

IDM UID 34SDZ5
VERSION CREATED ON / VERSION / STATUS 08 Feb 2013 / 4.0/ Approved
EXTERNAL REFERENCE

User Manual

CODAC Core System Overview

This document is an overview of the CODAC Core System software distribution. It is a part of the CODAC Core System documentation as well as a satellite document of the Plant Control Design Handbook (PCDH).

<i>Approval Process</i>			
	<i>Name</i>	<i>Action</i>	<i>Affiliation</i>
<i>Author</i>	Di Maio F.	08-Feb-2013:signed	IO/DG/DIP/CHD/CSD/CDC
<i>CoAuthor</i>			
<i>Reviewers</i>	Wallander A.	11-Feb-2013:recommended	IO/DG/DIP/CHD/CSD
<i>Approver</i>	Thomas P.	24-Mar-2013:approved	IO/DG/DIP/CHD
<i>Document Security: level 1 (IO unclassified)</i>			
<i>RO: Stepanov Denis</i>			
<i>Read Access</i>	AD: ITER, AD: External Collaborators, AD: Division - Control System Division - EXT, AD: Section - CODAC - EXT, AD: Section - CODAC, project administrator, RO		

<i>Change Log</i>				
<i>Title (Uid)</i>	<i>Version</i>	<i>Latest Status</i>	<i>Issue Date</i>	<i>Description of Change</i>
CODAC Core System Overview (34SDZ5_v4_0)	v4.0	Approved	08 Feb 2013	<p>Updated for version 4.0.</p> <p>Additions are marked with a bar: cubielle monitoring, COS support, SDN API, example suppression...</p> <p>Reworded to reflect that from version 4.0, version 2.0 is not anymore supported.</p>
CODAC Core System Overview (34SDZ5_v3_1)	v3.1	Approved	17 Jun 2012	<p>Changes for the 3.1 release + alignment with other documents.</p> <p>Add version 3.1 in history</p> <p>Remove PC specifications from this document while the installation manual should be the reference.</p> <p>Rename "Mini-CODAC" chapter to "CODAC Services" according to the CODAC DDD terminology.</p> <p>Separate development tools in a dedicated chapter</p> <p>Add a section on EPICS IOC development</p> <p>Add a section ITER Maven commands, insert new SVN commands.</p> <p>Remove the "Future releases" chapter. Information on new features and future releases shall be in the release notes and roadmap.</p> <p>Add the digital IO board as in 3.1.</p> <p>Update the list of documents.</p> <p>Update fig 1 with CODAC DDD schema (with DAN).</p>
CODAC Core System Overview (34SDZ5_v3_0)	v3.0	Approved	14 Feb 2012	<p>Updated for vs3.0:</p> <ol style="list-style-type: none"> 1) add the 3.0 version in history (section 1.5) 2) mention system profiles (section 2.2) 3) mention shared repository for CSS contributions (section 2.3) 4) Update RHEL version and other component versions (chapter 3) 5) add new SDD tools (parser sync) and develop

				<p>SDD web application (Chapter 4)</p> <p>6) mention new project structure and CBS (chapter 4)</p> <p>7) add icon library for BOY (section 5.3)</p> <p>8) add C/C++ and Java to Python as programming languages (section 5.6.2)</p> <p>9) extend VDCT usage with SDD parser (section 5.6.3)</p> <p>10) add fast controller sample (section 7.2)</p> <p>11) new text on future releases (chapter 8)</p> <p>12) list of documents updated</p> <p>In addition:</p> <ul style="list-style-type: none"> - Changes to make the document more version neutral - Suppression of some details (ex: RHEL licences type, user groups) - Suppression of references to version 1 and to version 1 components (EDM qualified deprecated).
CODAC Core System Overview (34SDZ5_v2_6)	v2.6	Approved	12 Jul 2011	<p>Change the recommended resolution for displays from 1400x1050 to 1280x1024.</p> <p>Include the 2.1 release and update the list of documents according to the changes in the 2.1 release.</p>
CODAC Core System Overview (34SDZ5_v2_5)	v2.5	Approved	10 Feb 2011	<p>Final for 2.0.0 & PCDH 6.1.</p> <p>Added PCDH satellite document section and SDN/TCN API.</p>
CODAC Core System Overview (34SDZ5_v2_4)	v2.4	Signed	24 Jan 2011	<p>Updated according to the final features of release 2.0.</p> <p>Updated the Self-description chapter (chapter 4), add illustrations.</p> <p>Suppress the empty CODAC IDE section (component has been de-scoped)</p> <p>Added description of System Monitoring (section 6.2)</p> <p>Updated the ICH sample section (section 7.1)</p> <p>Updated the PLC sample section (section 7.2)</p> <p>Added the plant system I&C health monitoring</p>

				(section 7.3). Updated all figures from latest code.
CODAC Core System Overview (34SDZ5_v2_3)	v2.3	Signed	07 Jan 2011	Version for review. Includes changes from J.Poole.
CODAC Core System Overview (34SDZ5_v2_2)	v2.2	Signed	16 Dec 2010	Develop the features, integrate internal comments. For beta4.
CODAC Core System Overview (34SDZ5_v2_1)	v2.1	In Work	30 Nov 2010	2nd draft for inclusion into 2.0b3.
CODAC Core System Overview (34SDZ5_v2_0)	v2.0	In Work	24 Sep 2010	1st draft for version2.0.
CODAC Core System Overview (34SDZ5_v1_5)	v1.5	Approved	19 Jul 2010	Minor changes to cope with the existence of vs 1.1.
CODAC Core System Overview (34SDZ5_v1_4)	v1.4	Approved	12 Feb 2010	Minor editions + added last section on user doc.
CODAC Core System Overview (34SDZ5_v1_3)	v1.3	Signed	10 Feb 2010	Updated from comments on vs 1.2
CODAC Core System Overview (34SDZ5_v1_2)	v1.2	Signed	05 Feb 2010	Version for review.
CODAC Core System Overview (34SDZ5_v1_1)	v1.1	In Work	13 Jan 2010	No changes, just PDF version
CODAC Core System Overview (34SDZ5_v1_0)	v1.0	In Work	11 Dec 2009	

