

jddd

Java DOOCS Data Display

Editor for a Graphical User Interface of DOOCS & TINE
and Runtime Engine

Elke Sombrowski
Kay Rehlich

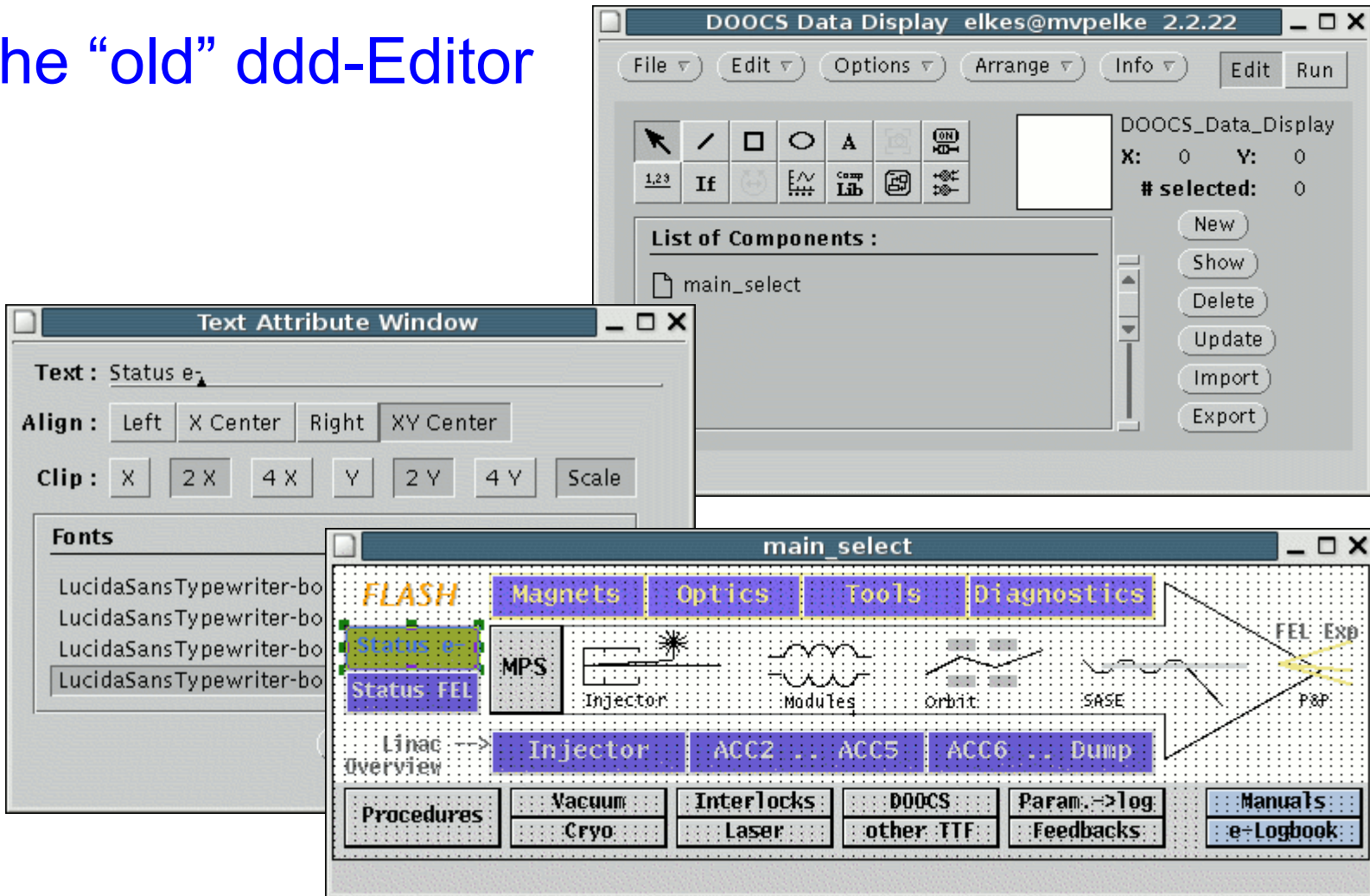


Content

- Motivation
- The “old” ddd (DOOCS Data Display)
- Considerations about what we need in future
- The new jddd (Java DOOCS Data Display)
 - The jdddEditor
 - Reusable components in jddd
 - Special features of jddd
 - How to start jddd
- Experience & Outlook

- **We don't want do design all panels in the controls group**
 - Non-programmers / subsystem experts should develop graphical controls panels
- **We already have ddd (C++) since 10 years**
 - synapical displays editor
 - but ddd needs an “update”

The “old” ddd-Editor



The screenshot displays the DOOCS Data Display editor interface. The main window, titled "DOOCS Data Display elkes@mvpelke 2.2.22", features a menu bar (File, Edit, Options, Arrange, Info) and a toolbar with various drawing and editing tools. A central canvas shows a schematic diagram of the FLASH accelerator components, including the Injector, Modules, Orbit, SASE, FEL Exp, and P&P. The diagram is overlaid with a grid and various colored boxes representing different components. A "List of Components" panel on the right lists "main_select".

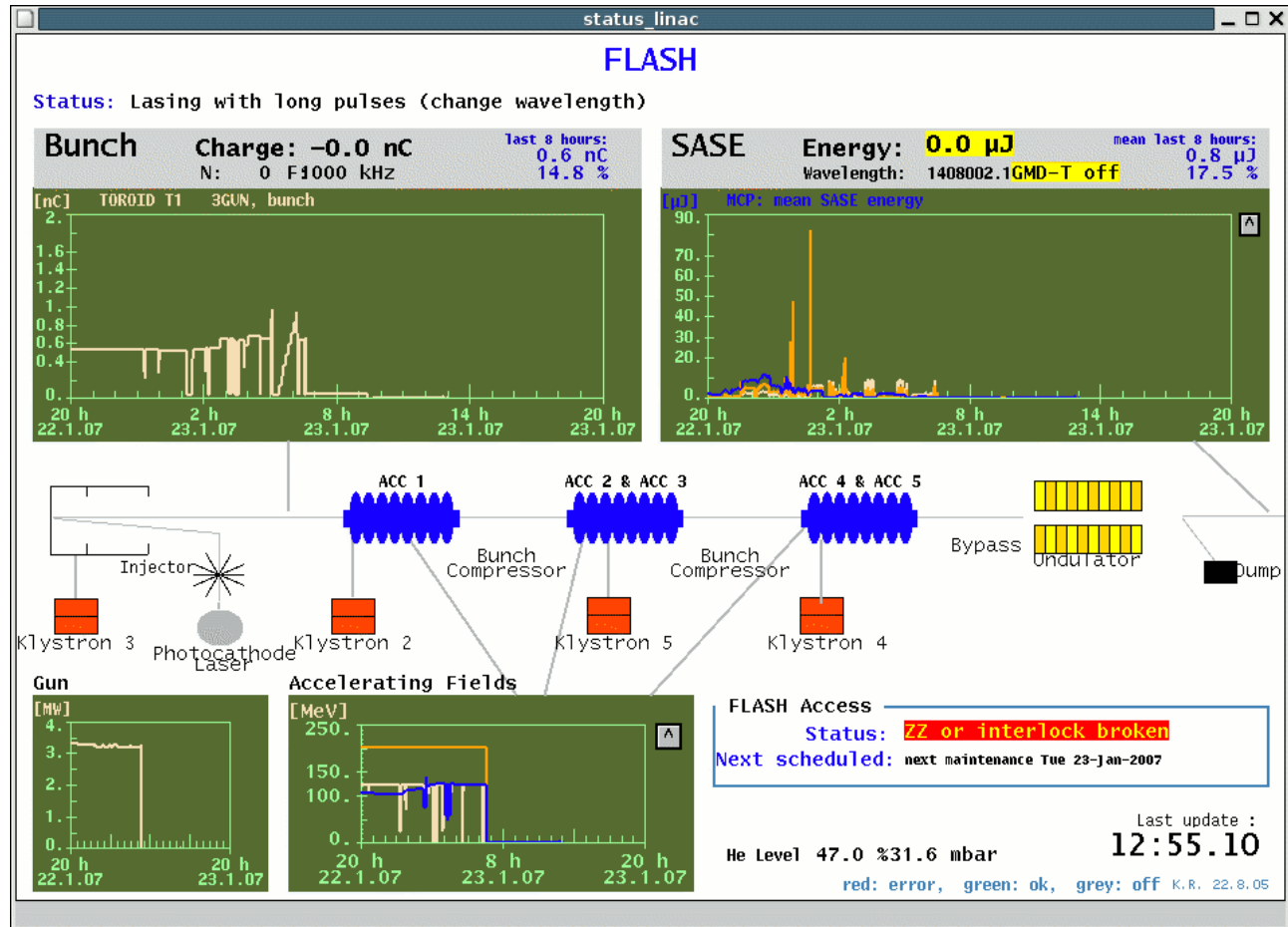
Overlaid on the main window are two other windows:

- Text Attribute Window:** This window allows for text formatting. It shows the text "Status e_" and provides options for alignment (Left, X Center, Right, XY Center), clipping (X, 2 X, 4 X, Y, 2 Y, 4 Y), and scaling (Scale).
- Fonts:** A list of fonts is displayed, including "LucidaSansTypewriter-bo" and "LucidaSansTypewriter-bo".

