

# jddd: A TOOL FOR OPERATORS AND EXPERTS TO DESIGN CONTROL SYSTEM PANELS

E. Sombrowski, A. Petrosyan, K. Rehlich, W. Schütte, DESY Hamburg, Germany

## Abstract

jddd, a graphical tool for control system panel design, has been developed at DESY to allow machine operators and experts the design of complex panels [1-5]. No knowledge of a programming language nor compiling steps are required to generate highly dynamic panels with the jddd editor. After 5 years of development and implementing requirements for DESY-specific accelerator operations, jddd has become mature and is now commonly used for standard operation. The focus meanwhile has changed from pure feature development to new tasks as archiving/managing a huge number of control panels, finding panel dependencies, book keeping/evaluation of panel usage and collecting possible failures in an automatic manner. Therefore technologies of the existing control system infrastructure like Servlets, JMS, Lucene, SQL, SVN are used. The concepts and technologies to further improve the quality and robustness of the tool are presented in this paper.

## INTRODUCTION

jddd has been developed at DESY as a successor for the previous synoptic display editor “ddd” (docs data display). The editor is designed similar to other standard graphical editors used in NetBeans or Eclipse. It offers many predefined widgets for control panel design. All generated displays can be reused as generic library components and can be added to other displays.

jddd has an interface which supports many control systems used at Desy: DOOCS, Tine, Tango and Epics. Currently more than 110 people working in different groups at Desy are designing panels with the jddd editor. Regular training courses for beginners and advanced users ensure that people learn how to use the editor and its widgets and follow the common standards for good panel design. An example for good panel design is the “Main Taskbar” window, which is displayed in Figure 2.

More than 3600 jddd panels have already been developed at DESY, approximately 1100 of them for FLASH (Free-electron -LASer in Hamburg).

## PANEL ARCHIVING

All panels are stored as XML files in a central revision control system (Apache SVN repository). Each file contains the author name, a description text, the panel content and a thumbnail screenshot. The history of the files is kept in the change log of the SVN browser and if required an old version of a panel may be recovered.

For to the huge number of panels stored in the SVN a powerful file search engine is required. For this purpose the text search engine library Apache Lucene is used, which creates an index of the file names and paths. The Lucene search result for a certain filename is provided via a Java servlet. In case there are several panels with the same filename the servlet returns the path, which is next to the path of the base panel. This way searching files in the SVN is very comfortable and takes only 30-50 milliseconds.

Lucene indexes not only the file names but also the file dependencies. In the jddd editor a file dependencies dialog (displayed in Figure 1) containing useful information about the file connections is available. The dialog has three tabs showing:

- The location(s) of a panel name in the SVN repository
- The filenames of all included panels and panels opened on a button click
- The occurrence of the filename in other panels

This dialog provides all information for a panel designer to check whether a change in a certain panel affects other panels.

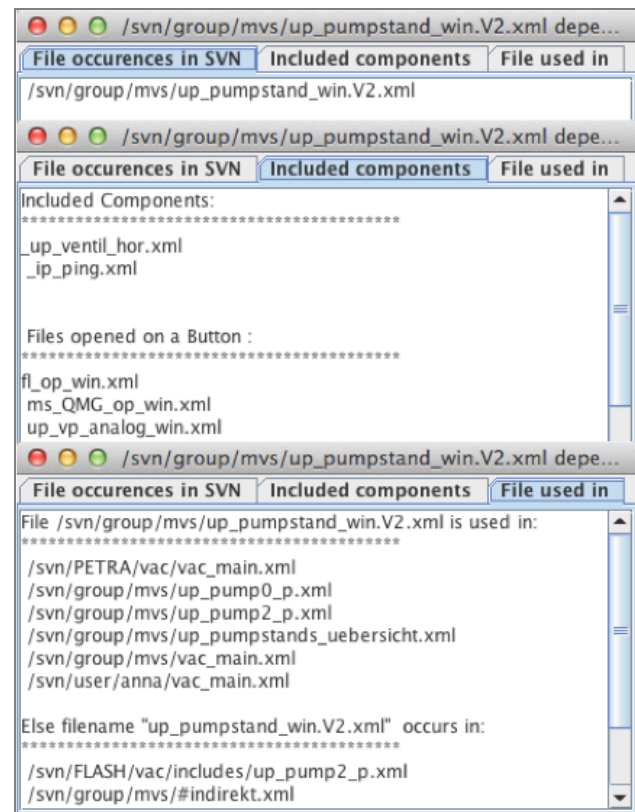


Figure 1: File dependencies of a selected panel.

