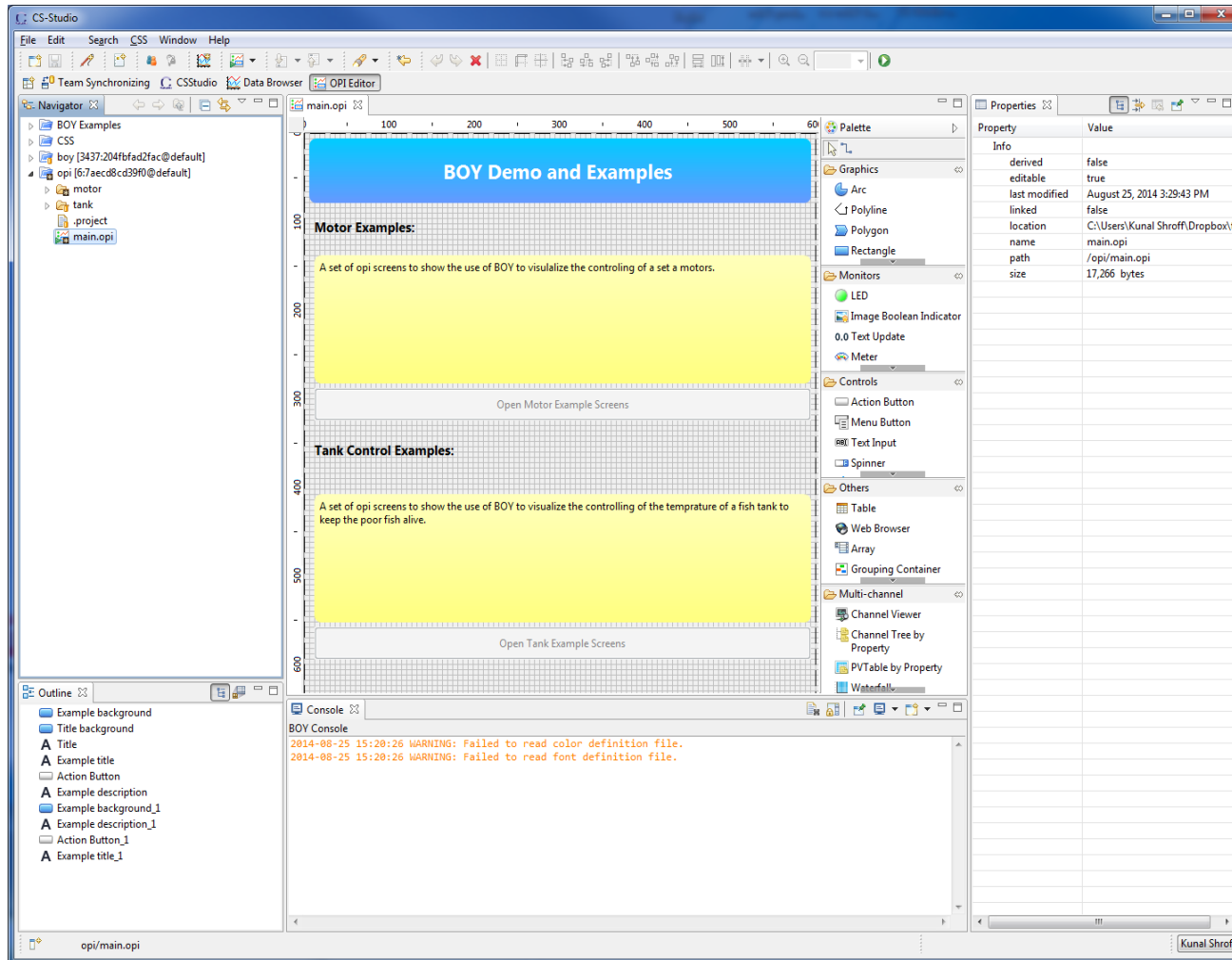
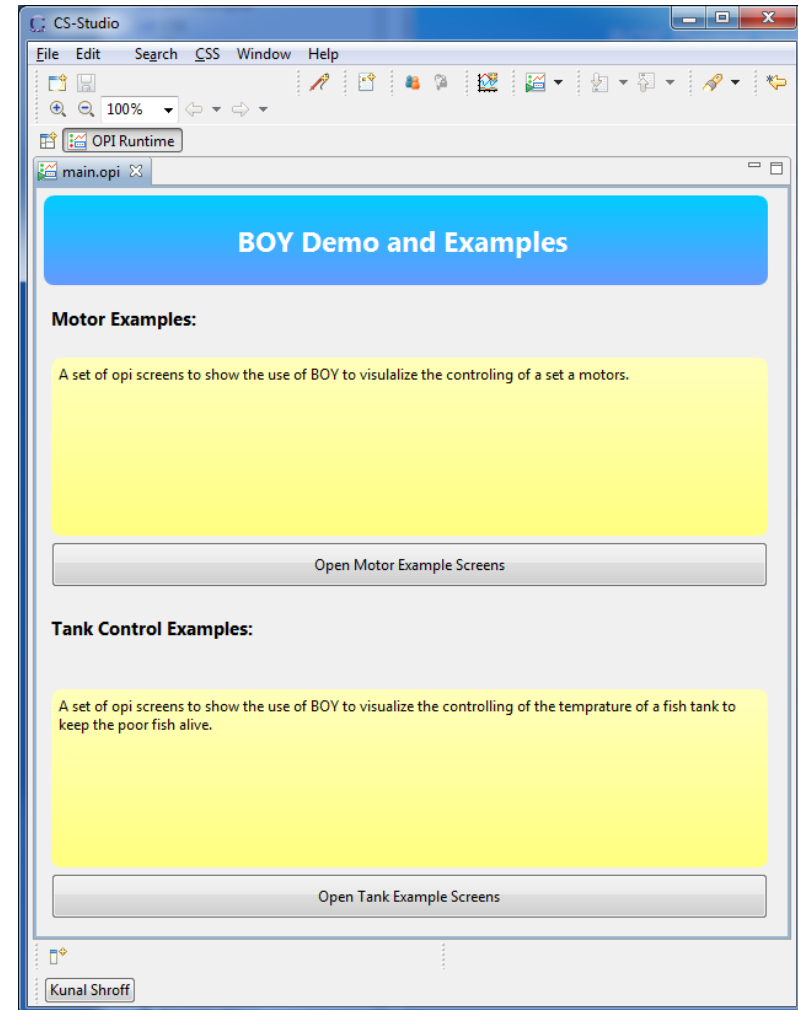


BOY

# OPI Editor





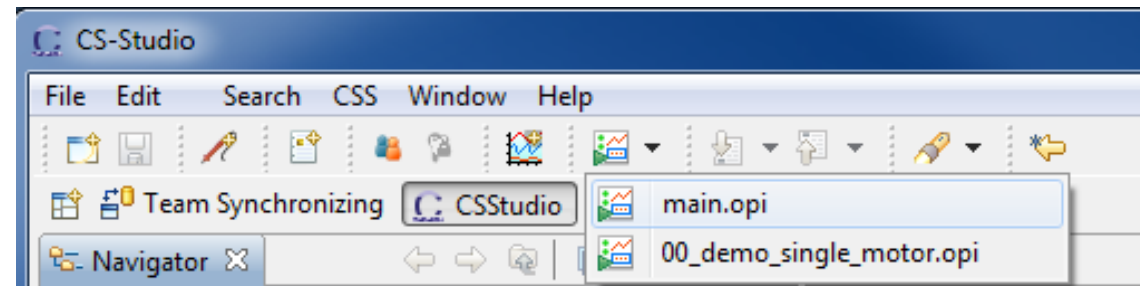
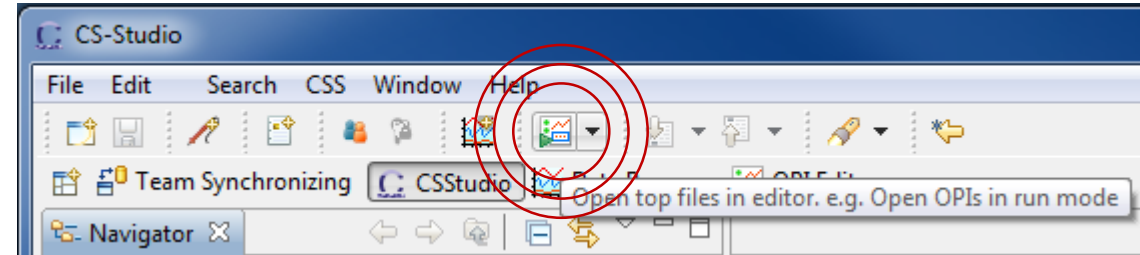
# OPI Runtime



