

# Display Builder

Kay Kasemir

July 2026

ORNL is managed by UT-Battelle, LLC for the US Department of Energy

# Display Builder

- Operator Interface Editor and Runtime
- Builds on ideas from EPICS edd/dm, medm, edm (~2000), ..
- Very compatible with CS-Studio 'BOY' from ~2009
- Started ~2015 in CS-Studio/Eclipse, now in CS-Studio/Phoebus
- Since 2019 also available as Web Runtime

**BL4A User Motors**

**Sample & Detector**

	Destination Pos	Current Pos	Status	Scan
SANGLE	0.5000 deg	0.5012 deg	Green	Scan
SampleX	-8.9618	-8.9618	Green	Scan
Beam Stop	0.0362 mm	0.0362 mm	Green	Scan
Sample Changer	Undefined	-87.0008 mm	Green	Scan
DANGLE	13.0000 deg	13.0015 deg	Green	Scan

**Slits - Collimation**

	Destination Pos	Current Pos	Status	Scan
S1Width	0.500 mm	0.501 mm	Green	Scan
S1VHeight	30.000 mm	29.998 mm	Green	Scan
S2Width	3.000 mm	2.996 mm	Green	Scan
S2VHeight	30.000 mm	30.010 mm	Red	Scan
S3Width	0.500 mm	0.500 mm	Green	Scan
S3VHeight	40.000 mm	40.000 mm	Green	Scan

**Slits - Background**

	Destination Pos	Current Pos	Status	Scan
SSHWidth	0.000 mm	0.000 mm	Red	Scan
LSlit4	-1.0092 mm	-1.0145 mm	Green	Scan
TDetSlit	0.0539 mm	0.0521 mm	Green	Scan
LDetSlit	-0.1279 mm	-0.1276 mm	Green	Scan
RSlit4	-58.5160 mm	-58.5165 mm	Green	Scan
BDetSlit	0.0438 mm	0.0455 mm	Green	Scan
RDetSlit	-4.9926 mm	-4.9927 mm	Green	Scan

**Stop Motors**

SampleX Mode  
SampleX->HexaX

**Phoebus (on bl12-dasrv1.sns.gov)**

**Instrument Status**

- Beam Power (kW): 1.166 kW
- Primary Shutter: Green
- Secondary Shutter: Green
- Acquisition Software Status: Green
- Data Reduction Status: Purple

**Proposal Information**

- Proposal #: IPTS-21677
- Proposal Title: Commissioning-TOPAZ
- Team Members: 5XW:FCT,JIU

**Run Information**

- Scan Status: Use
- Run Status: Use
- Run Number: 31017
- Run Time: 2061.1 s
- Total Neutron Counts: 20017936
- Count Rate (counts/s): 0
- Total Proton Charge: 0.4585 C
- Beam Monitor 1 Counts: 2280665
- Beam Monitor 2 Counts: 1599739

**Main Detector XY Plot (4x4 Binned)**

Min: 400, Max: 324, Mean: 238.375, Total: 68652, Rate: 0 e/s

**Scan Monitor**

ID	Created	Name	State	%	Runtime	Finish	Command	Error
105	11:11:07	Amp/20...	Aborted		02:04:30	13:15:38	- end -	Aborted
104	09:53:50	Amp/20...	Aborted		01:16:45	11:10:35	- end -	Aborted
103	09:45:09	Amp/20...	Aborted		00:02:02	09:47:12	- end -	Aborted
102	09:40:58	Amp/20...	Aborted		00:02:26	09:43:25	- end -	Aborted
101	09:39:11	Amp/20...	Aborted		00:01:08	09:40:20	- end -	Aborted
100	09:36:27	Amp/20...	Aborted		00:02:18	09:38:45	- end -	Aborted
99	09:33:13	Amp/20...	Aborted		00:02:57	09:36:11	- end -	Aborted
98	09:26:30	Amp/20...	Aborted		00:04:16	09:30:47	- end -	Aborted
97	09:24:02	Amp/20...	Aborted		00:01:35	09:25:38	- end -	Aborted

Scan Server Heap: 66.8 / 1024.0 MB (6.5 %), Non-Heap: 87.5 MB

# Examples: SNS Accelerator

## SNS Central Control Room

Sep 13, 2018 10:56:05

**Foil Image**

**937 Turns**

MEBT BCM02

**Energy: 1010 MeV**

**Rep Rate**

**59.9 Hz**

**Beam Targ**

**Target Image**

**Shutter Status**

1B NOMAD	2 BASIS	3 SNAP
4A Magnetism	4B Liquids	5 CNCS

**12-Hr Power on Target**

**RTBT30 Beam Size**

**Diagnostics**

Video Foil Monitor

**Camera Setup**

Camera Select: Primary / Secondary

Overlay: On / Off

Primary Lamp: Off / On / Off

Primary Insta-View: Off / On / Off

Secondary Lamp: Off / On / Off

Secondary Insta-View: Off / On / Off

Calibrate: Off / On / Off

**Neutral Density Filter**

Primary	Secondary
100% (blank)	100% (blank)
25%	25%
100% (blank)	100% (blank)
25%	25%
25%	25%

**Horizontal**

Amplitude: 0.0000

Mean: 1.0000 mm

Std.Dev.: 0.0000 mm

Offset: 0.0000

Slope: 0.0000

Area: 0.0000

**Vertical**

Amplitude: 0.0000

Mean: 0.0000 mm

Std.Dev.: 0.0000 mm

Offset: 0.0000

Slope: 0.0000

Area: 0.0000

**Primary Camera Image:**

Plunge: 0.0000

Foil: 0.0000

## SNS E&RF Main Monitor

09/13/18 10:56:48 E&RF Mag. Health RFQ Fields Click LEDs for Detail

**Power on Target**  
**1377.20 kW**

**Rep Rate**

**59.9 Hz**

**Avg Current**

**27.5 mA**

**Availability**

Current Shift

Max Achievable: 99.91%

Running Total: 99.79%

Previous Shift

Total: 99.51%

**RFQ**

MEBT

DTL

CCL

**MEBT Amplifiers**

X1 X2 X3 X4 X5 X6 X1 X2 X3 X4

**Recent Downtimes**

2018/09/12 16:52:10	2018/09/12 16:54:25	0.0
2018/09/12 13:21:46	2018/09/12 13:22:37	0.0
2018/09/12 13:10:16	2018/09/12 13:14:01	0.1

**Faults**

Ring Mag EKick08\_CT fault

SCL LLRF HPM22a fault

**MPS Latest MPS Fault**

Sep 13 2018 10:48:32 RFQ\_LLRF:HPM1:FPAR\_MEBT\_BS

Chain: MEBT Chassis: RFQ\_HPRF1A (Col, Row) (1, 8)

**E&RF Systems (Mobile View)**

HPRF Main HVCM Main LLRF Main

E&RF WIKI ELog

**Cavities**

XMTR	Modulator	Cavities																						
01	SCL 01	01a	01b	01c	02a	02b	02c	03a	03b	03c	04a	04b	05	SCL 05	04c	05a	05b	05c	06a	06b	06c	07a	07b	07c
09	SCL 09	08a	08b	08c	09a	09b	09c	10a	10b	10c	11a	11	SCL 12	11b	11c	12a	12b	12c	12d	13a	13b	13c	13d	
14	SCL 14	14a	14b	14c	14d	15a	15b	15c	15d	16a	16b	17	SCL 15	16c	16d	17a	17b	17c	17d	18a	18b	18c	18d	
20	SCL 18	19a	19b	19c	19d	20a	20b	20c	20d	21a	21b	21	SCL 21	21c	21d	22a	22b	22c	22d	23a	23b	23c	23d	

CS-Studio 'BOY' \*.opi files



# Browse the Examples

- Start CSS/Phoebus
- Your setup might have a menu entry
  - File, Top Resources, Examples
- If not, or if you'd like to inspect and edit the examples
  - Applications, Display, Examples, Install Example Displays

Main Application Toolbar  
Menu Window, Show Toolbar

Tab  
Hover mouse,  
open Context Menu,  
Close

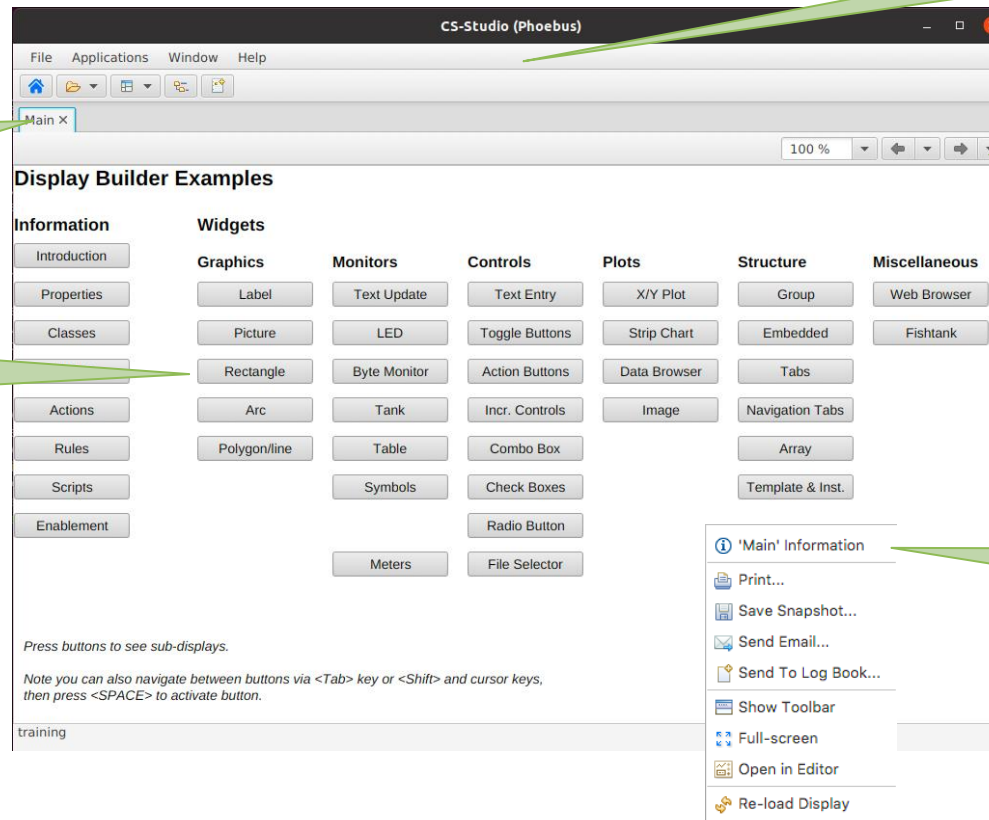
Display Runtime Toolbar  
Context Menu Window,  
Show / Hide Toolbar

Navigate back/forward  
also via Alt-Left, Alt-Right cursor keys

Zoom  
to view large control room displays on  
office computer

Example Display  
Push any of the buttons

Context Menu  
Details change with widget  
on which menu was invoked

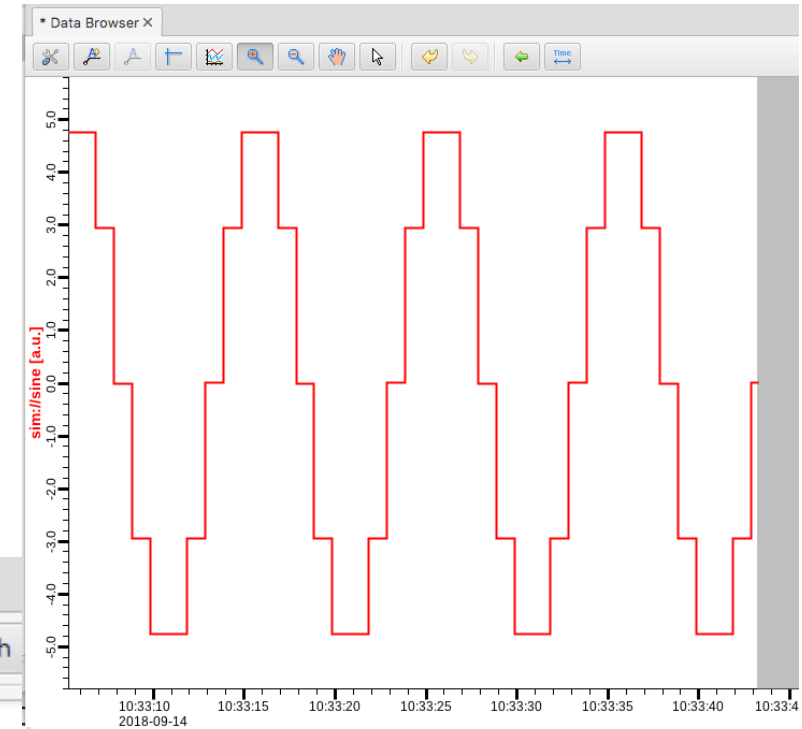


# Send PV to other Tools

Context menu opens other tool with PV

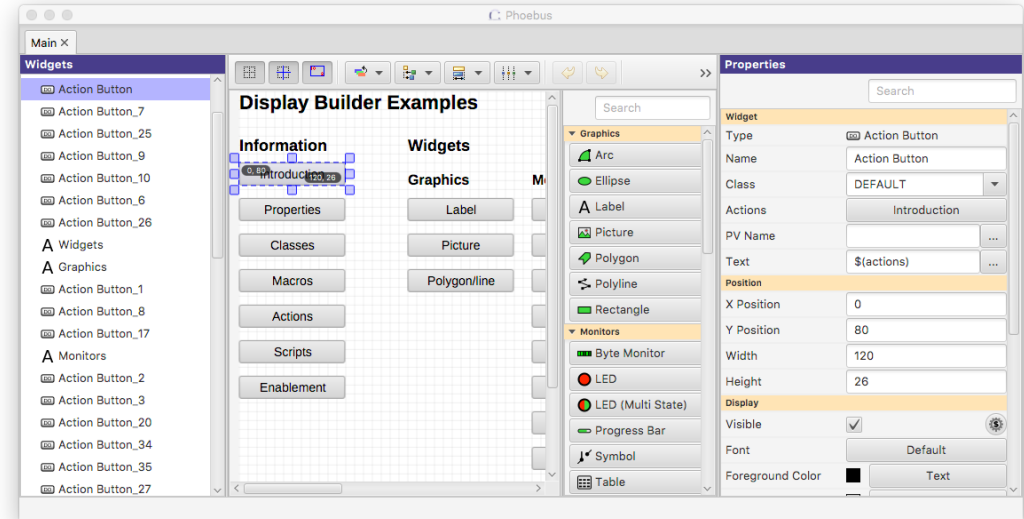
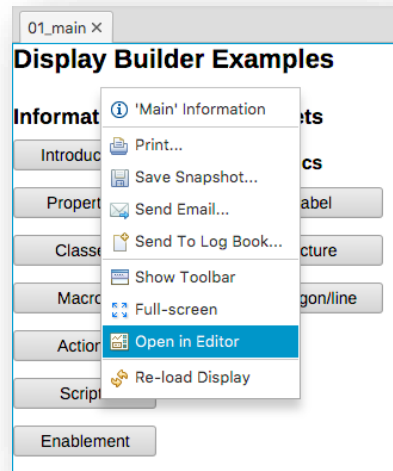
The image shows a context menu with the following items: 'Text Update4' Information, Copy PV to Clipboard, Data Browser, PV Table, PV Tree, and Probe. A blue arrow points from the 'Data Browser' item to the right. Another blue arrow points from the 'Probe' item to a 'Probe X' window. The 'Probe X' window displays the following information:

PV Name:	sim://sine	Search
Value:	4.755282581475438	
Alarm:	MAJOR - HIHI	
Time Stamp:	2018-09-14 10:33:25.854354000	
Metadata:	Units : a.u. Format : 0.123456789 Range : -5.0 .. 5.0 Warnings: -3.0 .. 3.0 Alarms : -4.0 .. 4.0	

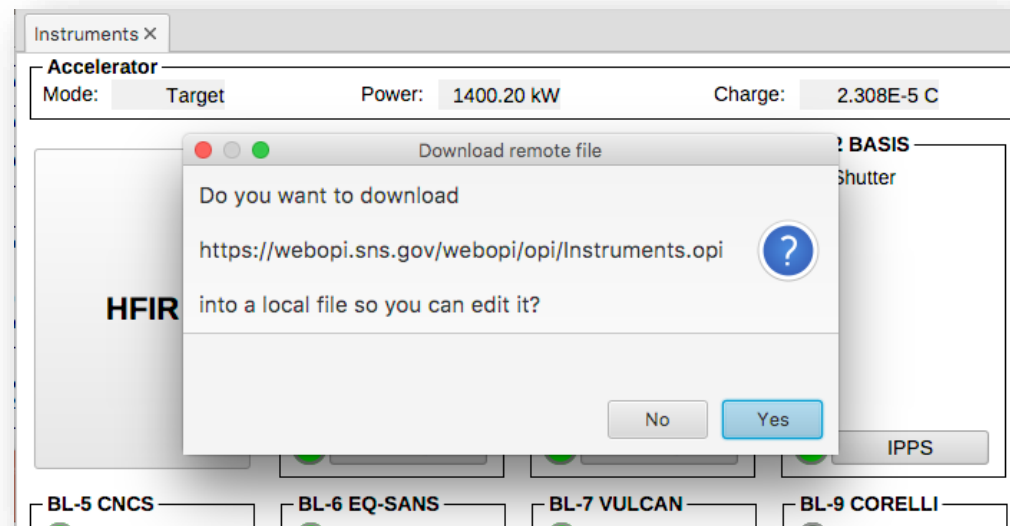


# Open Existing Display In Editor

- Context menu can open any display in Editor



- Downloads remote files



# Create New Display

Menu Applications, Display, New Display

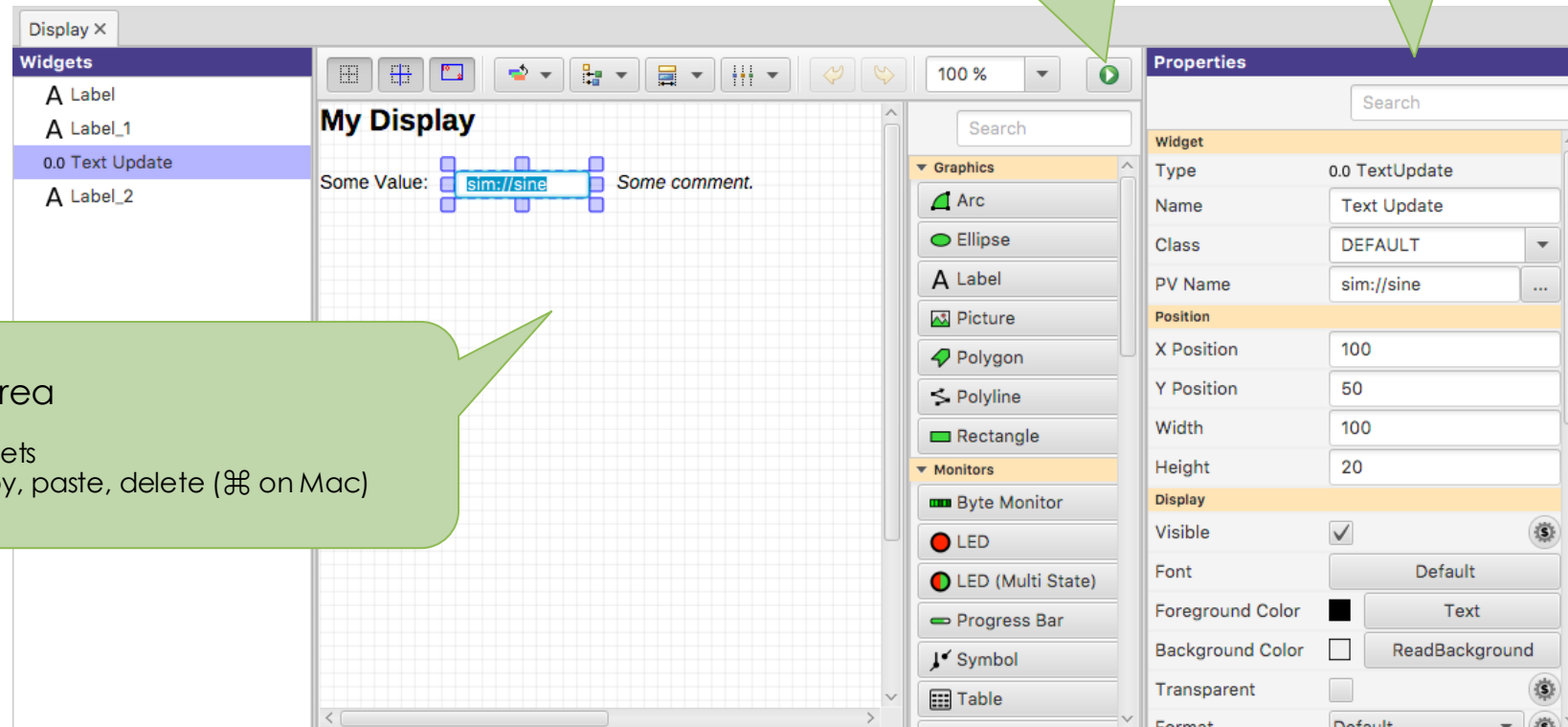
- Enter a name with .bob file extension

Save & Execute the Display

Property Panel  
Edit properties of  
selected widgets

Main Editor Area

Select Widgets  
Move, resize widgets  
Ctrl-C, V, X to copy, paste, delete (⌘ on Mac)



# Editing a Display

## Selecting Widgets

- a) Click single widget
- b) Ctrl-click to add widget (⌘ on Mac)
- c) Drag 'rubberband' around widgets
- d) Click or Ctrl/⌘ click in widget list

## Widget Palette

Drag widget into editor

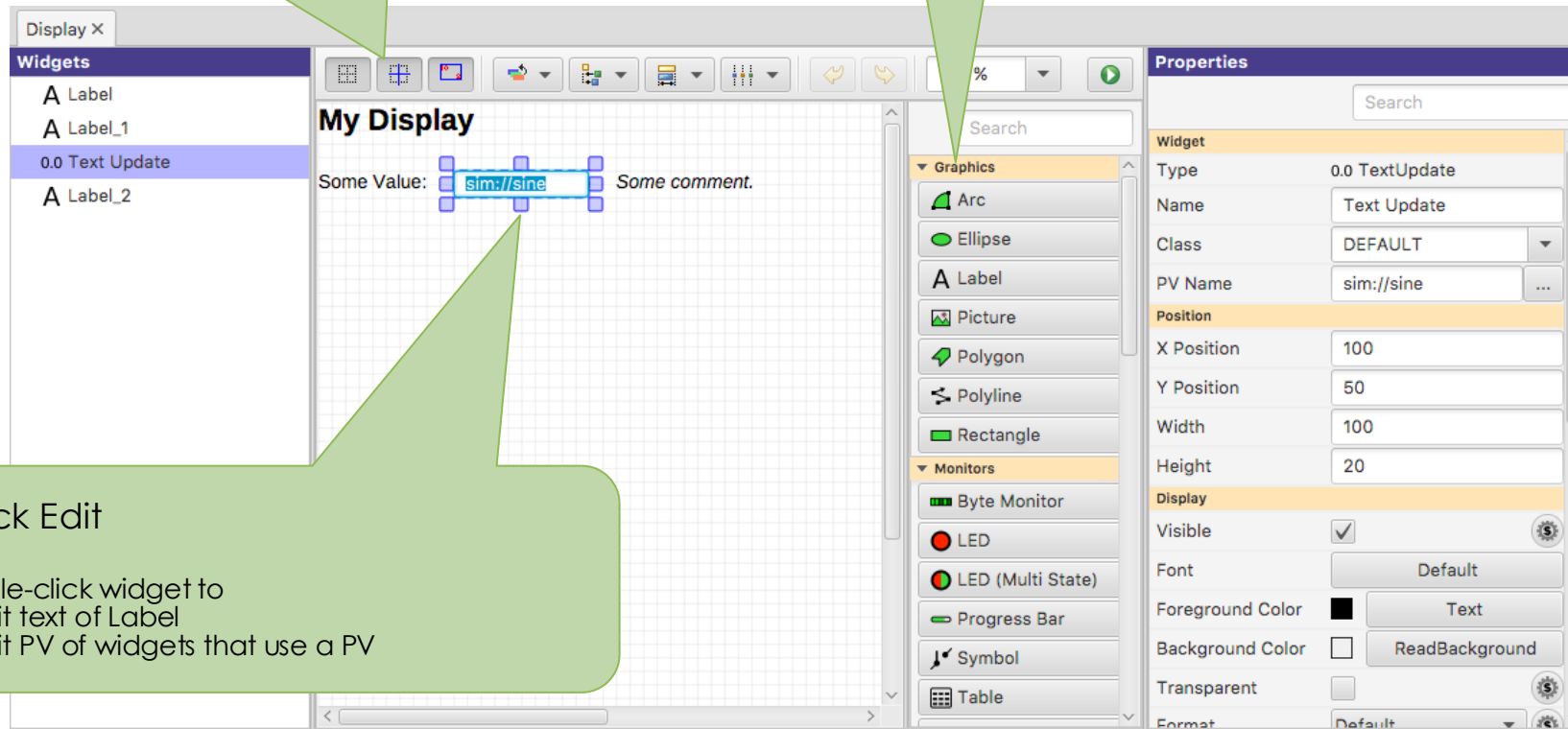
- or -

- 1) Select Widget Type
- 2) Draw rectangular area in display

## Quick Edit

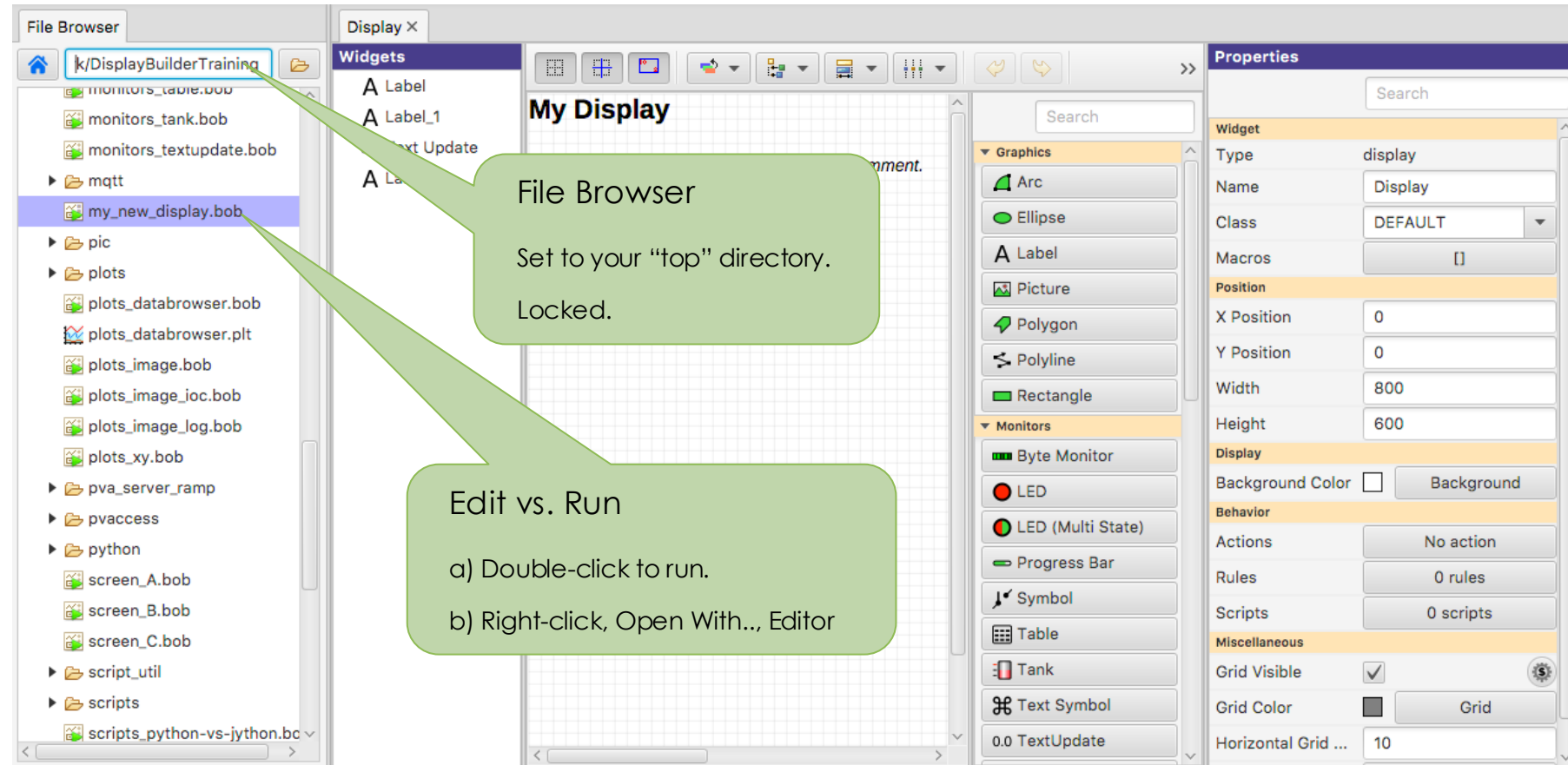
Double-click widget to

- a) Edit text of Label
- b) Edit PV of widgets that use a PV



# Suggested Setup for Editing

- Pick a top directory, for example where you installed the example files
- Open Applications, Utility, File Browser
  - Set it to your top directory
  - On file browser tab, open context menu, “Split Horizontally”, then “Lock Pane”
- Menu Window, Save Layout As..
  - “Editing”
- Menu Applications, Display, New Display
  - Create new file in your top directory



