

Control System Studio Training - EDM Converter

Kay Kasemir

ORNL/SNS

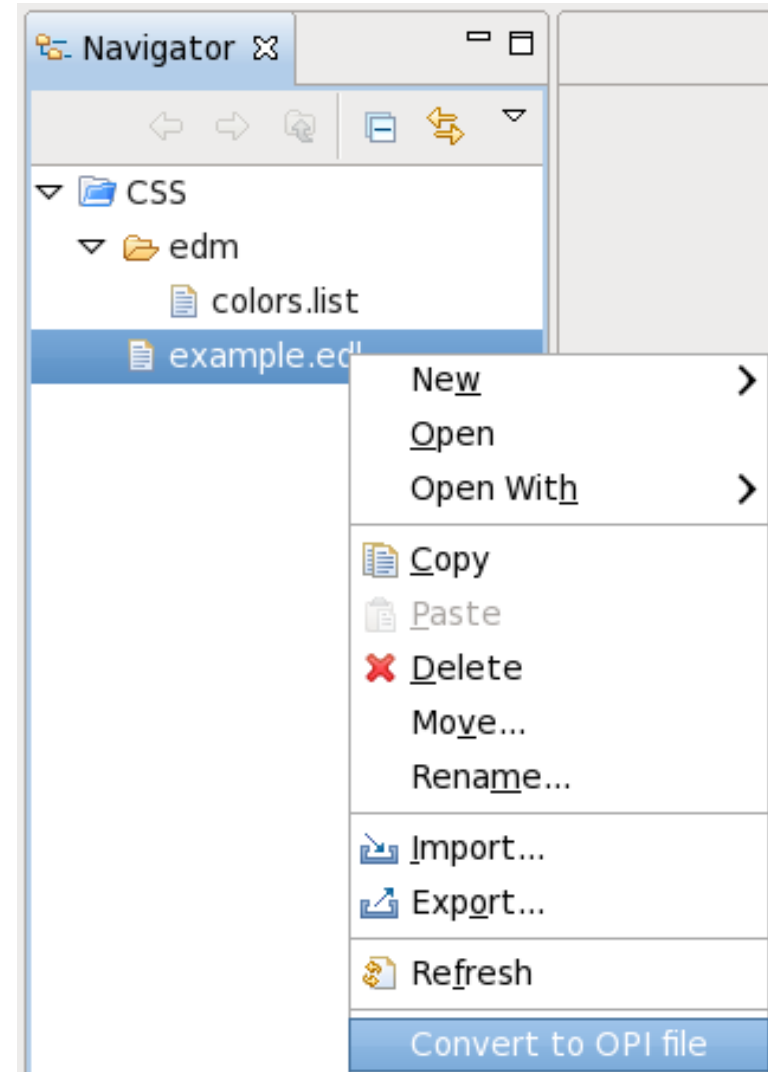
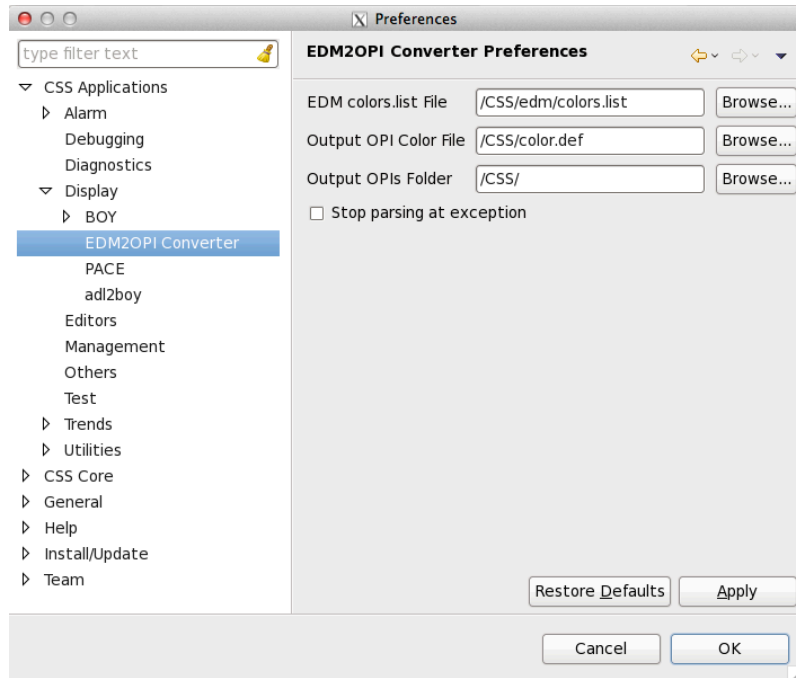
kasemirk@ornl.gov