The main window also includes a "main_select" window showing a detailed schematic of the accelerator components, including the Injector, ACC2, ACC5, ACC6, and Dump. The schematic is overlaid with a grid and various colored boxes representing different components.

ddd Screenshot

Flash: ~ 1300 control panels



UNIX / C++ version:

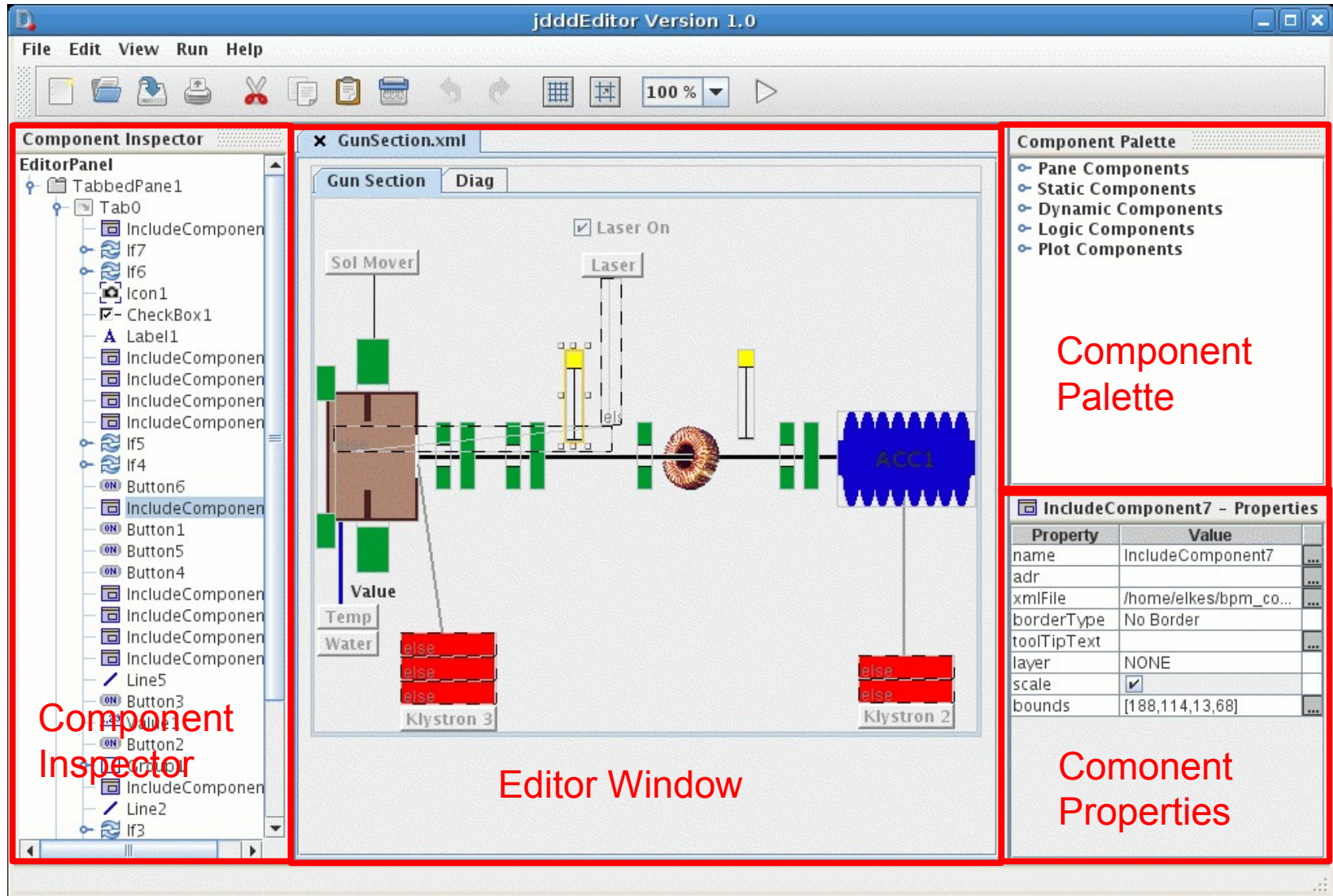
- + Editor: simple creation of control panels with a set of standard components: text, buttons, graphical components, values, dials, plots, “If”, animated components
- + Save control panels in text-files, ComponentASCIIFiles-files are parsed at runtime
- Some new editor features are difficult to implement (e.g. undo functionality)
- Open look library is no longer supported

What we need in Future:

- Keep the good things of ddd: simple Editor
- Editor with modern look&feel and functionality
- Improved components (plots with math functions, ...)
- New components (TabbedPane, ...)
- Platform independence
- More flexible usage:
 - Reuse of displays in other applications
 - Web interface (Applet)
- Convertibility of the old CAF-files to a new format

jddd implementation:

- Use JAVA
- Standard JAVA technology (Swing, Java beans)
- jddd editor with standard functionality, options:
 - use Eclipse or Netbeans and write Plugins
 - develop a GUI Editor based on the Netbeans visual library
- **we decided to write a completely new editor, because we want to be independent of external libraries and we want to have the highest flexibility**
- Save panels in standard xml file format



The screenshot shows the jddd Editor Version 1.0 interface. The main window displays a beamline diagram with components like Sol Mover, Laser, Klystron 3, and ACC1. The interface is divided into several panels:

- Component Inspector:** Located on the left, it shows a tree view of the current document's structure, including 'TabbedPane1', 'Tab0', and various 'IncludeComponent' and 'Button' elements.
- Editor Window:** The central area where the beamline diagram is edited. It includes a toolbar at the top with icons for file operations and a zoom level of 100%.
- Component Palette:** Located on the right, it lists categories of components: Pane Components, Static Components, Dynamic Components, Logic Components, and Plot Components.
- Component Properties:** Below the palette, a table shows the properties for 'IncludeComponent7':

Property	Value
name	IncludeComponent7
adr	
xmlFile	/home/elkes/bpm_co...
borderType	No Border
toolTipText	
layer	NONE
scale	<input checked="" type="checkbox"/>
bounds	[188,114,13,68]

Component Inspector

Component Palette

Editor Window

Component Properties

Pane Components:

- **TabbedPane**
- LayeredPane
- IncludeComponent

Static Components:

- Label
- Line
- Oval
- Rectangle
- Triangle
- Icon

Dynamic Components:

- Button
- Value
- Dial
- **CheckBox**
- **ProgressBar**
- **Slider**
- StatusRegister
- **Audio**
- ColouredIndicator
- **LocationChooser**
- CameraImage
- **TextArea**

Logic Components:

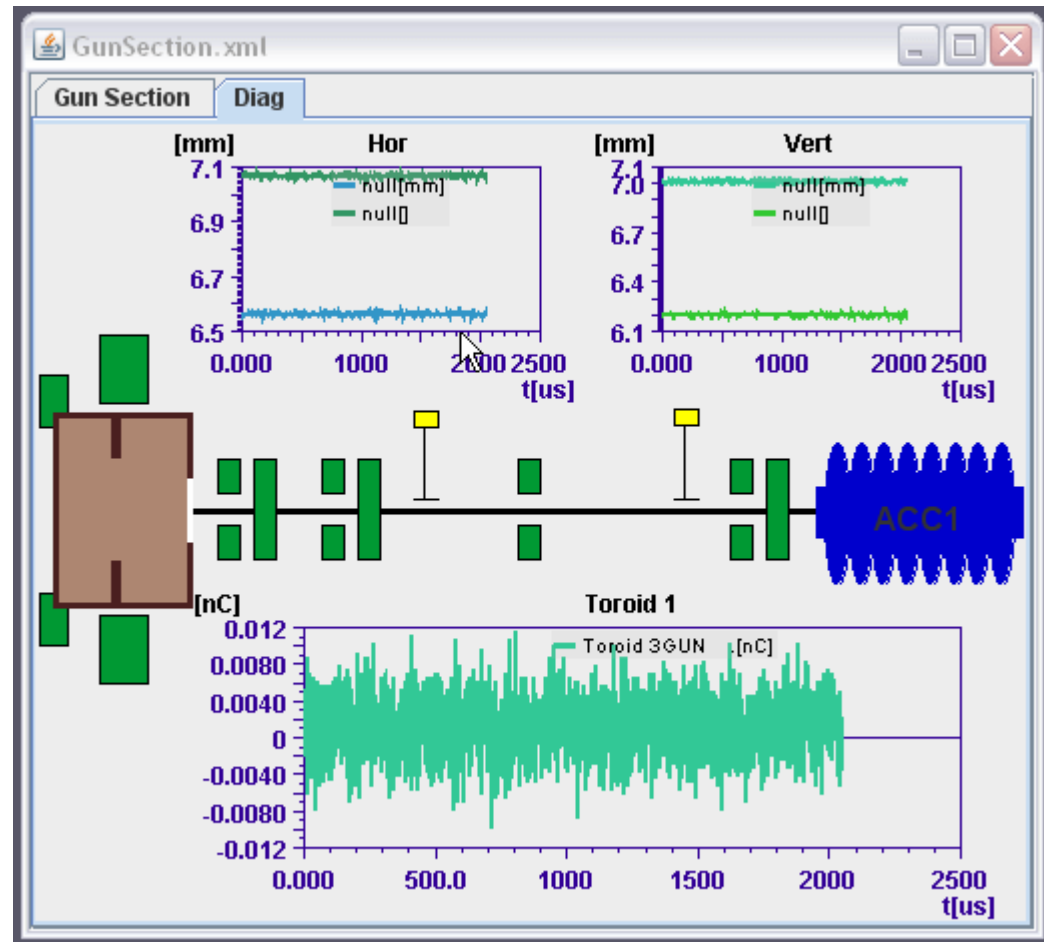
- If
- **Switch**

Plot Component:

- PlotSpectrum

New Components

In run mode



jdddEditor Version 1.0

File Edit View Run Help

Component Inspector

orPanel

- TabbedPane1
 - Tab0
 - IncludeComponent1
 - If7
 - X != 0
 - else
 - Line1
 - If6
 - Icon1
 - CheckBox1
 - Label1
 - IncludeComponent1
 - IncludeComponent1
 - IncludeComponent5
 - IncludeComponent8
 - If5
 - If4
 - Button6
 - IncludeComponent7
 - Button1
 - Button5
 - Button4
 - IncludeComponent6
 - IncludeComponent5
 - IncludeComponent4
 - IncludeComponent3