Table of contents

1	INTRODUCTION	2
1.1	Purpose	2
1.2	Scope	2
1.3	Reference Documents	3
1.4	Acronyms	3
1.5	History	3
1.6	PCDH Context	4
2	USERS, DISTRIBUTION AND SUPPORT	5
2.1	ITER users	5
2.2	Distribution for ITER users	5
2.3	Public users	5
2.4	User support	6
2.5	Release scheme	6
3	SYSTEM	7
3.1	RHEL	7
3.2	EPICS	7
3.3	Other system components	7
4	SELF-DESCRIPTION DATA TOOLKIT	8
5	CODAC SERVICES	10
5.1	Mini-CODAC	10
5.2	Control System Studio	10
5.3	HMI	10
5.4	Alarms	11
5.5	Archiving	11
6	DEVELOPMENT TOOLS	12
6.1	EPICS IOC development	12
6.2	ITER Maven commands	12
6.3	SNL Sequence Editor	13
6.4	VDCT	13
6.5	Support for C/C++, Java and Python	14
7	PSH AND FAST CONTROLLERS	14
7.1	PLC integration	14
7.2	System Monitoring	14
7.3	Timing and Synchronization	15
7.4	Support for I/O boards	15
7.5	Application Programming Interfaces for SDN (version 4)	16
7.6	Common Operating State (COS) support (version 4)	16
8	DOCUMENTATION	17

1 INTRODUCTION

1.1 Purpose

This document is a public document to provide the ITER users with an overview on the software package named *CODAC Core System* which is distributed to the suppliers in charge of the development and delivery of the Instrumentation and Control (I&C) for the ITER plant systems.

This document is also a satellite document for the ITER Plant Control Design Handbook (PCDH) as a reference for the I&C software environment.

1.2 Scope

CODAC designates the central control system responsible for operating ITER.

The different plant systems that constitute the ITER device will be driven by local instrumentation and control (I&C) designated as *plant system I&C*.

The plant system I&C contains local controllers that are either PLCs or rack mounted computers controlling PCIe I/O and designated as *fast controllers*.

Each plant system I&C also includes one *Plant System Host (PSH)*, supplied by IO and implementing standard functions not requiring plant system specific software.

During development and tests, the CODAC central infrastructure, that will include servers and operator workstations, is replaced with a computer designated as *Mini-CODAC* which implements a subset of the CODAC functions. The plant system I&C physical architecture is illustrated in Figure 1.

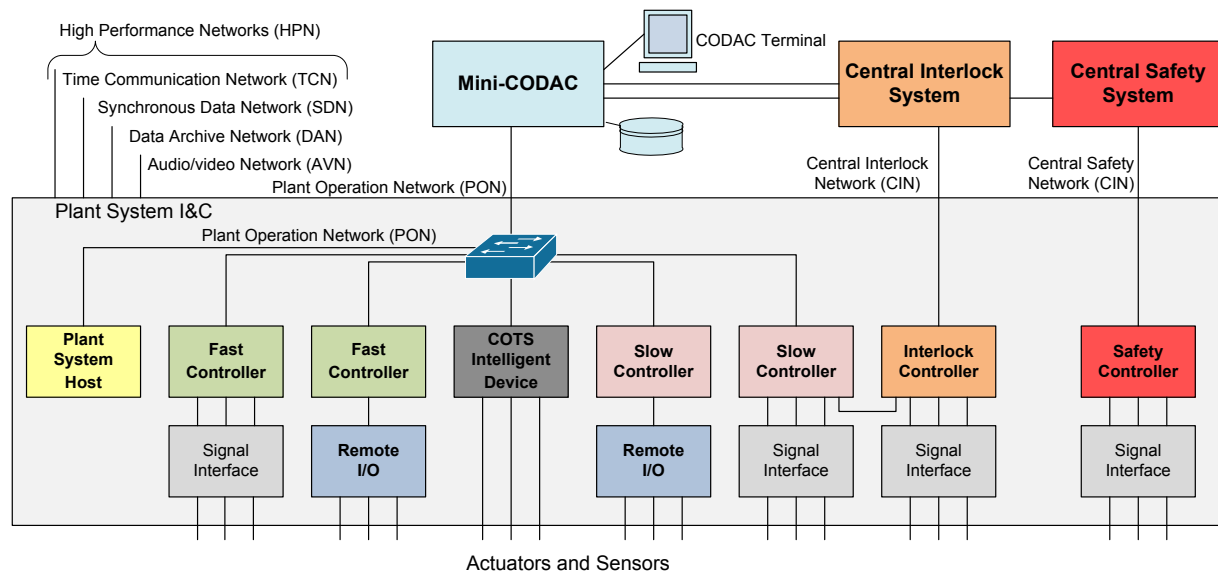


Figure 1 – Plant System I&C Physical Architecture

The *CODAC Core System* is a software package that is distributed by the IO CODAC Section for the development of the plant system I&C. It includes the software for Mini-CODAC and PSH and it provides the plant system I&C developers with the environment required to develop and test the software in a way that complies with the ITER requirements.

The software is based on **EPICS** [RD1], an open-source control system that is widely used in experimental physics. The software of PSH and fast controllers is based on the EPICS framework and utilities. The Mini-CODAC software is based on EPICS tools, developed within the EPICS community and extended by ITER.

In this version, statements that only apply to the latest major version (from 4.0) are signalled with a bar in the left margin.