# Keep It Simple

1. Add a Widget
2. Enter Label's Text or Widget's PV Name
3. Done

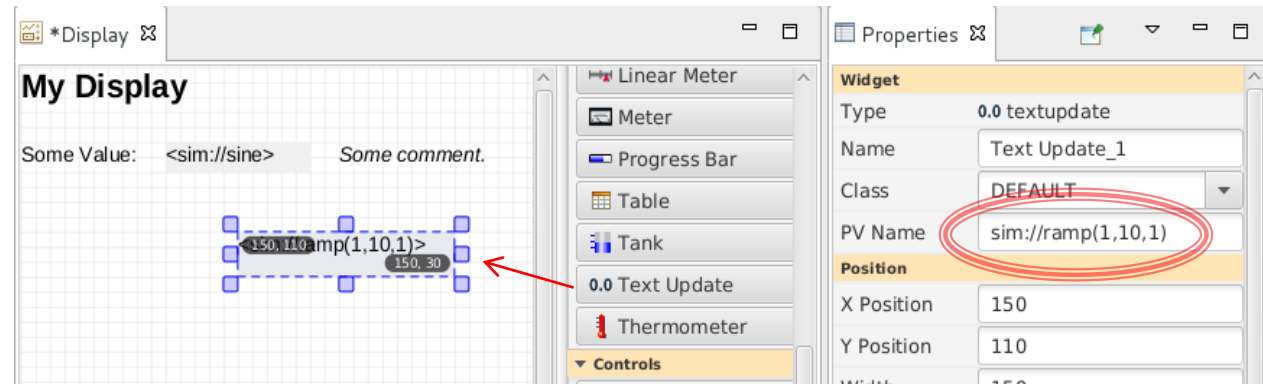
Basic Number:	-4.76 a.u.
Disconnecting channel:	<sim://intermittent>
Basic Text:	AAAAA

## At Runtime, widget will

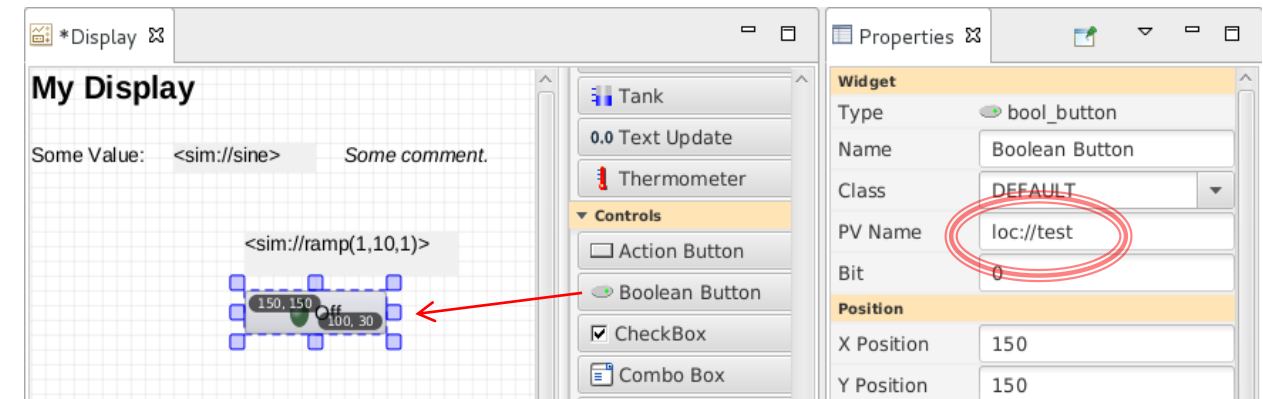
- ✓ Show PV's value, formatted, with units
- ✓ Indicate alarm, disconnect
- ✓ Show tool-tip with PV name and value
- ✓ Combo options read from Enum PV, slider range from numeric PV
- ✓ Disabled when 'control' widget has no PV write access

# Extend the First Display

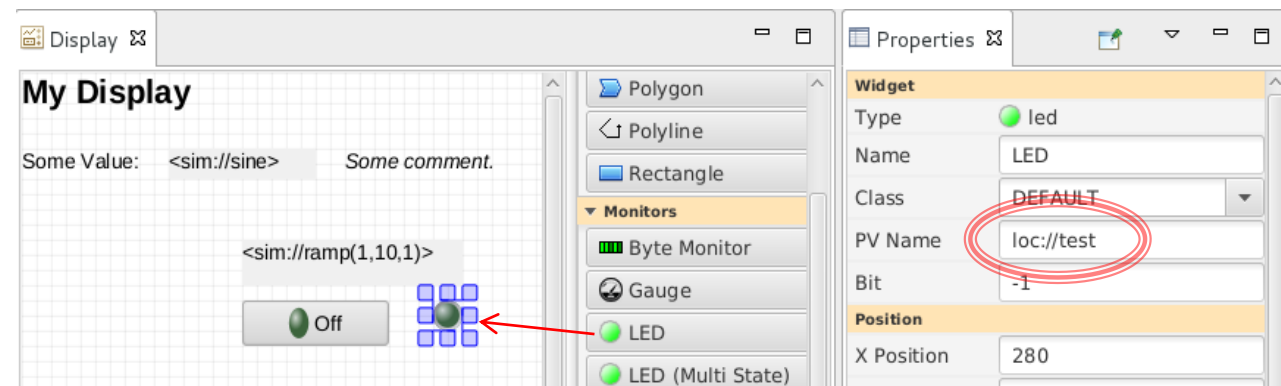
- Drag a “Text Update” from the palette
  - Enter PV name “sim://ramp(1, 10, 1)”.  
Note PV name auto-completion popup.



- Add “Boolean Button”
  - PV name “loc://test”
- Add “LED”
  - PV name “loc://test”.  
Note name in PV History.



- Execute the display
  - Toolbar Button or Context Menu

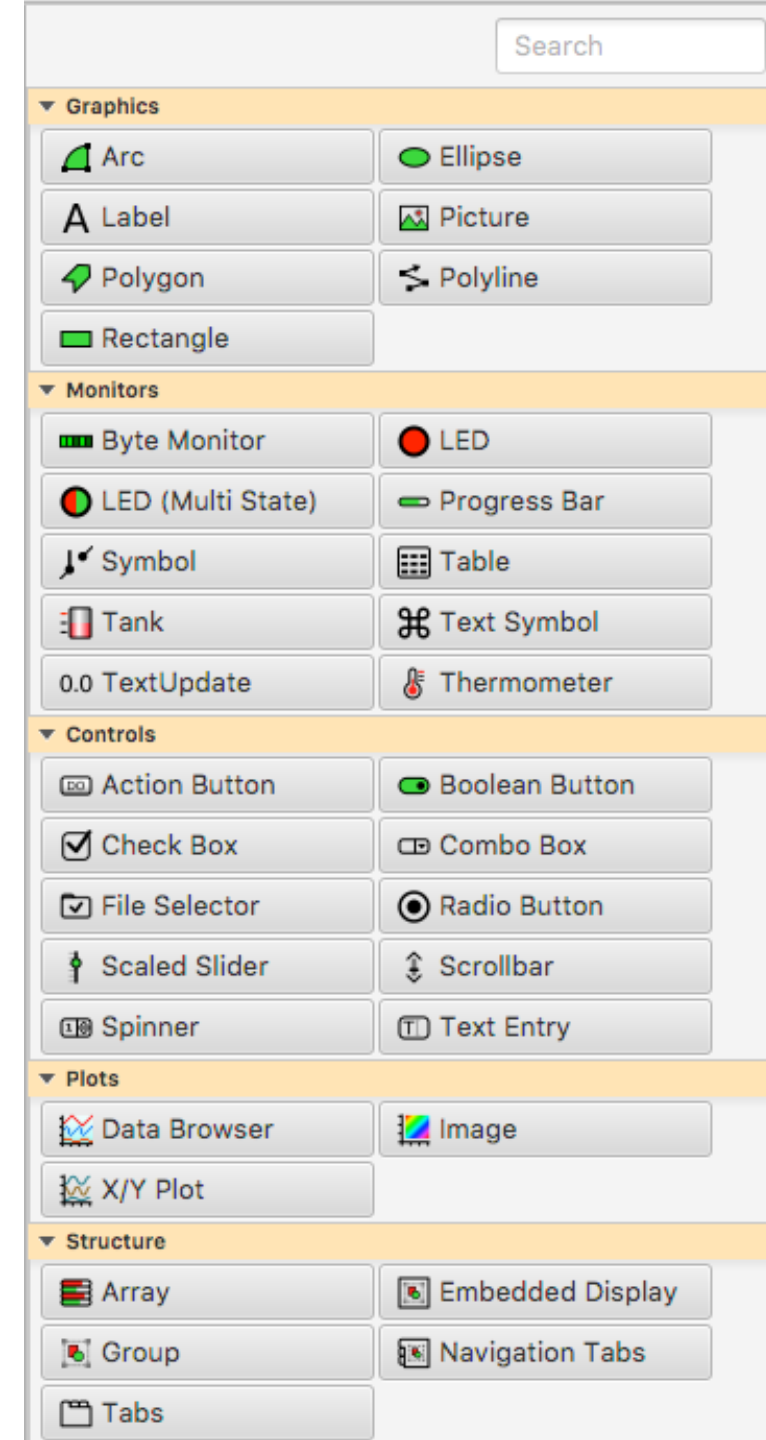


# PV Names

- `ca://some_pv_name`
  - EPICS Channel Access PV
- `some_pv_name`
  - Typically same, since “ca://” is the default
  - Eventually, “pva://” may become the default
- `sim://sine`
  - Simulated PV. See auto-completion hints
- `loc://x(4)`
  - Local PV. See auto-completion hints
- `pva://x`
  - EPICS PV Access

# Widget Palette

- Shows all available widgets
  - Enter name for "Search"
  - Hover mouse for description
  - Drag -or- Select & Rubberband
- Categories
  - Graphics show static label, picture, ..
  - Monitors update based on reading a PV
  - Controls read a PV and can write to the PV
  - Plots tend to read from one or more (waveform) PVs
  - Structures group widgets, embed sub-displays



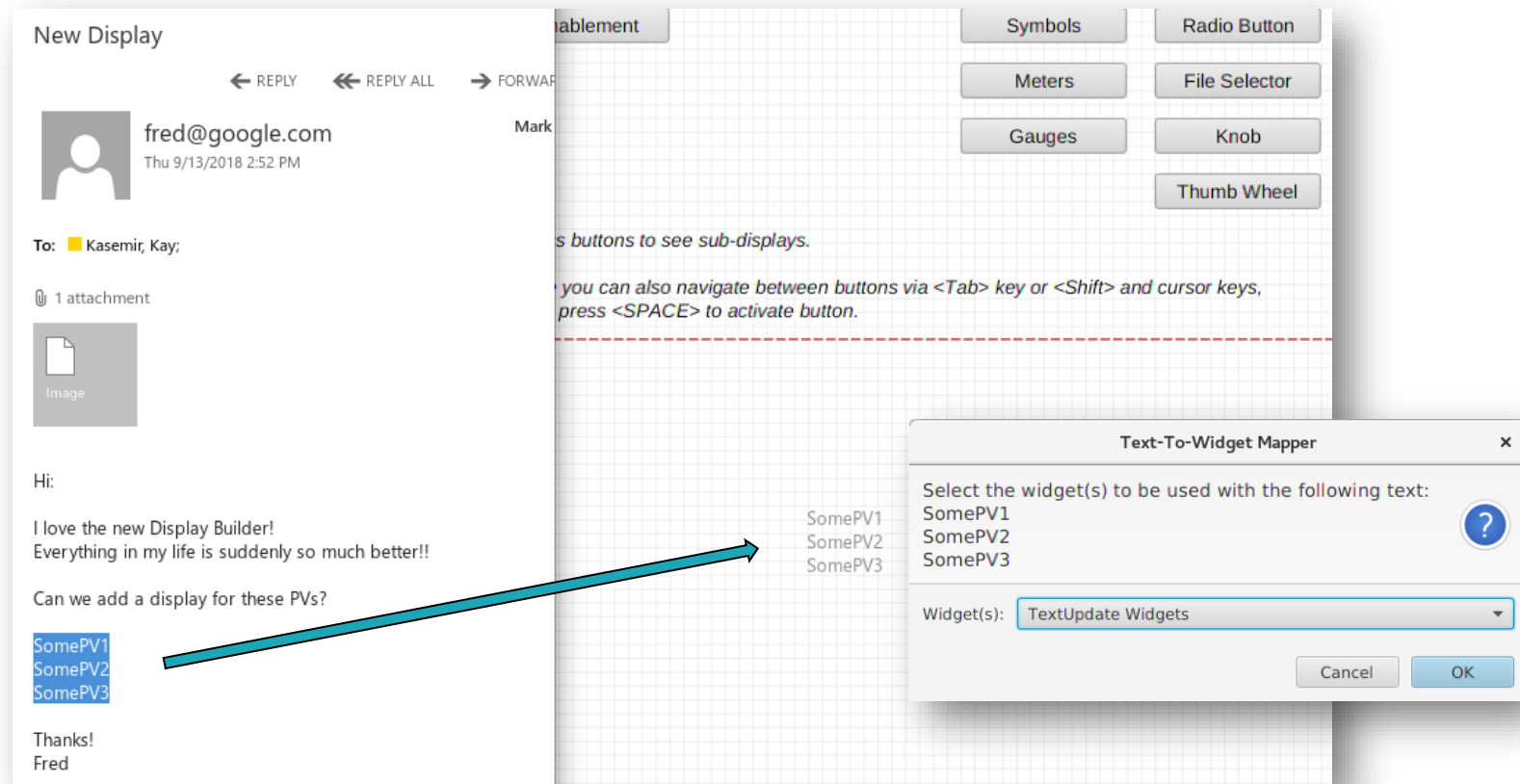
# Create Widgets via Drag/Drop from other Apps