BOY Runtime

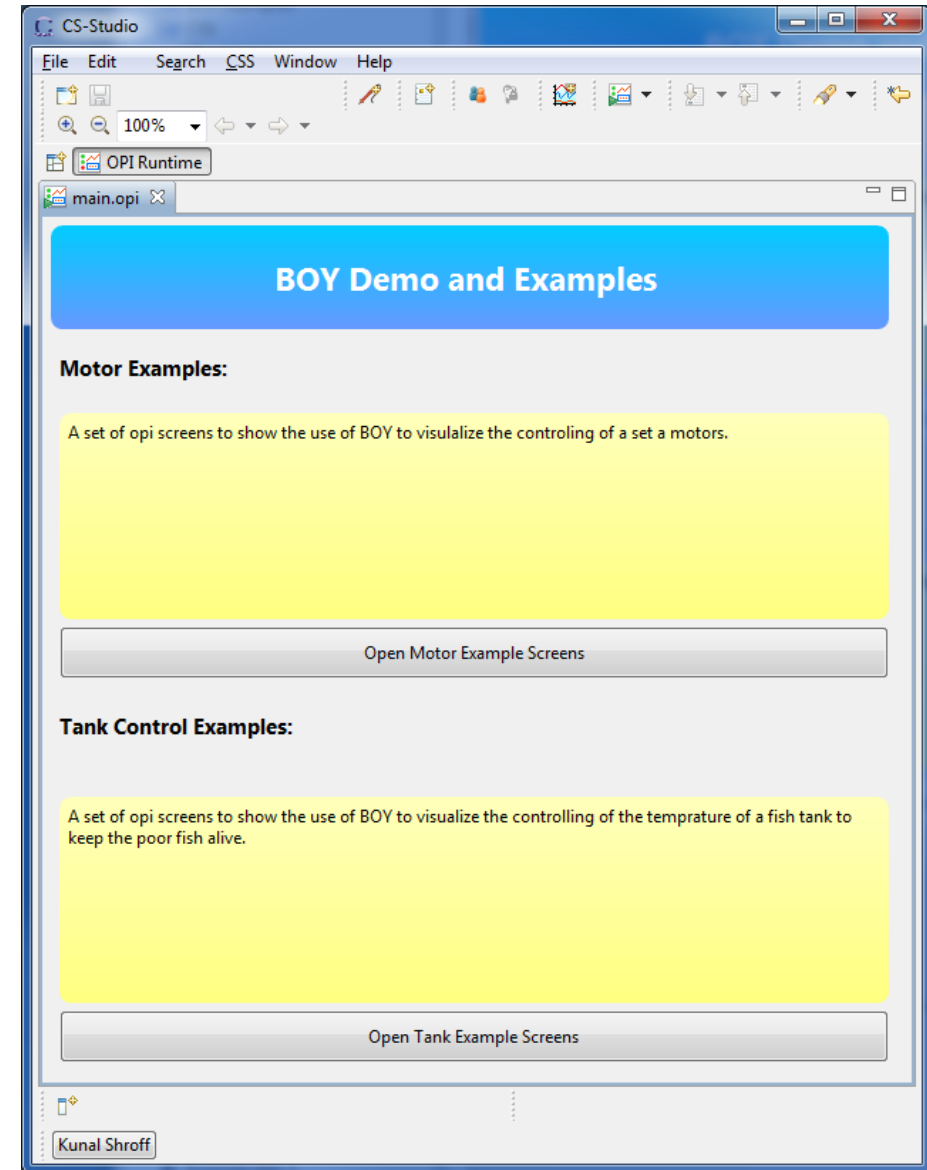
# Running your OPI screen

- Toolbar →  Open top files  
Opens the 1<sup>st</sup> top opi defined in the preferences  
Usually this can be configured to the main overview page
- Toolbar →  Open top files dropdown  
List various top level opi's  
Example: Beamline Status Page, Storage Ring Overview Page



# OPI Runtime: Perspective

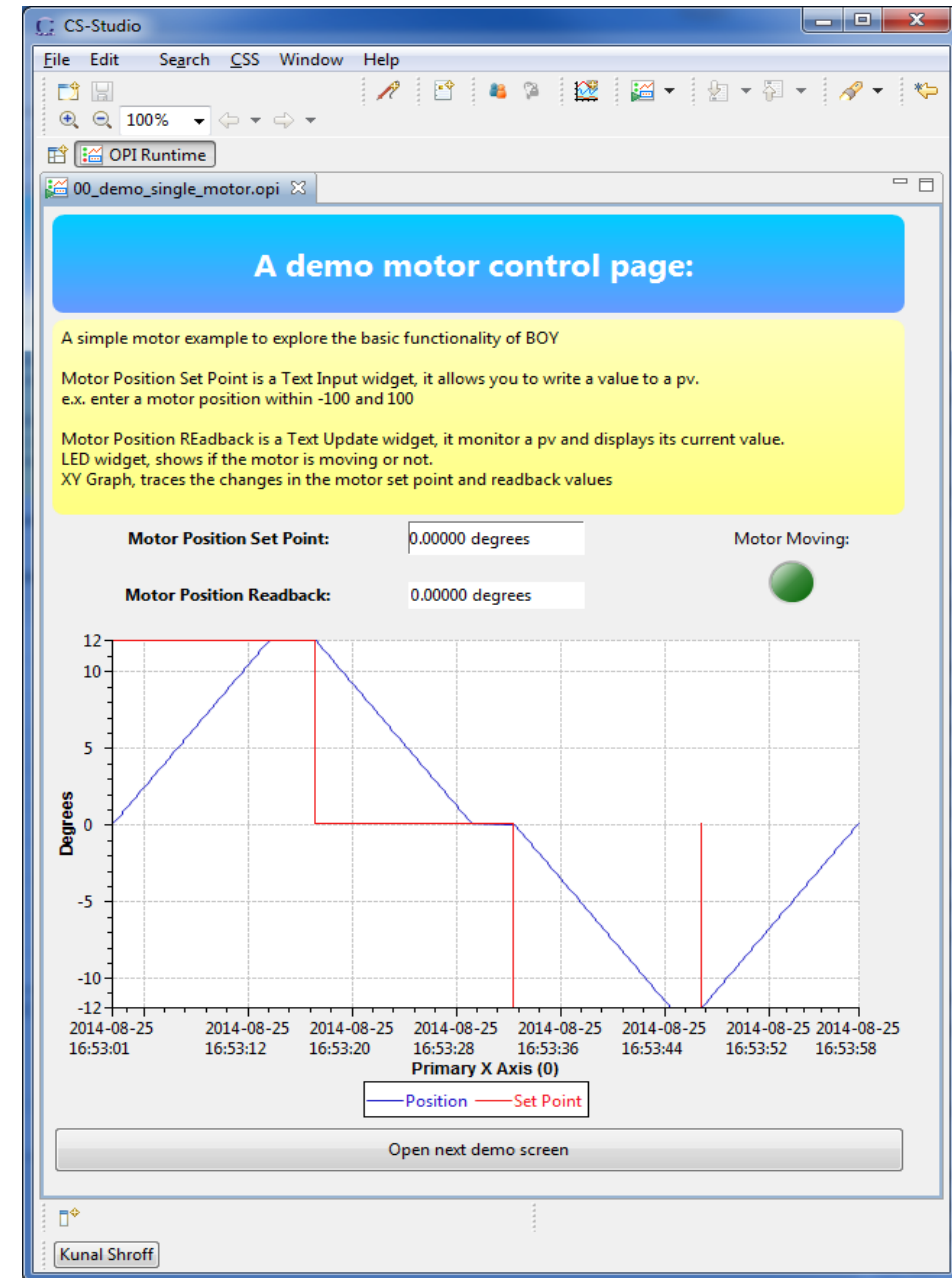
- An Environment to run user interfaces files which connect to the various systems (controls system, ..)
- Minimum clutter
  - No additional views
  - Toolbar consists of navigation, scaling and debugging buttons
- Compact mode
  - Even less clutter; no toolbar, perspective bar, status bar...
  - Enter/Exit compact mode using
    - Hot key (F8)
    - Context menu (right click → compact mode)