GunSection.xml

Gun Section | Diag

Laser On

Sol Mover

Value

Temp

Water

else

else

else

Klystron 3

Laser

else

Klystron 2

ACC1

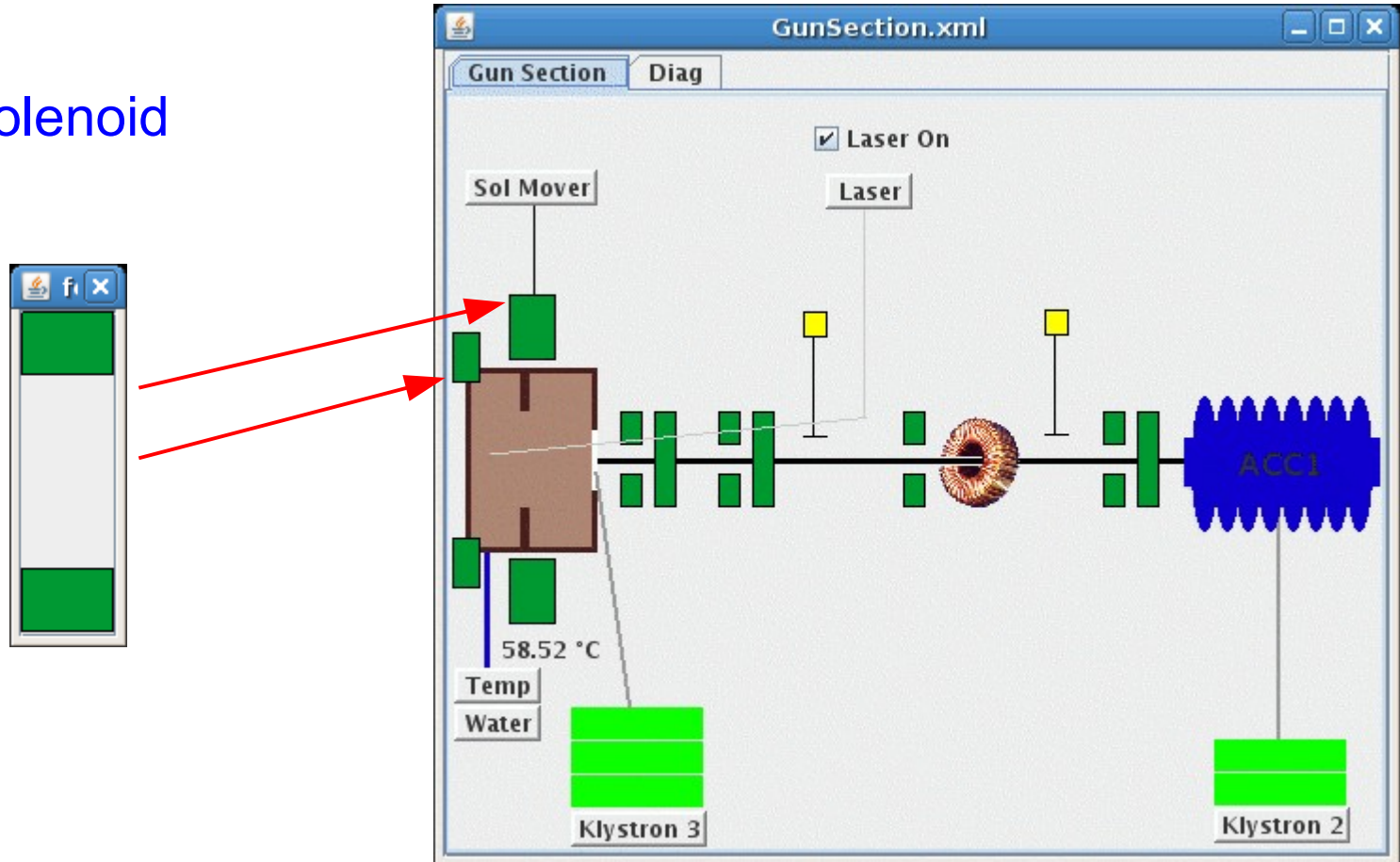
Component Palette

- Pane Components
 - LayeredPane
 - TabbedPane
 - Tab
 - IncludeComponent
- Static Components
 - Label
 - Line
 - Oval
 - Rectangle
 - Triangle
 - Icon
- Dynamic Components
 - Button
 - Value

If7 - Properties

Property	Value
name	If7
adr	TTF2.UTIL/LASER/...
borderTy...	No Border
toolTipT...	
layer	NONE
scale	<input checked="" type="checkbox"/>
dataindex	0
bitMask	0
useBitM...	<input type="checkbox"/>
updateTi...	1.0