## Email with list of PVs?

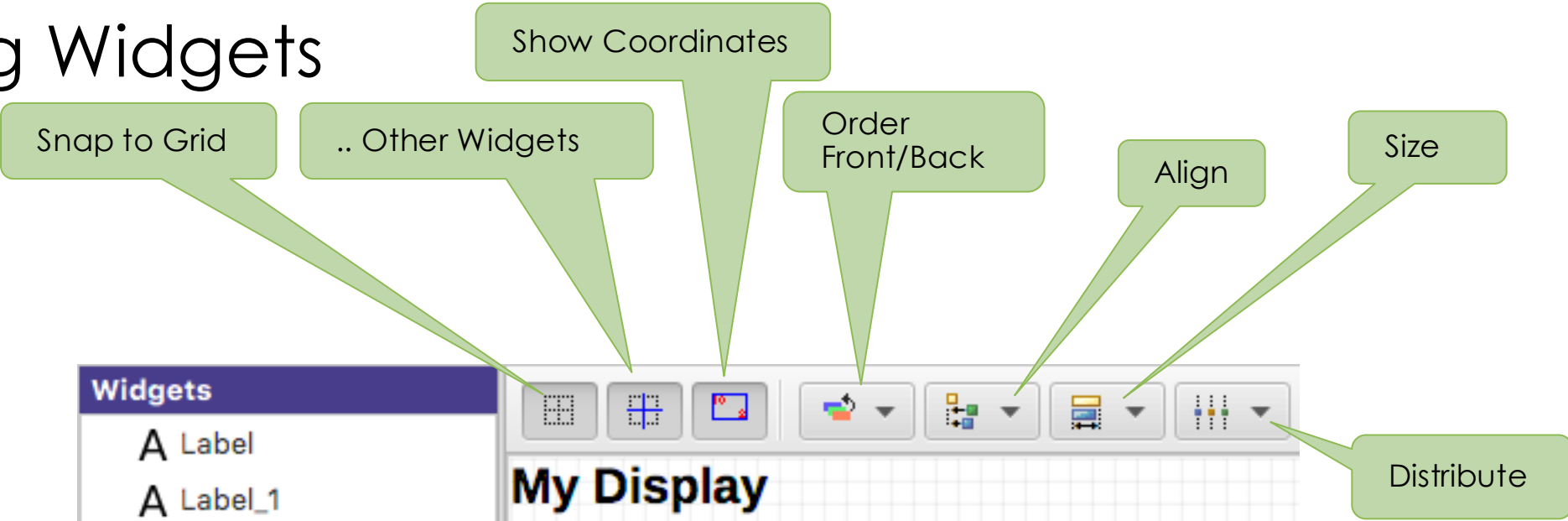
- Drag that text into Display Editor
- Select widget type

## Supported:

- Text → Label
- Text → PV Widget
- Image File → Picture Widget
- \*.bob File → Embedded Display Widget



# Manipulating Widgets



Widget List  
Select widgets  
Rename Widgets  
View/change their order.

**Widgets**

- A Label
- A Label\_1
- A Label\_2
- LED
- 0.0 Text Update
- ▼ [Group Icon] Group
  - Boolean Button
  - 0.0 TextUpdate

**My Display**

Some Value: 100, 50     *Some comment.*

Group

Off

138, 148

Selected Widgets Tracker  
Move or resize selected widgets

# Display Properties

Click on display background to select no widget for editing overall display properties

- Name
  - Shown in Tab
- Macros
  - Used by all widgets in this display
- Grid size
  - Can aid with placing widgets

Properties	
Search	
<b>Widget</b>	
Type	display
Name	Display
Class	DEFAULT
Macros	[]
<b>Position</b>	
X Position	0
Y Position	0
Width	800
Height	600
<b>Display</b>	
Background Color	<input type="checkbox"/> Background
<b>Behavior</b>	
Actions	No action
Rules	0 rules
Scripts	0 scripts
<b>Miscellaneous</b>	
Grid Visible	<input checked="" type="checkbox"/>
Grid Color	<input type="checkbox"/> Grid
Horizontal Grid Step Size	10
Vertical Grid Step Size	10

# Widget Properties

Select one (or more) widgets to edit their (common) properties

- Search
  - To find desired property
- PV Name
  - Most important property for most widgets

Details depend on the widget type

The screenshot shows a 'Properties' panel for a widget. The panel is organized into several sections:

- Widget**: Type (0.0 TextUpdate), Name (Text Update), Class (DEFAULT), PV Name (sim://sine).
- Position**: X Position (100), Y Position (50), Width (100), Height (20).
- Display**: Visible (checked), Font (Default), Foreground Color (black), Background Color (ReadBackground), Transparent (unchecked), Format (Default), Precision (-1), Show Units (checked), Horizontal Alignment (Left), Vertical Alignment (Top), Wrap Words (checked), Rotation (0 degrees).
- Behavior**: Actions (No action), Rules (0 rules), Scripts (0 scripts), Tool tip (\$(pv\_name)\$ (pv\_value)), Alarm Border (checked), Interactive (unchecked).
- Miscellaneous**: Border Width (0), Border Color (black).

# Common Widget Properties

Defaults tend to be reasonable:

- Format with precision set by PV
- Show units provided by PV
- Alarm-sensitive Border
- Fetch Items (Combo, ...) from PV

Instead of changing them,  
maybe the PV needs to be updated?

Still, can be adjusted as needed for the display.

Properties	
Search	
<b>Widget</b>	
Type	0.0 TextUpdate
Name	Text Update
Class	DEFAULT
PV Name	sim://sine
<b>Position</b>	
X Position	100
Y Position	50
Width	100
Height	20
<b>Display</b>	
Visible	<input checked="" type="checkbox"/>
Font	Default
Foreground Color	Text
Background Color	ReadBackground
Transparent	<input type="checkbox"/>
Format	Default
Precision	-1
Show Units	<input checked="" type="checkbox"/>
Horizontal Alignment	Left
Vertical Alignment	Top
Wrap Words	<input checked="" type="checkbox"/>
Rotation	0 degrees
<b>Behavior</b>	
Actions	No action
Rules	0 rules
Scripts	0 scripts
Tool tip	\$(pv_name)\$ (pv_value)
Alarm Border	<input checked="" type="checkbox"/>
Interactive	<input type="checkbox"/>
<b>Miscellaneous</b>	
Border Width	0
Border Color	Text

# Predefined "Named" Colors and Fonts

Use whenever possible!


Name	Text Update_4
Class	DEFAULT
PV Name	sim://sine
<b>Position</b>	
X Position	191
Y Position	351
Width	170
Height	20
<b>Display</b>	
Visible	<input checked="" type="checkbox"/>
Font	Default
Foreground Color	Text
Background Color	ReadBackground
Transparent	<input type="checkbox"/>
Format	Default
Precision	-1
Show Units	<input checked="" type="checkbox"/>
Horizontal Alignment	Left
Vertical Alignment	Top
Wrap Words	<input checked="" type="checkbox"/>
Rotation	0 degrees

<b>Position</b>	
X Position	191
Y Position	91
Width	170
Height	20
<b>Display</b>	
Visible	<input checked="" type="checkbox"/>
Font	Default
Foreground Color	Text

Font – Select a predefined font and/or customize it.

Predefined Fonts		Fonts Families			
Comment		Liberation Sans			
Default		Liberation Serif			
Default Bold		Libian SC			
Fine Print		LiHei Pro			
Header 1		LiSong Pro			
Header 2		Lucida Bright			
Header 3		Lucida Grande			
Oddball		Lucida Sans			
		Lucida Sans Typewriter			
Style:	Regular	Size:	14.0		
Preview					
Example Text					
Default		Cancel		OK	

Foreground Color – Select a predefined color and/or customize it.

Predefined Colors		Custom Color					
<input type="checkbox"/> DISCONNECTED		Color:	Black				
<input type="checkbox"/> Grid		Red:	0				
<input type="checkbox"/> Header_Background		Green:	0				
<input type="checkbox"/> Header_ForeGround		Blue:	0				
<input type="checkbox"/> INVALID		Alpha:	255				
<input type="checkbox"/> MAJOR							
<input type="checkbox"/> MINOR							
<input type="checkbox"/> Off							
<input type="checkbox"/> OK							
<input type="checkbox"/> On							
<input type="checkbox"/> Read_Background							
<input type="checkbox"/> STOP							
<input type="checkbox"/> Text							
<input type="checkbox"/> Write_Background							
Search							
							
		Default		Cancel		OK	

# Configuring Named Colors, Fonts

```
# -----  
# Package org.csstudio.display.builder.model  
# -----  
  
# Widget classes  
# One or more *.bcf files, separated by ';'   
# Defaults to built-in copy of examples/classes.bcf  
class_files=examples:classes.bcf  
  
# Named colors  
# One or more *.def files, separated by ';'   
# Defaults to built-in copy of examples/color.def  
color_files=examples:color.def  
  
# Named fonts  
# One or more *.def files, separated by ';'   
# Defaults to built-in copy of examples/font.def  
font_files=examples:font.def
```

Ideally set at start  
of project

```
color.def  
# Named colors  
#  
# Format:  
#   NameOfColor = red, green, blue [, alpha ] |  
#   PreviouslyDefinedNameOfColor  
#   with values in 0..255 range.  
#  
# Whenever possible, use named colors in displays  
# instead of arbitrary red/green/blue values.  
  
# ----- Predefined colors -----  
# May be overridden in here  
  
# Alarm related  
OK = 0, 255, 0  
MINOR = 255, 128, 0  
MAJOR = 255, 0, 0  
INVALID = 255, 0, 255  
DISCONNECTED = 200, 0, 200, 200  
  
# Default color for text  
Text=0,0,0  
  
# Default color for 'active' text that's being edited  
ActiveText=255, 255, 0  
  
# Display background  
Background = 255, 255, 255  
  
# .. for widgets that read/write a value  
Read_Background = 240, 240, 240  
Write_Background = 128, 255, 255  
  
# .. for buttons  
Button_Background = 210, 210, 210  
  
# ----- Examples for additional colors -----  
  
# Also show ideas for site-specific guidelines that  
# are required to make sense of the color names.  
  
# Styling  
Header_Background=77,77,77  
Header_ForeGround=255,255,255  
  
# Use alarm colors only when you mean to indicate an
```

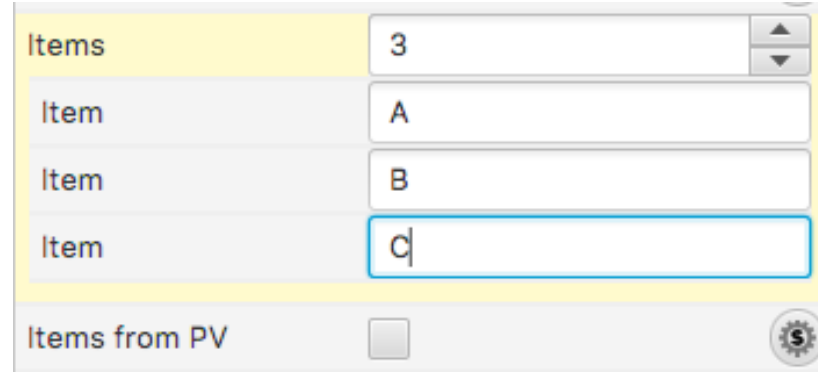
```
font.def  
// Named font definitions  
//  
// Entries in this file are read in sequence.  
// A later entry in the file can override  
// an earlier entry in the file.  
//  
// In a production setup, this file may be constructed  
// by concatenating a generic file with a more specific file,  
// and the specific entries would then override generic entries  
// of the same name.  
  
// Format:  
//   NamedFont['( OS )'] = Family '-' Style '-' Size | '@PreviouslyDefinedNamedFont'  
//  
// Family: Font family name "Liberation Sans", "Liberation Mono", "Liberation Serif"  
// Style: "regular", "bold", "italic", "bold italic"  
// Size: Font height in pixels  
// OS: "windows", "linux", "macosx"  
//  
// Leading/trailing spaces around each element are OK, but if the font family  
// is "Liberation Sans", it has to be typed with just that one space between  
// "Liberation" and "Sans"  
//  
// Examples of named fonts  
//  
//   Default      = Liberation Sans - regular - 14  
//   Default Bold = Liberation Sans - bold    - 14  
//   Header 1     = @Default Bold  
//  
// Speaking of "Liberation Sans":  
// The display builder includes the "Liberation" fonts  
// from https://fedorahosted.org/liberation-fonts.  
// Their use is encouraged because the resulting displays  
// will always render correctly.  
// When using other fonts, for example "Arial" on Windows,  
// the font might not be available to a display builder  
// runtime that is executing on Mac OS or Linux.  
  
// Predefined fonts that this file could re-define  
Default      = Liberation Sans - regular - 14  
Default Bold = Liberation Sans - bold    - 14  
Header 1     = Liberation Sans - bold    - 22  
Header 2     = Liberation Sans - bold    - 18  
Header 3     = Liberation Sans - bold    - 16  
Comment      = Liberation Sans - italic - 14
```

# Widget Notes

- Text Entry, Text Update:
  - Set Format = String for “long string” waveforms. Default will show array.
- LED, Boolean Button, Checkbox
  - Boolean PV
  - Numeric PV 0 or not 0 (when “Bit” set to default of -1)
  - Bit in a numeric PV (when “Bit” set to 0, 1, 2, ...)
- Multi-State LED
  - Enumerated or numeric PVs
  - Defaults to using state values 0, 1, 2, 3, ..

