CSS Training - BOY

Xihui Chen, Kay Kasemir
chenx1@ornl.gov
Spring 2012 EPICS Meeting
What is BOY?

- BOY (Best OPI, Yet) – An Operator Interface (OPI) development and runtime environment

Similar to EDM, MEDM, SDS, DM2K
BOY Highlights

- A set of plugins built on Eclipse
- CSS integration
- Modern Graphical Editor
- Simple things are simple
- Dynamic via rules or scripts (support JavaScript and Python(Jython) script)
- Modern web browser style Runtime (tab, CTRL, SHIFT click)
- Comprehensive types of widgets and extensible
- Sitewide deployment friendly (macro, color, font, schema, http)
- Good ideas and contributions from all around the world…
Examples-Virtual Linac

Virtual Linac

Beam On !!!

Full Open

Close

Beam on Target!

Beam Position Control

H1/V1

H2/V2

H3/V3

H4/V4

H5/V5

Beam Off

Beam On

15.48 mA

13.98 mA

12.62 mA

9.13 mA

7.98 mA

7.16 mA

X 0.02 mm

X 0.03 mm

X 0.71 mm

X -0.01 mm

X -0.00 mm

Y -0.06 mm

Y 0.02 mm

Y 1.33 mm

Y 0.43 mm

Y -0.07 mm

0.00

6.58

SNS
Examples-SNS

• SNS top-level displays created by operators
BOY OPI Examples-ITER
Let’s Start!

• Open OPI Editor Perspective:
  – Menu CSS, Display, OPI Editor Perspective

![OPI Editor Perspective](image1)

• Install BOY Examples
  – Menu CSS, Display, Install OPI Examples

• Navigator Context menu on CSS: New, OPI File, call it “first.opi”
  – Or Menu File, New, BOY, OPI File

![Navigator Context Menu](image2)
Exercise: Edit First OPI

1. Drag widget Knob from palette to editor
2. Input “sim://noise” as the PV name in Properties view
3. Click the “Run” button (or Ctrl+G) to execute!

- sim://noise is a simulation PV inside CSS (See CSS help->CSS Core->Process Variables), so you don’t need to run an IOC to try this examples. You can replace it with a real PV name such as css:sine if your training soft IOC is running.
Exercise: Run First OPI

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

💡 Main Point: Simple Things are Simple
OPI Editor Perspective

• Your All-In-One workbench for OPI editing
Demo: Customize OPI Editor Perspective

- Every View in the workbench can be dragged around, detached, minimized, maximized or closed.
- Recover the default perspective by resetting it.
OPI Editor

• What You See Is What You Get (WYSIWYG)

• Comprehensive editing functions on toolbar and context menu
  – Copy/Paste/Delete
  – Drag & Drop
  – Undo/Redo
  – Alignment & Distributing
  – Snap to G (Grid/Geometry/Guide)
  – Zoom In/Out
  – Copy/Paste Properties
  – Changing Orders
  – …

As easy as editing PowerPoint, maybe better?
Properties View

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

- Common Properties:
  - Name
  - Background/Foreground color
  - Border
  - Font
  - Position

- Widgets that read/write PVs:
  - Basic: PV Name
  - Border: Alarm Sensitive
  - Behavior: Limits from PV
  - Display: Background/Foreground color alarm sensitive
Outline

• Tree View Outline
  – Click on widgets on the tree will select widgets in editor
  – Change widgets order by drag & drop

• Overview Outline
  – An overview of the who OPI
Demo : OPI Editing Skills

• Create a new OPI named “editor.opi”
• Add three rectangle widgets to editor
• Select widgets and move them around
  – Select all widgets using Ctrl+A
• Arrange widgets
Demo: OPI Editing Skills

• Moving widgets with key board
  – Press “.” dot key several times until the cursor switches to the direction you want
  – Use arrow keys on key board to move it one pixel a time

• Duplicate widgets by dragging widgets with holding $\text{Ctrl}$ key ($\text{alt}$ key for Mac).
  – Or use Ctrl+C and Ctrl+V to copy and paste
Demo : OPI Editing Skills

• **Group widgets**
  – Select multiple widgets
  – Select “Create Group” from context menu

• **Lock/Unlock Children**

• **Set Group border**

• **Ungroup widgets**
  – Select group widget
  – Select “Remove Group” from context menu
Demo: OPI Editing Skills

- Edit common properties for multiple widgets
  - Select multiple widgets
  - Properties sheet will automatically update to show the common properties
  - Edit properties as usual

- Copy/Paste Properties
Demo: OPI Editing Skills

• Add widgets by dragging text to editor
Demo: OPI Editing Skills

- Add connections
  - Select connection tool from Palette
  - Click on the two widgets to be connected
  - Press “Esc” to switch back to selection tool
  - Select the connection to configure its properties
Exercise: Context Help

• In OPI editor, press “F1” to open the Help View
  – Right click the view tab header and select “Detached” to detach it.