Jan. 2013

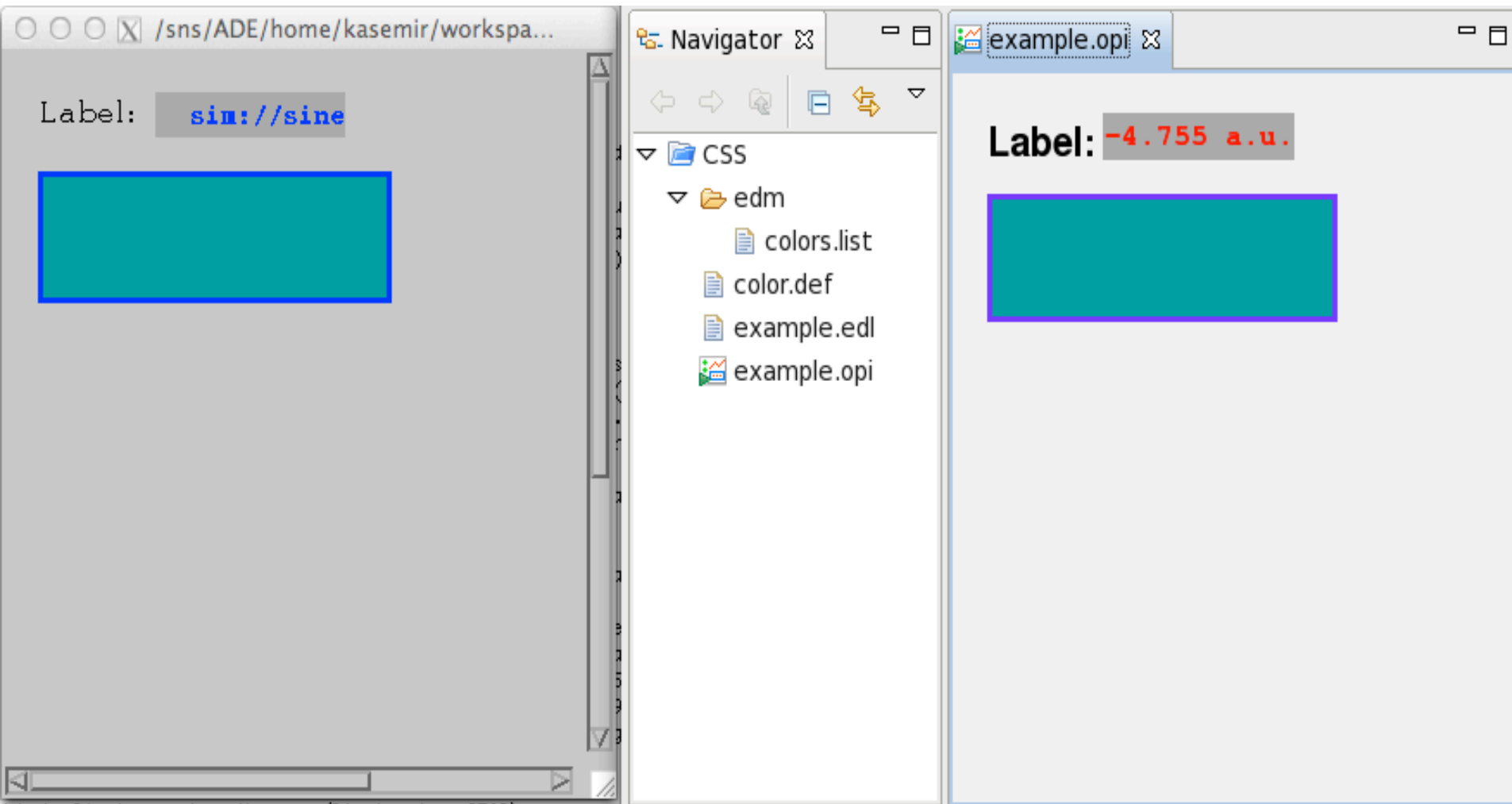
So far...