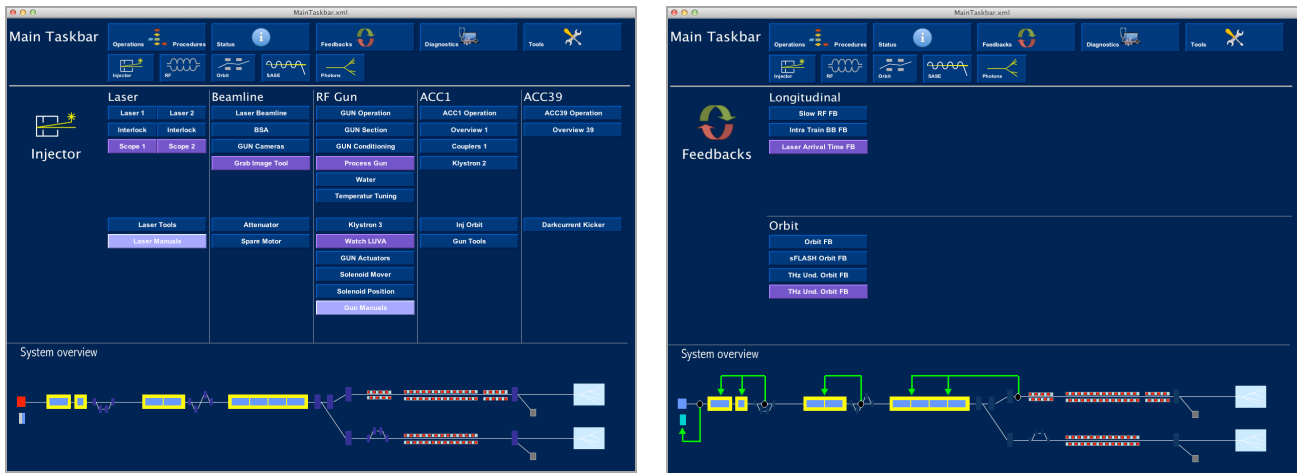


Figure 2: “Main Taskbar” in the new common style. The upper part of the display contains the main navigation buttons. The lower part of the display changes its content according to the selected task.

### EVALUATION OF PANEL USAGE

To collect panel usage statistics, the start and stop times of all panels are sent via JMS (Java Messaging Service) to an Oracle Database. These time stamps are evaluated using SQL and can be retrieved in the “About” dialog of each panel.

Figure 3 shows a screenshot of such an “About” window. It provides general information about the jddd display like file path, creation date, author name and description. The statistical information is read from an Oracle database: the number of runs, the total runtime in hours during the last 365 days and the date of the last known usage.

These statistical data help to find out which panels are no more used at all can be removed from the repository.

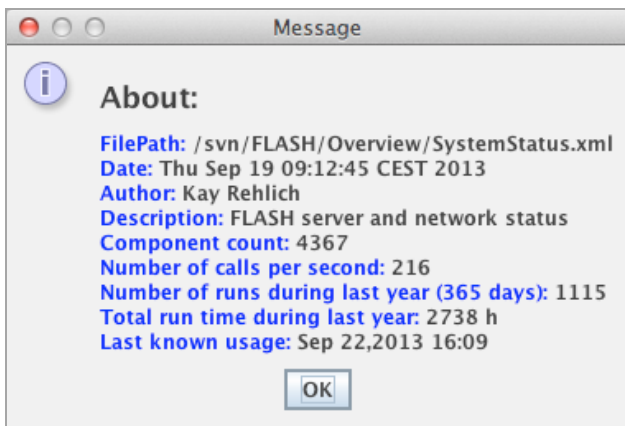


Figure 3: The “About” dialog displays panel information and runtime statistics.

### PERFORMANCE CHECK

Because the experiments are getting bigger and the number of control system values increases rapidly, the control panels are getting more complex and display a lot of information. Especially generic panels with dynamic jddd widgets like lists, which are automatically filled at runtime, may contain a huge number of components and may create many system calls.

To avoid performance problems the “About” window (see Figure 3) provides information about the component count and the number of calls per second for each panel. This way a jddd panel designer can easily check if a display is getting overloaded and if the panel design/structure has to be changed.

### UPDATE INFORMATION

If a panel has changed and is committed to the SVN a notification is automatically sent to a JMS message queue. All jddd applications will then receive this notification and if they are running an older version of this panel, an update info dialog will be displayed. This technology ensures that the operators always use the latest version of a display.

### IMPROVING CODE QUALITY

All possible failures/exceptions, which are generated by jddd applications running at Desy, are sent to a JMS message queue. This message queue serves as a central data collector for jddd shortcomings. The messages are evaluated by the jddd developers to improve the code quality and to find memory leaks.