• Help content will automatically update along with the selected widget

• Help view can be detached
Exercise: Extend first opi

• Open first opi with context menu Open With->OPI Editor in Navigator.

• Add a text update widget

• Select all two widgets, set PV Name to loc://test

• Select Knob, set limits From PV to no

• Add a Thermometer widget, set PV Name to sim://noise

• Move, resize and align them as below

• Run
Exercise: send PV name to other CSS tools

• Right click thermometer, select *Process Variable*->*Probe*
Exercise: Open OPI in Runtime

- Close first opi
- Run first opi from context menu Open With->OPI Runtime in Navigator
Actions

• Every widget can have actions
• Click on widget to execute the action
• Actions will also appear on widget’s context menu during runtime
• Can also be executed from script
  – `widgetController.executeAction(index);`
Exercise: Open OPI Action

- Create another OPI named *action.opi*
- Add an *Action Button* widget
- Click on *Actions* property
- Add an *Open OPI* action, set *File Path* to *first.opi*
- Run
Exercise: OPI navigation

- Similar to hyperlinks in a Web Browser
  - Open in new tab by Ctrl+click
  - Open in new Window by Shift+click
  - Or use context menu

- Zoom in/out
- Go Back/Forward
Demo: Open OPI in Detached View

- Set the size of first opi to 400*300
- Add an action button
- Add an Open OPI in View action as configured in right picture
Rules

- Easily make widget properties dynamic

• Condition depended property value
• Directly output PV value to a property
• Allow multiple rules on a widget
Demo: Rules

- Add a Label to first opi
  - **Text:** “Heater is on”
  - **Foreground color:** Red
  - **Transparent:** Yes
  - **Font:** size=16

- **Click on Rules property and add a rule as below**
Scripts
- Intelligenzize your OPI

- For more complex logic that Rules cannot achieved
- Support both JavaScript and Python(Jython) Script
- Triggered by PV value change or execute from actions
- Can do “anything”
  - access widgets and PVs
  - attach data to widget
  - call Java code
  - Call your custom Java library (custom script util)
- Option: Embedding script in OPI or separate script file
Demo: Scripts

```java
from org.csstudio.opibuilder.scriptUtil import PUtil
from org.eclipse.jface.dialogs import MessageDialog

# Name of the flag to show if dialog has been popped up.
flagName = "popped"

if widget.getExternalObject(flagName) == None:
    widget.setExternalObject(flagName, 0)

b = widget.getExternalObject(flagName);

if PUtil.getDouble(pvs[0]) > PUtil.getDouble(pvs[1]):
    # If dialog has not been popped up, pop up the dialog
    if b == 0:
        # Set popped flag to true
        widget.setExternalObject(flagName, 1)
        MessageDialog.openWarning(
            None, "Warning", "Heater is on!"
        )
    else:
        # Reset popped flag to false
        if b != 0:
            widget.setExternalObject(flagName, 0)
```

Access Widget
Access PV
Call Java code
PyDev for python script editing

• Syntax highlight, code auto-completion, error check
• See BOY Help->Script->Python Script for installation instruction
With script, BOY can be used for more purposes

- Beamline Experiments
With script, BOY can be use for more purposes

- PV based automated test
- General GUI applications

/BOY Examples/Miscellaneous/CalculatorExample/Calculator opi
Widgets

• Support various type of data
  – Double, Integer, String, Enum, Boolean, Waveform…

• Allow plugging in customized widgets to BOY
Exercise: XY Graph Widget

- Exercise 1: Plot scalar value trends (Update on Value Change)
  - Add Monitors/XY Graph widget
    - Trace Count: 2
    - Primary X Axis (0)
      - Time Format: Auto
    - Trace 0
      - Y PV: sim://noise
    - Trace 1
      - Y PV: sim://sine
Exercise: XY Graph Widget

- Exercise 2: Plot scalar value trends (Update from trigger PV)
  - Add XY Graph widget
    - Trace Count: 2
    - Trigger PV: `sim://ramp(0,100,1,0.5)`
      - This is a simulation PV updates every 0.5 second
    - Primary X Axis (0)
      - Time Format: Auto
    - Trace 0
      - Update Mode: Trigger
      - Y PV: css:setpoint
    - Trace 1
      - Update Mode: Trigger
      - Y PV: css:tank
Exercise: XY Graph Widget

- Exercise 3: Plot Waveform
  - Add Monitors/XY Graph widget
    - Trace Count: 2
    - Trace 0
      - Buffer Size: 200
      - Concatenate Data: no
      - Y PV: css:sineWave
    - Trace 1
      - Buffer Size: 200
      - Concatenate Data: no
      - Y PV: css:cosWave

Set following properties and try toolbar button to zoom in/out

- Primary X Axis (0)
  - Auto Scale: no
  - Maximum: 200
  - Minimum: 0

- Primary Y Axis (1)
  - Auto Scale: no
  - Maximum: 100
  - Minimum: -100
Exercise: XY Graph Widget

