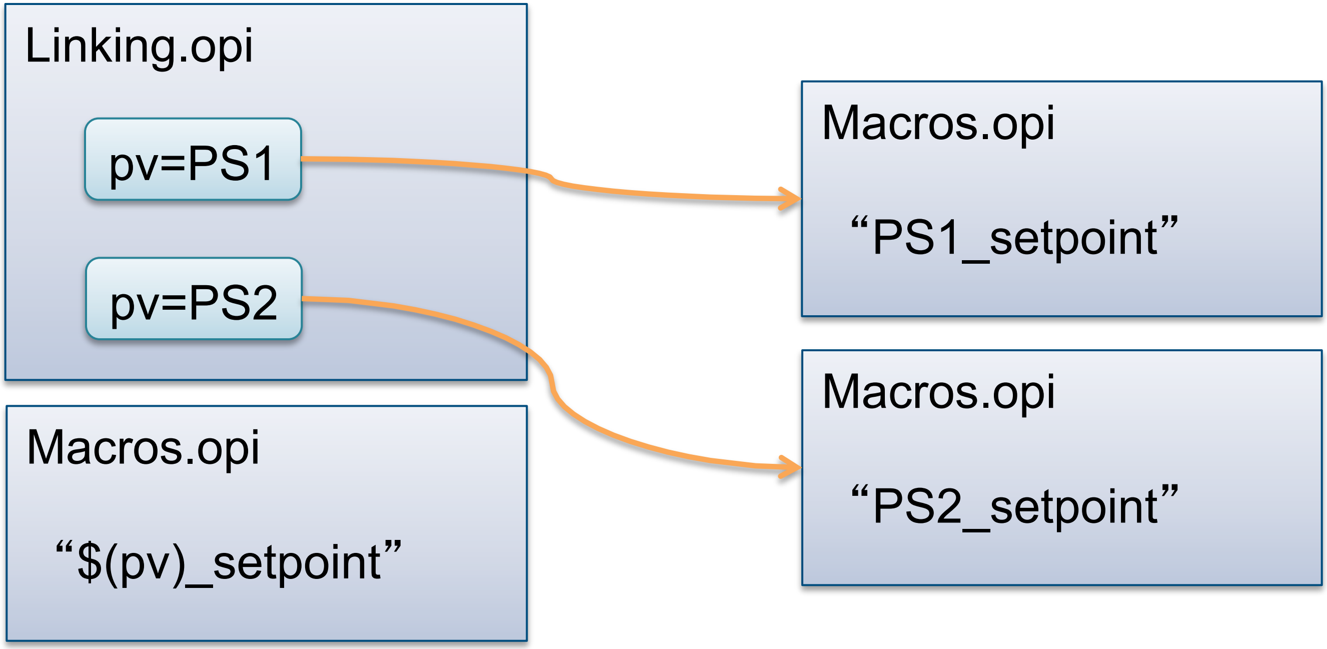
- Configure the properties of graph, axes or traces.
- Add/Remove Annotations. Annotations are moveable by dragging and dropping.
- Perform auto scaling.
- Zoom In/Out on plotting area or axes in different ways