- **John Hammonds (APS):
MEDM-to-BOY (*.adl to *.opi) converter**
 - Somewhat usable.
‘Asyn’ for example includes auto-converted displays.
- **SNS contract w/ Cosylab:
EDM-to-BOY (*.edl to *.opi) framework**
 - Only includes widget mapping for Label, TextUpdate, Rectangle.

Usage



Result



Fundamentally, Easy to Extend

The screenshot displays an IDE with the Package Explorer on the left and two code editors on the right. The Package Explorer shows the project structure for `org.csstudio.opibuilder.converter`, with the `src/main` directory expanded to show the `org.csstudio.opibuilder.converter.model` package. The `Opi_TextupdateClass.java` file is selected in the Package Explorer.

The main editor shows the `Edm_TextupdateClass.java` file. The class is defined as `public class Edm_TextupdateClass extends EdmWidget {`. It contains several attributes and methods. An orange box highlights the class declaration, and an arrow points to the `Opi_TextupdateClass` class in the adjacent editor. Another orange box highlights the `@EdmAttributeAn private String controlPv;` attribute, and an arrow points to the `new OpiString(context, "pv_name", t.getControlPv());` line in the `Opi_TextupdateClass` class.

The `Opi_TextupdateClass.java` file is also shown, which is a conversion class for the `Edm_TextupdateClass` widget. It contains the following code:

```

* Copyright (c) 2010 Oak Ridge National Laboratory.
package org.csstudio.opibuilder.converter.writer;

import org.apache.log4j.Logger;

/**
 * XML conversion class for Edm_TextupdateClass.
 * @author Matevz
 */
public class Opi_TextupdateClass extends OpiWidget {

    private static Logger log = Logger.getLogger("org.csstudio.opibuilder.converte
    private static final String typeId = "TextUpdate";
    private static final String name = "EDM Text Update";
    private static final String version = "1.0";

    /**
     * Converts the Edm_TextupdateClass to OPI TextUpdate widget XML.
     */
    public Opi_TextupdateClass(Context con, Edm_TextupdateClass t) {
        super(con, t);
        setId(typeId);

        context.getElement().setAttribute("version", version);

        new OpiString(context, "name", name);

        new OpiString(context, "pv_name", t.getControlPv());

        new OpiColor(context, "foreground_color", t.getFgColor());
        new OpiColor(context, "background_color", t.getBgColor());

        new OpiBoolean(context, "transparent", !t.isFill());

        new OpiFont(context, "font", t.getFont());
        if (t.getAttribute("fontAlign").isInitialized()){
            int align = 0;
            if(t.getFontAlign().equals("right"))
                align = 2;
            else if (t.getFontAlign().equals("center"))
                align = 1;

            new OpiInt(context, "horizontal_alignment", align);
        }

        if (t.getAttribute("lineWidth").isInitialized()) {
            new OpiInt(context, "border_width", t.getLineWidth());
            new OpiInt(context, "border_style", 1);
        }
    }
}
```

What Should be Easy

- **Add most widget types**
- **Add basic intelligence:**
 - **Replace EDM ‘visibility’ with BOY ‘Rule’**

What's Impossible

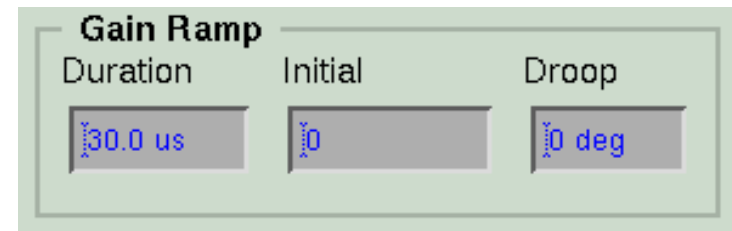
- **“Tabs” simulated by invisible Message Buttons writing to local PVs, ...**
- **Rectangles that are really meant to create a “Group”**
- **Invisible related display buttons that could now be “actions” attached to visible widgets**

Groups

‘Group’ in EDM:

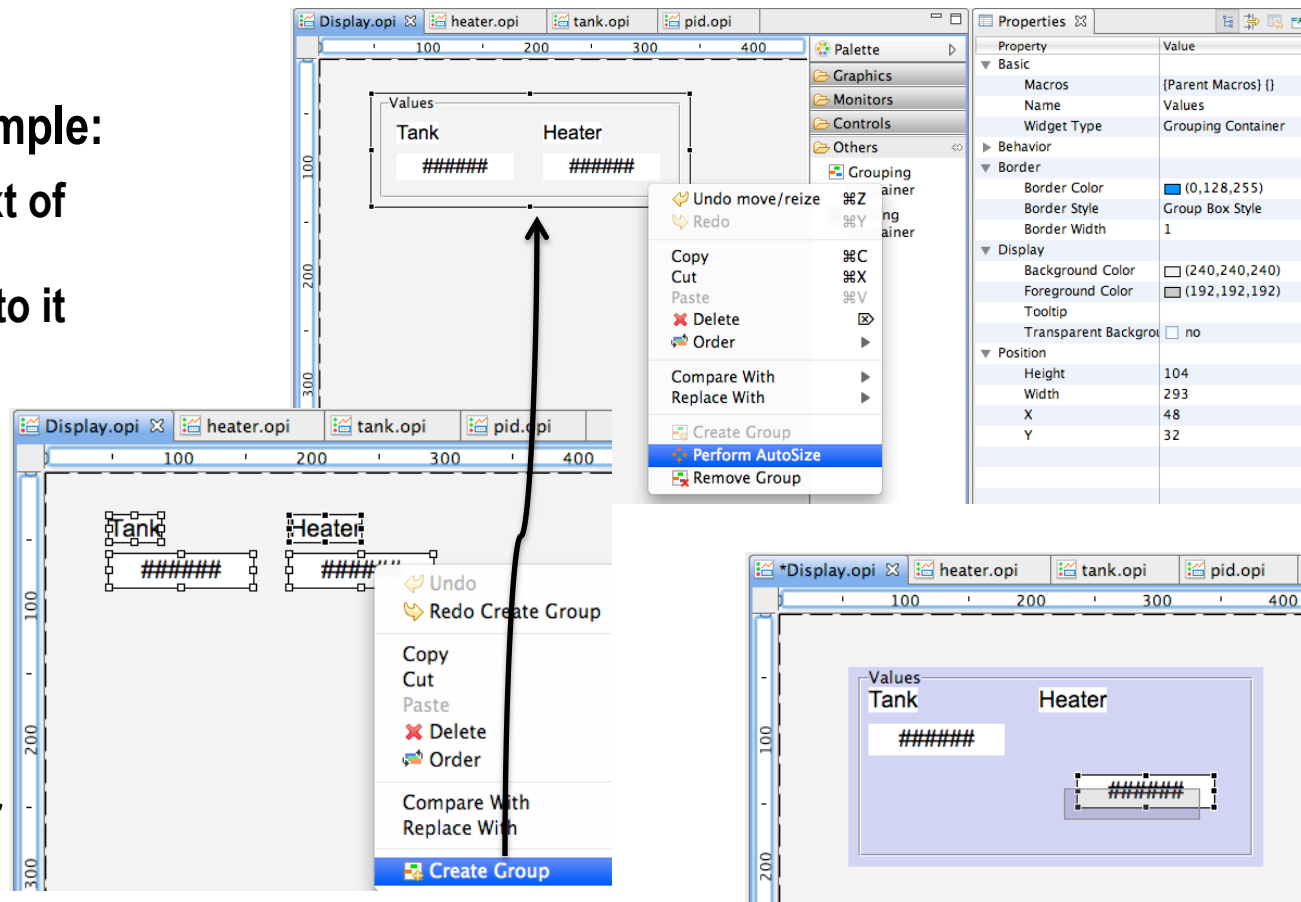
1. Rectangle around widgets
2. Label on top of Rectangle for group name

Looks OK, hard to edit/move



Grouping Container Example:

1. Create from context of selected widgets, or drop widgets into it
2. Configure name, move, (auto-)size, even remove.
3. When moving widgets, highlight indicates if they're still ‘inside’ container

