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).
## Change Log

### CODAC Core System Overview (34SDZ5)

<table>
<thead>
<tr>
<th>Version</th>
<th>Latest Status</th>
<th>Issue Date</th>
<th>Description of Change</th>
</tr>
</thead>
<tbody>
<tr>
<td>v1.0</td>
<td>In Work</td>
<td>11 Dec 2009</td>
<td></td>
</tr>
<tr>
<td>v1.1</td>
<td>In Work</td>
<td>13 Jan 2010</td>
<td>No changes, just PDF version</td>
</tr>
<tr>
<td>v1.2</td>
<td>Signed</td>
<td>05 Feb 2010</td>
<td>Version for review.</td>
</tr>
<tr>
<td>v1.3</td>
<td>Signed</td>
<td>10 Feb 2010</td>
<td>Updated from comments on vs 1.2</td>
</tr>
<tr>
<td>v1.4</td>
<td>Approved</td>
<td>12 Feb 2010</td>
<td>Minor editions + added last section on user doc.</td>
</tr>
<tr>
<td>v1.5</td>
<td>Approved</td>
<td>19 Jul 2010</td>
<td>Minor changes to cope with the existence of vs 1.1.</td>
</tr>
<tr>
<td>v2.0</td>
<td>In Work</td>
<td>24 Sep 2010</td>
<td>1st draft for version 2.0.</td>
</tr>
<tr>
<td>v2.1</td>
<td>In Work</td>
<td>30 Nov 2010</td>
<td>2nd draft for inclusion into 2.0b3.</td>
</tr>
<tr>
<td>v2.2</td>
<td>Signed</td>
<td>16 Dec 2010</td>
<td>Develop the features, integrate internal comments. For beta4.</td>
</tr>
<tr>
<td>v2.3</td>
<td>Signed</td>
<td>07 Jan 2011</td>
<td>Version for review. Includes changes from J.Poole.</td>
</tr>
<tr>
<td>v2.4</td>
<td>Signed</td>
<td>24 Jan 2011</td>
<td>Updated according to the final features of release 2.0. Updated the Self-description chapter (chapter 4), add illustrations. Suppress the empty CODAC IDE section (component has been de-scoped) Added description of System Monitoring (section 6.2) Updated the ICH sample section (section 7.1) Updated the PLC sample section (section 7.2) Added the plant system I&amp;C health monitoring (section 7.3). Updated all figures from latest code.</td>
</tr>
<tr>
<td>v2.5</td>
<td>Approved</td>
<td>10 Feb 2011</td>
<td>Final for 2.0.0 &amp; PCDH 6.1. Added PCDH satellite document section and SDN/TCN API.</td>
</tr>
<tr>
<td>v2.6</td>
<td>Approved</td>
<td>12 Jul 2011</td>
<td>Change the recommended resolution for displays from 1400x1050 to 1280x1024. Include the 2.1 release and update the list of documents according to the changes in the 2.1 release.</td>
</tr>
<tr>
<td>v3.0</td>
<td>Approved</td>
<td>14 Feb 2012</td>
<td>Updated for vs3.0: 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 SDD web application (Chapter 4) 6) mention new project structure and CBS (chapter 4) 7) add icon library for BOY (section 5.3) 8) add C/C++ and Java to Python as programming languages (section 5.6.2) 9) extend VDCT usage with SDD parser (section 5.6.3) 10) add fast controller sample (section 7.2) 11) new text on future releases (chapter 8) 12) list of documents updated In addition: - 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).</td>
</tr>
<tr>
<td>v3.1</td>
<td>Approved</td>
<td>17 Jun 2012</td>
<td>Changes for the 3.1 release + alignment with other documents. Add version 3.1 in history Remove PC specifications from this document while the installation manual should be the reference. Rename &quot;Mini-CODAC&quot; chapter to &quot;CODAC Services&quot; according to the</td>
</tr>
<tr>
<td>Version</td>
<td>Approved Date</td>
<td>Updates</td>
<td></td>
</tr>
<tr>
<td>---------</td>
<td>---------------</td>
<td>---------</td>
<td></td>
</tr>
<tr>
<td>v4.0</td>
<td>08 Feb 2013</td>
<td>Updated for version 4.0. Additions are marked with a bar: cubiclle monitoring, COS support, SDN API, example suppression... Rworded to reflect that from version 4.0, version 2.0 is not anymore supported.</td>
<td></td>
</tr>
<tr>
<td>v4.1</td>
<td>03 Jul 2013</td>
<td>Updated for the release of CODAC Core System vs 4.1 (July 2013). Addition of the 4.1 version in the History section (section 1.5) Addition of the handling of SDN data and SDN programs in SDD (section 4 and 7.5) Addition of web browser interface for alarms (section 5.4) Changes in version 4.0 and 4.1 and indicated with a left margin bar.</td>
<td></td>
</tr>
<tr>
<td>v4.2</td>
<td>10 Feb 2014</td>
<td>Minor changes for the CCS 4.2.0 release: - Update figure 1 with CODAC PDR version - Add version 4.2 in history - Add list of additional CSS tools (section 5.6) - Add support for NI FlexRIO (section 7.4) - Add section on DAN support (section 7.7) - Add SPSS and NI RIO documents in the list of documentation (chapter 8)</td>
<td></td>
</tr>
<tr>
<td>v4.3</td>
<td>26 Jun 2014</td>