# First Demo OPI screen

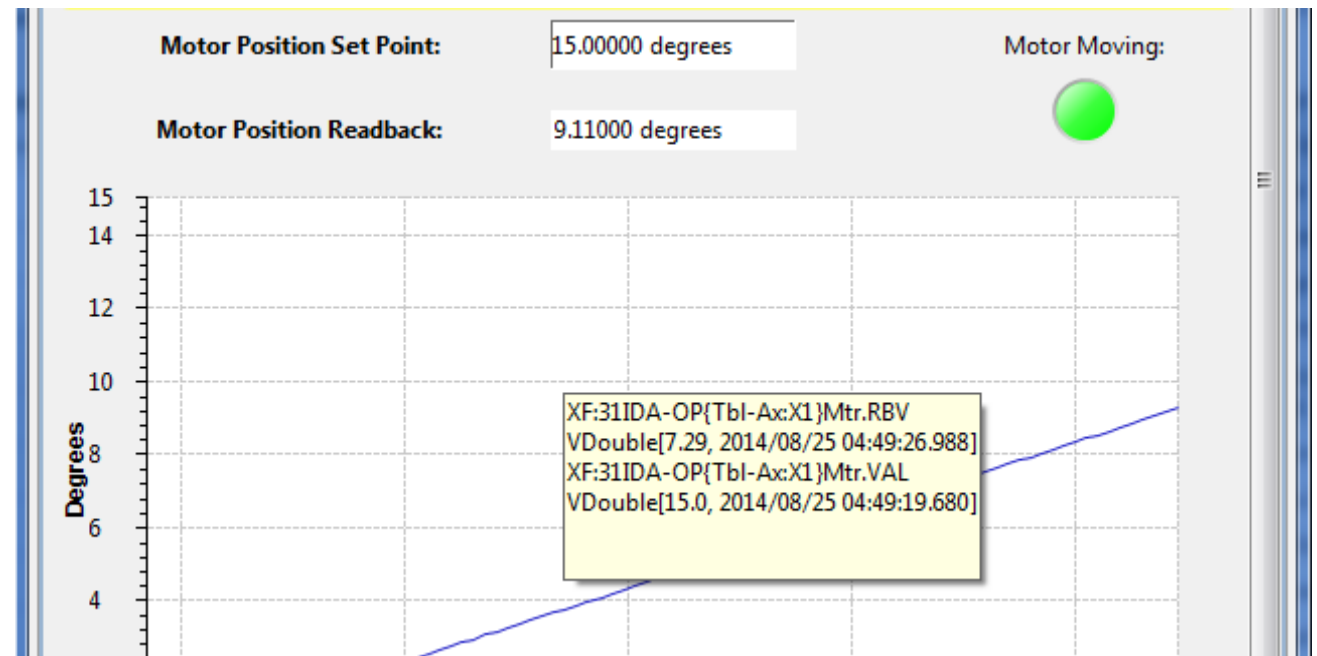
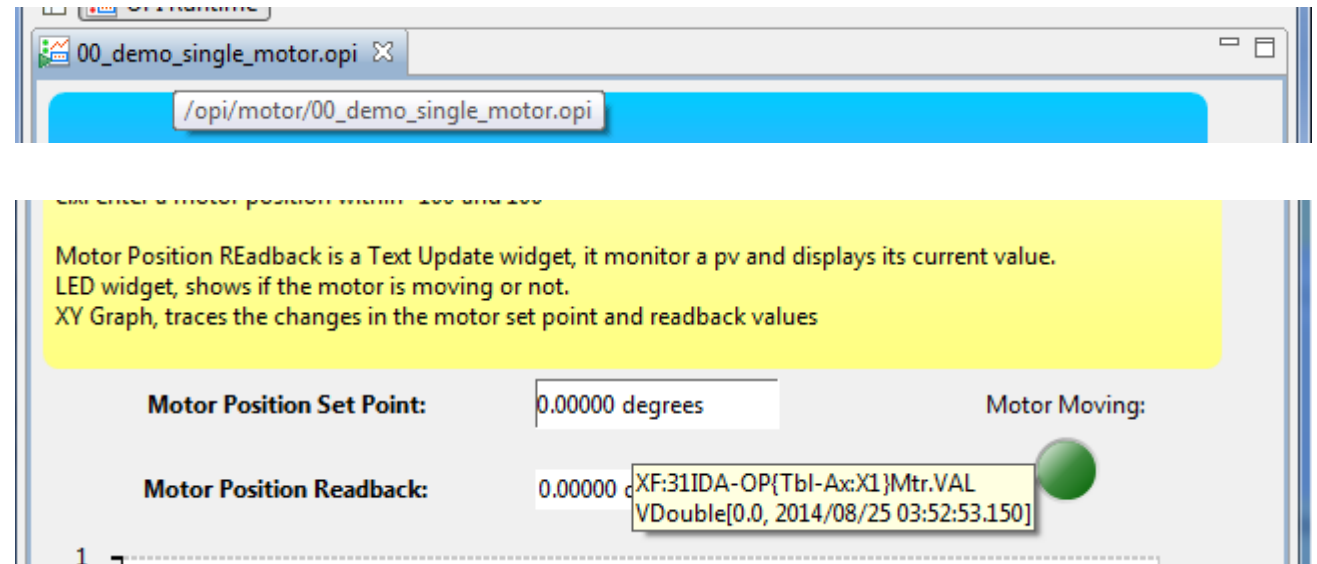
The screen allows you to enter a new position(in degrees) for the motor and then tracks the motor movement using various widgets.

- Input PVs
  - Motor Position Set Point
- Monitor PVs
  - Motor Position Readback
  - Motor Moving LED
  - XY graph



# OPI Runtime: Tooltip


- Status summary, Brief descriptions, Hints
  - “live” widget: PV value, PV connection status, Alarm information
  - Action / Menu button: Description of action to be performed
  - OPI screen tile: the complete file path

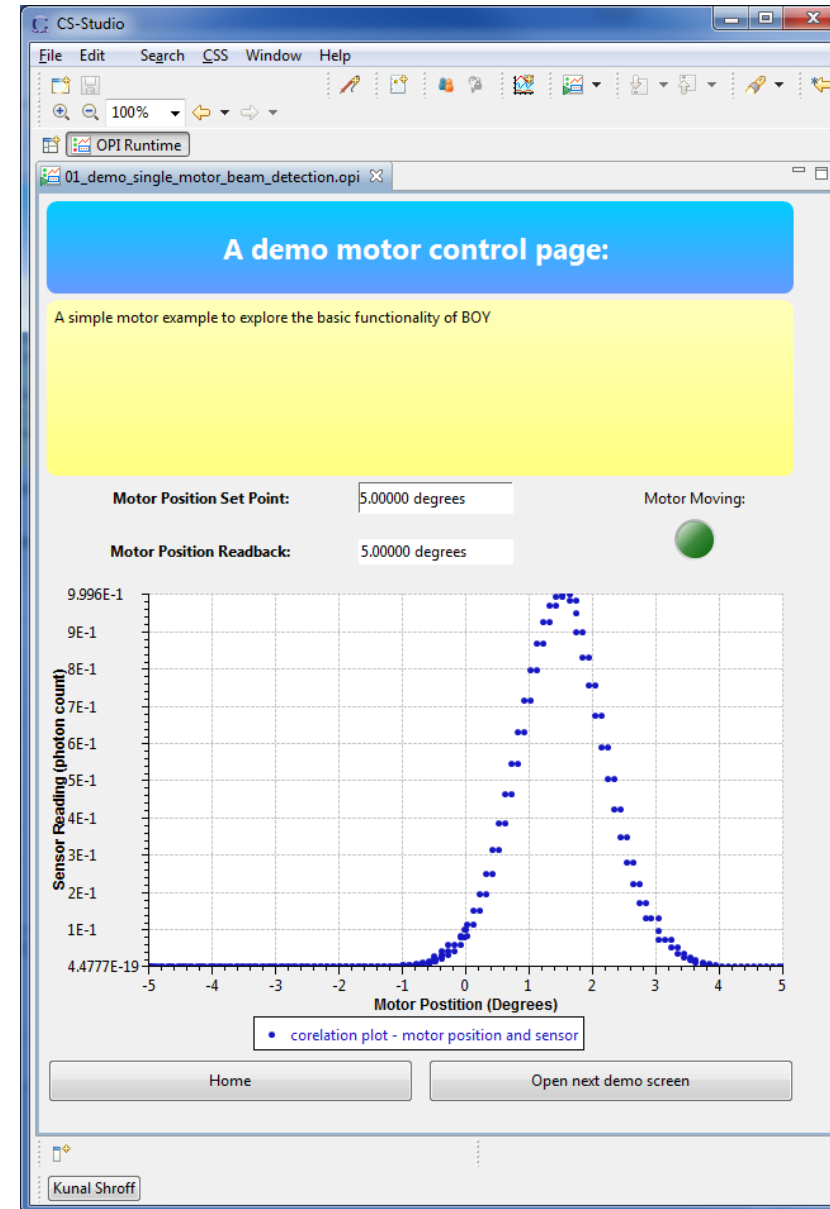


# Second Demo OPI screen

Moving the motor between -5 and 5 degrees while plotting the co-relation with sensor reading allows determining the location of the peak.

## Navigation

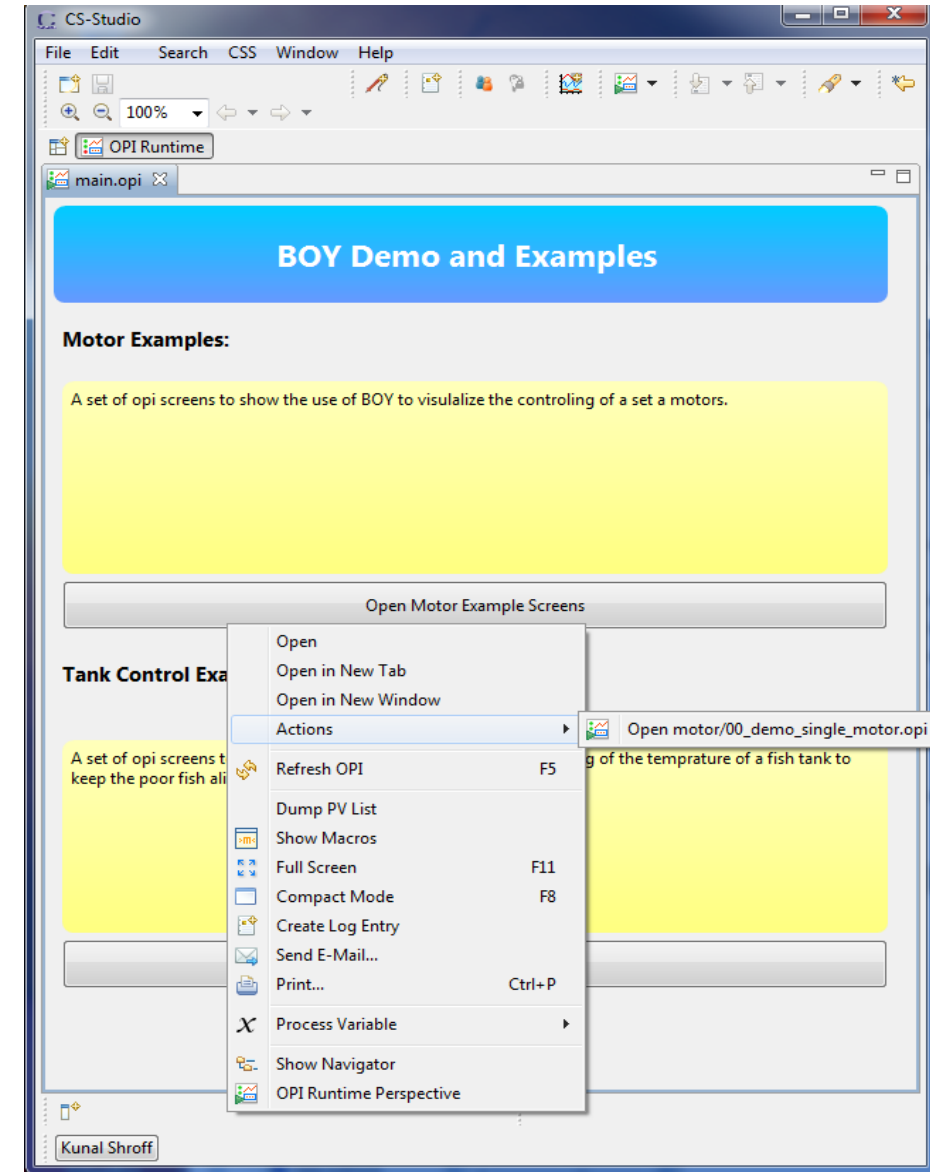
- Navigation buttons: Home, Next
- Navigation bar:  