Solenoid



Use the editor to create library components to be used in multiple panels

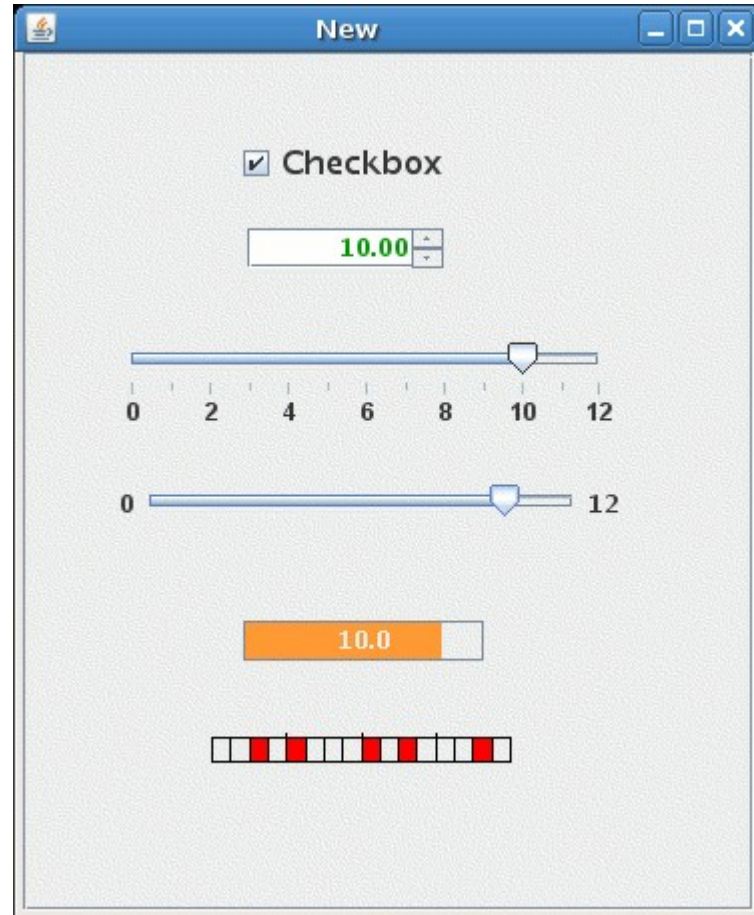
Checkbox

Dial

Slider

ProgressBar

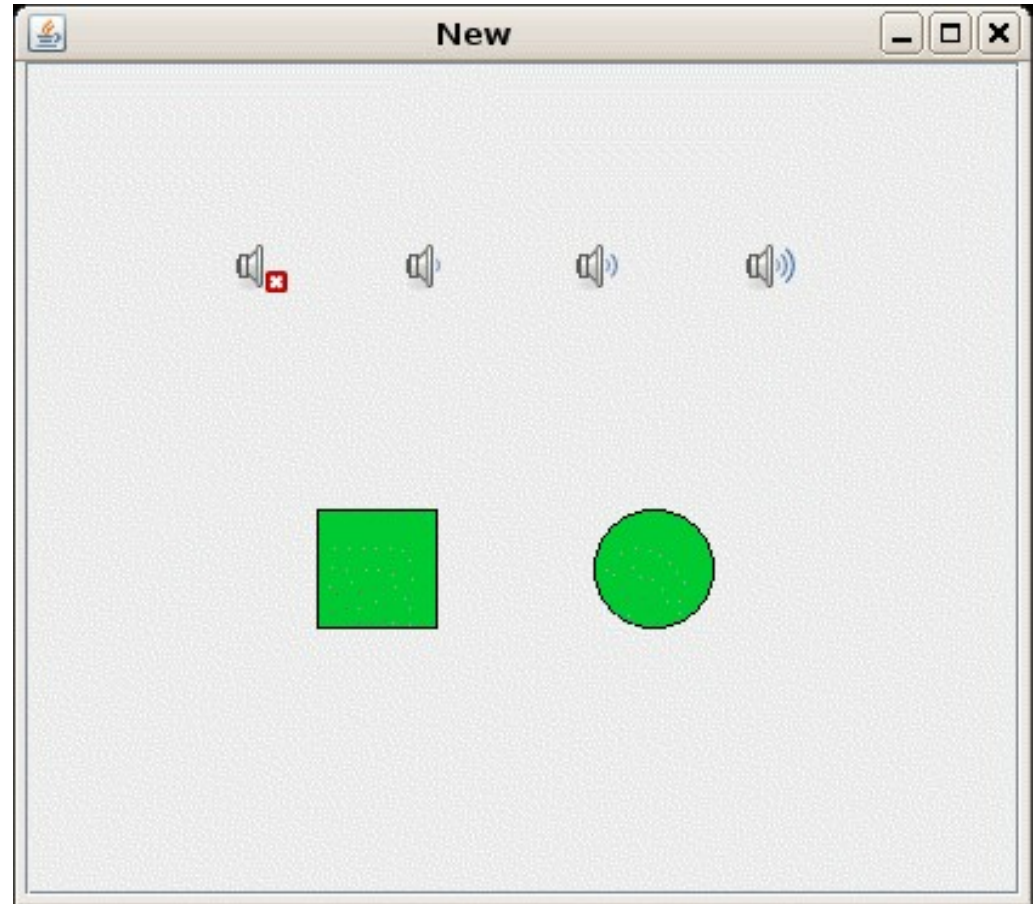
StatusRegister



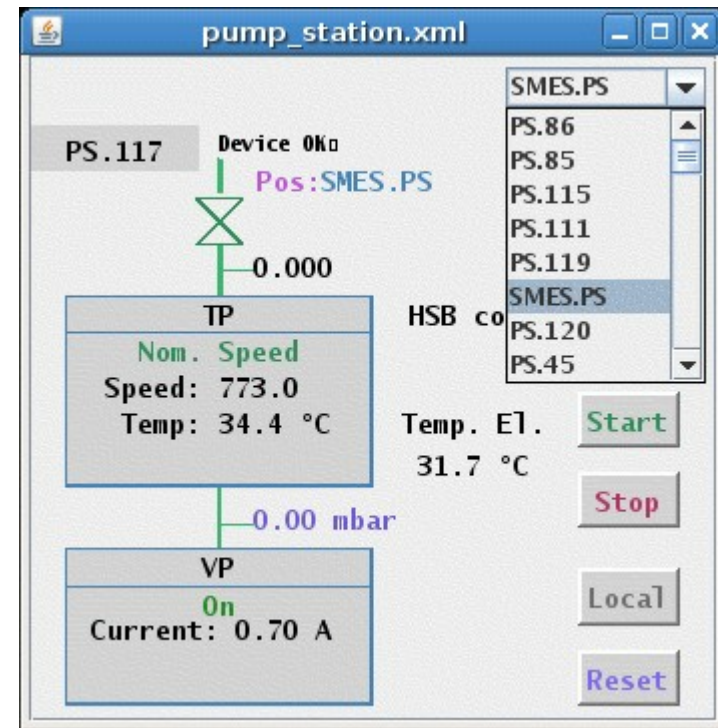
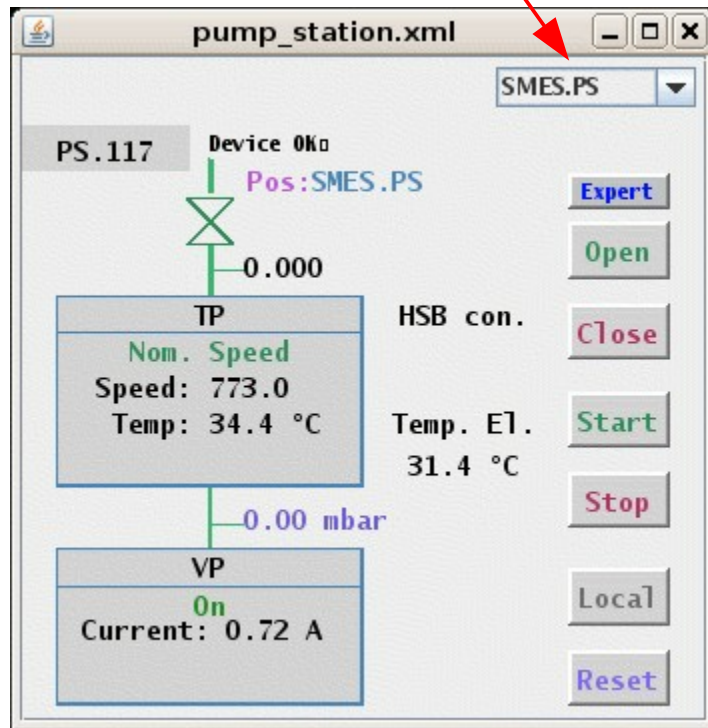
Audio Component

- plays .wav file
- plays dynamic beep
f(variable in contr. sys)

