Control System Studio: BOY

Kay Kasemir
ORNL/SNS
kasemirk@ornl.gov

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

June 2014
BOY – Best OPI, Yet

Operator Interface Editor

Runtime
Examples: SNS

- Top-level displays
Examples: SNS
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!

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 display limits set the knob’s default range
✓ Indication of ‘disconnected’ state via a pink border
✓ Widget will be greyed-out if read-only
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

• 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
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
View Online Help

Find
- CSS Core, Process Variables
- CSS Applications, Display, BOY, 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, Fanning, Auto Scale, Add/Remove Annotations, Undo/Redo, Take snapshot.
- Configure properties at Runtime, such as changing 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
- ...

Trends
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.
PV Names

- **ca://some_pv_name**
  - EPICS Channel Access PV
- **some_pv_name**
  - Typically same, since “ca://” is the default
- **sim://sine**
  - Simulated PV. Read online help for details
- **loc://x(4)**
  - Local PV. Read online help for details
- **pva://x**
  - EPICS V4 pvAccess
Formula Support

`='some_pv_name' * 2`
- Start with `='
- Enclose PV names in single quotes

`=3.14`
- Formula with constant value
  (replaces previous `const:\\3.14`)  

`="I like CS-Studio"`
- Enclose strings in double quotes

`loc://x(4)`
- Local PV. Read online help for details

Check online help, see CSS/Debugging/Formula, note auto-completion hints.

Beware:
Don’t use formulas for conversions that should happen on the IOC!
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

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

• Installed via Menu CSS, Display, Install OPI Examples

Note new project named BOY Examples

Explore the examples

Double-click on main opi file to open

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

Try with OPI Examples: Open in tab, … Window

OPIs in ‘Tabs’
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
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"

Linking.opi
pv=PS1
pv=PS2

Macros.opi
"$(pv)_setpoint"

Macros.opi
"PS1_setpoint"

Macros.opi
"PS2_setpoint"
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”

• 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
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:

- 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...