- Hot Keys: Alt + left, Alt + right





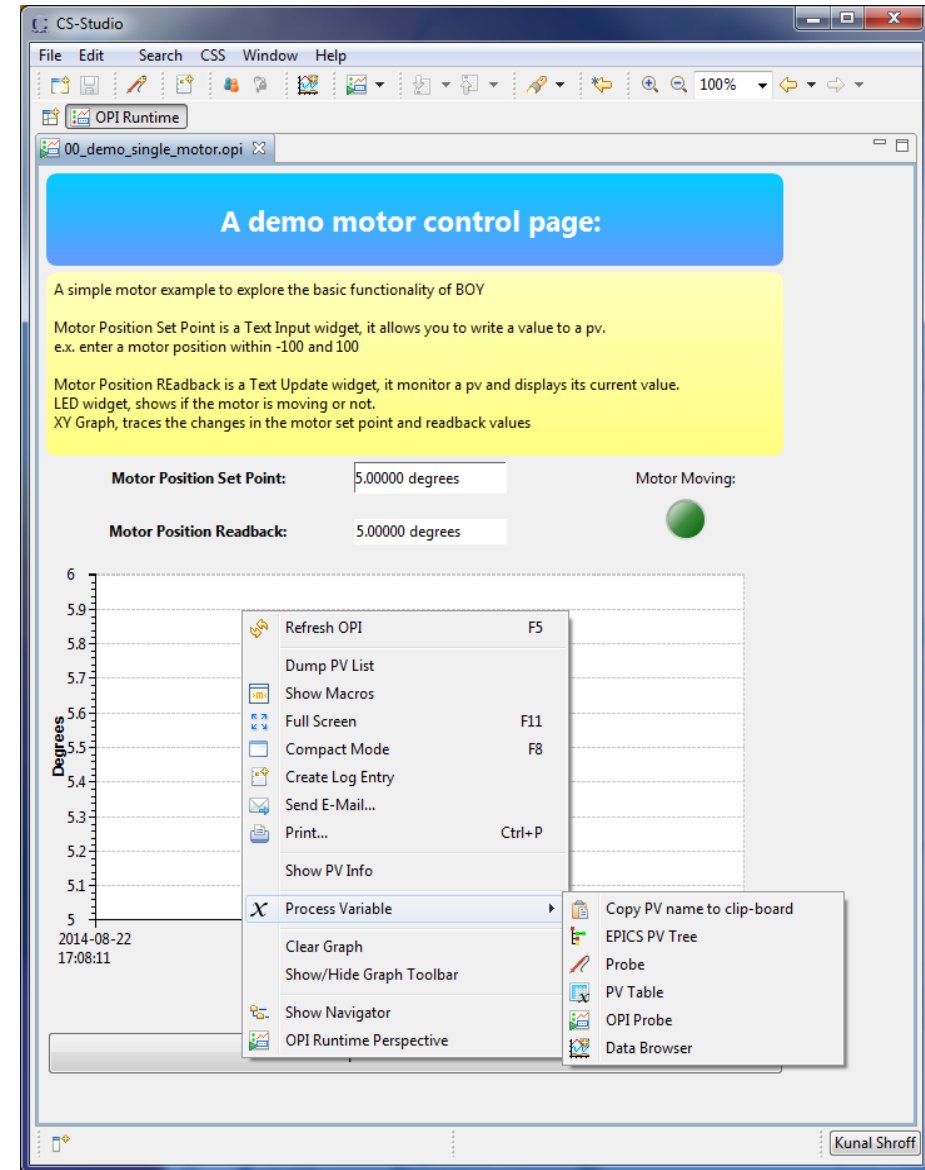
# OPI Runtime: Context Menu

- To open the context menu *Right Click* anywhere on the screen
- The context menu consists of a list of actions and information based on where you have clicked  
e.g.
  - Right click on the “Open Motor Example Screen” button, shows a context menu which lists the actions associated with this button under “Actions”.
  - Right click anywhere other than the button and the context menu does not contain the “Actions” or the Open \* commands



# OPI Runtime: Context Menu

- Context menu associated with “live” widgets
  - Show PV info
  - Process Variables menu to send these pv's to different applications
- Exercise:
  - Open the pvs associated with the plot in Probe
  - Open the pvs in Data Browser to see the historic values of these pvs



