

CONTROL SYSTEM STUDIO INTEGRATED OPERATING, CONFIGURATION AND DEVELOPMENT

THC002

M. Clausen, J. Hatje, M. Moeller, H. Rickens,
DESY, Hamburg, Germany

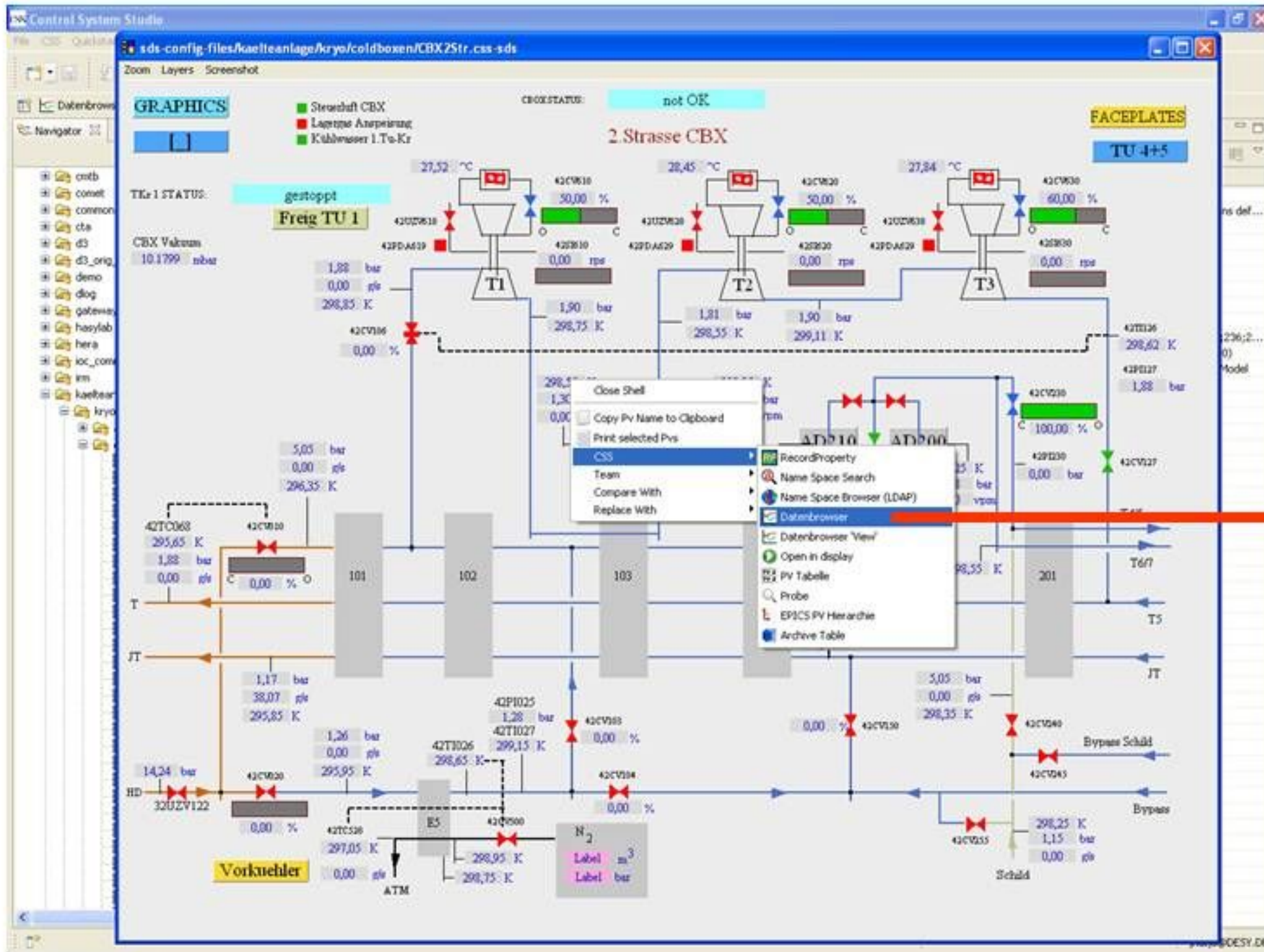
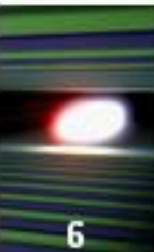
- Control System Studio Overview
- Operational Tools
 - Synoptic Display Studio (SDS)
 - Data Browser
 - Alarm Displays
- Configuration Tools
 - Database Creation Tool
 - Device Database
 - Digital Logic Editor and Simulator
 - Configuration of the Alarm Management System
- Development Editors
 - State Notation Language Editor
- Outlook