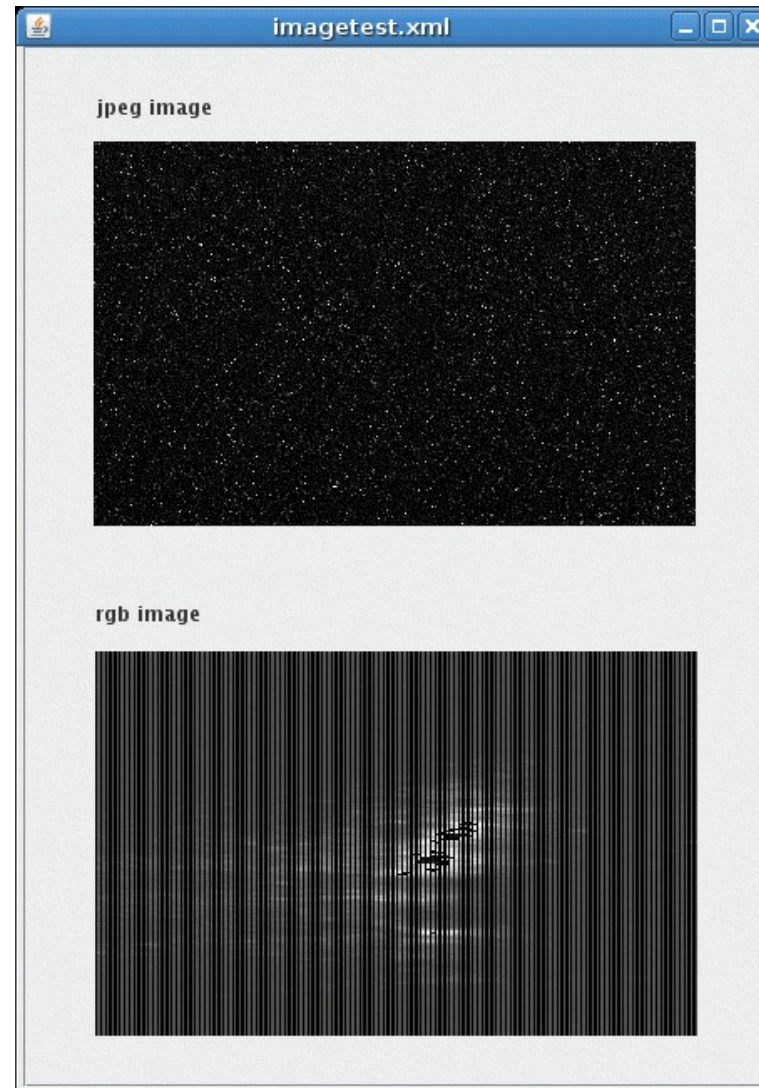
ColouredIndicator



LocationChooser



CameraImage



TextArea

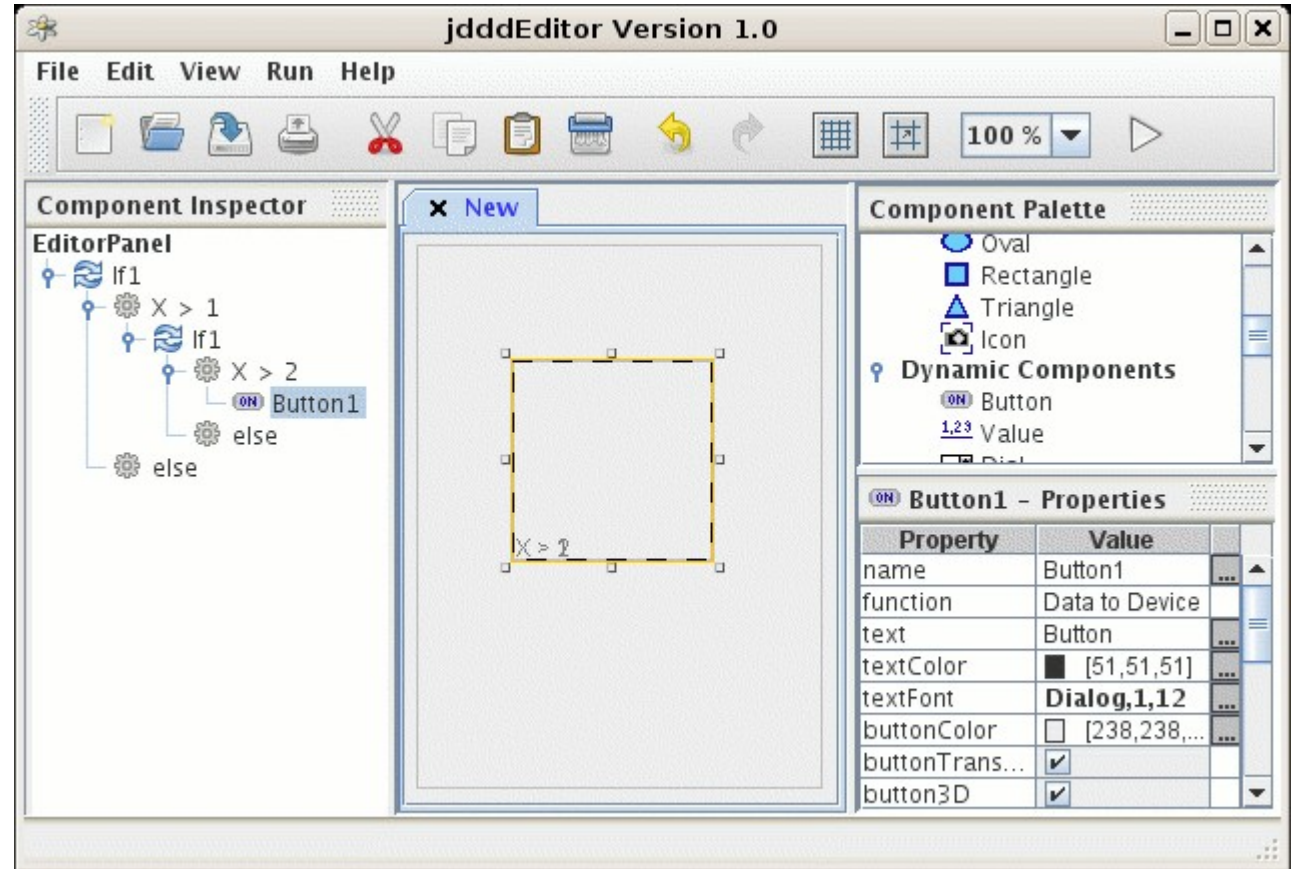
e.g. To display log files



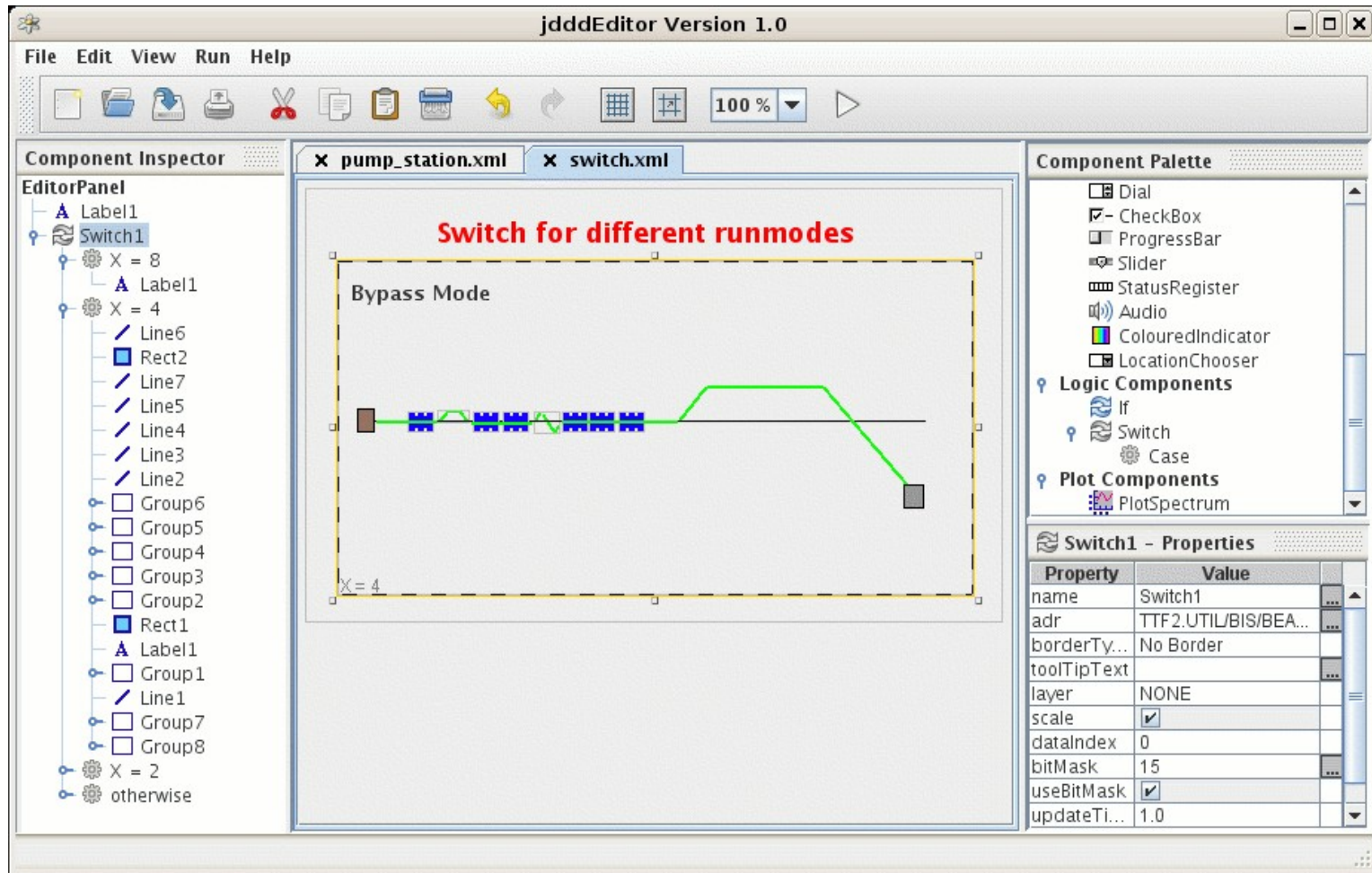
```
tail -f -n 20 server.log

Mpulse:42168849
Server: TTF2.DAQ/ENERGY.DOGLEG
tmask:0x0
Status: 2048
Nchan: 0
Time: 1190177808 sec 106316 usec
Total size: 36596
Mpulse:42168849
Server: TTF2.DAQ/LLRF.ML
tmask:0x0
Status: 0
Nchan: 22
DAQFSM Too many filled blocks, skipping
```

The "If" Component



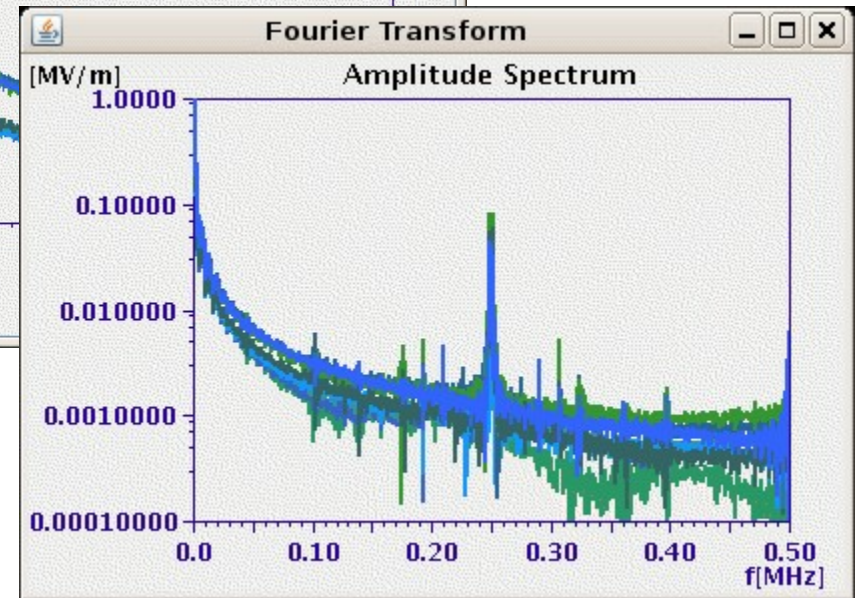
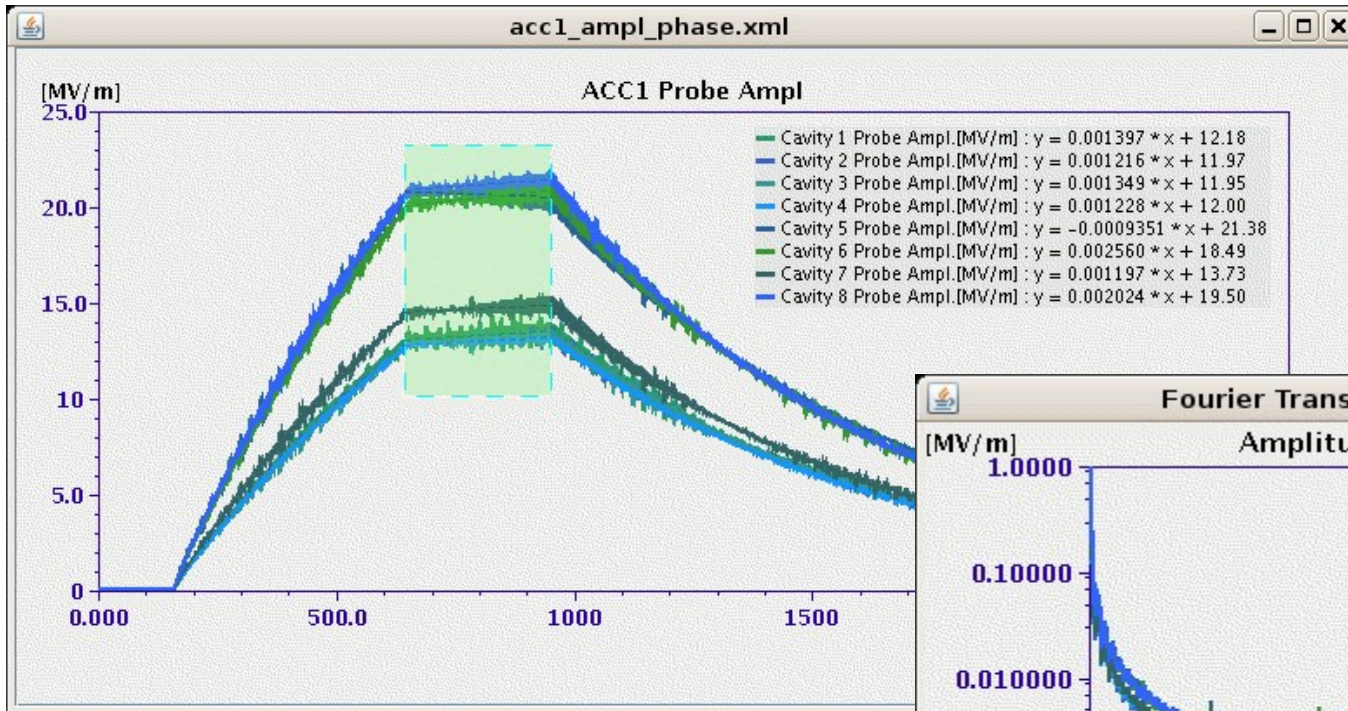
Switch



Switch1 - Properties

Property	Value
name	Switch1
adr	TTF2.UTIL/BIS/BEA...
borderTy...	No Border
toolTipText	
layer	NONE
scale	<input checked="" type="checkbox"/>
dataIndex	0
bitMask	15
useBitMask	<input checked="" type="checkbox"/>
updateTi...	1.0