<td>Minor update for the 4.3.0 release: - Add version 4.3 in history - Expand Maven editor (figure) - Removed VDCT - Confirm &amp; clarify the 1 micro-second accuracy for system time with TCNd - Added PXIe-63868 support - Update list of document (chapter 8)</td>
<td></td>
</tr>
<tr>
<td>v5.0</td>
<td>09 Feb 2015</td>
<td>Updated for CCS 5.0 release - Announce the change for support (reducing 3.x support on 5.0.0 release). - Update history, DAN description, hardware list and documentation table. - Update the release plan by replacing major/minor with A/B. - Remove history in features.</td>
<td></td>
</tr>
<tr>
<td>v5.1</td>
<td>09 Jul 2015</td>
<td>Update to align with CCS 5.1 and other changes since the CCS 5.0 release: 1) Add Plant Other Controller (POC) in acronyms (section 1.4) 2) Add the CCS 5.1 release with main changes in the history (section 1.5) 3) Suppress the section &quot;Public Users&quot; in chapter 2 (distribution suppressed) 4) Replace quarters with months for the planned dates of releases A&amp;B (section 2.4) 5) Add Install/remove commands in the list for ITER Maven tools (section 6.2) 6) Add &quot;Support for Programs&quot; section (7.8) 9) Update the documentation list with the new documents</td>
<td></td>
</tr>
<tr>
<td>v5.2</td>
<td>28 Jan 2016</td>
<td>Improvements from the review of the last version: update of references, missing acronyms, more overview (chapter 2), structure improved, screen dump updated, etc. Updates for the CCS 5.2.0 release: 5.2 in history (section 1.5), release B may be a maintenance version (section 2.2), SDD web application role reduced (chapter 5), operational applications added (section 6.1), documentation list updated (chapter 9)</td>
<td></td>
</tr>
<tr>
<td>Version</td>
<td>Status</td>
<td>Date</td>
<td>Updates</td>
</tr>
<tr>
<td>---------</td>
<td>--------</td>
<td>------------</td>
<td>--------------------------------------------------------------------------------------------</td>
</tr>
</tbody>
</table>
| v5.3    | Approved | 14 Jun 2016| Update for CCS 5.3.0 release  
- Add new version in history (section 1.6)  
- Announce that only one version / year from 2017 (section 2.2)  
- Add 2 documents in chapter 9 (SDN Archiver & installation manager)  
+ Minor text improvements to adjust with scope changes. |
| v5.4    | Approved | 08 Feb 2017| Update for CCS 5.4.0 release:  
- add release of CCS 5.4.0 in section 1.6 (history)  
- add list of device support in section 4.2 (EPICS)  
- limited support for PXI-6682 in section 8.3 (Timing & synchronization)  
- update documentation list in chapter 9 |
| v6.0    | Signed  | 13 Feb 2018| Update for the release of CCS 6.0  
1) add the new release in the history (section 1.6)  
2) clarify the release scheme (section 2.2)  
3) announce support change for older versions (section 3.3)  
4) suppress SDD web application (chapter 5)  
5) add PXIe-6363 support (section 8.4)  
6) update list of documents (chapter 9) |
| v6.1    | Approved | 15 Feb 2018| Same changes as in previous version for CCS 6.0:  
Update for the release of CCS 6.0  
1) add the new release in the history (section 1.6)  
2) clarify the release scheme (section 2.2)  
3) announce support change for older versions (section 3.3)  
4) suppress SDD web application (chapter 5)  
5) add PXIe-6363 support (section 8.4)  
6) update list of documents (chapter 9)  
and also:  
7) update RHEL version (section 4.1)  
8) update EPICS version (section 4.2) |
| v6.2    | Approved | 27 Feb 2019| - Text improvements at various places + some icons  
- Add release of 6.1 in history (Mar-2019)  
- Merge the SDD chapter into a general chapter on development tools (chapter 5)  
- Add the workflow presentation in this chapter (section 5.3)  
- Add SVN in this chapter (section 5.5)  
- Complete the set of SDD data components  
- Update CS-Studio presentation and illustration, add e-log  
- Mention UDA in the 2 archiving sections (CS-Studio & DAN)  
- Extend the scope of Chapter 7 to all supported controllers, including PSH.  
- Develop the PLC section to cover development in PLCs and SPSS (section 7.2)  
- Add a section about program (7.8) |
| v6.3    | Approved | 09 Oct 2020| - update and improve wording in many places  
- update reference documents (new URLs, new versions)  
- add version 6.2 to history  
- update figure 2-1 (physical architecture)  
- update description of release scheme (2.2)  
- update EPICS logo (4.2)  
- update figure 5-2 (development workflow)  
- add OPC UA integration for PLC at the same level as SPSS based integration  
- update list of available user documentation |
Table of Contents