- CSS is an Eclipse runtime environment with an enhanced set of core functionalities specific to control system environments
 - Locale setting (e.g. to Japanese) are possible for all strings in CSS
- CSS releases consist of CSS core and a set of control application plug-ins
They can be copied from the DESY ftp server.
- CSS sources are free available from the DESY cvs repository under the Eclipse Public License (EPL) policy (ask us for a DESY cvs account)
- Several sites create their own set of CSS products according to their desire

- CSS 1.2.0 is available since two weeks
 - Based on Eclipse 3.5
 - Java 1.6 (in a 1.5 compatible manner – to avoid conflicts with MAC users)
 - Using the Eclipse Communication Framework (ECF) for remote management
 - Bug fixes in CAJ
Thread safety, synchronization ...
 - SDS and ADL-Converter
 - Converting stripTool config files into dataBrowser config files
 - Calling dataBrowser from a related display button

- The three most prominent applications:
 - Synoptic Display Studio (SDS)
 - Data Browser
 - Alarm Displays
- Data Interfaces
 - Data Access Layer (DAL)
 - Archive API (AAPI)
 - Java Message Service (JMS)

Operational Tools – Synoptic Display Studio (SDS)



Based on GEF

The Eclipse
Graphical Editing Framework

Edit Mode

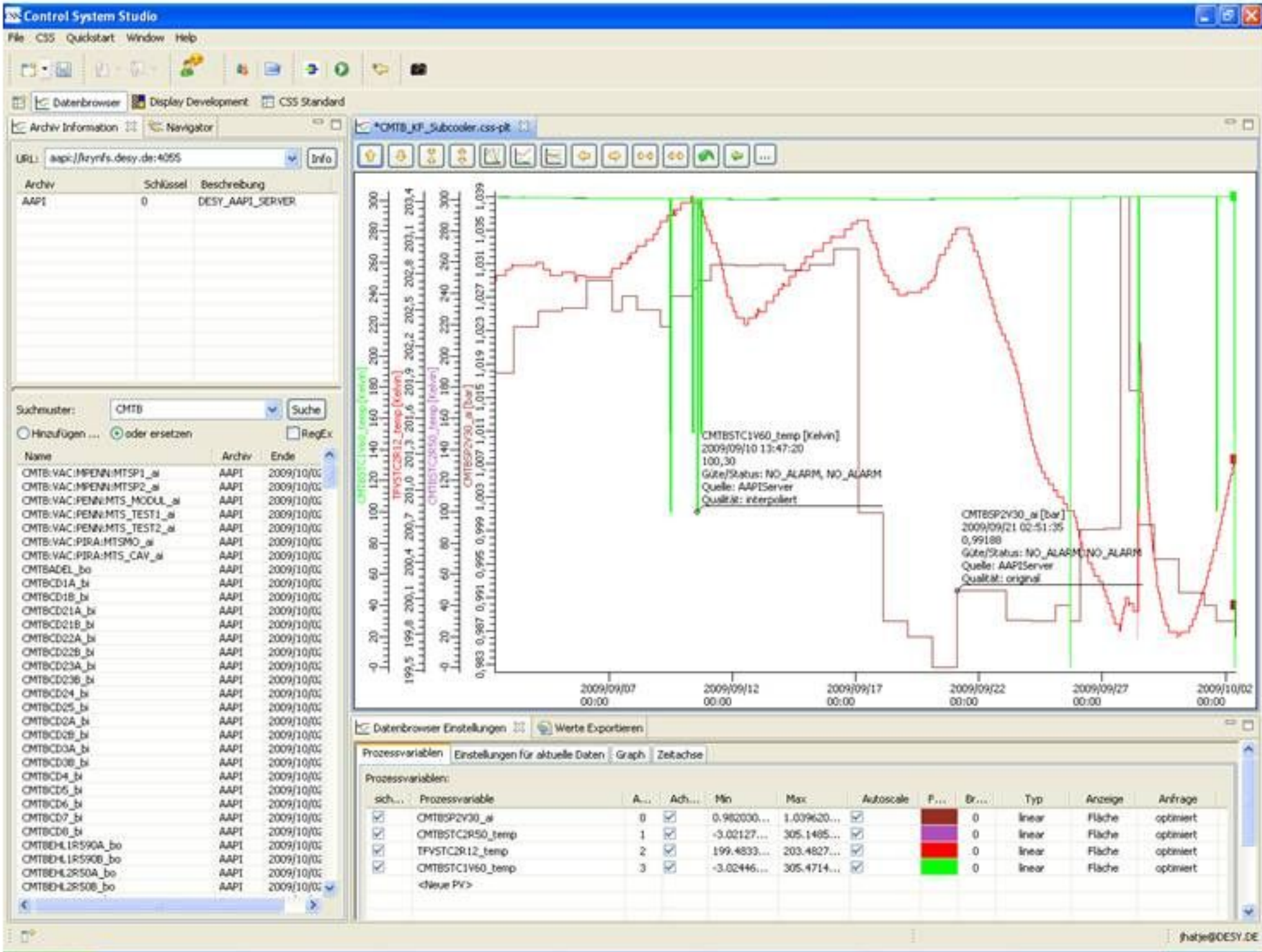
any property
can be dynamic

Runtime Mode

Contribution:

-> to DataBrowser

Operational Tools – Data Browser

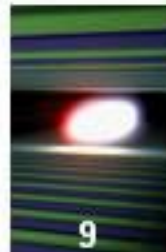


- Alarm- and Log-Tables are registered with **JMS** topics
 - ALARM (general alarm topic)
 - More topics can be configured in the Alarm Management System and are filled from the alarm filter system
 - Log topics: SYS_LOG, SNL_LOG, PUT_LOG

- Configuration of Table settings in Preferences
 - Color Coding (not only for EPICS alarms)
 - Column Labels – and their order

- Alarm Trees are defined in LDAP
(LDAP is also used as the EPICS name server at DESY)

TUP017: Managing Alarms and (Log)Messages - the CSS Way



Alarm Displays – Alarm(Log) Table(s)

The screenshot displays the Control System Studio interface. On the left, the 'Alarm Tree' shows a hierarchical structure of components like 'Quench' and 'Heater'. The main area shows a 'Log Table' with columns for 'TYPE', 'EVENTTIME', and 'NAME'. A dialog box titled 'Start- und Endzeit' is open, showing two calendar views for October 2009 and filter conditions for 'NAME' and 'SEVERITY'. The 'Archive Table' at the bottom shows a table with columns for 'ACK', 'TYPE', 'EVENTTIME', 'NAME', 'STATUS', and 'SEVERIT...'. Red arrows point from the text labels on the right to the corresponding elements in the screenshot.