1.3 Reference Documents

- [RD1] EPICS - Experimental Physics and Industrial Control System
www.aps.anl.gov/epics/
- [RD2] Plant Control Design Handbook ([ITER_D_27LH2V](#))
 Downloadable from <https://www.iter.org/org/team/chd/cid/codac>
- [RD3] CODAC Core System public web page:
<https://www.iter.org/org/team/chd/cid/codac/Pages/CoreSystem.aspx>
- [RD4] CODAC DDD ([ITER_D_6M58M9](#))
- [RD5] EPICS Application Developer's Guide
<http://www.aps.anl.gov/epics/base/R3-14/12-docs/AppDevGuide/>

1.4 Acronyms

COS	Common Operating State
CSS	Control System Studio
HMI	Human-Machine Interface
I&C	Instrumentation and Control
I/O	Input / Output
IO	ITER Organization
NI	National Instrument
PCI	Peripheral Component Interconnect
PCIe	PCI Express
PCDH	Plant Control Design Handbook [RD1]
PLC	Programmable Logic Controller
PSH	Plant System Host
RHEL	Red Hat Enterprise Linux
RO	Responsible Officer
SDD	Self-description Data
TCN	Time Communication Network

1.5 History

- ◆ 15-Feb-2010 - Version 1.0, preliminary release with first PLC integration and SDD tools.
- ◆ 22-Jul-2010 - Version 1.1, minor improvements of packaging and SDD components.
- ◆ 15-Feb-2010 - Version 2.0, stabilized release for developers, introducing the new environment for SDD and Mini-CODAC, stable data model and PLC interface.
- ◆ 13-Jul-2011 - Version 2.1, minor release to extend the support for fast controllers, to update the Control System Studio toolkit (BOY) and to distribute many bug fixes.
- ◆ 15-Feb-2011 - Version 3.0, consolidated release for I&C development: RHEL vs 6, project structure, updated components (ITER and external).
- ◆ 22-Jun-2012 – Version 3.1, minor release with new features in Control System Studio tools, SDD tools and Maven commands and with the support of a new digital I/O board.
- ◆ 15-Feb-2013 – Version 4.0: improvements in Control System Studio tools, SDD tools and Maven commands, addition of the SDN API, support for Common Operating State (COS) and support for redundant PLC.

From version 3.0, tools have been included in the distribution to allow the migration from one version to the next one (i.e. migration from 2.x to 3.x or 3.x to 4.x).

At the time the version 4.0 is released, the support for the 3.x versions will be maintained but the support for 2.x versions will be limited to assistance for migration. No support is provided for 1.x versions.

1.6 PCDH Context

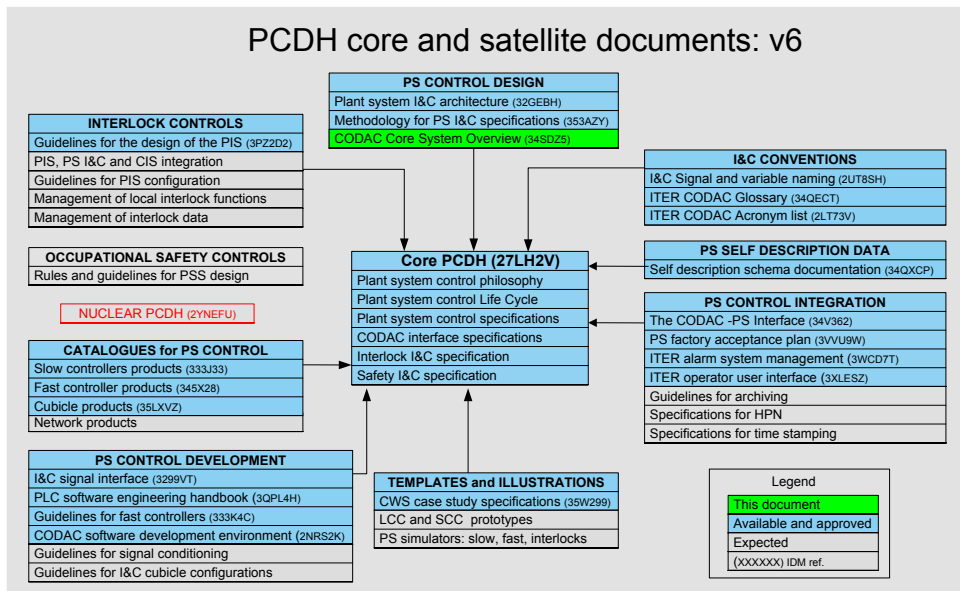


Figure 2 - PCDH Documentation Structure

The Plant Control Design Handbook (PCDH) [RD1] defines methodology, standards, specifications and interfaces applicable to ITER Plant Systems instrumentation and control (I&C) system life cycle. I&C standards are essential for ITER to:

- Integrate all plant systems into one integrated control system;
- Maintain all plant systems after delivery acceptance;
- Contain cost by economy of scale.

PCDH comprises a core document which presents the plant system I&C life cycle and recaps the main rules to be applied to the plant system I&Cs for conventional controls, interlocks and safety controls. Some I&C topics will be explained in greater detail in dedicated documents associated with PCDH as presented on Figure 2. This document is one of them.

2 USERS, DISTRIBUTION AND SUPPORT

2.1 ITER users

Users who contribute to the development of ITER I&C System such as ITER Domestic Agencies, institutes associated with them or industry working for ITER through contracts may become registered ITER community users, or in short, *registered users*.

The procedure for becoming a registered user is described in the ITER CODAC public web pages [RD3] and consists of completing and submitting a form which includes the name of an IO contact person and a justification. Requests are handled by the IO CODAC RO.

Registered users have access to all official ITER software releases and to full support services.

2.2 Distribution for ITER users

The software distribution is implemented through a server in the IO premises that provides all the required software packages (RPM packages on a YUM repository) for configuring the registered user's computers. Each computer on which the software must be installed has to be registered.

Different profiles are provided in order to allow configuring the different systems: Mini-CODAC, PSH, fast controller or CODAC terminal. Each profile is available in two variants, one for development and one for operation.

The documentation is included in the software distribution.

From version 4.0, the examples for I&C projects simulating some ITER equipment, such as ICH or CWS, are not anymore in the CODAC Core System distribution.

These are now distributed separately at the following link: <https://portal.iter.org/IPT/IC-IPT/Pages/IandC-Use-Cases.aspx>.

Only the examples that are provided as part of the user documentation for a component (ex: PLC driver, PXi-6259 EPICS interface...) are now in the distribution.

2.3 Public users

The ITER extensions to EPICS are made available for anybody outside of the ITER community that has an interest in these components.

These extensions are available as a public distribution by means of a zipped tar file downloadable from the ITER CODAC public web pages [RD3].

The EPICS packages that are used by ITER, as described in this document shall be installed separately.

It should be noted that the ITER software is developed with, and only tested with, the RHEL version and the versions of EPICS core and extensions that have been adopted in the ITER distribution.