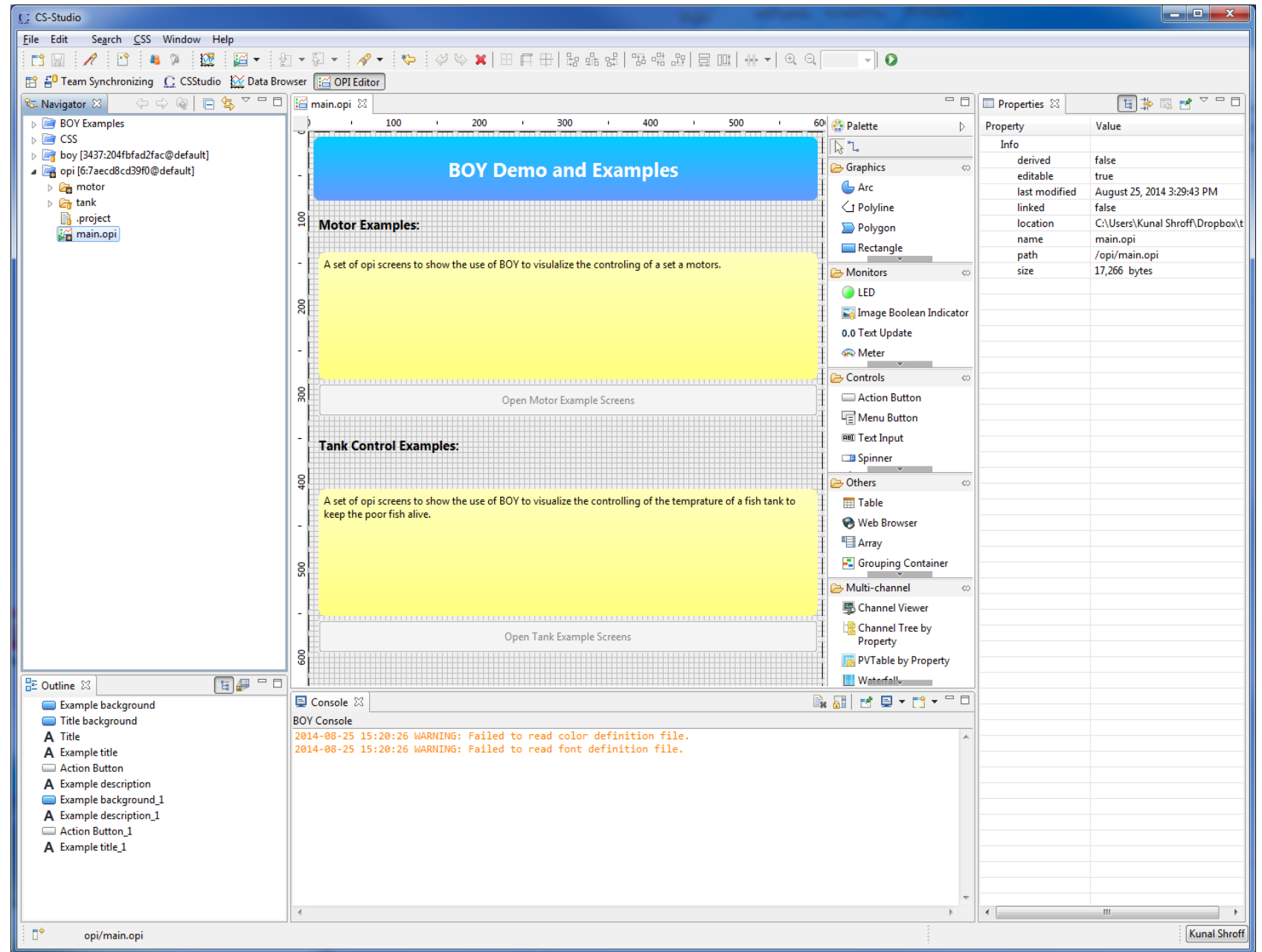
BOY Editor

# OPI Editor: Perspective

Window →

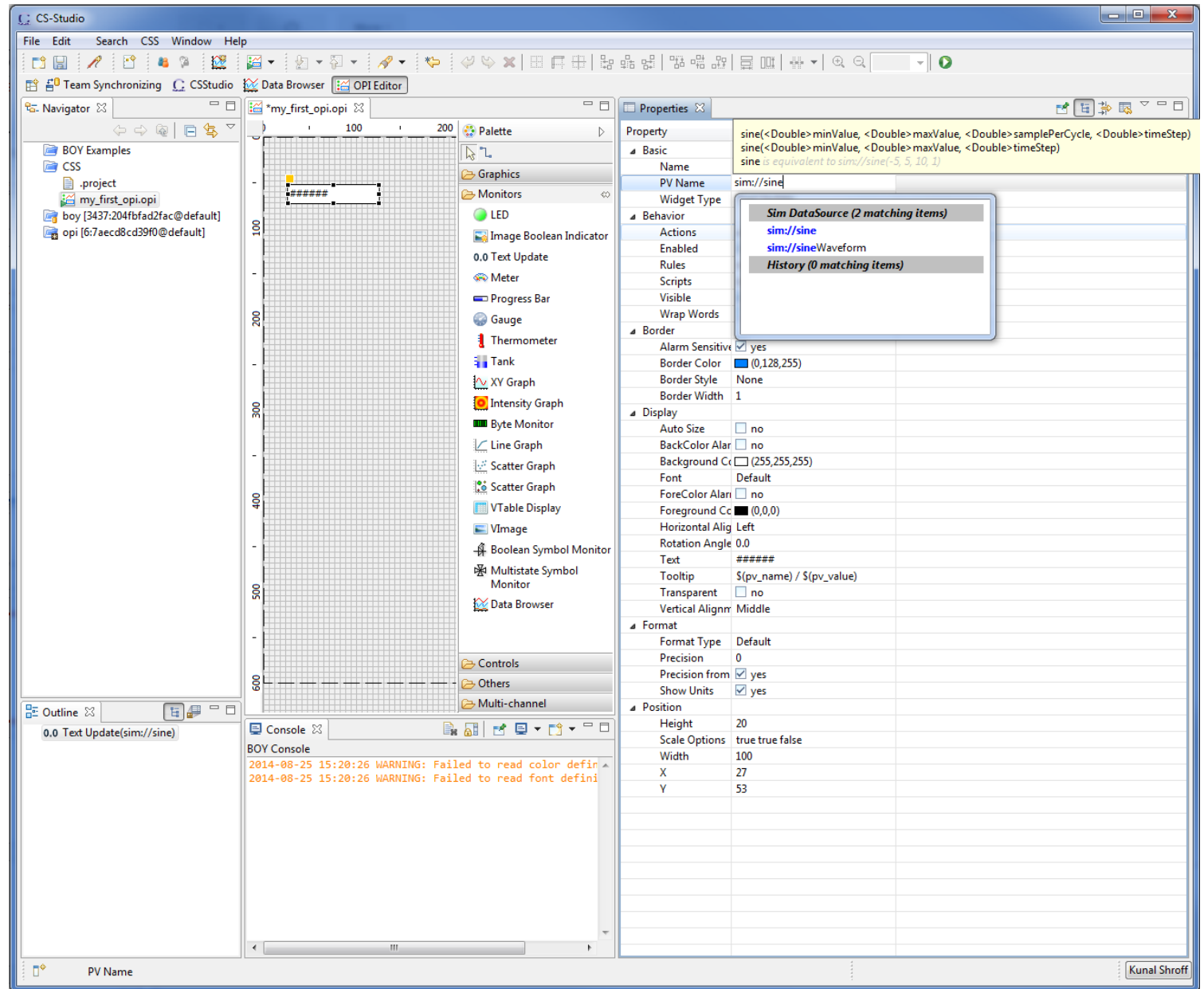
Display →

OPI Editor Perspective



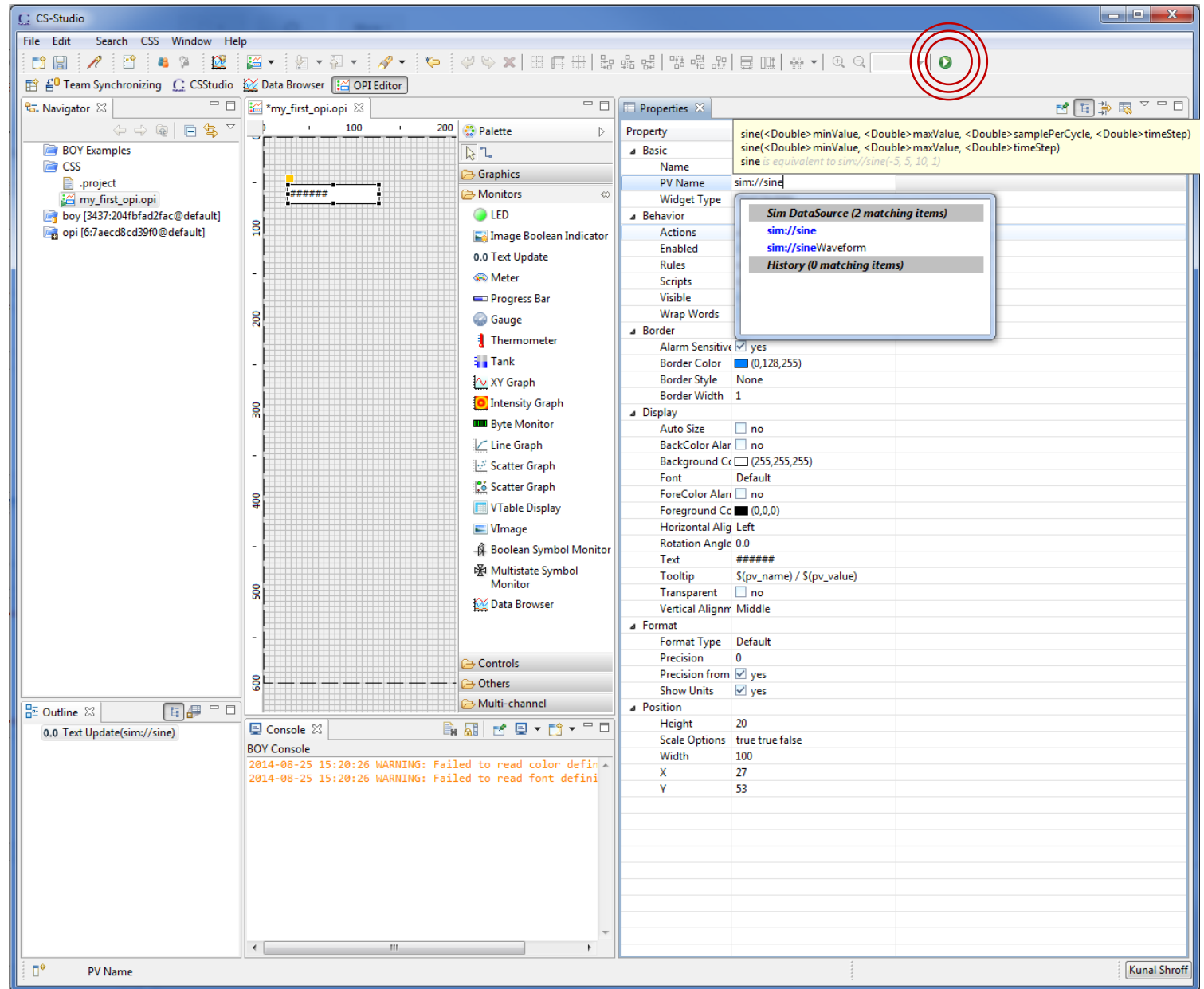
# Creating your first BOY screen

- File → New → BOY → OPI File
- Create a new OPI File Dialog  
Select the folder:  
CSS  
OPI File Name:  
<your\_name>\_demo
- Add a widget  
Drag and drop the text update widget from the palette  
Set property PV Name : sim://sine
- Save the file



# Running your first BOY screen

- Toolbar : Run opi button
- HotKeys: Ctrl + G
- Navigator: right click on file
  - Open With
  - OPI Runtime