Alarm – Tree
Log Table
Alarm Table
Archive Table

Contribution to Archive Table (default 24h)

Alarm Displays – Alarm Tree

The screenshot displays the Control System Studio (CSS) interface. The main window is divided into several panes:

- Navigator:** Shows a hierarchical tree of objects. The 'Heater' group is expanded, showing various temperature and pressure sensors. A context menu is open over the 'TMTHL1R44B' object, with options like 'Send Acknowledgement', 'Run Alarm Display', and 'Run Display'.
- Alarm Tree:** A pane showing the configuration of the selected alarm.
- Log Table:** A table showing the properties of the alarm configuration, such as 'alarm display', 'display', 'help guidance', etc.
- Alarm Table:** A table showing a list of active and historical alarms. The table has columns for 'COUNT', 'TYPE', 'EVENTTIME', and 'NAME'. The table is color-coded by severity: red for major, yellow for minor, and green for informational.
- Archive Table:** A table showing a list of archived alarms. It has columns for 'ACK', 'TYPE', 'EVENTTIME', 'NAME', 'VALUE', 'HOST', 'FACILITY', 'TEXT', 'SEVERITY', 'STATUS', and 'SEVERIT...'. The table is color-coded by severity.

- Configuration Tools
 - Database Creation Tool
 - I/O Configurator -> Device Database
 - Digital Logic Editor and Simulator (Diles)
 - Configuration of the Alarm Management System

- EPICS specific database creation tool
- Starting from a hierarchical approach of so called prototypes.
 - Prototypes can consist of records and other prototypes
- Names are created in the prototype hierarchy according to naming macro substitutions (rules)
- Instances are created by resolving the final level of macro substitution
- Persistence in XML file
- Output is an EPICS db file

- Plan:
 - *Record names and IO_NAMES are stored in a RDB*
 - *Graphical display of the prototype hierarchy - for documentation purpose only (for now)*

Configuration Tools - Database Creation Tool

The screenshot shows the Control System Studio interface with the following components:

- Navigator:** Shows a project tree with folders like >CSS [trysunc], >SOS, >Coldbox42 [trysunc], and >SOS [trysunc].
- ao-Record Properties:**
 - Common Settings:**

Identifier	a12a120c-ae5b-4300-b000-2971093d7120
Name	Ventil
Type	ao
Epics Name	\$(box)CV\$(nr12)\$(ifdnr) ao
Disabled	false
 - Fields:**

PINI	YES
DTYP	PBDP
DRVH	100
DESC	\$(desc)
OUT	>ioname(\$(box)CV\$(nr12)\$(ifdnr))
EOU	%
- Outline:** Shows a hierarchical tree of components under 'COLD42', including Prototypes like AnalogVentil_2Endschalter, Adsorber, and TurbinenDrumherum.