- Exercise 4: Plot XY Graph with two waveforms
  - Add Monitors/XY Graph widget
    - Trace 0
      - Buffer Size: 200
      - Concatenate Data: no
      - Name: cos-sine Ellipse
      - X PV: css:cosWave
      - Y PV: css:sineWave
Widget – Intensity Graph

- Display image-like data
- Can be used for video display
- Generate Profile data
- Zoom In/Out
Exercise: Intensity Graph Widget

• Add Monitors/Intensity Graph widget
  • PV Name: css:imageWaveform
  • Data Height: 200
  • Data Width: 200
  • Maximum: 100
  • Minimum: 100
Exercise: Explore BOY Examples

- If you haven’t done so, invoke Menu Edit->Display->Install OPI Examples

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

• Simple things are simple
  1. Add widget
  2. Enter PV Name
  3. Run
• Use Rules or Scripts only necessary
• For more features, Read The Friendly Manual
• Send your ideas to chenx1@ornl.gov
• You can contribute!
  — See you at CSS development session in afternoon
Thank you!

- **BOY Home Page**
  - Or Google “CSS BOY”
Continued...

• If you want to explore more after the training...
Macro

- Embedded in string based properties, such as *PV Name, Tooltip*
- Replace with its value at runtime
- **Format**: $(macro_name)$ or ${macro_name}$
- All properties can be accessed via macro: $(name)$, $(pv_name)$, $(height)$

In Editor

In Runtime
Where to define Macros

• BOY Runtime Preferences

• Open OPI Actions
  – It includes open OPI from actions, open OPI from command line and open OPI from top OPIs button.

• Macros property of Display

• Macros property of container widgets
  – such as Grouping Container, Linking Container and Tabbed Container.
Demo: Macro

- Create macro.opi
- Add a label widget, set text to **Use is $(user)**
- Add a Knob widget, set **PV Name** to $(user):setpoint
- Save (Ctrl+S)
- Add an action button in action.opi
- Add an Open OPI action
  - File Path: macro.opi
  - Macros: user=css
- Run
Color & Font Macro
- make consistent look to your OPIs

• Predefine colors or fonts in text files
• Using Color and Font macro will help you
  ✓ Giving a consistent look to the OPIs on your site
  ✓ Reuse some particular color or fonts
  ✓ Input once and change everywhere
  ✓ Apply different theme
Font Macro
- make consistent look to your OPIs

Support platform specified fonts

1//Default Title Font. It will used in case the OS specified
2Title = Arial-bold-18
3
4//Title Font for Linux GTK
5Title(linux_gtk) = _Sans-bold-18
6
7//Title Font for MacOS
8Title(macosx) = Lucida Grande-bold-18
9
10Header1 = Arial-bold-16
11Header1(linux_gtk) = Sans-bold-16
12
13Header2 = Arial-bold-14
14Header2(linux_gtk) = Sans-bold-14
15
16Text = Arial-regular-10
17Text(linux_gtk) = Sans-regular-10
18Text(macosx) = Lucida Grande-regular-10
19
20LinkText = Arial-italic-10
21LinkText(linux_gtk) = Monospace-italic-10
22
23LinkText2 = Arial-bold italic-10
24LinkText2(linux_gtk) = Monospace-bold italic-10
25

Title
Header1
Header2
Text
LinkText
LinkText2
Demo: Add a new Color Macro

- Edit `/BOY Examples/color.def`
- Add `MyColor = 123,231,0`
- Set color file in BOY Preference

- If necessary, reload the color file in color property dialog
Schema

• Don’t like the default widget properties value?

• Use schema!
  – Widgets’ properties value in schema opi will be used as the default properties value
Exercise: Schema

- Create a new OPI “MySchema.opi”
- Add a Rectangle widget
  - Background Color: green
- Save
- Set preference Schema OPI to MySchema.opi

- Go to first.opi
- Add a new rectangle
  - Its default background color will be green
Preferences: Sitewide settings

- Top OPIs: Appear in Toolbar

- Path names for color & font files, “Top” OPIs, Schema can be on the web
  - Instead of /BOY Examples/font.def use
    http://some.server.org/path/font.def
Exercise: Enum widgets

• Add Controls/Choice button
  – PV Name: css:sensor
  – Items From PV: Yes

• Add Controls/Combo Box
  – PV Name: css:sensor
  – Items From PV: Yes

• Add Monitors/LED
  – PV Name: css:sensor
  – Off Color: green
  – On Color: red

Similar for bo, mbbi, mbbo records
Demo: Linking Container and Macro

- Create a new OPI: embedded.opi
- Add a label widget with text *PV Name: $(pv)*
- Add an XY Graph widget
  - *PV Name: $(pv)*
- Save
Exercise: Linking Container and Macro (cont’d)

• Create a new OPI: linkingContainer.opi
  – Add an Others/Linking Container widget
    • Macros: pv=sim://noise
    • OPI File: embedded.opi
  – Duplicate extra two linking container widget
  – Set Macros to pv=sim://sine and pv=sim://ramp respectively
  – Run