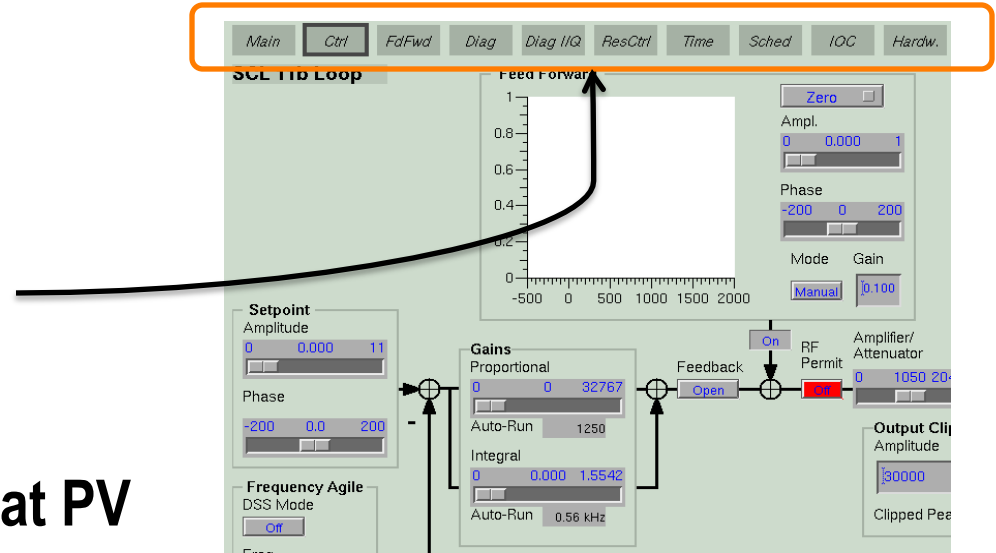


Optical feedback when moving widget in group

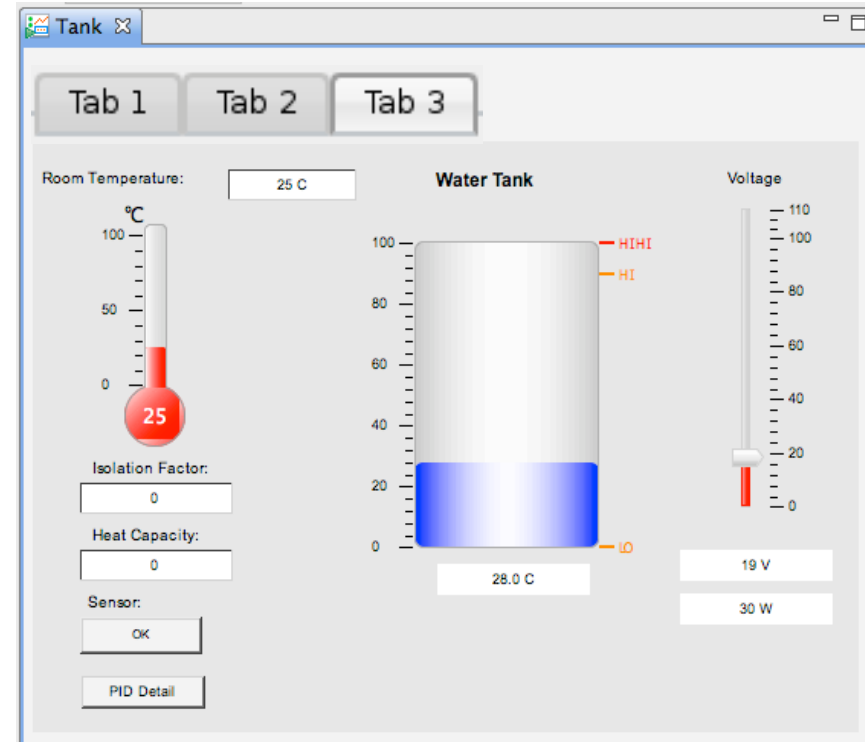
Tabs

EDM:

1. Invisible button writes to local PV
2. Rectangle etc. show/hide depending on value of that PV
3. Embedded display shows different file depending on PV



BOY: Tabbing Container



Summary

Decent skeleton, yet still just a skeleton.

Eagerly awaiting people with time to add all the widget conversion code.