

CSS – Control System Studio

TINE Workshop 2007

DESY, 26th September 2007

Jan Hatje, DESY / MKS-2

Overview

- Motivation
- CSS Design
- Eclipse RCP
- CSS Core Features
- CSS + TINE
 - Synoptic Display Studio
 - Databrowser
- Getting CSS

Motivation

DESY, MKS 2

- Responsible for the process control of the cryogenic facilities
- Control System is EPICS (www.aps.anl.gov/epics/)

Existing GUI Applications for EPICS

- Running mostly on UNIX
- Individual programs
- Different Look&Feel
- Data exchange via copy / paste of strings

Requirements

- Operation system independent
- Control system independent
- Common Look&Feel
- Easy data exchange between applications
- Easy to extend

→ **Rich client platform Eclipse**

- Provides configurable workbench, data exchange, deployment options, menus, ...
- CSS adds data types, management functionality, authorization, authentication, APIs, ...
- Plugin technology

CSS Development / Collaborating

Matthias Clausen, DESY - idea, coordination

University of Hamburg, C1-WPS (Scientific contract, CoEUD project) – CSS – Core, Synoptic Display Studio

DESY, ORNL (Kay Kasemir) – Applications: Databrowser, PV Table, Name Space Browser, Alarm System, ...

Cosylab (contract) – Data Access Layer

Josef Stefan Institute – Student exchange

CSS Design I

Java

- Running on several operation systems

Eclipse RCP

- Implementation of OSGI
- Based on plugins
- Common look and feel
- Drag and drop
- Object contribution
- Configurable workbench layout
- Update mechanism for plugins

CSS Design II

CSS Core Features

- Control system datatypes (e. g. process variable)
- Data Access Layer to use different control systems
- Logging service
- Common libraries (JDBC, JMS, ...)
- Authentication, authorization
- Management for CSS instances
- Preference store

Pros and cons

- + CSS is based on a widely used framework (Eclipse)
- + Technically a CSS application is an Eclipse plugin
- + Many levels of integration in CSS from
 - ‘Wrap my Java Code in an Eclipse plugin and run it in CSS’ to
 - ‘Using all CSS features for my plugin’
- + Quick creation of new plugins
- + Single JVM for all CSS applications / plugins

- The effort to get familiar with the Eclipse API is high
- It is difficult to use AWT / SWING for Eclipse plugins

Eclipse RCP – Standalone Applications

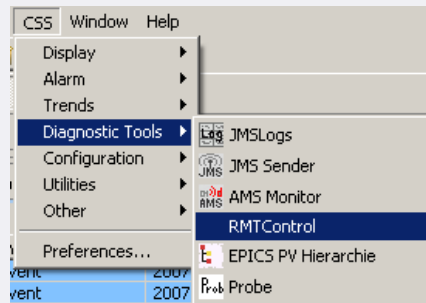
- Single virtual machine for all applications
- Eclipse loads plugins lazily
- Start / Stop of plugins during runtime
- CSS plugins can share CSS core services
- Initialization of CSS core services only on startup