1 INTRODUCTION .........................................................................................................................3
  1.1 Purpose .......................................................................................................................................3
  1.2 PCDH Context ............................................................................................................................3
  1.3 Structure of this Document .......................................................................................................4
  1.4 Reference Documents ..............................................................................................................4
  1.5 Acronyms ......................................................................................................................................4
  1.6 History .......................................................................................................................................6

2 CODAC CORE SYSTEM ...........................................................................................................8
  2.1 Overview .....................................................................................................................................8
  2.2 Release Scheme ..........................................................................................................................9

3 USERS, DISTRIBUTION, SUPPORT AND TRAINING ......................................................10
  3.1 ITER Users .............................................................................................................................10
  3.2 CODAC Core System Distribution .......................................................................................10
  3.3 User Support ..........................................................................................................................10
  3.4 Training ....................................................................................................................................10

4 SYSTEM COMPONENTS .........................................................................................................12
  4.1 RHEL .........................................................................................................................................12
  4.2 EPICS .........................................................................................................................................12
  4.3 Other System Components ....................................................................................................13

5 DEVELOPMENT TOOLS .........................................................................................................14
  5.1 EPICS Records .......................................................................................................................14
  5.2 Self-Description Data ..............................................................................................................14
    5.2.1 The Self-Description Data Concept ................................................................................14
    5.2.2 SDD Content ....................................................................................................................15
  5.3 The Development Workflow ..................................................................................................16
  5.4 ITER Maven Tools ..................................................................................................................16
  5.5 Source Code Repository .........................................................................................................17

6 CODAC SERVICES ....................................................................................................................18
  6.1 Mini-CODAC ...........................................................................................................................18
  6.2 Control System Studio ............................................................................................................18
  6.3 Operator Interface Development .........................................................................................18
  6.4 Alarm Handling .......................................................................................................................19
  6.5 Data Archiving ........................................................................................................................20
  6.6 Electronic Logbook ..................................................................................................................21

7 CONTROLLERS SUPPORT ......................................................................................................22
  7.1 System Monitoring ...................................................................................................................22
  7.2 Support for PLCs .....................................................................................................................22
    7.2.1 Integration using SPSS ......................................................................................................22
7.2.2 Integration using OPC UA

7.3 Timing and Synchronization

7.4 Support for I/O Boards

7.5 SDN Support

7.6 DAN Support

7.7 Common Operating State Support

7.8 Support for Programs

8 DOCUMENTATION
1 INTRODUCTION

1.1 Purpose

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

This document serves also a satellite document for the ITER Plant Control Design Handbook (PCDH) [RD1] as a reference for the I&C software environment.