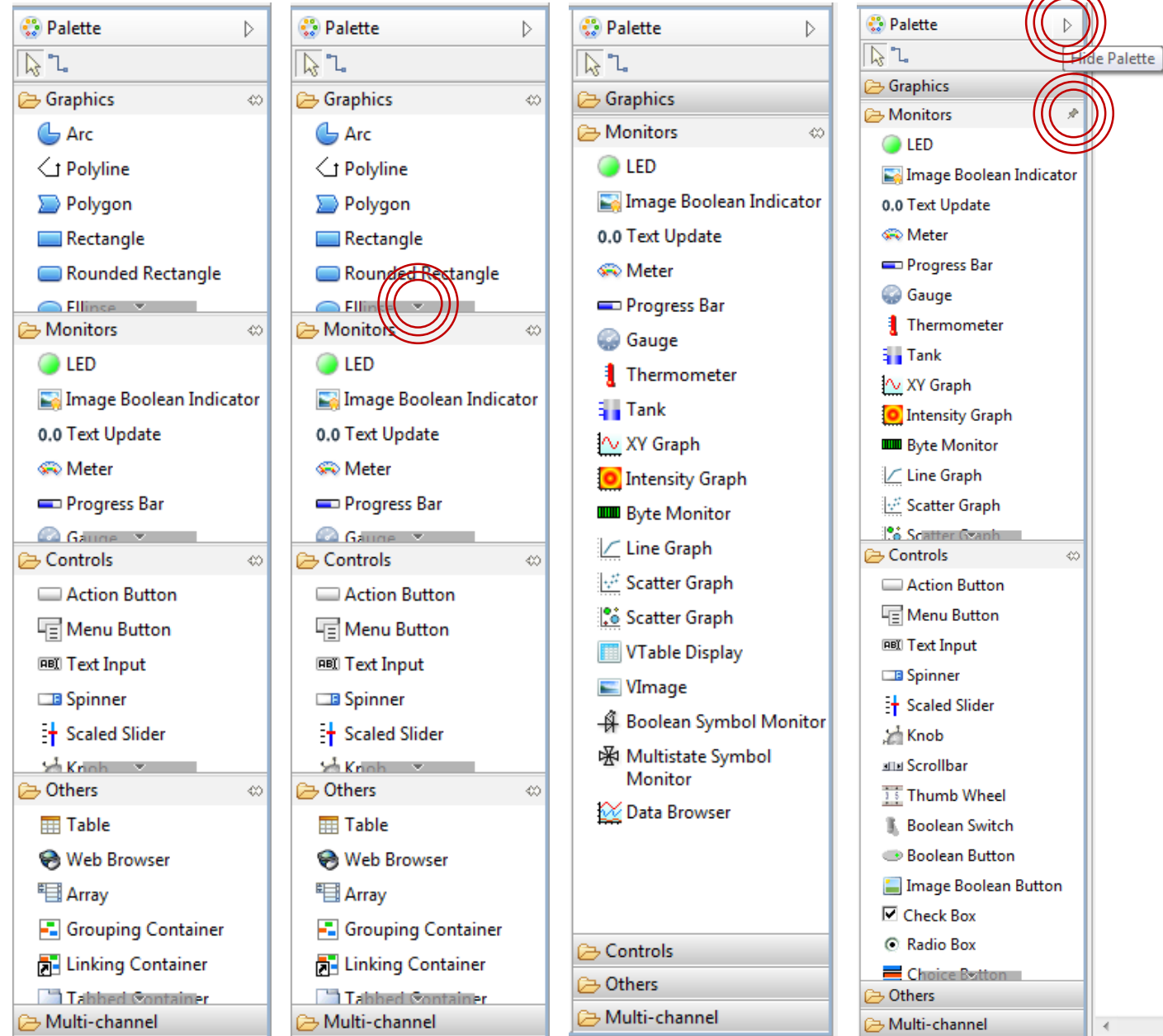
# Widget Palette

Displays the available widgets grouped by behavior

- Graphics
- Monitor
- Controls
- Others
- Multi-Channel

## Navigating the Palette

- Click section heading to minimize/maximize that section
- Hide Palette
- Pin the Palette (does not minimize when another section is selected)



# Widget Properties

Describe the visual and behavioral attributes of the widget

Help → Help Contents

CSS Applications → Display → Best OPI, Yet (BOY) → Widgets

| Property                  | Value   |
|---------------------------|---|
| Basic                     |   |
| Name                      | Text Update_1   |
| PV Name                   | sim://ramp(0,100000,11323,1)                                    |
| Widget Type               | Text Update   |
| Behavior                  |   |
| Actions                   | no action   |
| Enabled                   | <input checked="" type="checkbox"/> yes                         |
| Scripts                   | no script attached  |
| Visible                   | <input checked="" type="checkbox"/> yes                         |
| Border                    |   |
| Alarm Sensitive           | <input type="checkbox"/> no                                     |
| Border Color              | <span style="color: blue;">■</span> (0,128,255)                 |
| Border Style              | None  |
| Border Width              | 1   |
| Display                   |   |
| Auto Size                 | <input type="checkbox"/> no                                     |
| BackColor Alarm Sensitive | <input checked="" type="checkbox"/> yes                         |
| Background Color          | <span style="background-color: #cccccc;">■</span> (255,255,255) |
| Font                      | 1 Arial 9.0 0 WINDOWS 1 0 0 0 0 0 0 0                           |
| ForeColor Alarm Sensitive | <input type="checkbox"/> no                                     |
| Foreground Color          | <span style="color: black;">■</span> (0,0,0)                    |
| Horizontal Alignment      | Center  |
| Text                      | #####   |
| Tooltip                   | \$(pv_name) / \$(pv_value)                                      |
| Transparent               | <input type="checkbox"/> no                                     |
| Vertical Alignment        | Middle  |
| Format                    |   |
| Format Type               | Decimal   |
| Precision                 | 0   |
| Precision from PV         | <input checked="" type="checkbox"/> yes                         |
| Show Units                | <input checked="" type="checkbox"/> yes                         |
| Position                  |   |
| Height                    | 20  |
| Width                     | 100   |
| X                         | 555   |
| Y                         | 162   |



# Editing Widgets

- Select, Move or Resize
- Clone Widgets (copy paste)
- Alignment tools



Grid

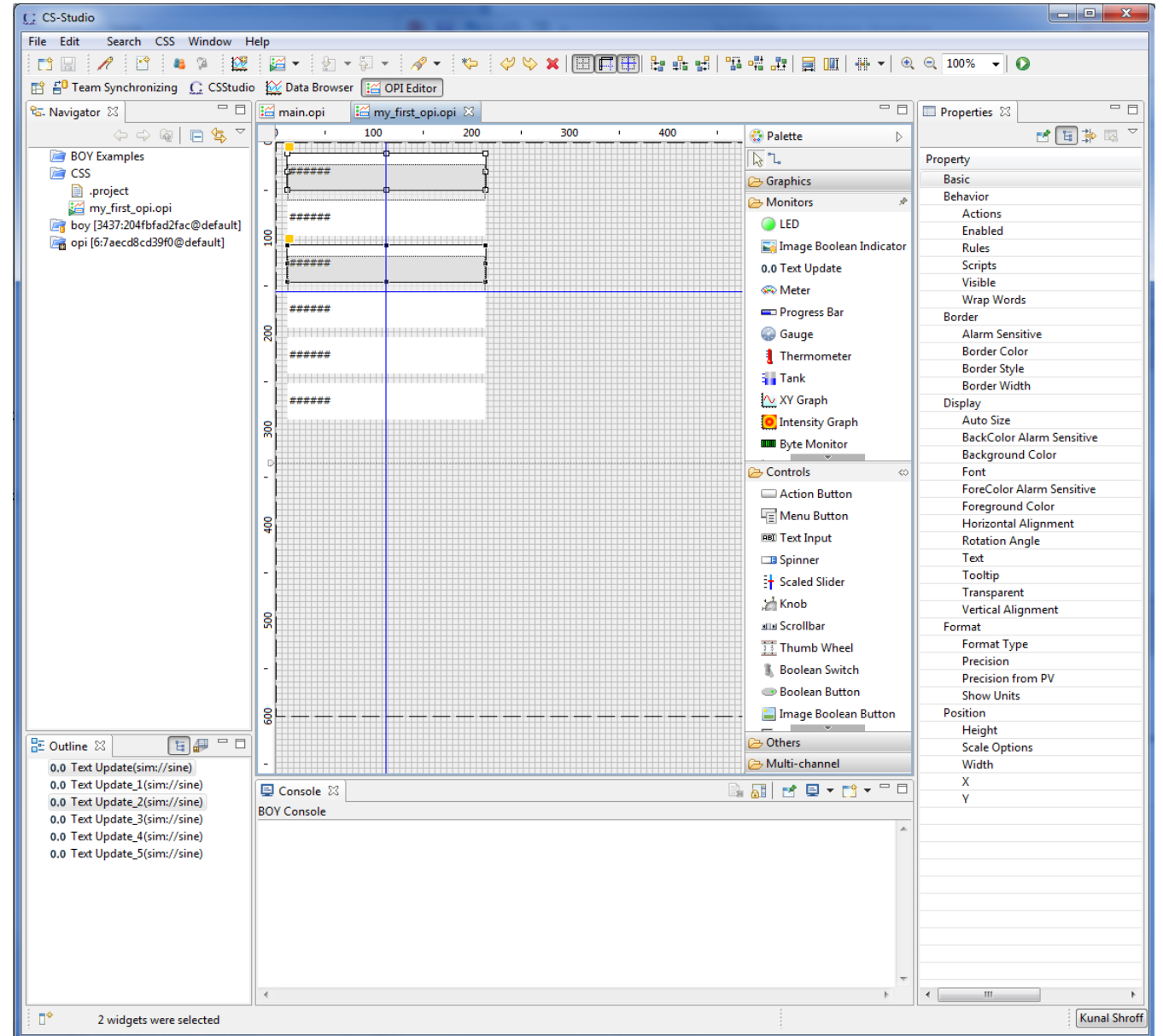
Snap to Grid

Snap to Geometry (sides and sizes of other widgets)