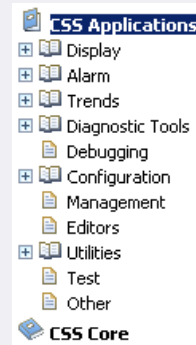
Eclipse Features

- CSS definitions for

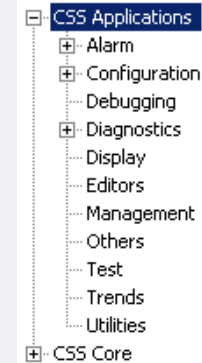
CSS menu



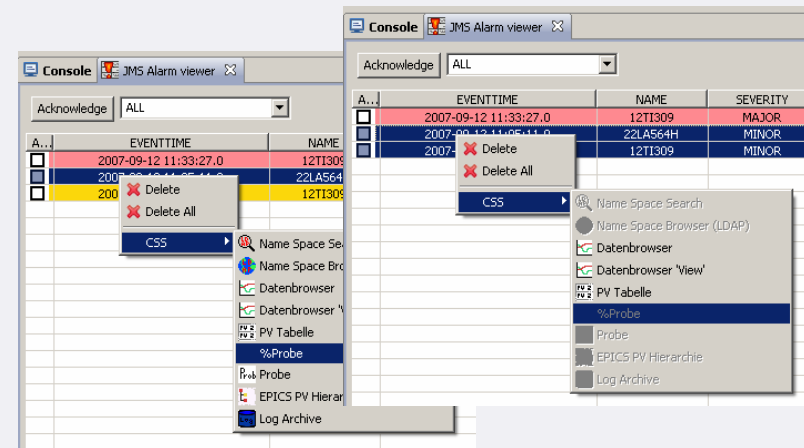
Help system



Preference pages

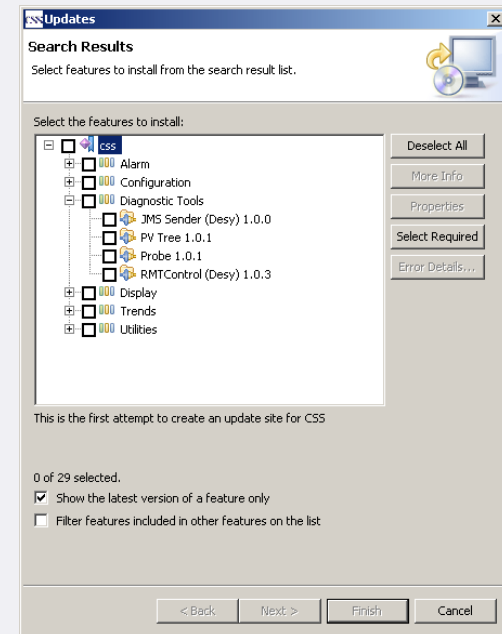


- CSS Datatypes for DnD and Object contribution



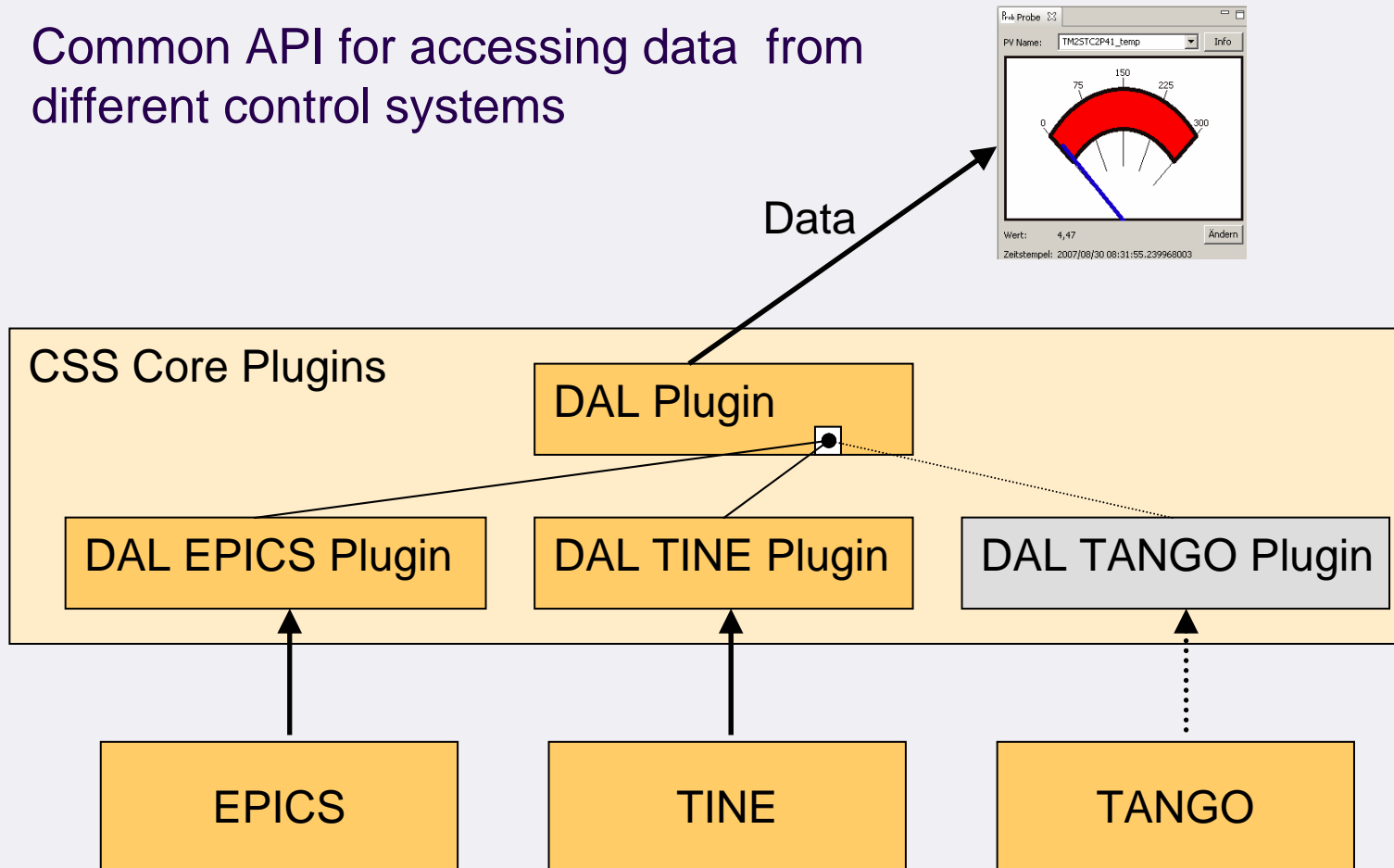
Eclipse update mechanism

- Get new plugins for your CSS installation
- Update current plugins
- Eclipse checks for dependencies of plugins
- Easy to create an update site for your own applications



Data Access Layer (DAL)

- Common API for accessing data from different control systems



Synoptic Display Studio - Design

SDS is based on GEF (Graphical Editing Framework)

- Model View Controller concept
- Grid
- Rulers
- Alignment support
- Widget palette
- Property view
- ...

Synoptic Display Studio

- Easy to draw / configure displays
- The displays are configured in XML files
- Converter for existing ADL files
