Display Builder Web Runtime

https://github.com/kasemir/pvws
https://github.com/kasemir/dbwr

Kay Kasemir, June 2019
General Web Display Idea

At runtime, network traffic similar to

```json
{    pv: "SomePV",    value: 3.14}
```
# EPICS Operator Interfaces

<table>
<thead>
<tr>
<th>Year</th>
<th>Interface 1</th>
<th>Interface 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>198x</td>
<td>edd/dm</td>
<td>X11, Xt</td>
</tr>
<tr>
<td>199x</td>
<td>medm, dm2k</td>
<td>Motif</td>
</tr>
<tr>
<td>200x</td>
<td>edm</td>
<td>Motif</td>
</tr>
<tr>
<td>2010</td>
<td>CS-Studio BOY</td>
<td>SWT</td>
</tr>
<tr>
<td>2017</td>
<td>CS-Studio Display Builder</td>
<td>JavaFX</td>
</tr>
</tbody>
</table>

Also: tcl/tk/ca, python/qt/ca, ..

UI libs change with technology and fashion
Web EDM
Ryan Slominski, JLab,
2017 EPICS Meeting

Very useful and performant for EDM displays and Channel Access IOCs

Limitations:
- No waveforms (plots, images)
- No metadata from non-IOC servers (LabView, Python)
- No PV Access support
PV Web Socket - [https://github.com/kasemir/pvws](https://github.com/kasemir/pvws)

- Based on Phoebus Stack
  - VType PV for loc://, sim://, ca://, pva://, ...
  - PV Pooling
  - RXJava Throttling

- Data Packaged as JSON
  - Sends metadata once
    - No separate connections to *.EGU, *.PREC, ...
  - Severity and value on change
  - Arrays packed as Base64-binary
  - JavaScript in client merges updates
Display Builder Web Runtime - https://github.com/kasemir/dbwr

Labels, LEDs, Text Updates

Groups, Embedded displays, Macros
'Static’ Widgets

Lines, Circles, Rectangles, ..

Limited ‘Rule’ Support:
Some colors, Hide/Show
Line and Detector plots

Summary
- Detector ROI: 2034565
- Beam Power: 145047 Watts
- Time-Of-Flight: 170937
- Beam Q: 1439369
- Beam Q: 1817 mks
- BM1 Counts: 422843
- BM2 Counts: 4132 mks
- Progress Bar: 100%
CS-Studio on Desktop vs. Web Runtime

- **Integrated Product**
- **Robust Development**
  - Type checking
  - Single-Language

- **Just Display Runtime**
- **Fragile Development**
  - No type checking
  - Client: HTML, CSS, JS
  - Server: Java, Python, ...
  - Different Web Browsers

Still:
Read-only web view of control system is extremely convenient and useful!
Status: New Project, but already very useful

- Label
- Rectangle
- Ellipse
- Arc
- Polyline
- Polygon
- Text Update
- Text Input
- Text formatting (precision, units, enum labels)
- LED
- Multi-State LED

- Action Button to open display or web link
- Combo
- Group with group border
- Embedded Displays
- Tabs
- XYPlot
- Image
- Macro support
- Alarm-sensitive border based on PV
- Limited Rule support: Color of rect/circle/label, visibility
- Caching
Summary

Display Builder Web Runtime offers web access to much of the Desktop version

https://github.com/kasemir/pvws
https://github.com/kasemir/dbwr

1. git clone
2. ant
3. copy *.war to Tomcat
4. Set environment: EPICS_CA_ADDR_LIST, ..

→ http://your_tomcat/dbwr
Limited Compatibility

**MEDM**

```plaintext
Text {
  object {
    x = 265
    y = 440
    width = 155
    height = 20
  }
  "basic attribute" {
    clr = 54
  }
  textix = "Hello"
}
```

**EDM**

```plaintext
object activeXTextClass
beginObjectProperties
x 265
y 440
w 155
h 20
font "arial-bold-r-10.0"
fgColor index 5
value {
  "Hello"
} endObjectProperties
```

**Qt Designer**

```xml
<widget class="caLabel" name="caLabel_0">
  <property name="foreground">
    <color alpha="255">
      <red>10</red>
      <green>0</green>
      <blue>184</blue>
    </color>
  </property>
  <property name="text">
    <string>Hello</string>
  </property>
  <property name="geometry">
    <rect>
      <x>265</x>
      <y>440</y>
      <width>155</width>
      <height>20</height>
    </rect>
  </property>
</widget>
```

Dump of tool's in-memory "model".

Why not just `<text>Hello`?
Simple, General Format

Display Builder

```xml
<widget type="label">
  <x>268</x>
  <y>440</y>
  <width>155</width>
  <height>20</height>
  <text>Hello</text>
  <foreground_color>
    <color red="0" green="0" blue="0"/>
  </foreground_color>
</widget>
```

- **XML**
  - Somewhat human-readable
  - Parsers for every programming language

- “label”, “text_update”
  - Not hinting at specific implementation