A blue text overlay in the center of the image reads: `>ioname($(box)CV$(nr12)$(ifdnr))`

Configuration Tools – I/O Configurator and Device Database

- Configuring the structure of I/O devices
 - First implementation is available for Profibus I/O
 - Necessary to configure Profibus I/O on NON-Windows Systems
 - Standard Tools only run **on** Windows and configure Profibus Systems running in PLCs or **on** Windows
 - It is using the Profibus configuration files provided by the hardware vendor GSD (Geräte Stamm Datei) to configure the actual installed hardware
 - *Second implementation planned for Siemens S7*
- Writing configuration into XML (not EPICS specific)
 - Parsed by Profibus driver on the EPICS IOC to configure the DPM memory in the Profibus controller card
- Central store for documentation ('information on your fingertip')
EPICS channel -> IO_NAME -> I/O device -> Documentation

Configuration Tools – Device Database

Storing Documentation and/or Configuration Data

The screenshot shows the 'Control System Studio' interface. The main window displays 'Profibus Slave Configuration' for a device named 'SystemNameNode'. A search window is open, showing a list of documents available for storage in the device database. The 'Available' section contains the following table:

Subject	M..	Crea...	Desc
Messstellenliste	xls	2009-09-01...	Messstellenliste LPE42□□□...
Anleitung zum Austausch	pdf	2009-08-27...	Anleitung zum Austausch eine...
Anleitung zum Laden	pdf	2009-08-27...	Anleitung zum Laden des CoD...
CoDeSys Software	zip	2009-08-27...	Dies ist die WAGO-Programmi...
Projekt: Sicherheitsschlei...	zip		Aktuelle Projektdatei für WAG...

A context menu is open over the 'CoDeSys Software' entry, showing options: Open, Save, and Update.

The console window at the bottom shows the following output:

```

CSS Console
2009-10-02 15:08:18,248 2009-10-02 15:08:18,248 ERROR [win] org.hibernate.tool.hbm2ddl.SchemaUpdate: ORK-02275: such a referential constraint already exist
Start up Fri Oct 02 15:12:29 CEST 2009
Start up Fri Oct 02 15:13:43 CEST 2009
difference: 73783
    
```

Store the internal logic program of the intelligent Profibus controller in the device database

Configuration Tools – Device Database Profibus Slave Configuration

Profibus Slave Configuration

General Information
 Created by: hrickens@DESY.DE
 Modified by: okorth@DESY.DE
 Created on: 2009-04-27 14:22:22.0
 Modified on: 2009-09-01 09:57:15.0
 Version: 5
 Database ID: 9063

PK	Name	Ext User Prm Data Const
	PK7	0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x02,C
1	Module0	0x80, 0x21, 0x00
2	750-431	0x23, 0x01, 0x00
3	750-431	0x23, 0x01, 0x00
4	750-431	0x23, 0x01, 0x00
5	750-431	0x23, 0x01, 0x00
6	750-501	0x20, 0x02, 0x00
7	750-501	0x20, 0x02, 0x00
8	750-454	0x20, 0x44, 0x11, 0x06
9	750-454	0x20, 0x44, 0x11, 0x06
10	750-454	0x20, 0x44, 0x11, 0x06
11	750-454	0x20, 0x44, 0x11, 0x06
12	750-454	0x20, 0x44, 0x11, 0x06
13	750-454	0x20, 0x44, 0x11, 0x06
14	750-454	0x20, 0x44, 0x11, 0x06
15	1 Byte PFC-Outp. (Boolean)	
16	1 Byte PFC-Outp. (Boolean)	
17	1 Byte PFC-Outp. (Boolean)	