EPICS users should also take into account that it is an ITER's objective to provide an integrated environment. As a result, there are some dependencies between ITER packages and imported packages. However, whenever possible, new ITER packages, such as I/O board support software, are intended to be shareable. Some ITER extensions, such as those in the Control System Studio applications, are also submitted in EPICS shared repositories (e.g. SourceForge) and should be obtained from these.

2.4 User support

A support service is available for all registered users. The support is available by e-mail and the service will operate continuously during IO opening hours.

An issue handling system, based on Bugzilla, has been set-up to keep track of reported bugs, known issues or planned extensions. This is the main tool for the follow-up of the enhancements and bug fixes, and for the planning of the releases.

Training is now organized in Cadarache on a regular base for ITER I&C developers.

From version 3, migration tools have been introduced to facilitate the migration of the I&C applications to the latest release of the distribution.

2.5 Release scheme

All releases are labelled with a sequence of 3 numbers that identify the major version, the minor release, starting from 0 for each major version, and the maintenance release starting from 0 for each minor release. With respect to software packages and documentation, the version identifier includes the minor release number (e.g. 2.0).

Examples: "2.0.1" has been a maintenance release for the 2.0 major release and 2.1 has been a minor release of version 2.

The release plan is the following:

- ◆ Major version: one version each year, in Q1 (February)
- ◆ Minor releases: zero to two releases, in Q2 and/or Q3, only if new features are required.
- ◆ Maintenance releases: only if bug fixes need to be distributed to many users. For single user, patch distribution will be supported.

Beta releases are produced before any official version but these can only be distributed to testers shall not be used for I&C development.

Two or more different versions can be installed on one computer, provided the RHEL version is the same, but only one is activated at a time. This feature is provided mainly to facilitate migration to the latest version by allowing easy reversibility.

It is strongly recommended to migrate to the latest version of the IO distribution. Tools are included for upgrading systems (from version 3.0) and for migrating applications. The support team can also assist users to facilitate such migration.

With each release (from version 3.1), the list of enhancement and bug fixes is recorded. This list is published in the release notes and the details can be obtained from the support. There is no retrofitting of bug fixes to previously released versions but patches can be provided if justified.

3 SYSTEM

3.1 RHEL

Red Hat Enterprise Linux (RHEL) is the standard operating system for Mini-CODAC, PSH and fast controllers. An officially supported version of RHEL is required for these systems. The operating system is included in the distribution and the first part of the installation will install the operating system on any target computer.

The architecture supported is linux-x86_64.

RHEL 6 has been introduced with version 3: RHEL 6.1 in version 3.x and RHEL 6.3 in version 4.x.

The real-time enabled version, MRG-Realtime (MRG-R), is also supported for fast controllers: MRG-R 2.0 in version 3.x and MRG-R 2.1 from version 4.0.

3.2 EPICS

The EPICS base is included in the distribution and is required for Mini-CODAC, PSH and fast controllers. The EPICS framework is the base for the fast controllers and PSH, and the EPICS communication protocol, Channel Access (CA) is the standard communication protocol for access to plant system I&C over the Plant Operation Network (PON).

EPICS version 3.14.12 has been introduced with version 2.

The following EPICS components are included:

- ◆ State Notation Language (SNL), for the development of state machines in PSH, fast controllers and mini-CODAC.
- ◆ VDCT (Visual Database Configuration Tool), for edition of the EPICS configuration files.
- ◆ Autosave for saving/reloading the configuration values to/from disks when persistency is required.
- ◆ The standard EPICS IOC error logging, for error and trace logging.

3.3 Other system components

The following software components are included and configured in the distribution:

- ◆ A relational database, built with PostgreSQL, for storing the plant system I&C data on Mini-CODAC.

- ◆ Eclipse, with plug-ins for Java, Python, SVN, and with all required plug-ins for Control System Studio.
- ◆ Apache ActiveMQ as the JMS implementation, as required for Control System Studio
- ◆ The Spring Framework and Hibernate, as required for SDD tools.
- ◆ The Apache Tomcat servlet container for web applications.

4 SELF-DESCRIPTION DATA TOOLKIT

Self-Description Data (SDD) is an ITER concept designating the static data that configure the plant system I&C. This data is produced by the plant system I&C designers and developers, using IO tools and according to an IO defined schema.

The data will be delivered by the plant system I&C developers into IO database upon software delivery. During development, it is stored in a local database (the SDD database) on a mini-CODAC system.

The SDD toolkit has been developed by IO in order to allow the user to configure the plant system I&C.

This toolkit now includes:

- ◆ The **SDD editor** (Figure 3) and the **SDD web application** (Figure 4) to define the plant system interface, the I&C components, the interfaced signals and to configure variables, alarms, archiving, etc.
The editor is an Eclipse RCP application while the web application is usable from a web browser.
- ◆ The **SDD translator** to convert the SDD into the required EPICS configuration data for Mini-CODAC, PSH and fast controllers and into the required STEP-7 files for PLCs. The translator is integrated with the editor and web application.
- ◆ The **SDD sync** tool to save and loading SDD data to/from XML files and to synchronize local SDD databases with IO databases as well as local files with the IO source repository.
- ◆ The **SDD parser** to parse user provided or user modified EPICS configuration files (EPICS record definition) and retrofit changes into the SDD database.

The SDD tools provide the user with creation, editing and saving features for:

- The list of signals interfaced by the plant system I&C
- The list of functions and variables implemented by the plant system I&C
- The list of control units (PSH, controllers) that belongs to the plant system I&C
- The communication between PSH and PLC
- The configuration for alarms
- The configuration for archiving
- The configuration for HMIs
- The configuration for the supported I/O boards
- The cubicles that shall be monitored (from 4.0)
- The mapping of Common Operating State variables into plant-system specific ones (from 4.0)

A project structure has been introduced in version 3, with migration tools for version 2 data. It allows creating I&C projects that correspond to a single delivery (ex: one Procurement Arrangement). An I&C project can cover more than a single plant system I&C.

The SDD tools enforce the usage of the Control Breakdown Structure (CBS) defined in the CODAC System Design Description (DDD) [RD4] and are configured with the official nodes of the 1st level.