Arrange group of widgets

- Copy properties



# Note: location of demo

On github

<https://github.com/shroffk/cs-studio-training>

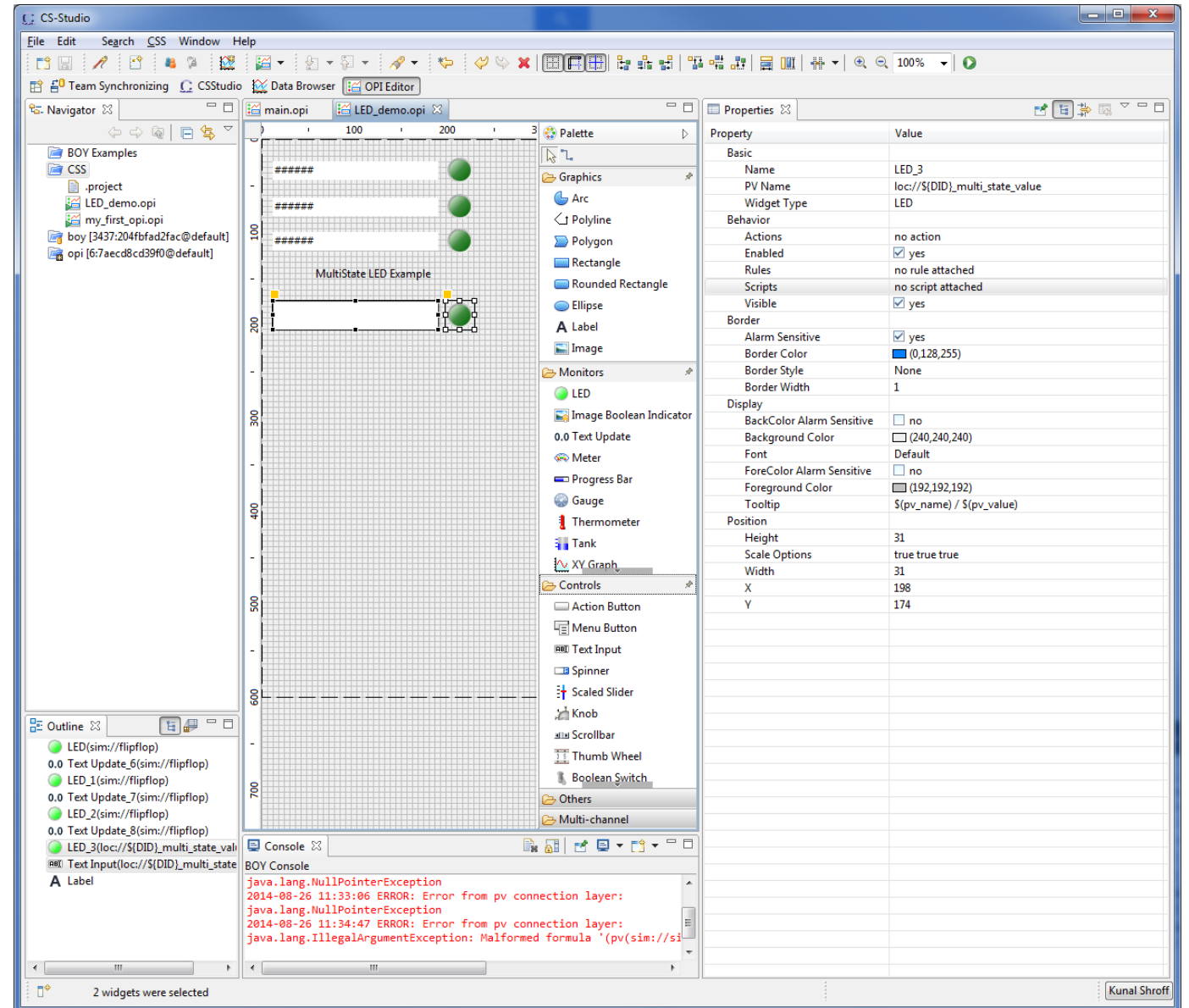
On train

/home/train/cs-studio-training/demo-opis

# LED

## Exercise:

- Boolean
  - Create a LED widget with PV Name *sim://flipflop*
  - Copy paste the LED to Create an Array of LEDs
  - Copy the PV Name property from the LED to a Text Update widget
- MultiState
  - Create a LED widget with  
PV Name = *loc://\${DID}\_multi\_state\_value*  
State Count = 4
  - Create a Text Input widget with  
PV Name = *loc://\${DID}\_multi\_state\_value*
  - Change the value in the Text Input



# Action Button

## Menu Button

An Action button widget which is used to execute an action with clicking the button

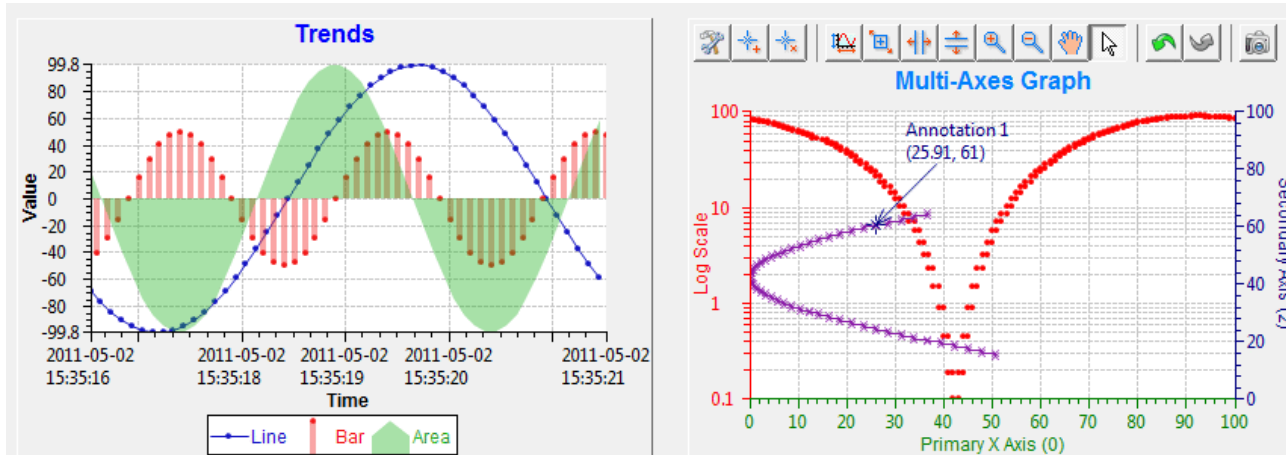
- Open OPI
- Write to PV
- Execute Command
- Play Wave file
- Open File

Menu buttons provide a list of Actions

Help → Help Contents → CSS Applications →  
Display → BOY → Widgets → Action/Menu  
Button

# XY Graph

A widget that is able to plot 1D or 2D data in an XY Graph

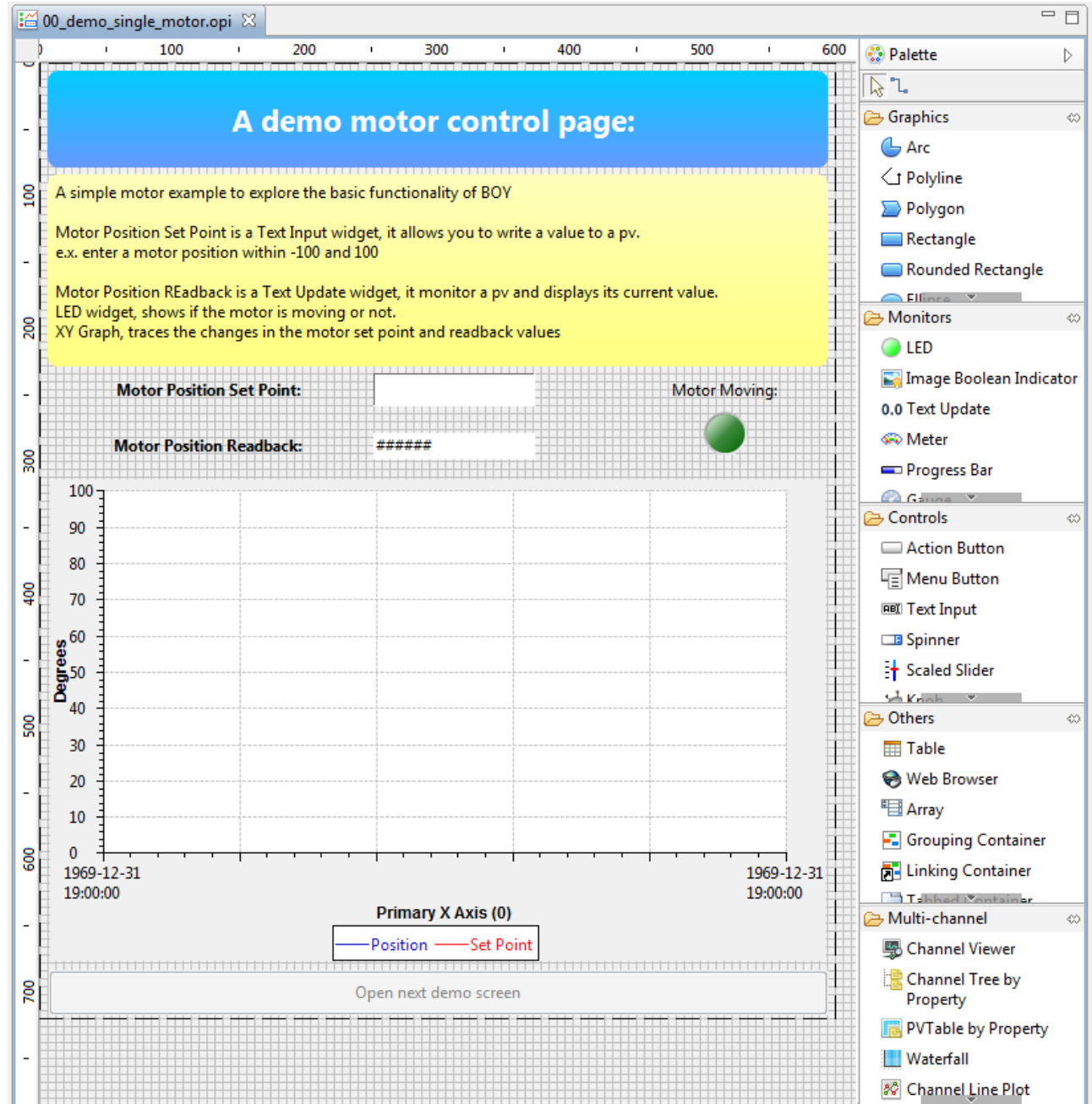


Help → Help Contents → CSS Applications →  
Display → BOY → Widgets → XY Graph

# Exercise

## Create a simple motor control screen

- A Text Input Widget (controls) for entering the set point  
PV Name = *XF:31IDA-OP{Tbl-Ax:X1}Mtr.VAL*
- A Text Update Widget (Monitor) for monitoring the readback  
PV Name = *XF:31IDA-OP{Tbl-Ax:X1}Mtr.RBV*
- A LED Widget (Monitor) to indicate the motor is moving  
PV Name = *XF:31IDA-OP{Tbl-Ax:X1}Mtr.MOVN*
- An XY graph (Monitor) to plot the changed in the motor setpoint and the motor position  
Trace 0  
Y PV = *XF:31IDA-OP{Tbl-Ax:X1}Mtr.RBV*  
Trace 1  
Y PV = *XF:31IDA-OP{Tbl-Ax:X1}Mtr.VAL*  
Primary X Axis (0)  
Time Format = HH:mm:ss