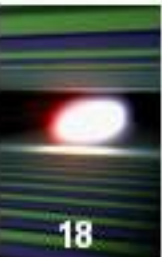
Groups
 1 2 3 4
 5 6 7 8

Configuration Tools – Device Database

IO Name:
42TAC622

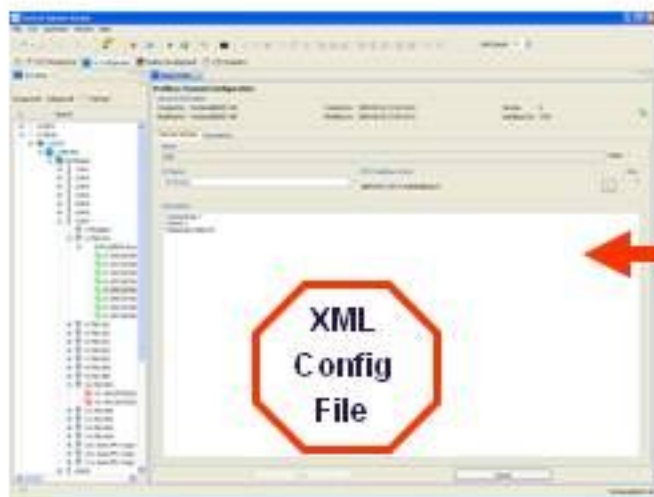
EPICS address string:
@PROFI1:7/0 'T=UNSIGN8,B=4'

Hardware Channel
I/O_NAME
EPICS address string

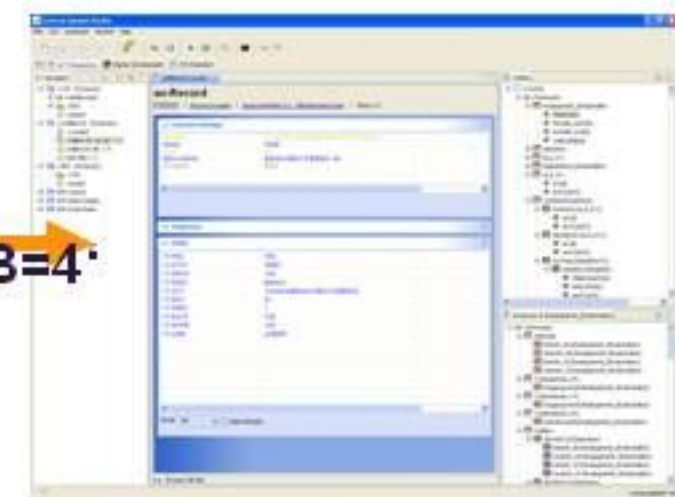


IO_NAME the Link between I/O Data and DCT

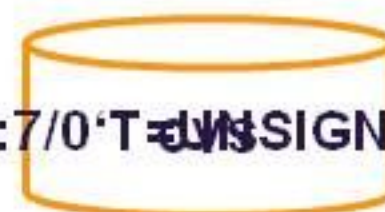
CSS I/O Configurator



CSS DCT



Device Database



Cvs repository

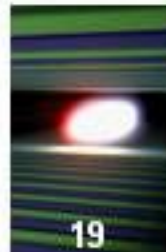
`RDP@PROF11:7/0'T=UNSIGN8,B=4'`

Create EPICS db file

INP: `@PROF11:7/0(T=UNSIGN)8,B=4'`



Configuration Tools – Digital Logic Editor and Simulator



Work in Progress

Initial implementation by a student from the Jozef Stefan Institute

Control System Studio

File CSS Quickstart Window Help

Datenbrowser Display Development CSS Standard

Navigator

- >CSS [Irykaunc]
 - Backup
 - Databrowser
 - project
 - Logic.dfiles
- >Coldbox42 [Irykaunc]
- >SDS [Irykaunc]
- SDS Cursors
- SDS Demo Display
- SDS Script Rules
- sds-config-files [Irykaunc]

Properties

Property	Value
Column	1
ID	1
Name	Input 2
Status	false

Table View

ID	Com...	Hardware Input	Hardware Output
0		true Input 1	false Output A
0		false Input 2	
1		true Input 3	
2		true Input 4	
3		false Input 5	
4			
5			
6			
7			
8			
9			

Properties

- Select
- Marquee
- Connection
- Input (HW)
- Output (HW)
- Command
- Status
- Analog Input
- Comparator
- Not
- And
- Or
- Xor
- Flip Flop
- TDE Timer
- TDD Timer

Table View

ID Com... Hardware Input Hardware Output

0 true Input 1 false Output A

0 false Input 2

1 true Input 3

2 true Input 4

3 false Input 5

4

5

6

7

8

9

jhaly@DESY.DE

Configuration Tools – AMS Configuration

- Configuration of the Alarm Management System
 - User
 - User-Groups
 - Filter conditions
 - Filter (set of conditions)
 - Actions
 - Short Message Service (SMS)
 - Into another JMS topic
 - Mail
 - Voice mail