Macros

Usage: $(macro) or ${macro}

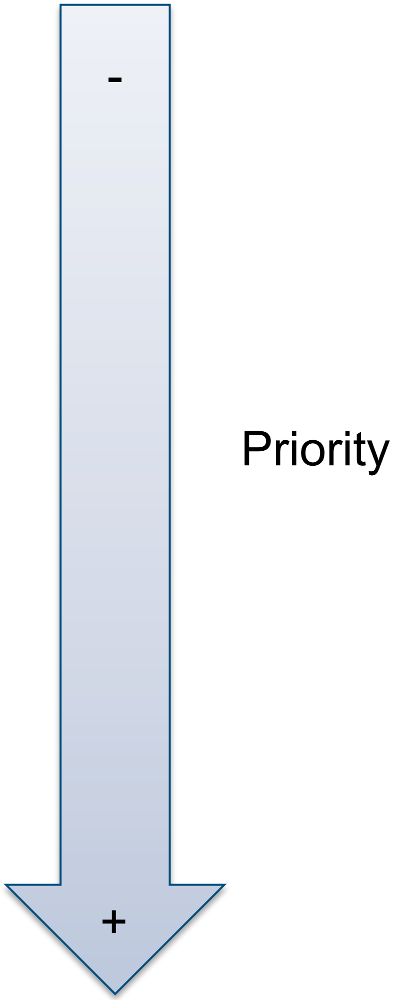
- **Wherever you enter a widget property**
- **Most often used for (partial) PV name:**
  - **$(pv)_setpoint**
  - **$(pv)_readback**

  Such a display can then be invoked with pv="PowerSupply1" or "PowerSupply2"



Macro Definition

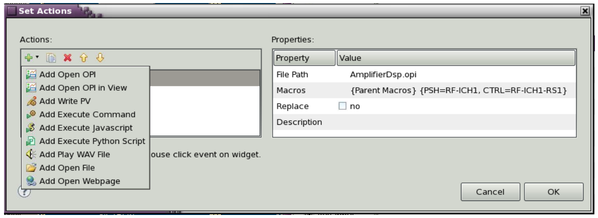
- Predefined Macros: Widget properties, see online help for name mapping
  - Property “X”: Macro $(x)
  - Property “Name”: Macro $(pv_name)
  - Automatic: Macro $(pv_value)
    - See default for the “Tool Tip” property
- User-defined:
  1. BOY Runtime Preference Setting (-pluginCustomization ….)
  2. User Preference settings (CSS, Pref…, ..App.., Display, BOY, OPI Runtime)
  3. Macro parameter of Action that opens the *.opi file
  4. Display *.opi file property “Macros”
  5. Grouping/Linking/Tabbed Container that wraps the widgets



Example:
Macro parameter of *Action* will
override *Preference* settings.

Exercise: Linking Displays with Macros

- Create display file “Macros.opi”
  - Label with Text “$(pv)”
  - Text Update with PV Name “$(pv)”
- Create display file “Linking.opi”
  - Action button with “Actions” to “Open OPI”
    - Use File Path for Macros.opi
    - Define Macros: pv= “sim://sine”
  - Add another action button (copy previous one)
    - Set macro to pv=“sim://ramp”

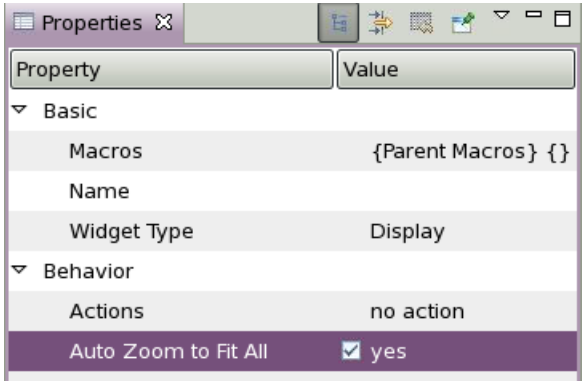


Property	Value
File Path	AmplifierDsp.opi
Macros	{Parent Macros} {PSH=RF-ICH1, CTRL=RF-ICH1-RS1}
Replace	□ no
Description	

- Execute. Check that you can open the linked display
- Extra: Check OPI Examples, “4. Actions”
  - Can have more than one “Open OPI”
  - Any widget can have “Action”. Try Label.
  - Try Linking Container to display Macros.opi within Linking.opi