- Cooperative usage
- Layer concept for widgets
- Wizards and dialogs for creating dynamic behavior
- Edit- and Run-mode for displays

Synoptic Display Studio - Perspective

The screenshot shows the Synoptic Display Studio interface in perspective view. The main workspace is a large grey area. To the left is the Workspace Explorer showing a tree view of the project structure. At the bottom are two Namespace Browser windows. On the right is the Widget Properties palette.

Annotations with arrows point to the following components:

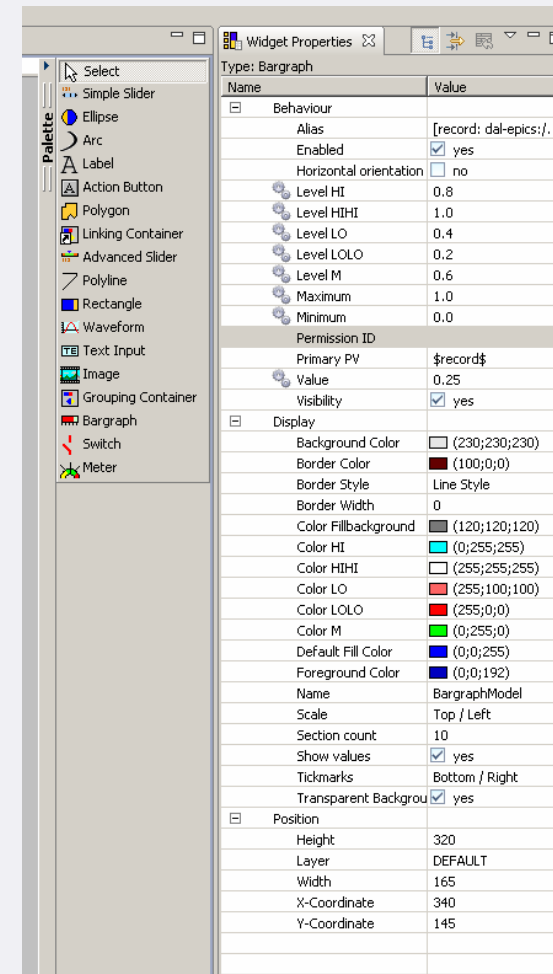
- Workspace**: Points to the Workspace Explorer on the left.
- Editor field**: Points to the main workspace area.
- Property View**: Points to the Widget Properties palette on the right.
- TINE Namespace-browser**: Points to the Namespace Browser (Time) window at the bottom left.
- Epics Namespace-browser**: Points to the Namespace Browser (LDAP) window at the bottom right.