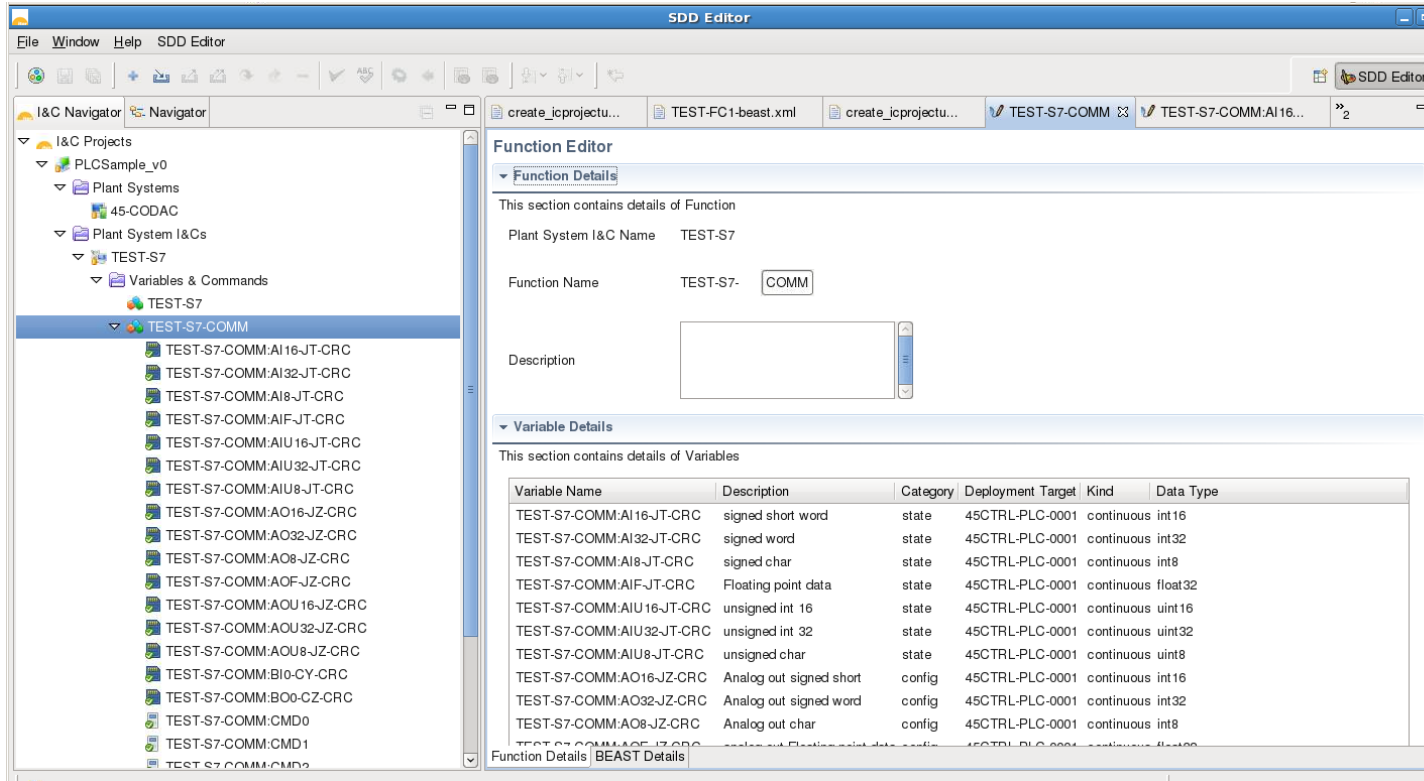


Figure 3 - SDD Editor: List of variables in a function of a plant system I&C.

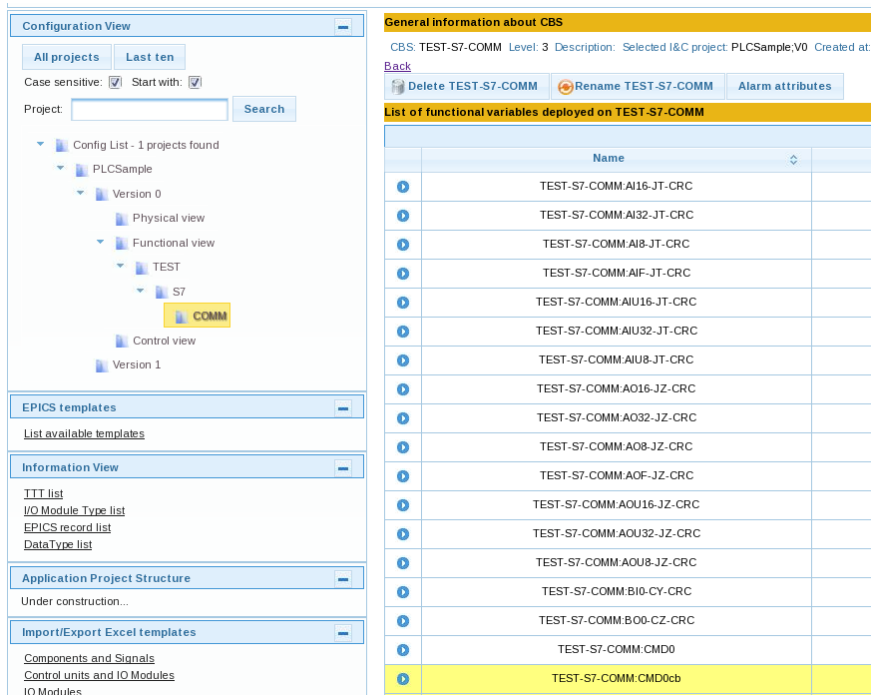


Figure 4 - SDD Web Application: variables in a function of a plant system I&C.

5 CODAC SERVICES

5.1 Mini-CODAC

During plant system I&C development, the Mini-CODAC system has two roles:

- ◆ A local CODAC server implementing the following CODAC service, with reduced functionality
 - ◇ HMI,
 - ◇ alarms handling,
 - ◇ data archiving,
 - ◇ error & trace logging.
- ◆ A development server with the SDD database and other shared file or file systems.

Developments and test, including Factory Acceptance Tests (FAT), shall also be executed with Mini-CODAC as a replacement for CODAC.

After integration, the CODAC services will be provided from CODAC servers and the development tools shall be used from on-site systems for the maintenance tasks. The configuration data produced with Mini-CODAC will be re-used in this final configuration.