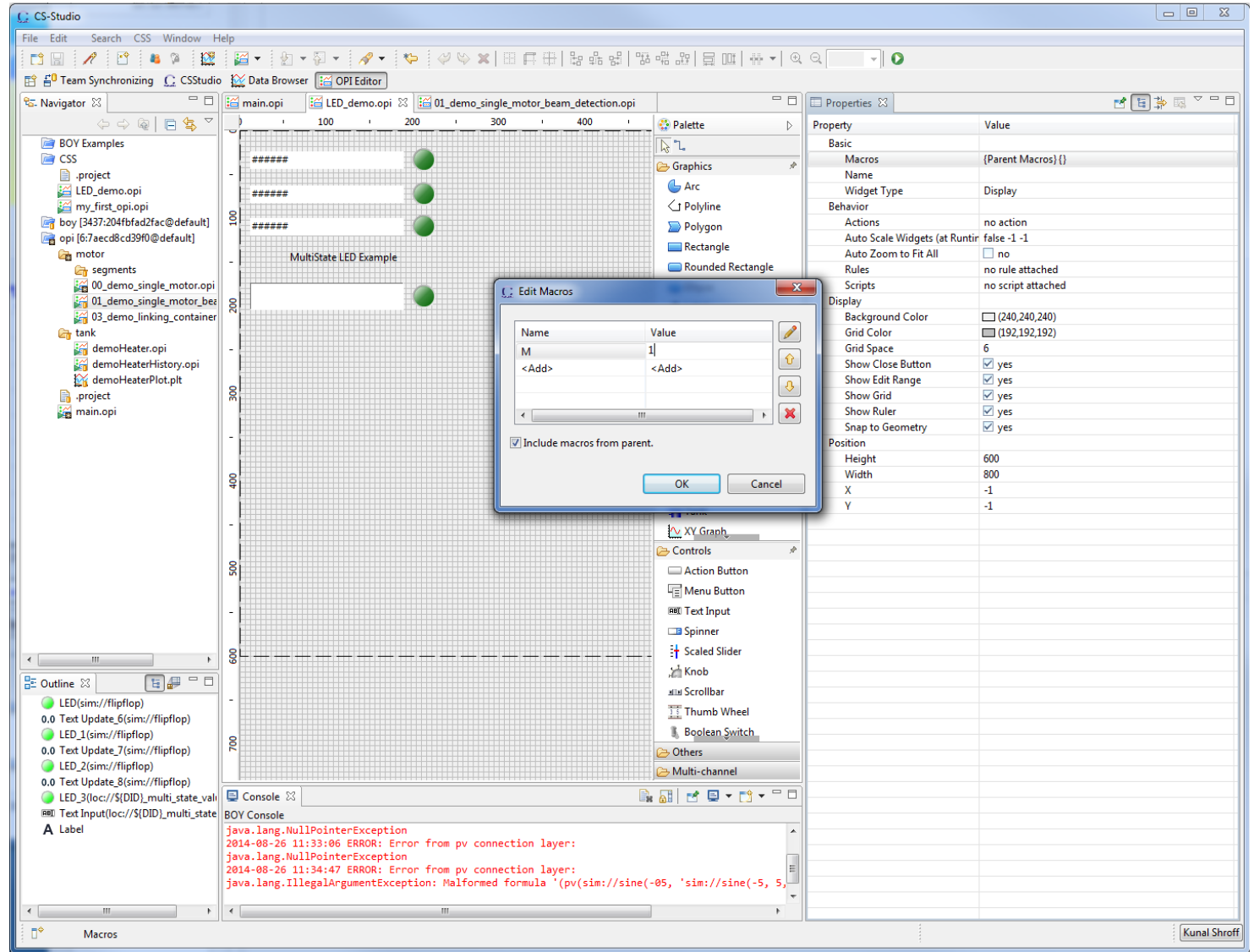
# Macros

Macros are some predefined strings that can be embedded in any string based properties such as "PV Name", "Text", "Tooltip", "Rules"

Macros are resolved at runtime when the screen is loaded

e.g.

PV Name = XF:31IDA-BI{Dev: \$(M)}E-I



# Containers

## Linking container

Create composite screens by embedding different opi screens or groups.

## Grouping container

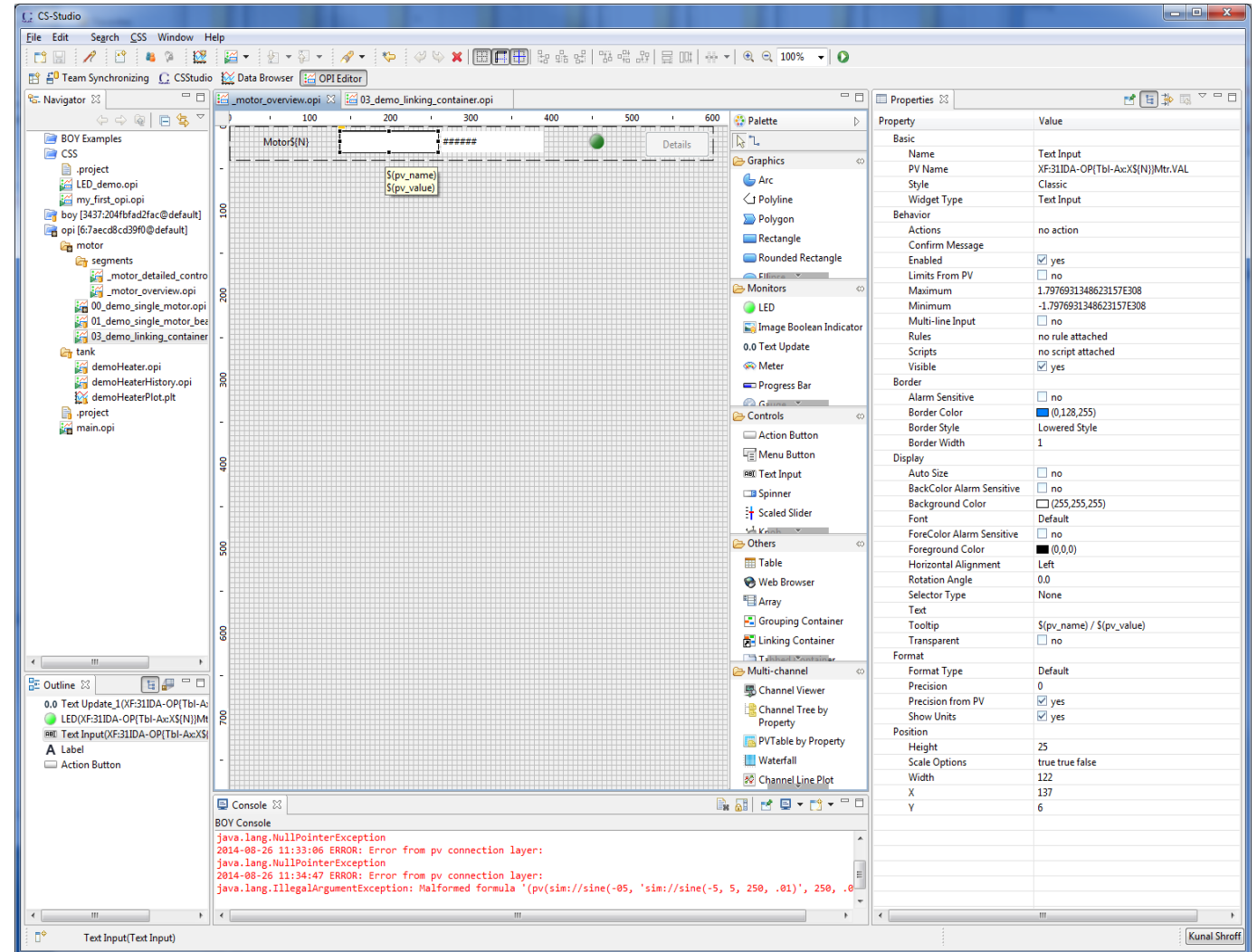
Groups a set of widgets together



# Exercise: Linking container & Macros

Create a simple screen (*\_motor\_overview.opi*) which represents a reusable segment with a

- Label  
Text = *Motor \$(N)*
- Text Input  
PV Name = *XF:31IDA-OP{Tbl-Ax:X\$(N)}Mtr.VAL*
- Text Update  
PV Name = *XF:31IDA-OP{Tbl-Ax:X\$(N)}Mtr.RBV*
- LED  
PV Name = *XF:31IDA-OP{Tbl-Ax:X\$(N)}Mtr.MOVN*



# Exercise: Linking container & Macros

Create another opi screen with 6 Linking container

- OPI File = Path to *\_motor\_overview.opi*
- Macros = for each linking container set N = 1, 2,...6