1.2 PCDH Context

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

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

The 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 the PCDH as presented in Figure 1-1. This document is one of them.
1.3 Structure of this Document

This document is structured as follows:

- Chapter 1 describes the purpose of this document and its context with regards to the Plant Control Design Handbook. It also lists references, defines abbreviations and describes the history of how this document has evolved.
- Chapter 2 gives an overview of the CODAC Core System as well as its release scheme.
- Chapter 3 describes how to become a registered user, how to obtain the CODAC Core System distribution and how to get support.
- Chapter 4 lists the system components of the CODAC Core System.
- Chapter 5 describes development tools that come packaged in the CODAC Core System.
- Chapter 6 describes the CODAC services, mostly for operators, that it implements.
- Chapter 7 provides information about the tools for controllers.

Since this document is intended to serve as an introduction to the CODAC Core System, more details are contained in the documentation listed in Chapter 8.

1.4 Reference Documents

[RD1] Plant Control Design Handbook (ITER_D_27LH2V)
https://www.iter.org/org/team/chd/cid/codac

[RD2] EPICS – Experimental Physics and Industrial Control System
https://epics-controls.org/

[RD3] ITER CODAC Acronyms List (2LT73V)
https://portal.iter.org/departments/CHD/CODAC/Pages/Acronyms.aspx

[RD4] CODAC Core System public web page:
https://www.iter.org/mach/codac/CoreSystem

[RD5] On-line Learning Centre – CODAC Core System Training
https://sharepoint.iter.org/support/learning/SitePages/CODAC%20Core%20System%20Training.aspx

[RD6] CODAC DDD (ITER_D_6M58M9)

[RD7] ITER catalogue for I&C products – Slow controllers PLC (333J63)
http://static.iter.org/codac/pcdh7/Folder%202/7-ITER_catalogue_for_I%26C_products-_Slow_controllers_hours_333J63_v4_1.pdf

[RD8] ITER Catalog of I&C products – Fast Controllers (345X28)

1.5 Acronyms

For a complete list of ITER CODAC abbreviations, please refer to [RD3].

<table>
<thead>
<tr>
<th>Acronym</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>API</td>
<td>Application Programming Interface</td>
</tr>
<tr>
<td>CA</td>
<td>Channel Access</td>
</tr>
<tr>
<td>Acronym</td>
<td>Description</td>
</tr>
<tr>
<td>---------</td>
<td>-------------</td>
</tr>
<tr>
<td>CBS</td>
<td>Control Breakdown Structure</td>
</tr>
<tr>
<td>CCS</td>
<td>CODAC Core System</td>
</tr>
<tr>
<td>CODAC</td>
<td>Control, Data Access and Communication</td>
</tr>
<tr>
<td>COS</td>
<td>Common Operating State</td>
</tr>
<tr>
<td>cRIO</td>
<td>Compact Reconfigurable Input Output</td>
</tr>
<tr>
<td>CS-Studio</td>
<td>Control System Studio</td>
</tr>
<tr>
<td>DAN</td>
<td>Data Archiving Network</td>
</tr>
<tr>
<td>DDD</td>
<td>Design Description Document</td>
</tr>
<tr>
<td>EPICS</td>
<td>Experimental Physics and Industrial Control System</td>
</tr>
<tr>
<td>GUI</td>
<td>Graphic User Interface</td>
</tr>
<tr>
<td>HMI</td>
<td>Human-Machine Interface</td>
</tr>
<tr>
<td>HPN</td>
<td>High Performance Network</td>
</tr>
<tr>
<td>I&amp;C</td>
<td>Instrumentation and Control</td>
</tr>
<tr>
<td>I/O</td>
<td>Input / Output</td>
</tr>
<tr>
<td>IDE</td>
<td>Integrated Development Environment</td>
</tr>
<tr>
<td>IO</td>
<td>ITER Organization</td>
</tr>
<tr>
<td>MRG-R</td>
<td>Messaging Realtime Grid - Realtime</td>
</tr>
<tr>
<td>NI</td>
<td>National Instruments</td>
</tr>
<tr>
<td>NTP</td>
<td>Network Time Protocol</td>
</tr>
<tr>
<td>OPC UA</td>
<td>Open Platform Communications (formerly &quot;OLE for Process Control&quot;) Unified Architecture</td>
</tr>
<tr>
<td>OPI</td>
<td>Operator Interface</td>
</tr>
<tr>
<td>PA</td>
<td>Procurement Arrangement</td>
</tr>
<tr>
<td>PCF</td>
<td>Plant Controller, Fast, or Fast Controller</td>
</tr>
<tr>
<td>PCI</td>
<td>Peripheral Component Interconnect</td>
</tr>
<tr>
<td>PCIe</td>
<td>PCI Express</td>
</tr>
<tr>
<td>PCDH</td>
<td>Plant Control Design Handbook</td>
</tr>
<tr>
<td>PLC</td>
<td>Programmable Logic Controller or Slow Controller</td>
</tr>
<tr>
<td>POC</td>
<td>Plant Other Controller</td>
</tr>
<tr>
<td>PON</td>
<td>Plant Operation Network</td>
</tr>
<tr>
<td>PSH</td>
<td>Plant System Host</td>
</tr>
<tr>
<td>PSOS</td>
<td>Plant System Operation State machine</td>
</tr>
<tr>
<td>PTP</td>
<td>Precise Time Protocol</td>
</tr>
<tr>
<td>PV</td>
<td>Process Variable</td>
</tr>
<tr>
<td>PXI</td>
<td>PCI eXtensions for Instrumentation</td>
</tr>
</tbody>
</table>
1.6 History