- Activation of this plug-in is controlled by the role based authentication/ authorization scheme which is part of the core CSS functionality

TUP015: A Framework for Authentication and Authorization in Plug-in-Based Control System Software

- Special Features
 - Language Sensitive Editor (LSE)
 - Syntax check
(no code completion – yet)
 - Colour coded keywords
 - Start compiler on save operation
 - Return compiler warnings into problem view
 - Outline View showing variables, event flags, state sets, states
Jump from Outline View back into editor
 - SNL Diagram Editor illustrates states and conditions

Development Editors – SNL Editor

Control System Studio

File CSS Quickstart Window Help

SNA Development Datenbrowser Display Development CSS Standard

Navigator

- >CSS [krykunc]
 - Backup
 - DataBrowser
 - .project
 - Logic.dles
- >Coldbox42 [krykunc]
- >SDS [krykunc]
- SDS Cursors
- SDS Demo Display
- SDS Script Rules
- >str12 [krykunc]
 - bin
 - generated
 - graphics
 - source
 - .cvsignore 1.1
 - Makefile 1.1
 - kuerberw.st 1.27
 - kuerberw_or.txt 1.1
 - kbetrieb.st 1.10
 - kuerberw.st 1.16
 - tukreis1.layout 1.1
 - tukreis1.st 1.20
 - .project 1.3
- sds-config-files [krykunc]

```

#define OK 1
#define NOT_OK 0
#define SUPERVISORY 0
#define CLOSED_LOOP 1
#define AUF 1
#define ZU 0
#define PID_MANUELL 1
#define PID_AUTO 0
#define STOP 0
#define START 1
#define CLOSE 0
#define OPEN 1
#define BETRIEB 3

#define EQS(x, y) (!strcmp( x, y))
#define PV_SET(pv, val) pv = val; pvPut (pv);
#define PV_ASSIGNED(pv, type, rname) type pv; assign pv to rname;
#define PV_ASSIGNED_N(pv, type, rname) PV_ASSIGNED(pv, type, rname) monitor pv;

#define NDK1 (kompBits & K1)
#define NDK2 (kompBits & K2)
#define NDKR12 (kompBits & NDR12) // korrekt?? Red. fuer JT-ND

#include <stdio.h>
#include <time.h>
time_t t_aba_sec;
#include <string.h>
#include <sniflog.h>
#include <alarm.h>

char text[40];

//Rezept-Var.
double tlimit;
assign tlimit to "KSS:SNL:KUR:TLIMIT_n1";
monitor tlimit;
double thyst;
assign thyst to "KSS:SNL:KUR:THYST_n1";
    
```