# Widget Notes

- Combo Box, Radio Button:
  - Best for enumerated PV: Enter PV name, done
  - Alternatively, un-check “Items from PV” and enter items

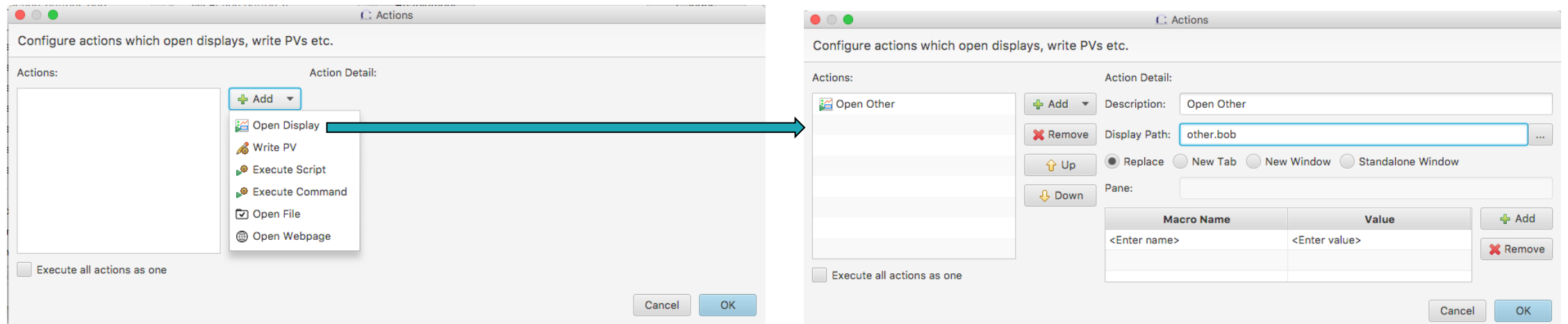


The screenshot shows a widget interface with the following elements:

- A yellow header bar with the label "Items" and a numeric input field containing the value "3".
- A list of three items, each with a label "Item" and a text input field:
  - Item 1: "A"
  - Item 2: "B"
  - Item 3: "c" (highlighted with a blue border)
- A footer bar with the label "Items from PV" and an unchecked radio button.
- A gear icon in the bottom right corner of the footer bar.

# Action Button

1. Add ActionButton
2. Configure “Actions” property, add “Open Display”



3. Run: Clicking button opens the “other” display.

*In principle, any widget can have ‘Actions’.  
They appear in the widget’s runtime context menu.  
But it’s not obvious to end users that for example a Label will have actions.*

# Screen Navigation

- Replace

- Suggested default.
- Allows back/forward navigation as in web browser



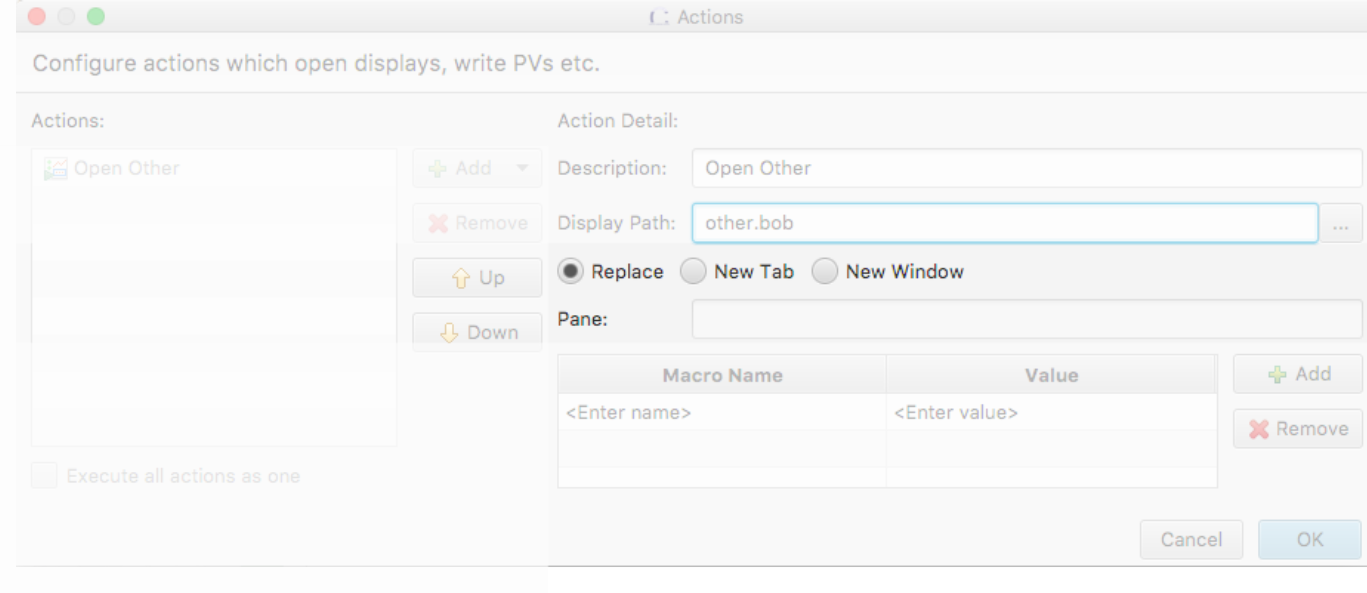
- Minimizes number of open screens

- New Tab

- Opens in new tab
- Allows specific Pane name

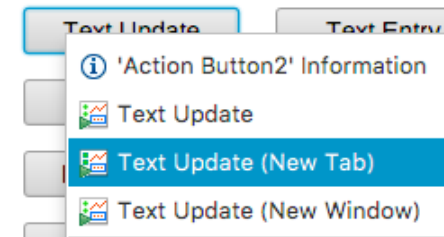
- New Window

- Opens in new window



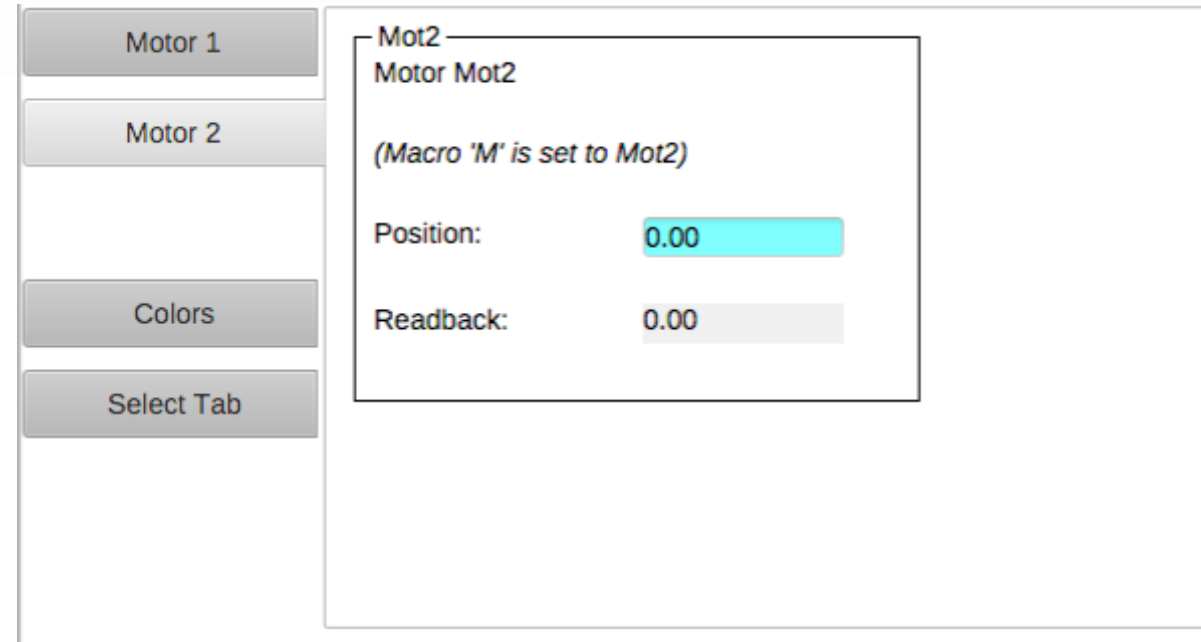
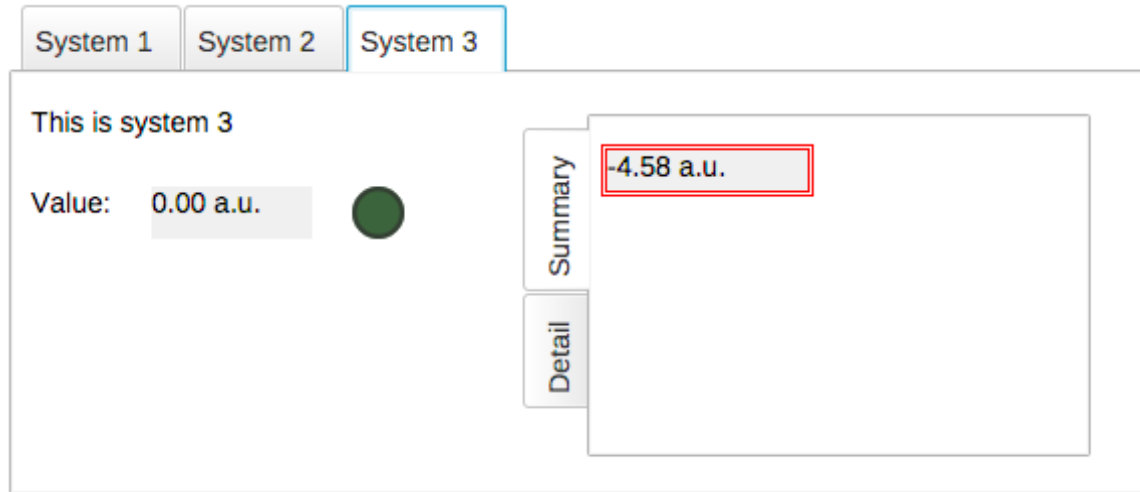
- With “Replace”, can still use

- Context menu



- Control (⌘ on Mac) for tab
- Shift-Control for window

# Screen Navigation: Tabs



## Tabs

Each tab is in-memory, same \*.bob

- Appears immediately when selected
- Uses CPU and memory when hidden

## Navigation Tabs

Tab is loaded from separate \*.bob when selected

- May need a little time to load
- No CPU and memory when hidden

# Macros

- Macros are passed into displays from
  1. Enclosing Group or Tab Widget
  2. Display
  3. Embedded widget container or Action that loaded the display
  4. Phoebus preferences
  5. Environment Variables

- To use:  $\$(NameOfMacro)$

.. or  $\${NameOfMacro}$ .  
EPICS \*.db files use  $\$(xx)$ ,  
SNL and shell use  $\${xx}$ ,  
so we support both conventions.

- Examples:

PV Name:             $\$(PV)$                             with PV=TheFullPVName

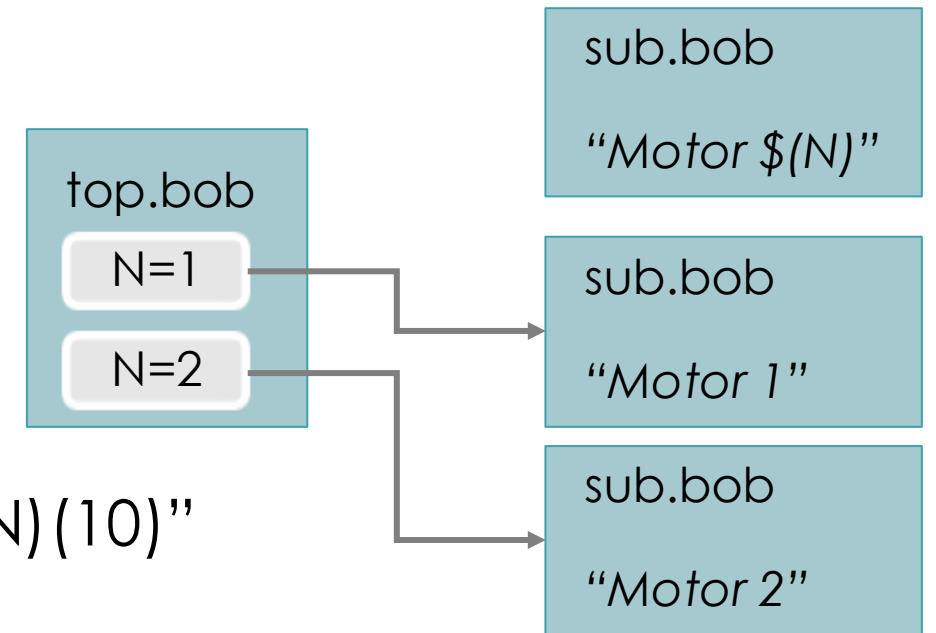
PV Name:            Motor $\$(N)$                             with N=1, 2, 3, ...

Width:               $\$(WID)$                             with WID=200

Visible:              $\$(SHOW)$                             with SHOW=true

# Macro Example

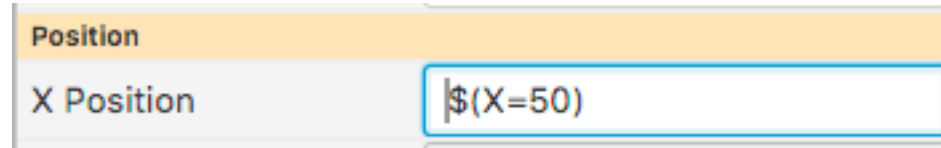
1. Create display "sub.bob"
  - Label with text "Motor \$(N)"
  - TextUpdate with PV "loc://pos\$(N)(10)"
  - ActionButton with PV Name "loc://pos\$(N)(10)" and Action to "Write PV" value 20
  - Copy that button, update to set PV to 30
2. Create display "top.bob"
  - ActionButton with Action to open sub.bob with N=1
  - Copy/paste the button, update to N=2
3. Execute top.bob, press buttons



# Macros

- Default values:  $\$(\text{MACRO}=\text{default})$

Allows standalone testing without passing values into display



A screenshot of a software interface showing a 'Position' section. The 'X Position' field is set to the macro  $\$(X=50)$ .

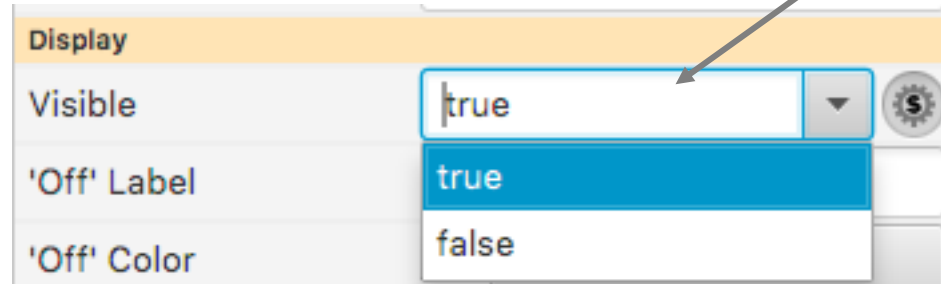
- To enter macro for Boolean  
Press the “\$” macro button

Select valid option from drop-down ...