The Widget Properties palette shows the following table:

Name	Value
Behaviour	
Alias	[record: dal-epics:/...
Enabled	<input checked="" type="checkbox"/> yes
Permission ID	
Primary PV	
Visibility	<input checked="" type="checkbox"/> yes
Display	
Background Color	(192;192;192)
Border Color	(100;0;0)
Border Style	Line Style
Border Width	0
Foreground Color	(0;0;0)
Name	DisplayModel
Position	
Height	880
Layer	
Width	1200
X-Coordinate	10
Y-Coordinate	60

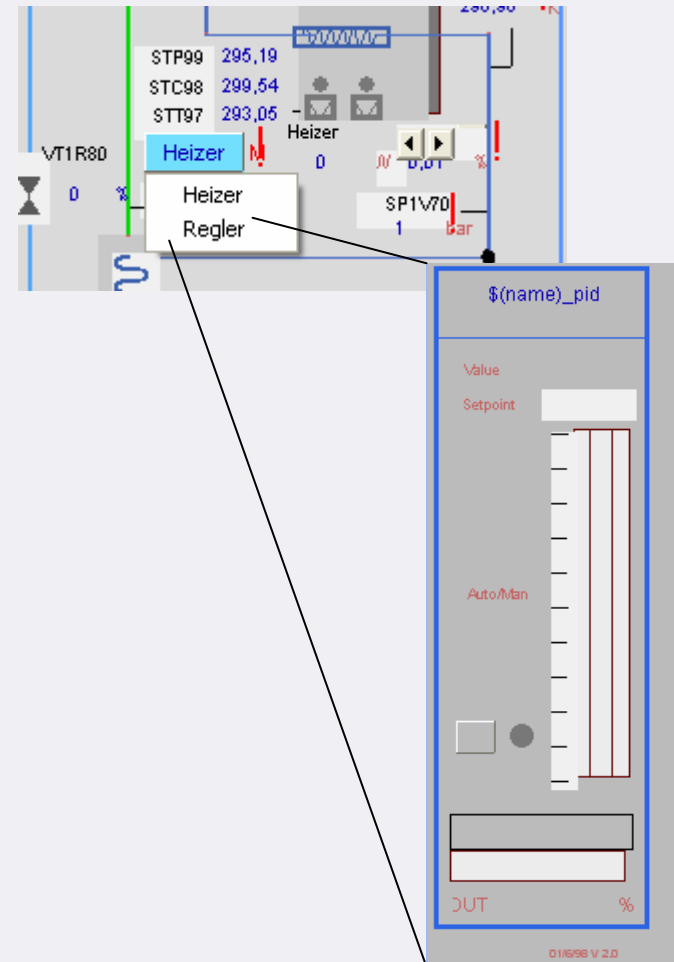
Synoptic Display Studio - Widgets

- Except of strip chart all MEDM Widgets are now available in SDS
- All properties can be dynamic
- Rules (Java Script) for dynamic behavior
- Initializer for widgets (available EPICS)
- Easy extensible



Synoptic Display Studio - Aliases

- Alias names for process variables
- Definition on different levels (display, widget)
- Same alias in widget overwrites alias in display
- Macro substitution (calling a display with an alias as parameter)