Outline

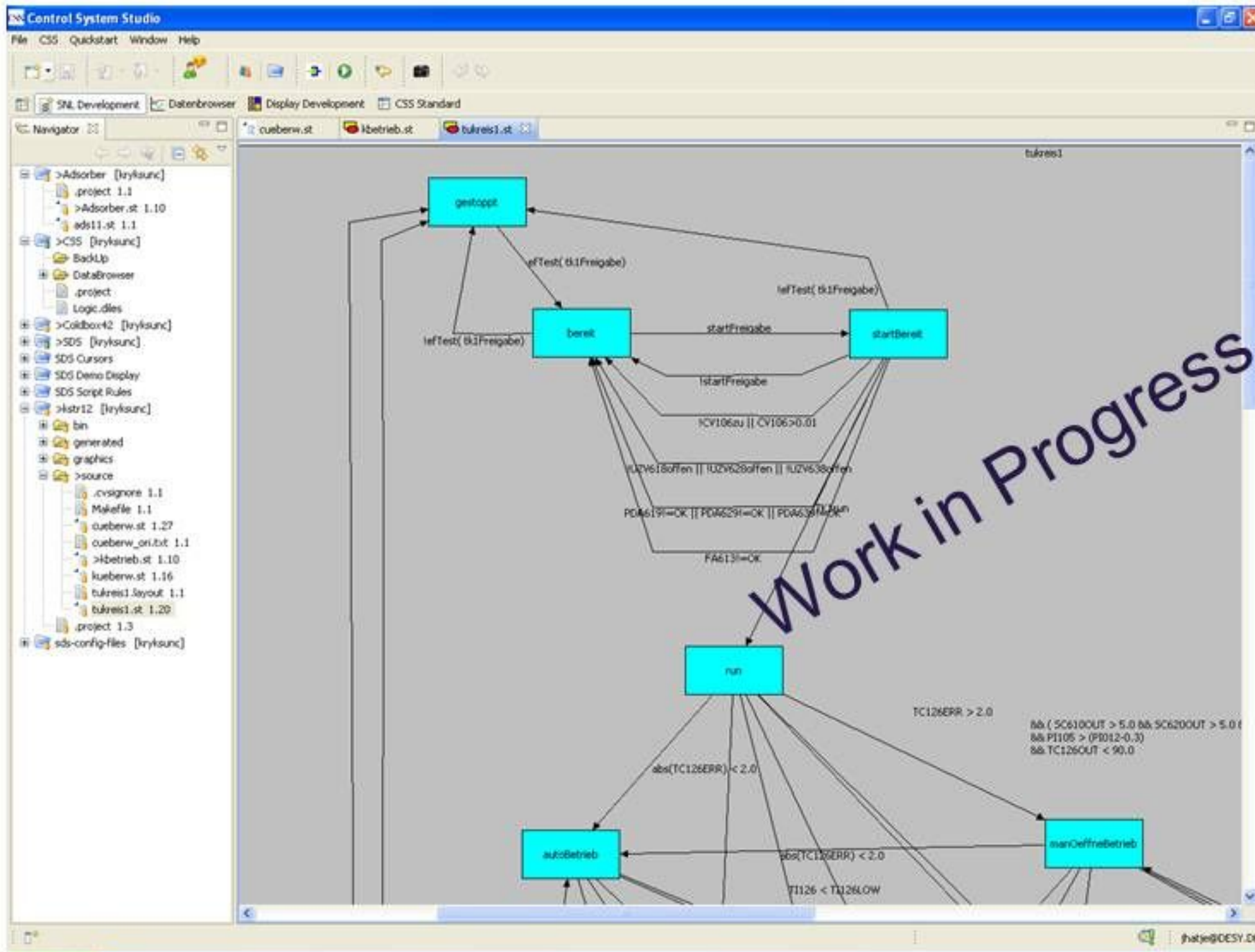
- Program kbetrieb
 - option -e
 - Variables
 - watchFlag (evtfag)
 - kompOKFlag (evtfag)
 - softStopEnableFlag (evtfag)
 - softStopFlag (evtfag)
 - state set: Ueberwachung
 - state inactive
 - state kompOK
 - Entry statement
 - when (HDoelTemp>tlim) -> kompFehler
 - when (NDoelTemp>tlim) -> kompFehler
 - when ((T6>T6thk)) -> kompFehler
 - when ((T1==NOT_OK) || (T2==NOT_OK) || (T5==NOT_OK) -> kompFehler
 - when (!elTest(watchFlag)) -> inactive
 - Exit statement
 - state kompFehler
 - state set: StopQuery
 - state set: Kompressoren
 - state initial
 - state NichtBereit
 - state Bereit
 - state KompStart
 - state Stabilize
 - state Warte_C0C
 - state LastBetrieb
 - Entry statement
 - when (!elTest (kompOKFlag)) -> Emergency
 - when (elTestAndClear(softStopFlag)) -> St
 - when (!(NDK1 66.ful25Y157 > OZ108R))
 - state BypassBetrieb
 - Entry statement
 - when (!elTest (kompOKFlag)) -> Emergency
 - when (elTestAndClear(softStopFlag)) -> St
 - when ((NDK1 66.ful25Y157 > OZ108Z) ||
 - state EmergencyStop
 - Entry statement
 - when (TRUE) -> Stop1
 - state Stop1
 - state Stop2
 - state Stop3
 - state Stop4
 - state kStop
 - state kStopC