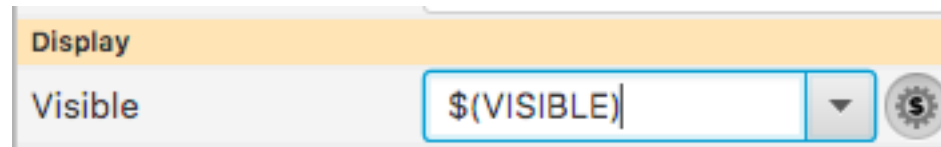


A screenshot of a software interface showing a 'Display' section. The 'Visible' checkbox is checked. A gear icon (macro button) is visible to the right of the checkbox.

.. or enter a macro



A screenshot of a software interface showing a 'Display' section. The 'Visible' field has a drop-down menu open, showing 'true' selected. A gear icon (macro button) is visible to the right of the drop-down menu. An arrow points from the gear icon in the previous screenshot to this one.



A screenshot of a software interface showing a 'Display' section. The 'Visible' field is set to the macro  $\$(\text{VISIBLE})$ .

# Macro Fallbacks

When macro is not defined, falls back to

- Widget Properties

- Uses the internal property name shown in tool-tip of Properties view
- Note how tooltip is usually preset to “\$(pv\_name)\n\$(pv\_value)”
- Action Button has PV Name property.  
It’s not used directly as in other widgets with PV name, but in “Write PV” the PV name is preset to \$(pv\_name)
- Action Button text is preset to “\$(actions)”

- Java Properties

- \$(os.name)

- Environment Variables

- \$(HOME), \$(USER)

# Predefined Macros

\$(DID): Unique display identifier, useful for per-display PVs

loc://x\$(DID)(10)

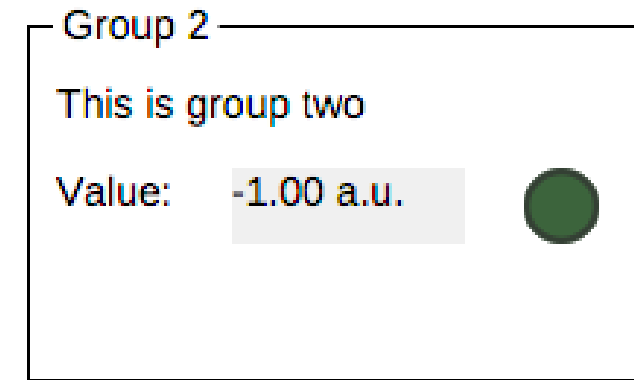
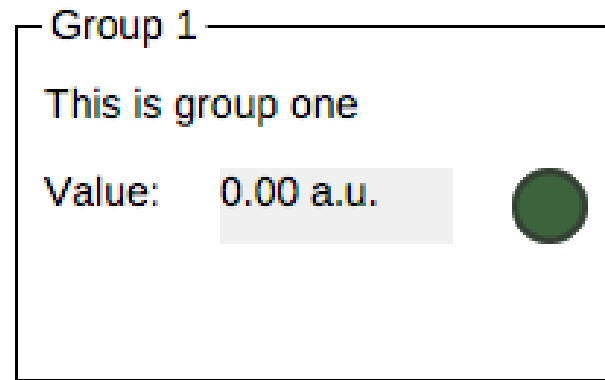
\$(DNAME): Display Name

# Group Widget

Contains other widgets

Visual Effect:

- Border, Name



Practical Effect:

- Group can define macros for contained widgets
- Group can be moved, copied/pasted as one unit in editor

# Group Widget


1) Add Group Widget

2) Move other widgets inside the Group

Active Group is highlighted

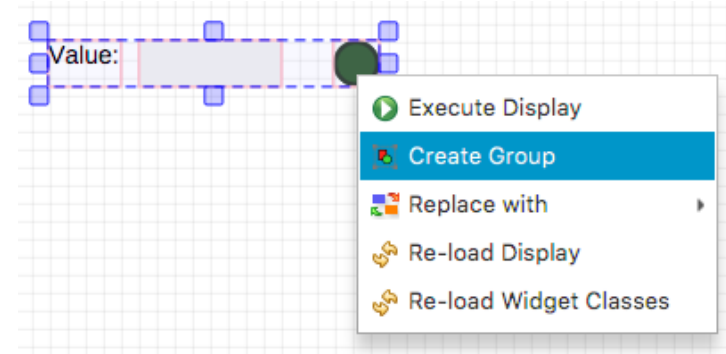
# Group Properties

- Name:  
Shown in border
- Style:  
“Group Box” for named border
- Macros:  
Passed to contained widgets

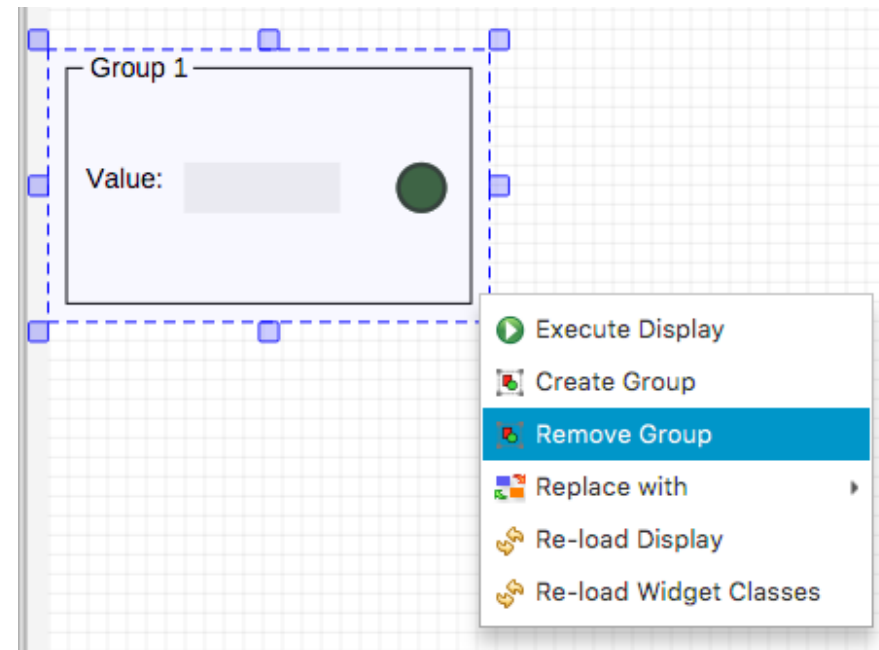
Properties	
<input type="text" value="Search"/>	
<b>Widget</b>	
Type	 Group
Name	<input type="text" value="Group 1"/>
Class	DEFAULT <span>▼</span>
Macros	[PV = 'sim://sine', TE...]
<b>Position</b>	
X Position	<input type="text" value="0"/>
Y Position	<input type="text" value="91"/>
Width	<input type="text" value="227"/>
Height	<input type="text" value="140"/>
<b>Display</b>	
Visible	<input checked="" type="checkbox"/> <span>⚙</span>
Style	Group Box <span>▼</span> <span>⚙</span>
Font	Default
Foreground Color	<input type="color" value="black"/> Text
Background Color	<input type="color" value="white"/> Background
Transparent	<input type="checkbox"/> <span>⚙</span>
<b>Behavior</b>	
Actions	No action
Rules	0 rules
Scripts	0 scripts
Tool tip	<input type="text"/> <span>⋮</span>

# Group Editing Shortcuts

1. Select Widgets
2. Context menu "Create .."



1. Select Group
2. Context Menu "Remove.."



# Embedded Display

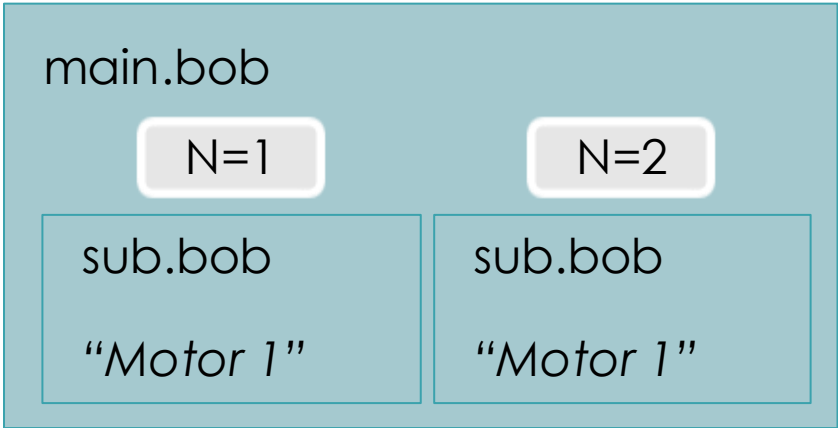
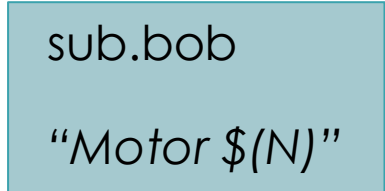
Hosts a complete \*.bob file within a widget

Allows composing higher-level displays from smaller displays:

- Per-device \*.bob
- Show multiple devices in one display

# Embedded Display Example

1. Create display “sub.bob” (or use the one created earlier)
  - Label with text “Motor \$(N)”
  - TextUpdate with PV “loc://pos\$(N)(10)”
2. Create display “main.bob”
  - Embedded Display, File sub.bob, Macros N=1
  - Copy/paste the Embedded Display, update to N=2
3. Execute main.bob



# Embedded Display Sizes

## a) Embedded Display Size

- Size of the widget that will host the \*.bob
- Defined by the widget Width and Height properties

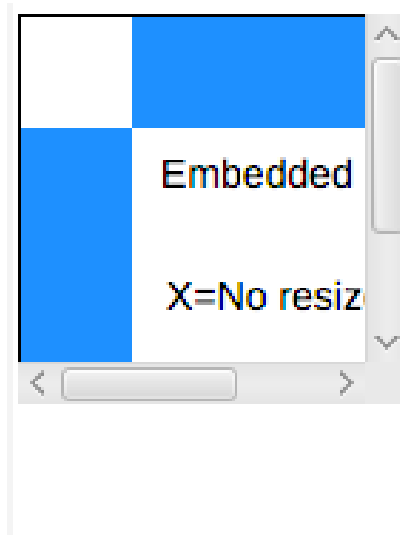
## b) Content Size

- Size of the \*.bob
- Defined by that Display Width and Height properties

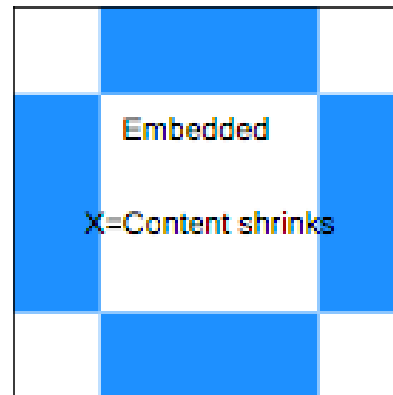
What if those sizes differ?