Synoptic Display Studio - Wizards

select CS

select widget

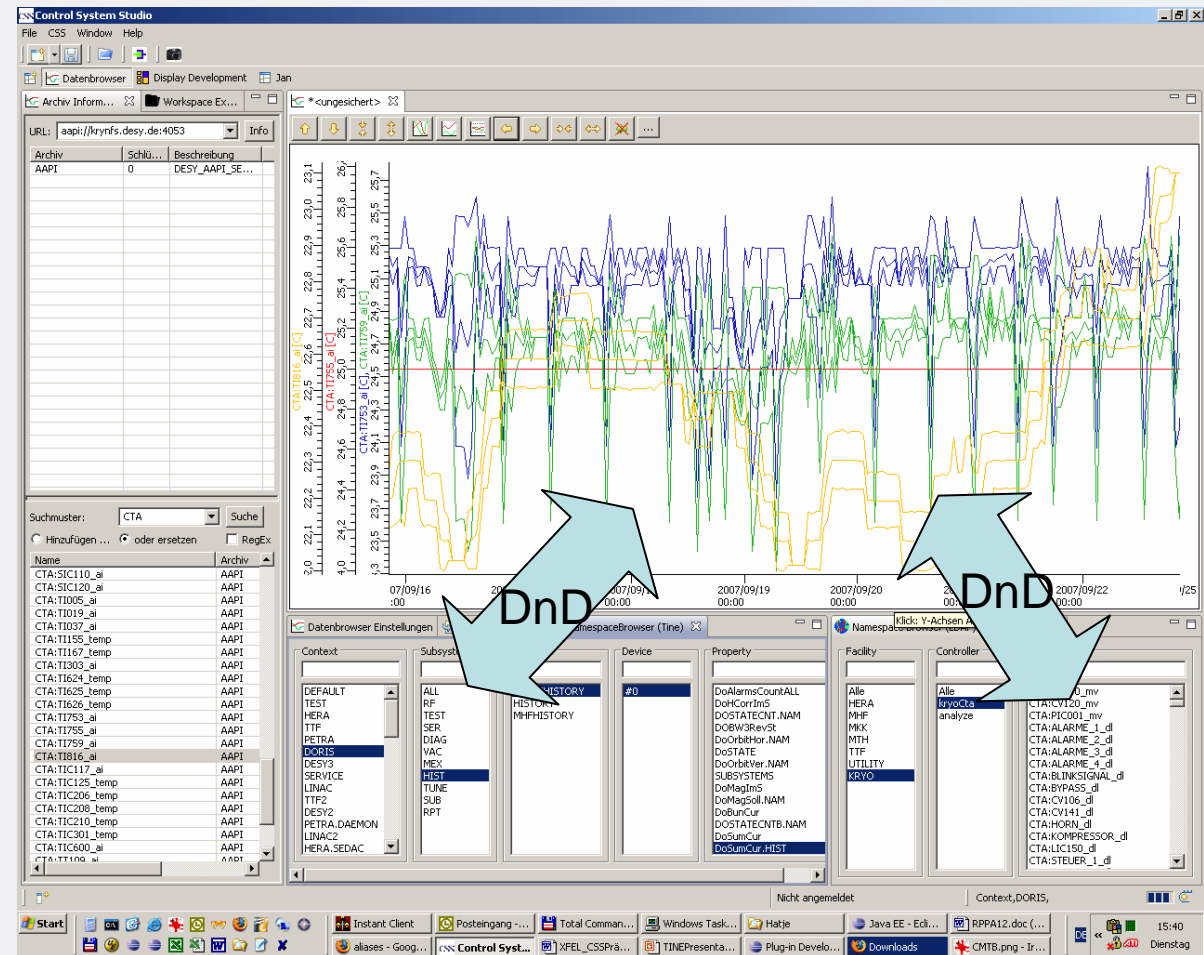
set alias

DnD directly on a widget

Aliases Name	Value
record	dal-epics//TWC2SP1R3_ai

CSS Applications – Databrowser (Kay Kasemir)

- Live- and archive data in one plot
- Multiple archive sources (EPICS, TINE, ...)
- Browsing / searching for archived PVs
- Different display modes (raw, min/max, interpolated)
- Zoom, autoscale, set colors, assign PVs to certain axis, ...



CSS Applications–Alarm System, Name Space Browser

- Tree view for alarm status of PVs
- Table view for alarm status of PVs
- Acknowledgment of alarm messages
- Searching for archived alarm messages
- Configuration tool for messages system
- Browse / search for PVs

The screenshot displays the Control System Studio (CSS) interface. The main window is titled "Control System Studio" and contains several panes:

- Alarm Treeview:** A tree view on the left showing the hierarchy of alarm sources. The "TTF" folder is expanded, showing sub-folders like "BATTERIE", "Kuehlumkreis", "Pumpenhaus1", "GLYKOLKREIS", "HAUPTKREIS", "Kuehlumkreis", "Pumpenhaus2", "TTF-Dump-Tunnel", "TTF-TDC-Halle", and "TTF-TDC-Tunnel".
- Console:** A table view showing active alarm messages. The table has columns for EVENTTIME, NAME, SEVERITY, STATUS, VALUE, TEXT, USER, HOST, and APPLICATI. The table is filtered by "Acknowledge" set to "ALL".
- Namespace Browser (LDAP):** A pane at the bottom left showing a tree view of the LDAP namespace. It is divided into "Facility" (Alle, HERA, PHF, PKK, MTH, TTF, UTILITY, KRYO) and "Controller" (Alle, vacuumCompress2, modulator, ttfkryolnac, ttfkryocb, ttfkryocmb, ttfkryofv, ttfkryolnac, ttfkryovc1, ttfkryovc2, ttfkryovcb).
- JMS Log viewer:** A pane at the bottom right showing a table of archived alarm messages. The table has columns for ACK, TYPE, EVENTTIME, and a description. The table is filtered by "ACK" set to "ALL".

Further information, getting CSS

- For more information or to download CSS please see the CSS website: <http://css.desy.de>
- To get the CSS applications please use the update mechanism: **Help** → **Software Updates** → **Find and Install** → **Search for new features**
- The CSS source code you find in the cvs repositories on kryksunc.desy.de:
`/afs/desy.de/group/m/mks/cvs/css-core,`
`/afs/desy.de/group/m/mks/cvs/css-applications`
(a DESY account is necessary)
- For questions please contact: matthias.clausen@desy.de,
jan.hatje@desy.de

- Thank you -