Plots with mathematical functions, autoscale

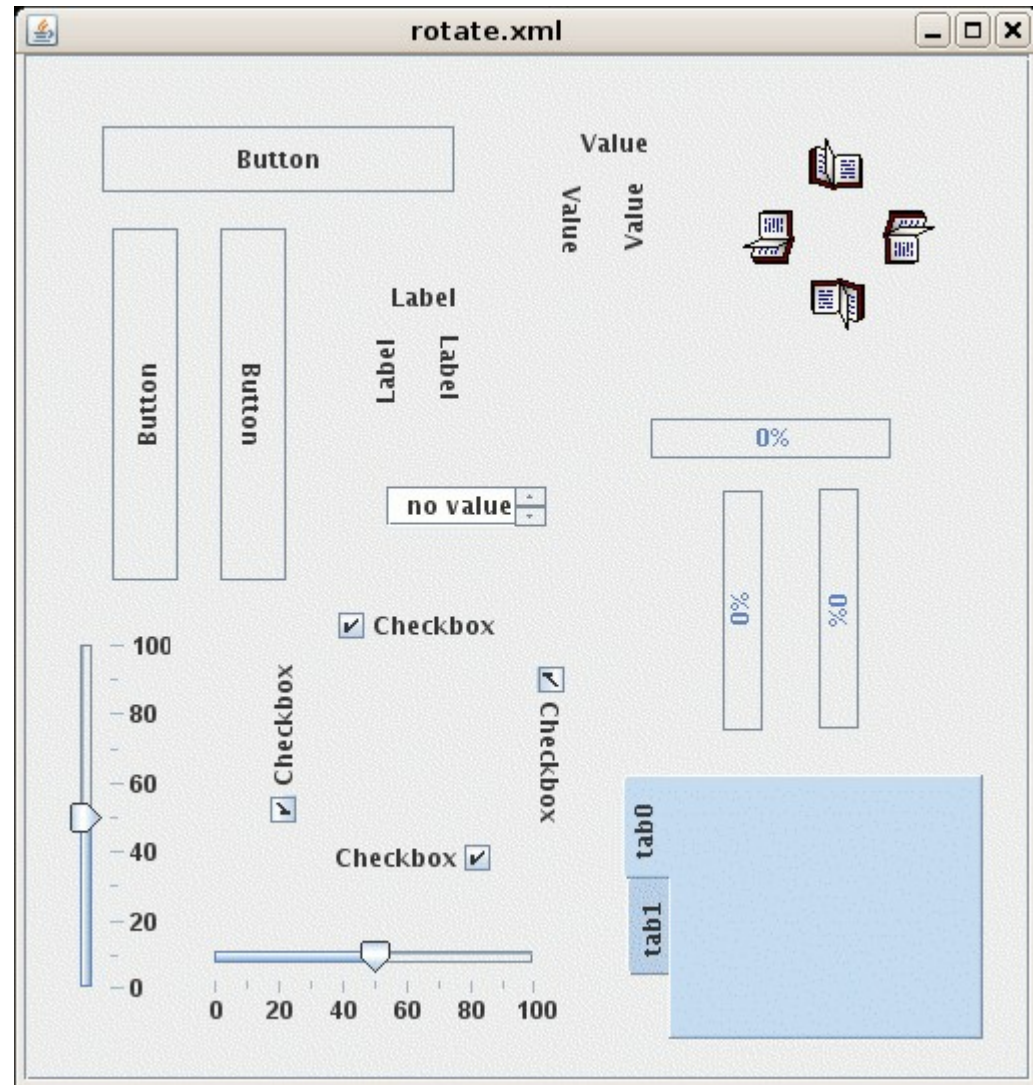




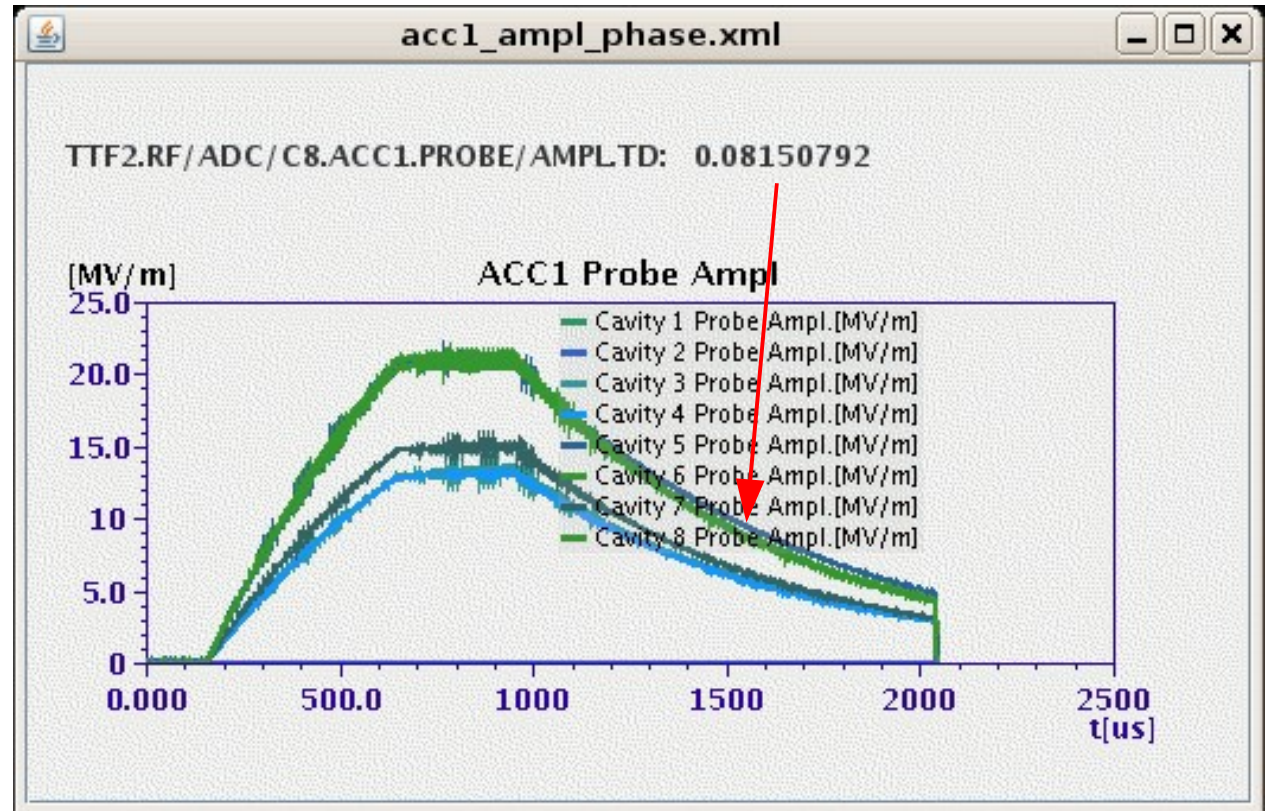
jddd special features

- Rotated components
- Drag & Drop
- Tooltips
- Layers
- Applet
- Use jddd for high level applications