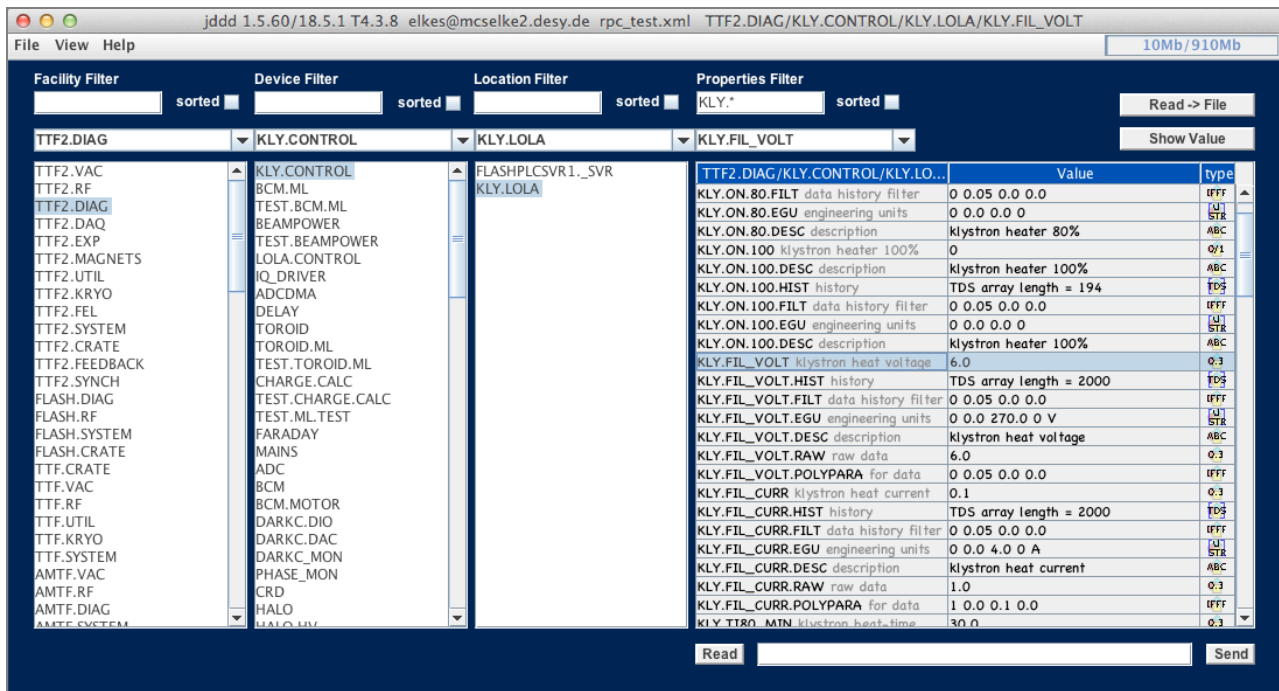


Figure 4: New jddd browser for control system values. Facility, device and location can be selected from lists or from a drop down menu. The properties and their values are displayed in a table. Regular expression filters are used to display a subset of the available addresses.

## THE NEW jddd CONTROL SYSTEM BROWSER

jddd provides all widgets needed for a simple DOOCS control system browser. Figure 4 shows a screenshot of this browser. It displays the address structure and allows the setting of single control system values. Regular expression filters help to sort and find a dedicated address. A mouse click on the properties name opens a special display for the value, for example a spectrum or history plot window or a camera image.

## SUMMARY

jddd is a tool for machine operators and experts, which offers a convenient synaptic editor for control system panel design. It is well accepted and widely used at DESY.

Technologies of the existing control system infrastructure like Servlets, JMS, Lucene, SQL, SVN were used to implement helper tools for jddd developers and designers.

These tools are needed for the management of the huge number of displays at DESY. We are well prepared for an increasing number of panels coming with the XFEL.

## REFERENCES

- [1] <http://jddd.desy.de>
- [2] E. Sombrowski, A. Petrosyan, K. Rehlich, W. Schütte, "jddd, a state-of-the-art solution for control panel development", ICALEPCS'11, Grenoble, France, October 2011.
- [3] E. Sombrowski, P. Gessler, J. Meyer, A. Petrosyan, K. Rehlich, "jddd in action", ICALEPCS'09, Kobe, Japan, October 2009.
- [4] E. Sombrowski, K. Rehlich, "First Experiences with jddd for Petra Vacuum Controls", PCAPAC'08, Ljubljana, Slovenia, October 2008.
- [5] E. Sombrowski, A. Petrosyan, K. Rehlich, P. Tege, "jddd: A Java Doocs Data Display for the XFEL", ICALEPCS'07, Knoxville, Tennessee, October 2007.