5.2 Control System Studio

Eclipse has been adopted as the platform for Mini-CODAC software.

Control System Studio (CSS) is a recent Eclipse-based suite of tools developed within the EPICS community. It provides an integrated environment based on the Eclipse plug-in technology and therefore is easily extensible.

New tools developed in this environment have been adopted for the Mini-CODAC services: HMI, alarms, and archives.

5.3 HMI

An EPICS display manager is a tool that allows the development of dynamic displays by means of an interactive editor and the run-time management of such dynamic displays.

The new display manager, **BOY**, developed with the Control System Studio platform, has been included in the distribution since version 2. In addition to its integration into the CSS environment, BOY offers a rich set of graphical widgets and dynamic configuration features that can both be extended and customized to satisfy future requirements.

A dedicated library of symbols with graphical widgets displaying these with dynamic attributes has been added for ITER applications (I&C HMIs).

BOY is continuously extended by its users within a collaboration. New features are added in each release. Extensions are now listed in the release notes.

From version 4/0, it is also possible to start BOY displays in a web browser.

Note:

EDM (Extensible Display Manager), a display manager using the X-window technology, was included in the distribution in version 1 and is maintained in the distribution to give the possibility of running existing EDM screens. However, EDM is deprecated and should not be used for developing operator displays.

5.4 Alarms

The new alarm system, **BEAST**, developed with the Control System Studio platform, has been included in the distribution since version 2.

BEAST is a distributed alarm system consisting of alarm servers that monitor alarms from the EPICS IOC processes via Channel Access and user interface for viewing the current alarms as well as acknowledging alarms and browsing the alarm history. Alarms logs and configuration are stored in the Mini-CODAC relational database. Figure 5 is an illustration of the new alarm display.

The screenshot shows the Control System Studio (ITER) interface. The main window displays a list of alarms in a table format. The table has columns for SEQ, TIME, DELTA, TYPE, TEXT, NAME, STATI, SEVER, CURRI, VAL, CREATETIM, USER, HOST, and APPLICATION-II. The current alarms section shows several entries with 'MINOR' severity and 'STATE_AL OFF' status. The acknowledged alarms section shows one entry with 'minor-ack' status and 'STATE_AL ON' status.

SEQ	TIME	DELTA	TYPE	TEXT	NAME	STATI	SEVER	CURRI	VAL	CREATETIM	USER	HOST	APPLICATION-II
1	2011/01/05 15:37:08.003	00:00:0	alarm	STATE	RF-ICH STATE	MINOR	OK	OFF	utzeln apps1	AlarmServer			
2	2011/01/05 15:37:07.999	00:00:0	alarm	STATE	RF-ICH STATE	MINOR	OK	OFF	utzeln apps1	AlarmServer			
3	2011/01/05 15:37:06.003	00:00:0	alarm	STATE	RF-ICH STATE	MINOR	OK	OFF	utzeln apps1	AlarmServer			
4	2011/01/05 15:37:05.999	00:00:0	alarm	STATE	RF-ICH STATE	MINOR	OK	OFF	utzeln apps1	AlarmServer			
5	2011/01/05 15:37:04.621	00:00:0	alarm	STATE	RF-ICH STATE	MINOR	OK	OFF	utzeln apps1	AlarmServer			
6	2011/01/05 15:37:04.618	00:00:0	alarm	STATE	RF-ICH STATE	MINOR	OK	OFF	utzeln apps1	AlarmServer			
7	2011/01/05 15:37:04.615	00:00:0	alarm	STATE	RF-ICH STATE	MINOR	OK	OFF	utzeln apps1	AlarmServer			

PV	Description	Alarm Time	Alarm Sev	Alarm Sta	Alarm Val	Current Se	Current St
RF-ICH1-RS1:HFB12-CY-C	RF Power Validation ON/OFF Stat	2010/12/21 17:16:02	MINOR	STATE_AL OFF	OK	OK	OK
RF-ICH1-RS1:HFB11-CY-C	RF Power Validation ON/OFF Stat	2011/01/05 15:32:54	MINOR	STATE_AL OFF	OK	OK	OK
RF-ICH1-RS1:BDC123-C2	DC Screen Grid Trig HVA2	2010/12/21 17:01:01	MINOR	STATE_AL OFF	OK	OK	OK
RF-ICH1-RS1:BDC113-C2	DC Screen Grid Trig HVA1	2010/12/21 17:00:59	MINOR	STATE_AL OFF	OK	OK	OK

PV	Description	Alarm Time	Alarm Sev	Alarm Sta	Alarm Val	Current Se	Current Status
RF-ICH1-RS1:SIMULATION	IOC Simulation Mode (ON/OFF)	2010/12/21 17:09:32	minor-ack	STATE_AL ON	MINOR	STATE_ALARM	

Figure 5 – Alarm handler (BEAST) user interface

From version 4.0, alarm descriptions can be converted to speech.

5.5 Archiving

The new data archiver, named **BEAUTY**, has been included in the distribution since version 2. It is also a Control System Studio application. The data are stored into a relational database on the Mini-CODAC system.

A data viewer, the **CSS DataBrowser**, is available in Control System Studio that allows retrieval of archived data as well as monitoring of live data.

Data archiving requires that the set of variables to be stored is configured with SDD tools.

From version 4.0, it is also possible to visualize archive data in a web browser.

6 DEVELOPMENT TOOLS

6.1 EPICS IOC development

EPICS includes a framework for developing control processes, named IOC process. The IOC acronym (Input Output Controller) designates controllers in the EPICS community. Each IOC process is built as a database of records that belong to a set of pre-defined record types.

All EPICS development tools and procedure described in the EPICS Application Developer's Guide [RD5] are valid. However, the EPICS build tools have been encapsulated into ITER specific commands (cf. ITER Maven commands, below) and the development of EPICS configuration files should be done with the SDD tools in order to satisfy the requirements for integration. This includes naming conventions and convention on files that imported EPICS software shall satisfy.