Miscellaneous

- Display has an “Auto Zoom” property
  - Size will adjust to fit window



Properties

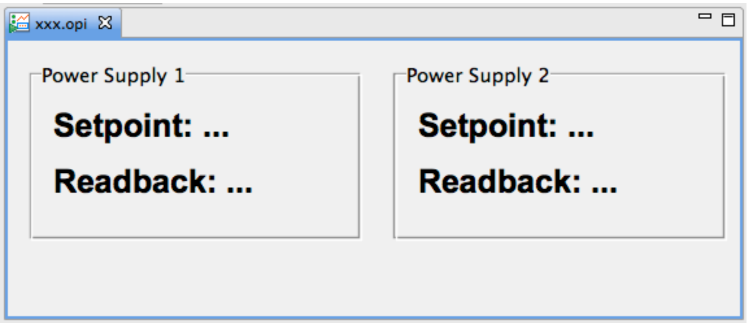
Property	Value
Basic	
Macros	{Parent Macros} {}
Name	
Widget Type	Display
Behavior	
Actions	no action
Auto Zoom to Fit All	☑ yes

Exercise: Grouping Container

In EDM, MEDM, … we needed lines and rectangles to visually group related displays.

In BOY there is the Grouping Container

- Create a display with Grouping Containers that look like this:



xxx.opi

Power Supply 1

Setpoint: ...

Readback: ...

Power Supply 2

Setpoint: ...

Readback: ...

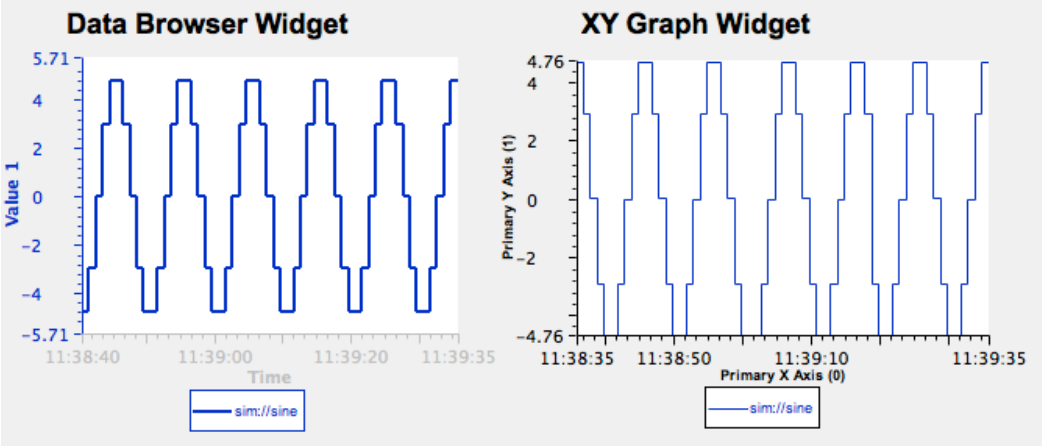
  - Border Style=Group Box Style
  - Name = Power Supply 1, Power Supply 2
  - Add Labels “Setpoint:…”, “Readback:…”

- Note how you can
  - Move the Grouping Container an all its content
  - Move Labels inside and out of the container

Exercise: “Striptool” type Plots of PV over Time

Try both options

- Data Browser Widget
  - New Data Browser Plot, add PV
  - Set desired axis and time range
  - Save as *.plt
  - Add Data Browser Widget to BOY
  - Set its File Name to the *.plt
- XYGraph Widget
  - Behavior, Trigger PV: “sim://noise”
    - This PV updates once a second and will trigger plot updates
  - Primary X Axis(0), Time Format: “HH:MM:ss”
    - To get a “time” axis
  - Trace 0, Trace Type: Step Horizontally
  - Trace 0, Update Mode: Trigger
  - Trace 0, Y PV: Name of PV to plot



✓ Can also display archived data

✓ PV can be ‘monitored’, showing brief spikes

□ Fewer display options

✓ Has many more display options

□ Cannot show archived data

□ PV scanned at update rate, can miss brief spikes

TO BE
CONTINUED...→