# Embedded Display Resize Options

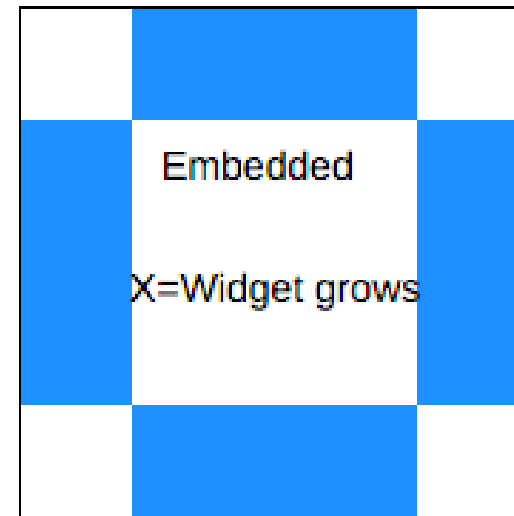
No Resize



Size content  
to fit widget

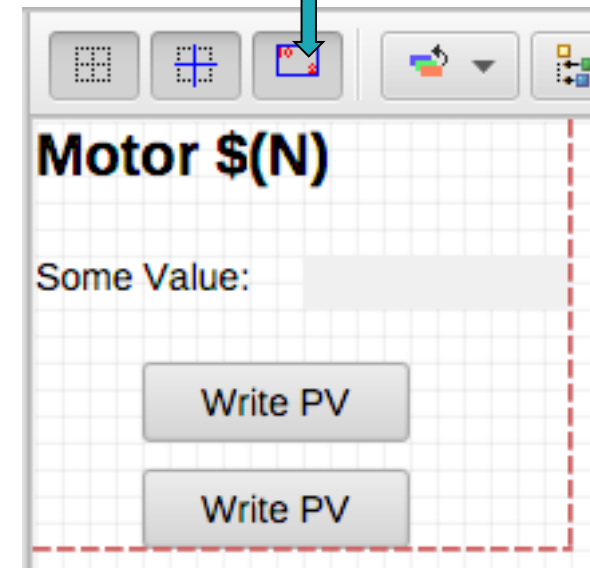
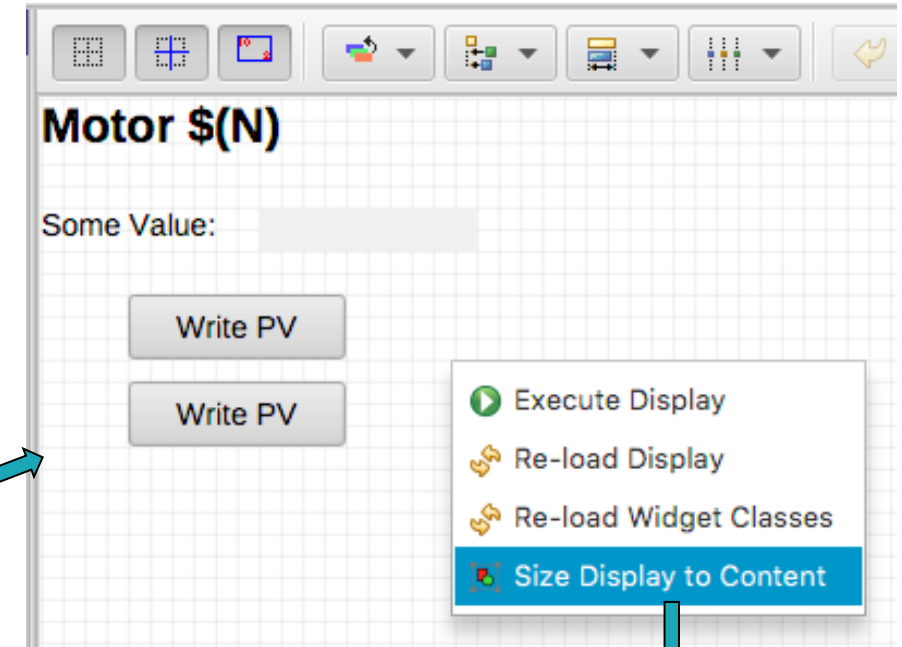
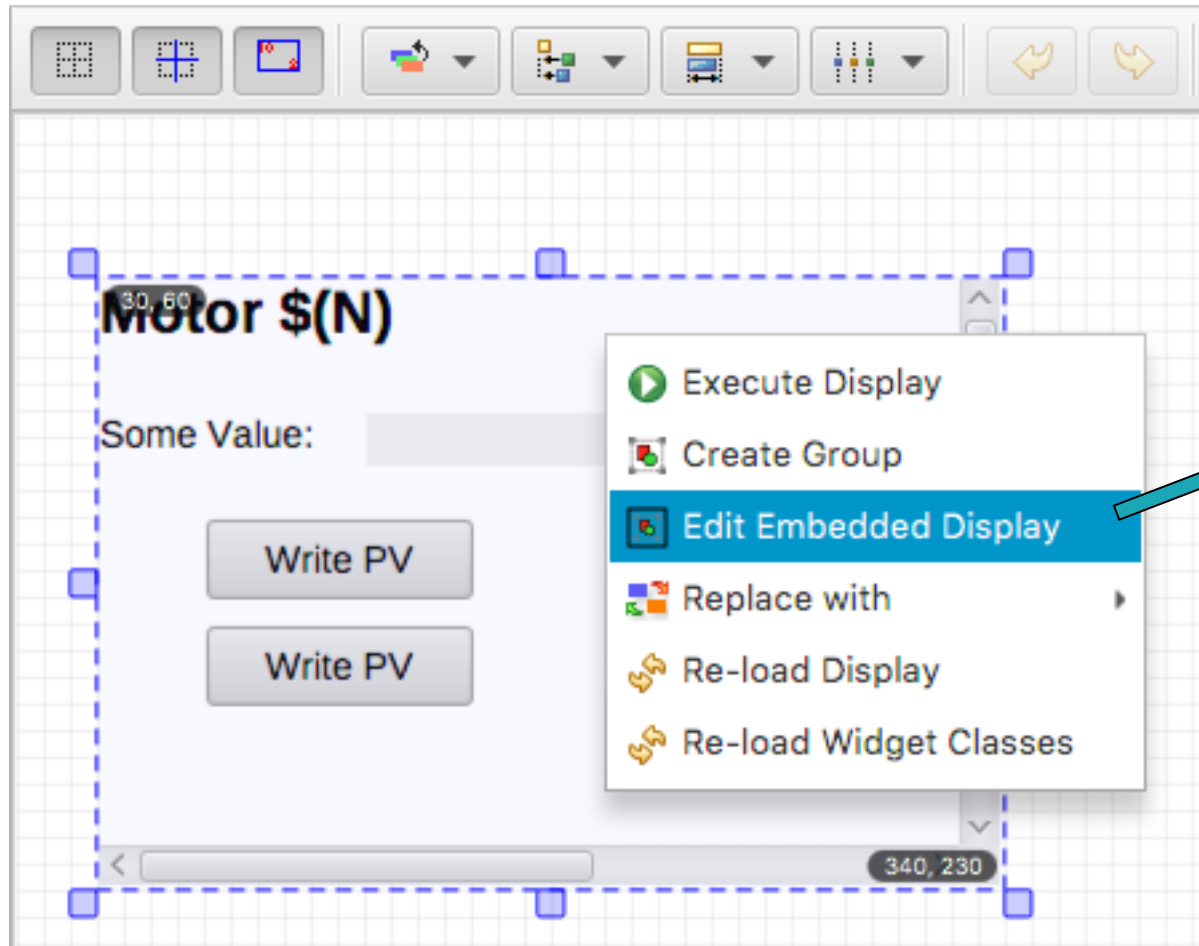


Size widget  
to fit  
content



- ✓ No Resize usually best. Scrollbars appear as needed.
- Resizing results in odd font sizes or widgets that outgrow their initial space.

# Embedded Display Editing



**TO BE  
CONTINUED...** 

# Top Resources

## See Help, Preferences Listing

```
# -----  
# Package org.phoebus.ui  
# -----
```

```
# Top resources to show in "File" menu and toolbar  
#  
# Format:  
# uri1 | uri2,Display name 2 | uri3,Display name 3  
top_resources=examples:/01_main.bob?app=display_runtime,Example Display | pv://?sim://sine&app=probe,Probe Example | pv://?sim://sine&loc://x(10)&app=pv_table,PV Table Example | http://www.google.com?app=web, Google
```

## Start phoebus with “-settings /path/to/my\_settings.ini”:

```
org.phoebus.ui/top_resources=/home/controls/displays/main.bob, Start Page |  
http://controls.my.site/displays/main.bob, Start Page
```

- File system: Use NFS or ‘git pull’ to distribute files
- http: All users always see the same set of files

# Many Widgets and Properties

Compared to earlier EPICS display tools,

- **Group Widget** instead of Lines
- **LED** instead of Circle-with-changing-color
- **Tab/Navigation Tabs** instead of buttons, local PVs, conditional visibility,..

Display describes **Meaning**:

- Group of related widgets
- LED for binary PV, not circle that happens to change color

*Files with meaning are easier to translate into the next tool*

# Widget Classes

- Instead of creating a Label with large font, define a “TITLE” class for the Label
- Instead of creating an LED with Orange color, define a “WARNING” LED class

```
# -----  
# Package org.csstudio.display.builder.model  
# -----  
  
# Widget classes  
# One or more *.bcf files, separated by ';'   
# Defaults to built-in copy of examples/classes.bcf  
class_files=examples:classes.bcf
```

# Editing \*.bcf Widget Class Files

*Slightly different editor behavior*

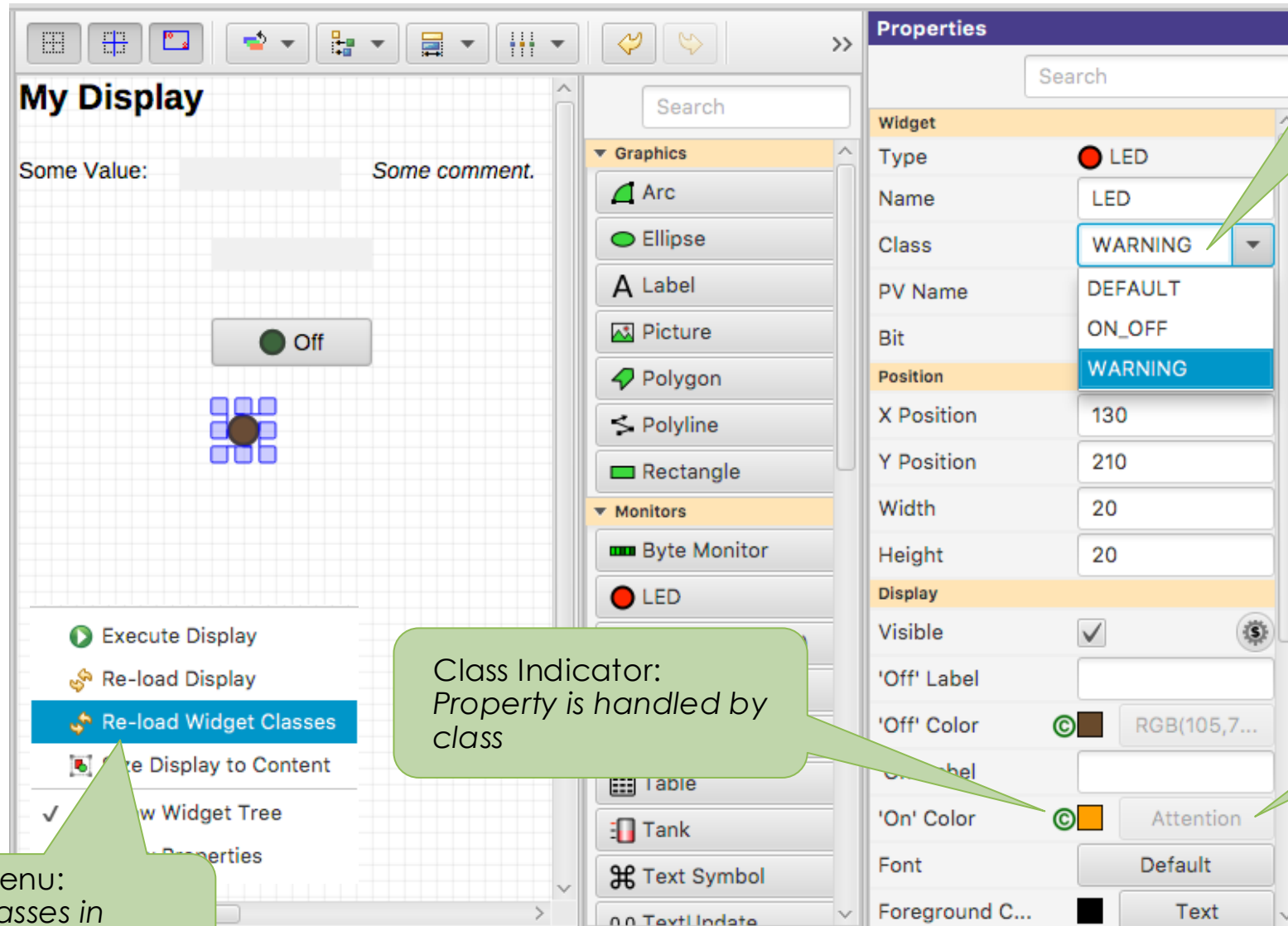
**Name** Defines a widget Class:  
'WARNING' LED,  
'TITLE' Label,  
...

The screenshot shows a software interface for editing widget class files. The interface is divided into several sections:

- Widgets List:** A list of widget classes on the left, including TITLE, COMMENT, ON\_OFF, WARNING (selected), SECTION, and several unlabeled 'A' entries.
- Grid Editor:** A central workspace with a grid. It contains text labels for 'TITLE', 'SECTION', 'COMMENT', 'ON\_OFF', and 'WARNING'. The 'WARNING' widget is visually represented by a 3x3 grid of blue squares with a red circle in the center.
- Toolbar:** A toolbar with various drawing tools like Arc, Ellipse, Label, Picture, Polygon, Polyline, Rectangle, Byte Monitor, LED, LED (Multi State), Progress Bar, Symbol, Table, and Tank.
- Properties Panel:** A panel on the right showing the properties of the selected 'WARNING' widget. The 'Name' is 'WARNING'. The 'Type' is 'LED'. The 'On' color is set to 'Attention' (orange) and is checked. A callout points to this checked property with the text: "Checked Property: Value becomes part of class definition".

**Checked Property:**  
Value becomes part  
of class definition

# Using Widget Classes



Select Widget Class

Class Indicator:  
*Property is handled by class*

Disabled:  
*Cannot change the class-based property*

Context Menu:  
*Re-load classes in case \*.bcf is changed while editing display*

# Class Details

- \*.bcf files define widget classes
  - Label of class *TITLE* uses font XYZ
- When editing a \*.bob file, classes are applied.  
Add Label, select Class *TITLE*:
  - Font is set to XYZ
  - Can no longer change the font
  - File is saved with font=XYZ, marked as “use\_class”
- \*.bob files use widget classes, if they are defined.  
Open a file with Label of class *TITLE*, and
  - a) *TITLE* is a known class:  
Whatever that class defines is used. If it sets font=EFG, that'll be used.
  - b) *TITLE* is not a known class:  
Using font=XYZ as saved in file.

# Compare \*.bcf and \*.bob to \*.css and \*.html

\*.bcf classes are similar to \*.css style settings,  
\*.bob files are similar to \*.html content

a) Have same \*.bcf/\*.css

→ Display looks the same

b) Use different \*.bcf/\*.css

→ Display looks as requested in my \*.bcf/\*.css

c) Have no \*.bcf/\*.css

→ \*.html turns into rubbish, lacking any description of what to look like.

\*.bob display looks as seen by last person who edited it,  
since the class settings effective at that time are in the \*.bob file.