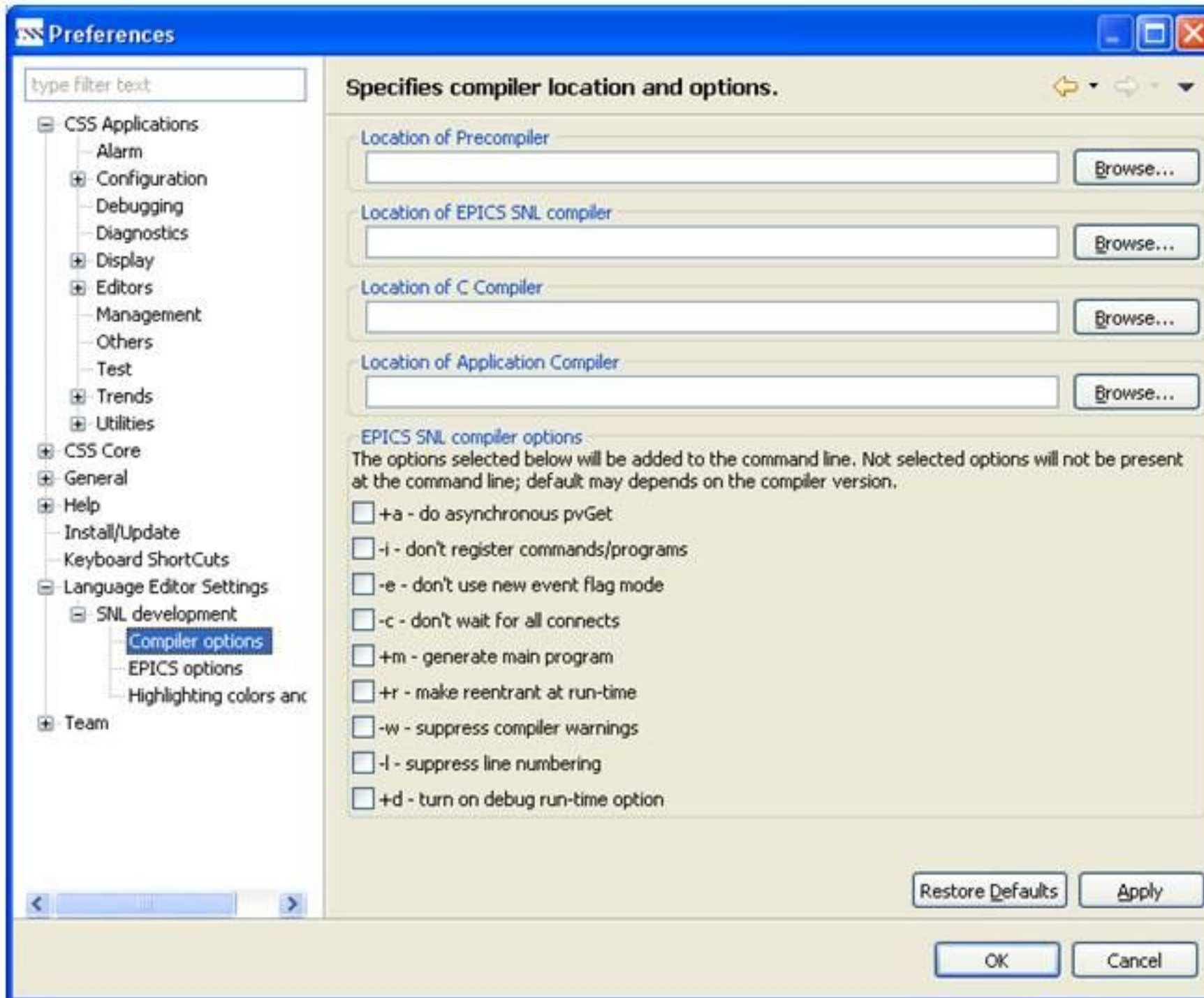
Problems

0 errors, 2 warnings, 0 others

Description	Resource	Path	Local...	Type
Warnings (2 items)				
No assign statement found for 'pv'	kbetrieb.st	/str12/source	line 51	Problem
No variable definition found for 'pv'	kbetrieb.st	/str12/source	line 51	Problem

Matthias@CESY.DE





- Preferences
 - Compiler Options (Linux)
 - EPICS Base
 - Colour coding

- Preparation for CSS 1.2.1
 - Collecting requirements based on the experience during the current commissioning phase with the configuration tools and the synoptic displays
 - Change requests from other CSS/SDS users
 - Merge in the SDS enhancements shown during the EPICS meeting
 - Adding DAL plugs (basic-TANGO, CA-V4?)

- Collect requirements for an EPICS Integrated Configuration Environment (EPICS-ICE)

An initial implementation by Kenneth Evans might be a good starting point
<http://aps.anl.gov/epics/eclipse/plugins/epicsIde/epicsIde.html>
- Closely following the 'Eclipse Way to the Web'

THP109: Eclipse RCP on the Way to the Web

- CSS core provides an excellent platform to integrate new applications.
- New CSS configuration and editing plug-ins were successfully used to improve the development cycles for the ongoing commissioning of the (former HERA) now FLASH cryogenic plant.
- Decoupling the definition of the I/O address space from the EPICS database configuration by unique IO_NAMES reduces the potential risk of address mismatches.
- A new EPICS ICE would help to organize the configuration of bigger installations.
- CSS 1.2.0 is now available
requirements for 1.2.1 are currently collected

Thank you for listening



CSS 1.2.0 is available from:
<http://css.desy.de>