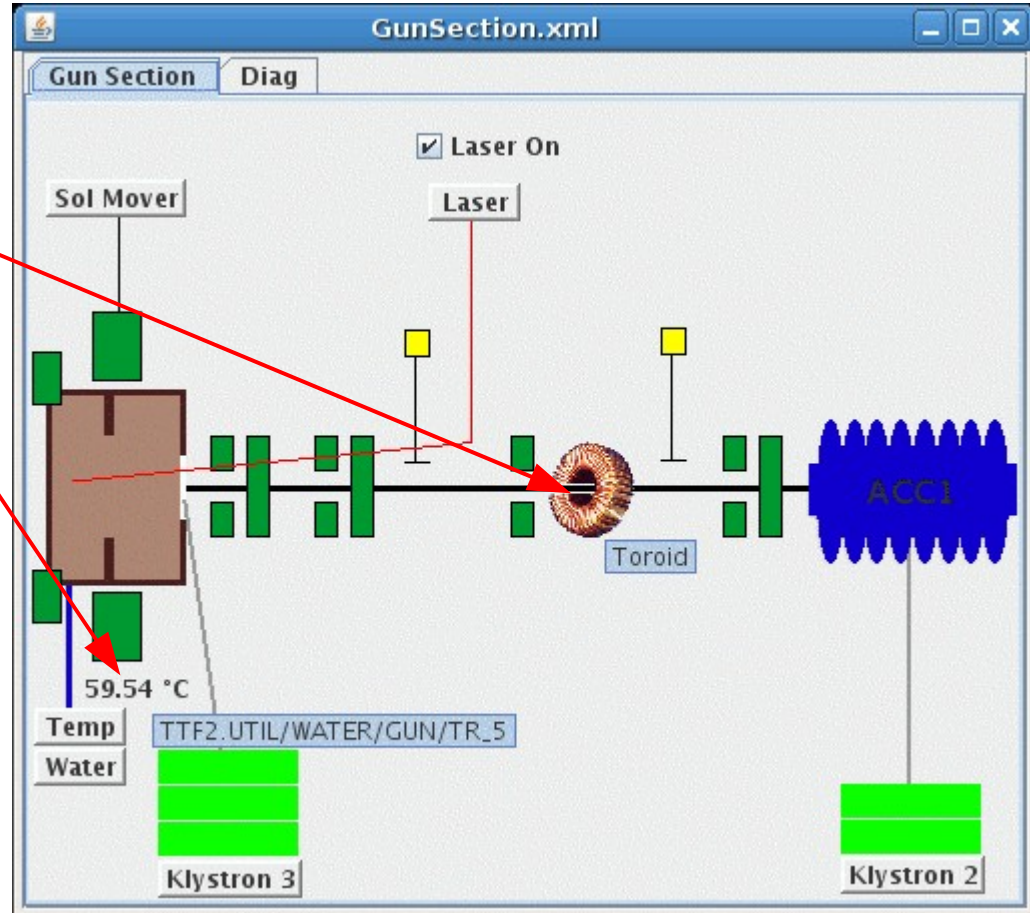
Rotated Components



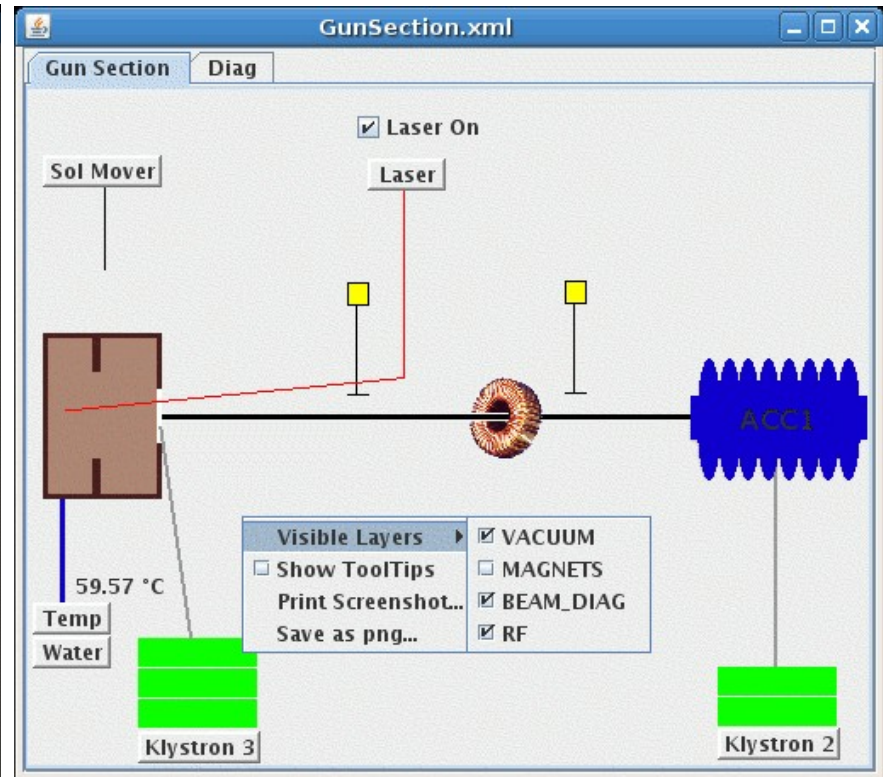
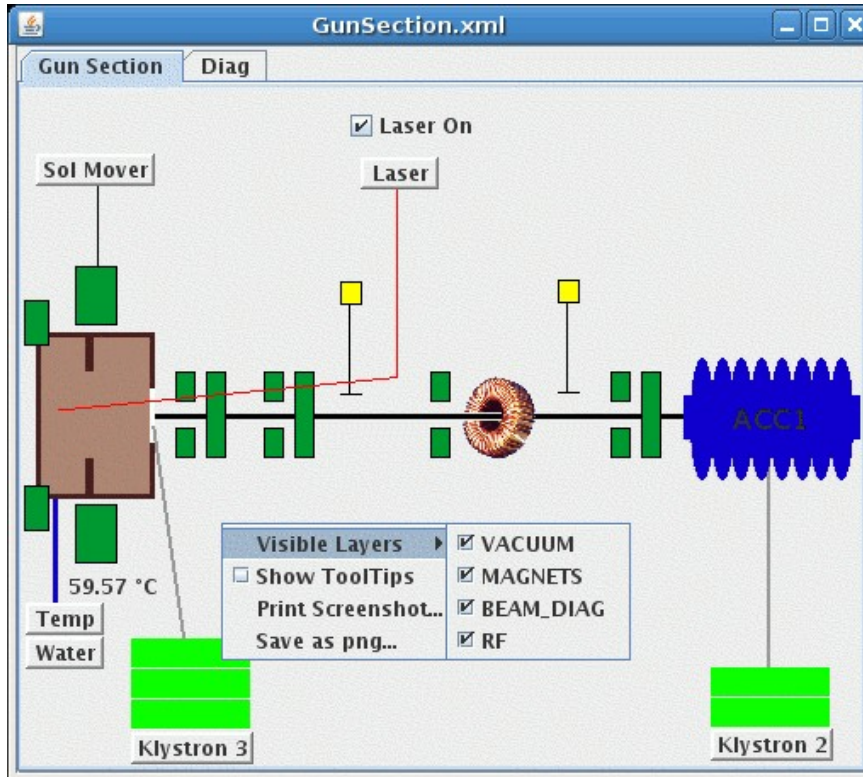
Drag & Drop



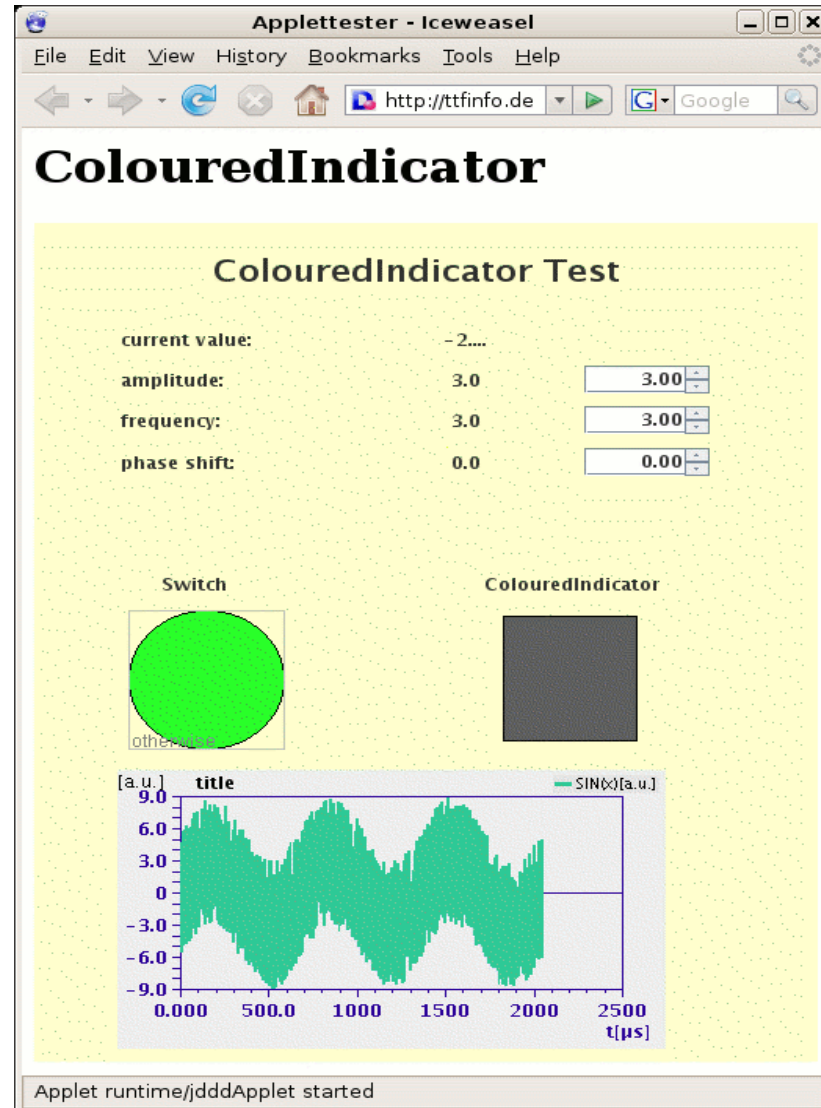
Tooltips



Layers



jddd Applet:





jddd special features

jddd for high level applications:

- Exports a JAVA program code as a Swing
 - JFrame
 - JPanel

Disadvantage:

Panels can't be revised with the jdddEditor after the export.

jddd for high level applications:

Use jddd panels as Java Beans:

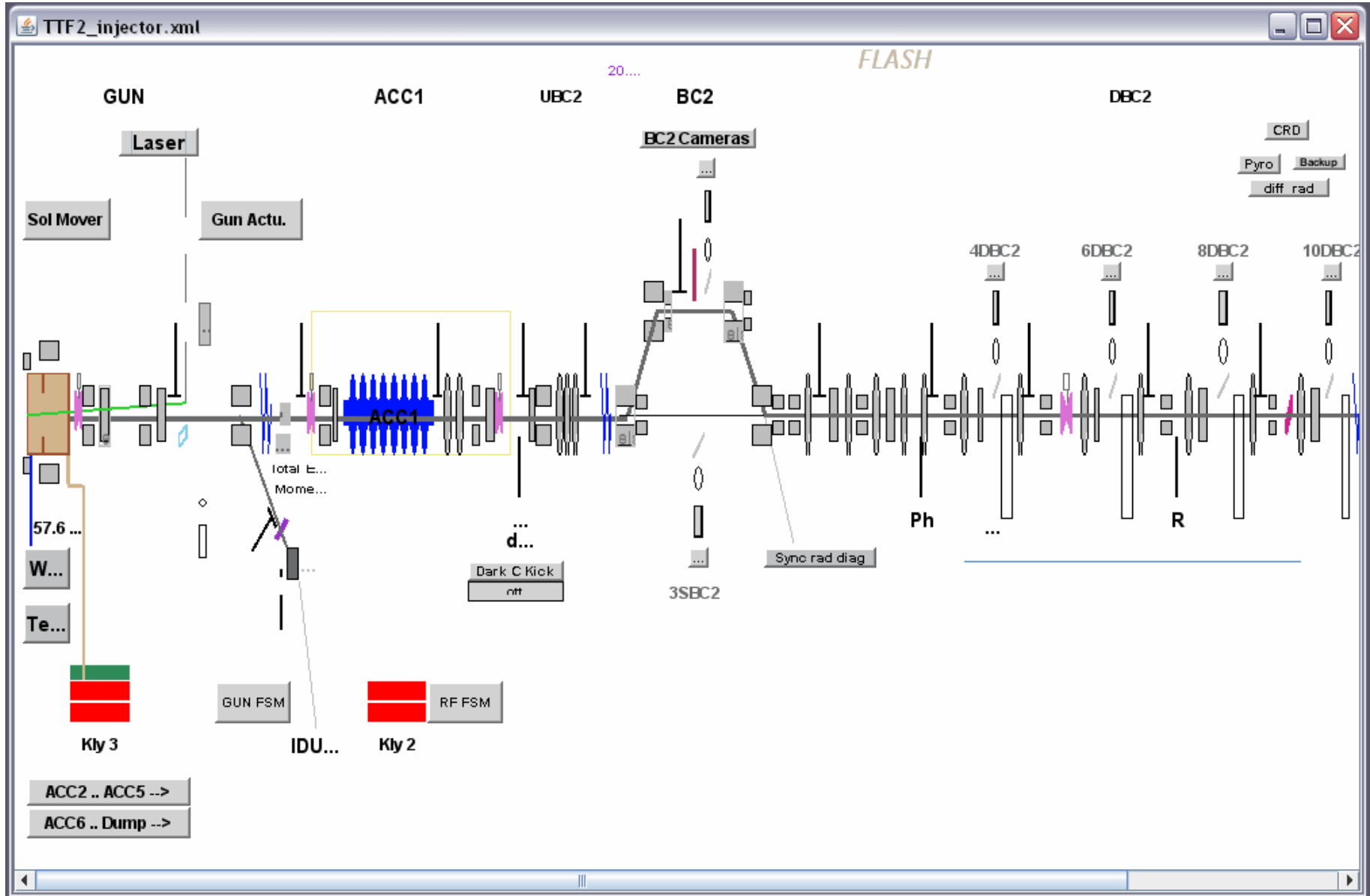
- Create a java application and add the following lines to insert a jddd panel as Java Bean:

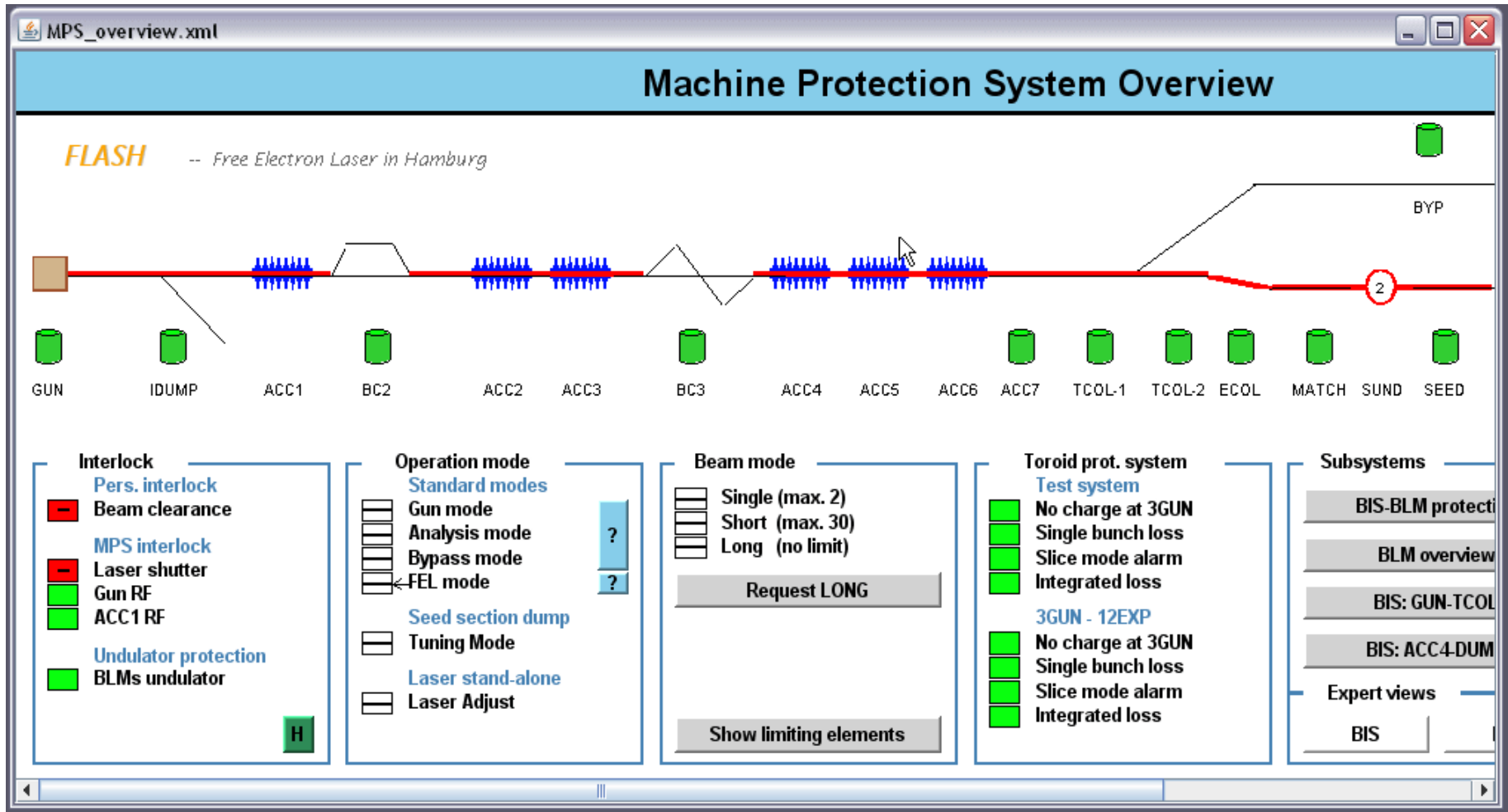
```
private void initComponents() {  
    jdddPanel panel = new jdddpanel();  
    panel.setXmlFile(new java.io.File("/home/ttflinac/jdddFileName.xml"));  
    ...  
}
```

- To access components of the jddd panel:

```
DOOCSOval oval1 = (DOOCSOval)panel.getDoocsComponent("LayeredPane1/Oval1");  
oval1.setDoocsFillColor(Color.orange);
```

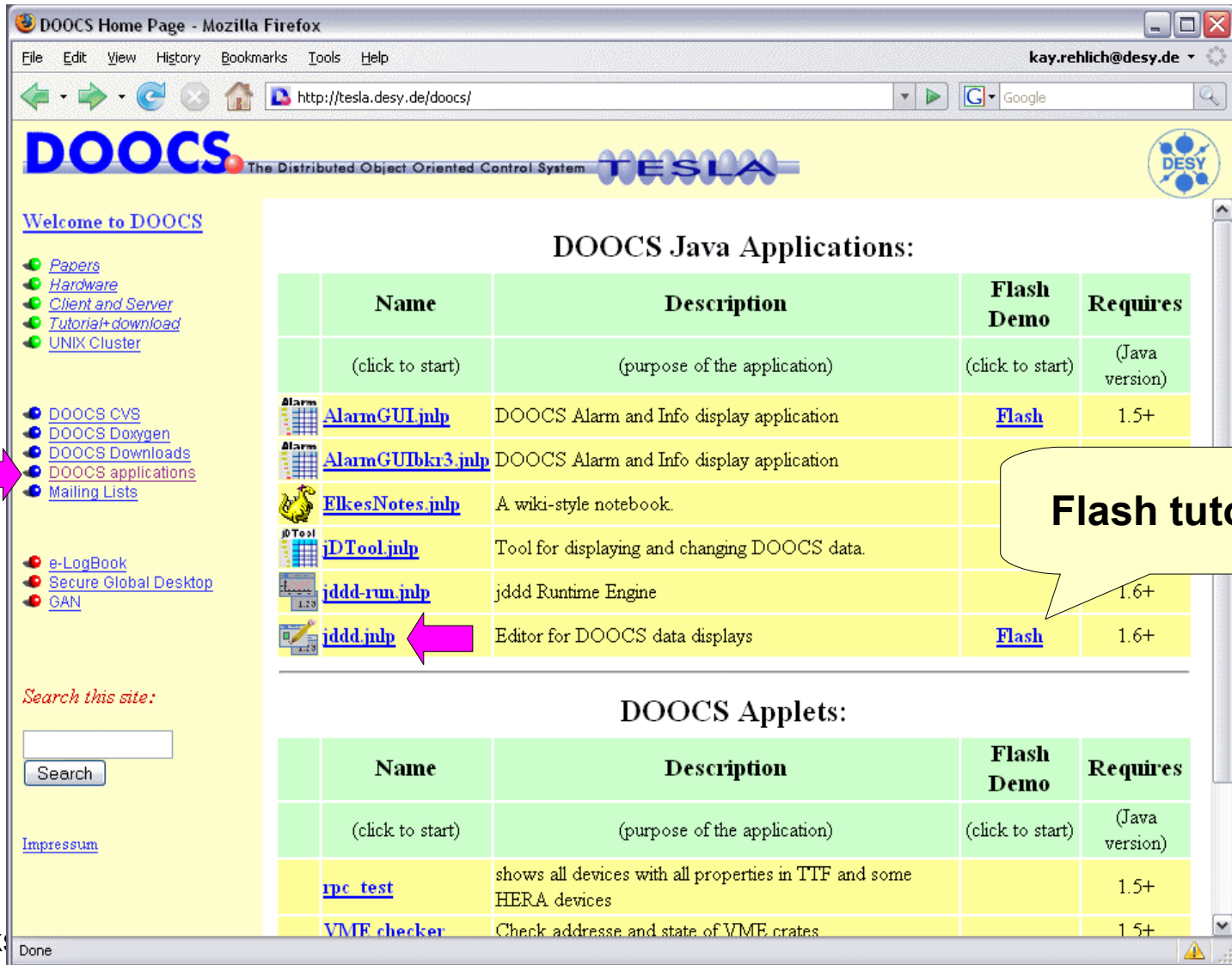
Import FLASH Panels







How to start jddd

http://tftinfo2.desy.de/common/applications/index.jsp



DOOCS The Distributed Object Oriented Control System  

Welcome to DOOCS

- [Papers](#)
- [Hardware](#)
- [Client and Server](#)
- [Tutorial+download](#)
- [UNIX Cluster](#)







- [DOOCS CVS](#)
- [DOOCS Dxygen](#)
- [DOOCS Downloads](#)
- [DOOCS applications](#)
- [Mailing Lists](#)

- [e-LogBook](#)
- [Secure Global Desktop](#)
- [GAN](#)

Search this site:

[Impressum](#)

DOOCS Java Applications:

Name	Description	Flash Demo	Requires
(click to start)	(purpose of the application)	(click to start)	(Java version)
 AlarmGUI.jsp	DOOCS Alarm and Info display application	Flash	1.5+
 AlarmGUIbkr3.jsp	DOOCS Alarm and Info display application		
 ElkesNotes.jsp	A wiki-style notebook.		
 jDTool.jsp	Tool for displaying and changing DOOCS data.		
 jddd-run.jsp	jddd Runtime Engine		1.6+
 jddd.jsp	Editor for DOOCS data displays	Flash	1.6+

DOOCS Applets:

Name	Description	Flash Demo	Requires
(click to start)	(purpose of the application)	(click to start)	(Java version)
ipc_test	shows all devices with all properties in TTF and some HERA devices		1.5+
VME_checker	Check addressse and state of VME crates		1.5+

General Experience:

- Java has proved to be a good choice for GUI development
- Java / jdoocs API is fast enough
- Java Webstart works fine

Outlook:

- jdoocs API improvements
- Find bugs, improve existing components
- Save xml-files in a subversioning system
- Connect jddd to the DAQ system
- ... and many more ideas



jddd

Thank you for your attention!

<http://doocs.desy.de>

--> 'DOOCS applications'