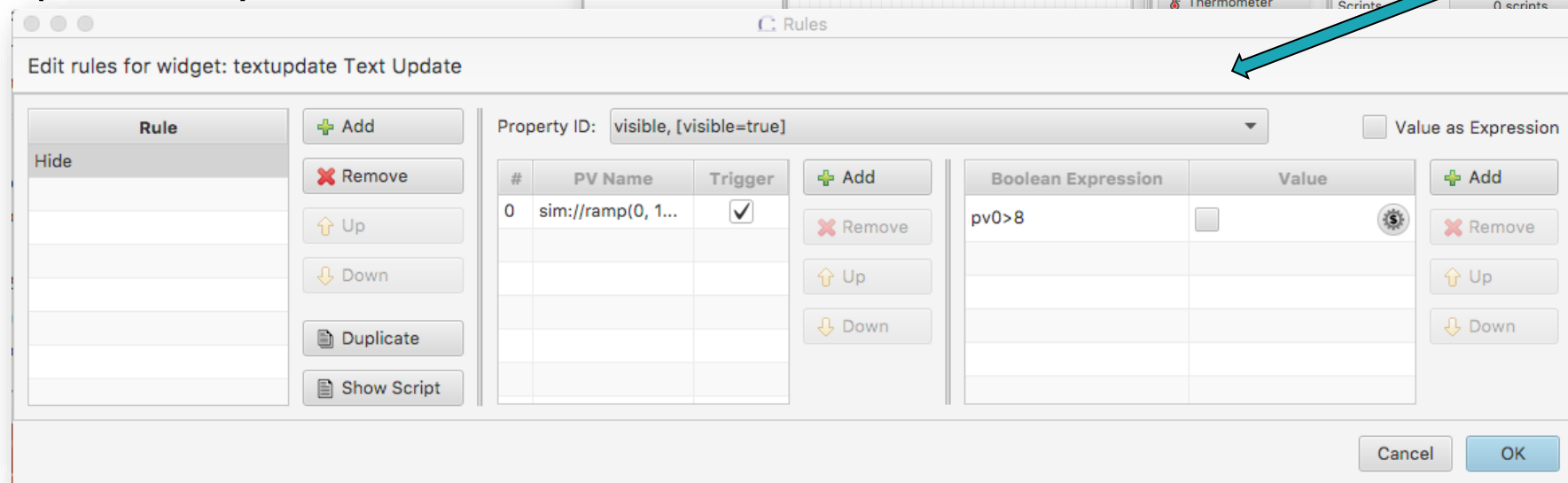
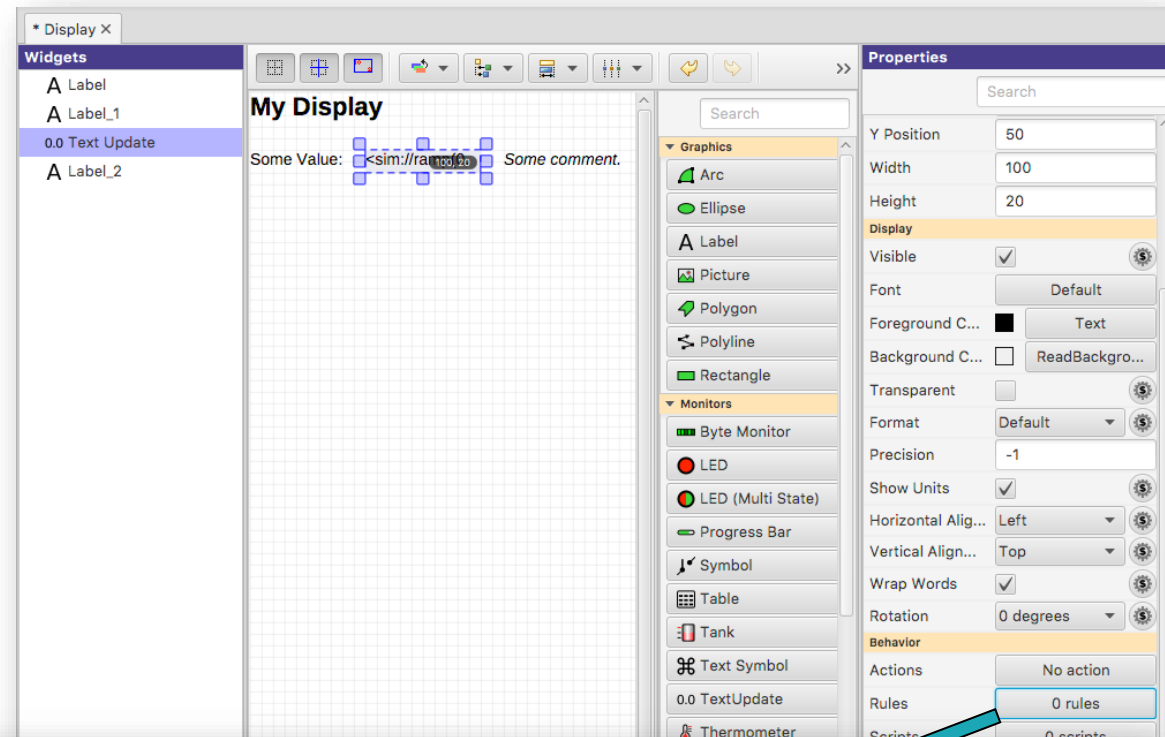
# Rules

- Ideally, use widgets' built-in functionality
  - Value of PV displayed in TextUpdate, LED, ..
  - Alarm indicated via Border
- Sometimes useful to for example hide a widget, i.e. change visibility based on a PV
  - Rules can accomplish this
  - .. But functionality may not be obvious to the next person who needs to maintain a display



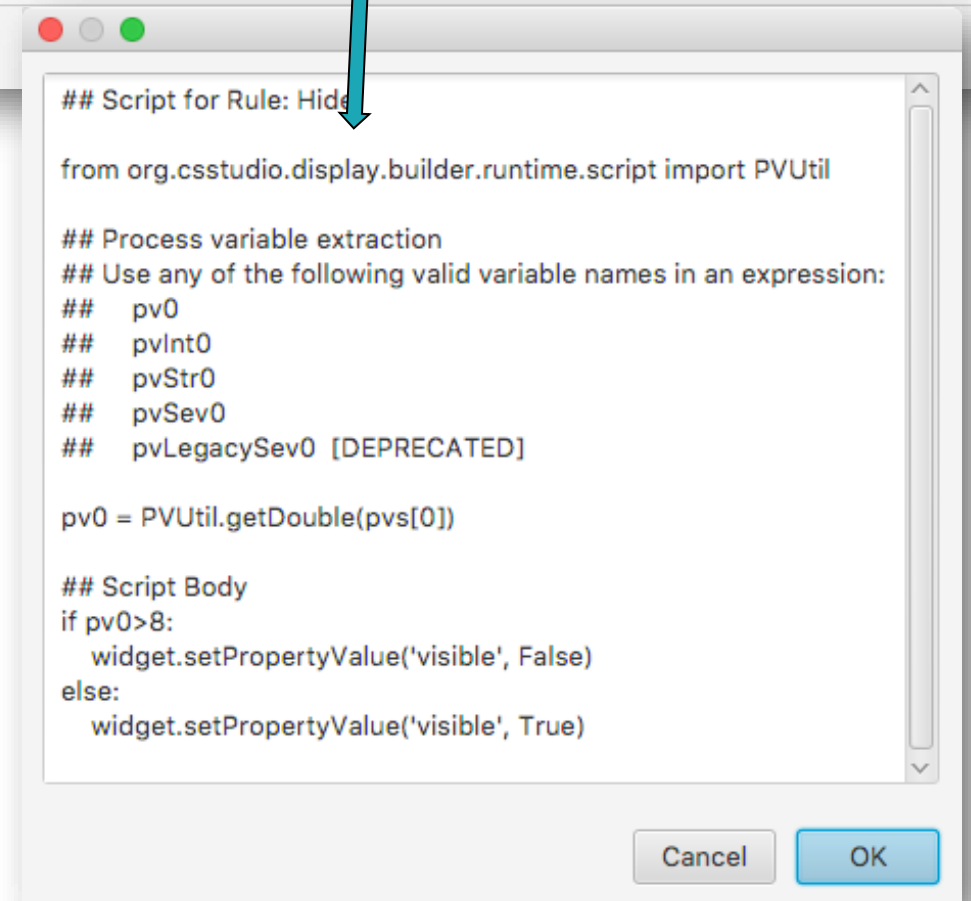
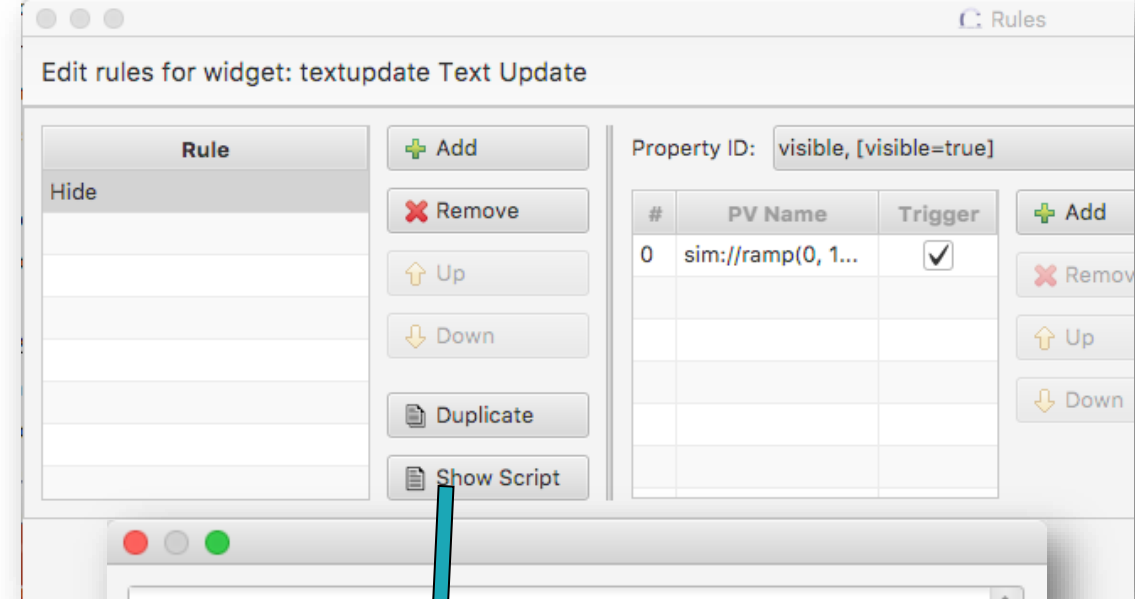
# Adding a Rule

- Add TextUpdate widget
- Set PV to `sim://ramp(0, 10, 1)`
- Open Widget's Rules
- Add Rule, name it "Hide"
- Select "visible" property
- Add PV `sim://ramp(0, 10, 1)`
- Add Boolean Expression  
"pv0>8"
- Un-check value
- Run



# Rules Detail

- Triggered by at least one PV
  - May use additional non-trigger PVs
- Expressions use pv0, pv1, ..., pvStr0, pvStr1, .. to access PVs' values
- Rule internally converted to Jython
  - Use preview to debug
- “else: ..” sets property to original value



# Scripts

- Scripts are attached to a widget
- Triggered by at least one PV
  - May use additional non-trigger PVs
- Invoked with
  - pvs[]      – Array of requested PVs
  - widget    – The widget
- Script can
  - Read & write the received PVs
  - Set widget properties
  - Locate other widgets in the display
  - Invoke any Java code in the product
  - Be very powerful
  - Result in an unmaintainable mess
- One Script Executor per \*.bob file, Runs in background thread
  - Slow scripts do not block the UI
  - One script per display at a time
    - a) Many short-duration scripts
    - b) One that never quits

# Rules vs. Scripts

- Both are in the end Jython code
- Both should be the exception.  
Plain displays don't need them.  
But can be powerful,  
replacing separate custom Java/Python/C/C++ applications.
- Prefer Rules because they describe meaning, easier to maintain

# When to use a script

- It's simple, well documented, and tremendously improves the UI
- Would be a one-of, specialized, hard to maintain, separate application anyway.  
With a script, at least its integrated into the operator UI

## Examples:

- Turn scalar PVs into `loc://waveform` for guideline in XYPlot
- Fill display with 50 widgets based on config file, `examples/template_and_script`
- Add information from web service to display

# When not to use a script

- It adds logic to the display that should be on the IOC
  - Display should only display PVs and allow user to write PVs.
  - Display must never do anything
- You have to ask for help implementing the script
  - If you can't implement it, you can't maintain it, either

## Examples

- Open relieve valve when pressure too high.  
Ramp Power Supply.
  - What if somebody closes the display? Opens two displays?
- Wiggle something on the display
  - It's not a video game



# Summary

Display Builder is powerful Editor and Runtime with many Widgets, Macros etc.

Keep it Simple

1. Add a Widget
2. Enter Label's Text or Widget's PV Name
3. Done

**Sample & Detector**

	Destination Pos	Current Pos		
SANGLE	0.5000 deg	0.5012 deg	●	Scan
SampleX	-8.9618	-8.9618	●	Scan
Beam Stop	0.0362 mm	0.0362 mm	●	Scan
Sample Changer	Undefined	-87.0008 mm	●	Scan
DANGLE	13.0000 deg	13.0015 deg	●	Scan

**Slits - Collimation**

	Destination Pos	Current Pos			Destination Pos	Current Pos			
S1HWidth	0.500 mm	0.501 mm	●	Scan	S1VHeight	30.000 mm	29.998 mm	●	Scan
S2HWidth	3.000 mm	2.996 mm	●	Scan	S2VHeight	30.000 mm	30.010 mm	●	Scan
S3HWidth	0.500 mm	0.500 mm	●	Scan	S3VHeight	40.000 mm	40.000 mm	●	Scan

**Slits - Background**

	Destination Pos	Current Pos		
RSlit4	-58.5160 mm	-58.5165 mm	●	Scan
BDetSlit	0.0438 mm	0.0455 mm	●	Scan
RDetSlit	-4.9926 mm	-4.9927 mm	●	Scan

**Pump Maintenance (RP)**

	Current Running (Hours)	Alarm Setpoint	Maintenance Require if RED
RP01	25	8000	●
RP02	25	8000	●
RP03	25	8000	●
RP04	26	8000	●
RP05	128	8000	●
RP06	127	8000	●
RP07	3244	8000	●
RP08	0	8000	●
RP09	3245	8000	●
RP10	3245	8000	●

**Pump Maintenance (TMP)**

	Current Running (Hours)	Alarm Setpoint	Maintenance Require if RED
TMP01	2204	8000	●
TMP02	26	8000	●
TMP03	26	8000	●
TMP04	25	8000	●

**X1Y ROI**

	Min	Max	Mean	Total	Total +/-	Rate
Signal	11	363	116.194	22774	150.911	0 e/s
Background	0	1425	71.161	160824	401.029	0 e/s
S/B	0.000	0.255	1.633	0.142	0.001	0.000

**QIE ROI**

	Min	Max	Mean	Total	Total +/-	Rate
Signal	44	8996	778.156	105051	324.116	0 e/s
Background	6	2090	182.778	24675	157.083	0 e/s
S/B	7.333	4.304	4.257	4.257	0.030	0.000

**Data Collection**

Total Counts	4890418	0 e/s
Proton Charge	4.0063559E-1 C	
Beam Power	1402537 Watts	
Data Collection State	Idle	
Data Collection Pause	Not Paused	

**QIE Axes & ROI Position Details**

	ROI Start	ROI Size	Start	End	Bin Size	
Q Signal	4.268	1.057	Q Axis	0.0000	13.9396	0.0697
E Signal	0.000	7.500	E Axis	-80.0000	80.0000	0.8000
Q Background	2.134	1.057				
E Background	15.000	7.500				