6.2 ITER Maven commands

I&C projects are developed using an ITER specific workflow that is supported by the SDD tools and by commands implemented using the Apache Maven build tool

The development workflow is composed of the following steps:

- 1) Edition of the I&C project with the SDD Editor and/or web application.
- 2) Generation of the EPICS/CSS/STEP-7 configuration files with the SDD translator.
- 3) Creation/update of the software unit with dedicated commands. The sequence of calls is produced by the SDD translator, according to the I&C project definition.
- 4) Edition of user-defined files with test editor or specific editors, such as the CSS SNL editor and VDCT.
- 5) Compilation of the EPICS IOC processes and, from version 4.0, of the real-time programs with the Maven *compile* command.
- 6) Test of the project with the *start*, *stop*, *status* and *test* commands.
- 7) Creation of the software packages for distribution with the *package* command.

From version 3.1, new commands (*checkout*, *commit*, *import*, *update*) allows saving, restoring and synchronizing the project's files with a SVN source repository, either the ITER one or a local one during off-line development phases.

From version 4.0, a graphical tool, the maven-editor, provides the user with a graphical HMI for executing these commands. This is also integrated in SDD tools so user can build/test/package the applications from the SDD editor or the SDD web application.

6.3 SNL Sequence Editor

The **SNL Sequence Editor**, a standard EPICS tool, is included to allow the development of EPICS state machines using the EPICS SNL language.

A language sensitive editor as well as the environment for testing the sequence and building a graphical representation of a state diagram is included in Control System Studio, as illustrated in Figure 6.

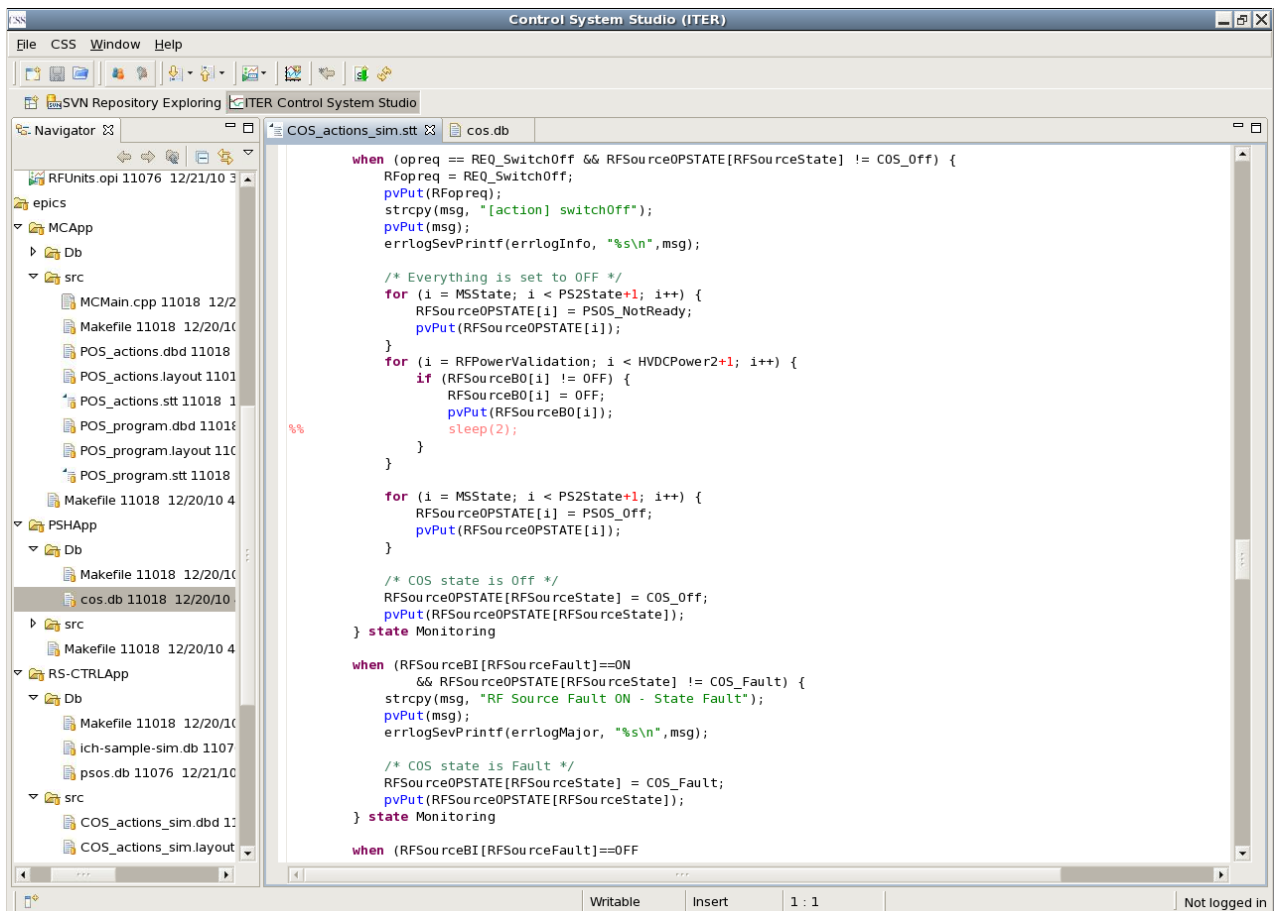


Figure 6 – SNL Sequence Editor (Control System Studio)

SNL programs can be developed and added to EPICS IOC processes for implementing state machine, such as plan-system operating state (PSOS) management on a fast controller.

6.4 VDCT

VDCT (Visual Database Configuration Tool), is a widely used tool for editing the EPICS database in the EPICS community.

This tool is included in the distribution. The recommended ITER tools for editing records are SDD tools, not VDCT, but it is possible to edit SDD-generated record files using VDCT, or other editor, and retrofit the changes into the SDD database using the SDD parser.

6.5 Support for C/C++, Java and Python

C and C++ are the programming languages for controller-level software and for real-time applications, Java is the main language for Mini-CODAC software and Python is the recommended language for rapid prototyping and for testing.

For each of these languages, the Channel Access client library is provided as well as Eclipse configuration for development. Example will be provided in each of these languages.

7 PSH AND FAST CONTROLLERS

7.1 PLC integration

PLCs used in plant system I&C shall be SIMATIC S7 PLCs and these are programmed using the Siemens development tool, SIMATIC STEP 7.

The CODAC Core system distribution includes the tools to build the EPICS software that is necessary for integrating the PLCs into the CODAC infrastructure. The communications with PLCs are implemented by an EPICS IOC process deployed, by default, on the PSH, alternatively on a fast controller.

These communications consist of synchronizing EPICS variables on the PSH with the PLC variables. The PLC variables must be defined using the SDD tools. Each PLC command or variable is mapped into an EPICS process variable in the PSH and into a program variable in the PLC.

Automatically generated data are also inserted in the frames transmitted (header, trailer, length, version, timestamp, heartbeat) in order to detect communication errors and data inconsistencies. This data is also generated by SDD translator.

Support for redundant PLC (ex: interlock systems) has been added from version 4.0. This includes the transparent connection to the active master CPU and the reconnection to the new one in case a switch is triggered because of a failure.

7.2 System Monitoring

System monitoring features are included for monitoring the status of the PSH and fast controllers as well as the status of the control tasks (EPICS IOC processes) running in these control units.

Each IOC process is built with a monitoring module that provides, by means of automatically generated PVs, information about the process (process ID, start time, port used...) and its resource usage (CPU, memory, registered CA clients ...). Each process also provides a heartbeat counter to detect missing or blocked process.

In addition, for each PSH and controller, an IOC process is built to provide information about the computer (boot time, software version) and its resources (CPU load, memory usage, file descriptors...). A set of automatically generated PV also provides user with an interface to these data.

Diagnostic displays using these variables can be developed or generated in an automatic manner. The system monitoring variables are also intended to be used for generating alarms when errors are detected on processes or computers.

From version 4.0, the monitoring of the cubicles environment (ex: temperature), as implemented by dedicated PLCs in standard cubicles, is also interfaced via such variables.

7.3 Timing and Synchronization

IEEE-1588, also called Precision Time Protocol (PTP), has been selected for the Time Communication Network (TCN).

PCI/PXI boards compliant with this protocol are supported for fast controllers as well as PCI boards for PSH and Mini-CODAC.

The timing and synchronization board for fast controller is the **N.I. PXI-6682** board. The software for this board is included in the distribution.

It contains:

- ◆ The Linux device driver
- ◆ The EPICS device support
- ◆ SDD extensions for the board
- ◆ Documentation and examples

The following features are supported:

- ◆ Accurate system time using PTP. The accuracy depends on the local infrastructure. With a proper master clock and distribution, it will be better than one micro-second, as per the 1588 standard. On the ITER TCN network, the accuracy will be 50 ns RMS.
- ◆ Generation of events and clock signals using the precise time distributed by TCN.
- ◆ Time-stamping incoming signals.

7.4 Support for I/O boards

I/O boards selected for fast controllers by IO will be integrated into the CODAC core system software. The distribution will include Linux driver, EPICS driver, SDD integration, test programs and documentation.

The software will follow the evolution of the ITER catalogue that specifies such I/O boards. This catalogue is a satellite document of PCDH [RD1].

The first supported board is a multi-function data acquisition board, **NI PXI-6259**, that has following channels:

- ◆ 16/32 Analog Input channels (16 bits), 1 MSamples/s total for multi-channel, 1.25 MSamples/s for a single channel
- ◆ 4 Analog Output channels (16bits), 2.8 Ms/s each
- ◆ 48 bi-directional Digital channels
- ◆ 4 counters (32 bits)

From version 3.1, a digital I/O board, **NI PXI-6528**, is also supported with the following channels:

- ◆ 24 optically isolated input channels
- ◆ 24 solid-state relay output channels.

7.5 Application Programming Interfaces for SDN (version 4)

A draft version of the Application Programming Interfaces (APIs) for the SDN network has been distributed in version 2.x and 3.x. This is now deprecated.

From version 4.0, the 1st version of the final SDN API is included in the distribution. This is a library for C++ programs. It allows defining messages to be exchanged via the SDN network as well as publishing such messages or subscribing to those.

7.6 Common Operating State (COS) support (version 4)

COS variables are a mandatory part of the plant system I&C interface with CODAC. It provides the supervision system with a standard interface for each plant system I&C. The variables allow acquiring the current state and requesting a transition.

The COS variables are automatically added to any plant system I&C and the implementation is configured by means of a mapping table between the generic COS variables and the specific ones that the I&C developer should implement. These specific variables can be implemented either on a fast controller or on a PLC.

8 DOCUMENTATION

The following list of document is included in the distributions. These are also available from the ITER documentation management system (IDM). Versions are given in the release notes.

Overview

ITER_D_34SDZ5	CODAC Core System Overview (This document)
ITER_D_DUJH86	CODAC Core System Version 4 Release Notes

Installation & Support

ITER_D_33JNKW	CODAC Core System Installation Manual
ITER_D_7JCFUD	CODAC Core System Migration Guide
ITER_D_33T8LW	CODAC Core System Development Manual
ITER_D_763KTT	CCS Public Release Installation Manual

User Manuals

ITER_D_43PSH9	CODAC Core System User Manual
ITER_D_32Z4W2	SDD Editor User Manual
ITER_D_42NFCY	SDD Web Application User Guide
ITER_D_46AAXR	SDD Synchronization Guide
ITER_D_7MT2YC	Maven Editor User Guide (version 4 only)

Development & Testing

ITER_D_43PSH9	CODAC Core System User Manual
ITER_D_B7SKFU	SDN Software User Manual (version 4 only)
ITER_D_7367JQ	OPI - CSS BOY Edition & Runtime (version 4 only)
ITER_D_A69URK	OPI - CSS Symbol Library (version 4 only)

I/O Modules support

ITER_D_6KFJEJ	S7 PLC EPICS Driver Extensions
ITER_D_2PLQ4P	NI Sync Linux Device Driver User's Guide
ITER_D_33Q5TX	NI Sync EPICS Driver User's Guide
ITER_D_32GTJY	NI PXI-6259 Linux Driver User's Guide
ITER_D_3DEY52	NI PXI-6259 EPICS Driver User's Guide

ITER_D_3ZHxQ9	NI PXI-6528 Linux Device Driver User's Guide
ITER_D_433VEW	NI PXI-6528 EPICS Driver User's Guide

Samples

ITER_D_2N8C3M	PLC Sample Application Guide
---------------	------------------------------