- Feb-2010 – Version 1.0, preliminary release with first PLC integration and SDD tools.
- Jul-2010 – Version 1.1, improvements of packaging and SDD components.
- Feb-2011 – Version 2.0, stabilized release for developers, introducing the new environment for SDD and mini-CODAC, stable data model and PLC interface.
- Jul-2011 – Version 2.1, minor release to extend the support for fast controllers, to update the CS-Studio toolkit (BOY) and to distribute bug fixes.
- Feb-2012 – Version 3.0, consolidated release for I&C development: RHEL vs 6, project structure, updated components (ITER and external).
- Jun-2012 – Version 3.1, minor release with new features in CS-Studio tools, SDD tools and Maven commands and with the support of a new digital I/O board.
- Feb-2013 – Version 4.0, improvements in CS-Studio tools, SDD tools and Maven commands, addition of the SDN API, support for Common Operating State (COS) and support for redundant PLC.
- Jul-2013 – Version 4.1, improvements in CS-Studio tools, SDD tools and Maven commands. Support for SDN configuration and redundant PLC in SDD. New version of the timing software with support for new boards.
- Feb-2014 – Version 4.2, addition of CS-Studio tools, extension of Maven commands, improvement of existing CS-Studio and SDD tools, support for FlexRIO, draft DAN API.
- Jul-2014 – Version 4.3, extensions of the CS-Studio, SDD and Maven tools, support for the PXie-6368 board, new version of the TCN library and TCNd, and improvements of the SDN and DAN support.
- Feb-2015 – Version 5.0, new EPICS version (3.15.1), upgrade of the OS (RHEL 6.5, MRG-R 2.5), extensions of SDD, Maven and CS-Studio tools, new driver for PLCs, support for CompactRIO, improvements in TCN, SDN and DAN support.

- Jul-2015 – Version 5.1, addition of the support for Plant Other Controllers (POC), of events in PLC communications and of EPICS interface for the N.I. RIO boards; extensions for OPIs and for the PXIe-6368 board support; corrections and enhancements in all but very few components.

- Feb-2016 – Version 5.2, addition of operational applications for FAT/SAT, corrections and enhancements in the existing components.

- Jul-2016 - Version 5.3, CBS4/CBS5 support, support for huge projects, corrections and enhancements in the existing components

- Feb-2017 - Version 5.4, extension of EPICS device support, improved PTPd support implementation, corrections and enhancements in the existing components

- Feb-2018 – Version 6.0, new RHEL version (RHEL 7.4), new EPICS version (EPICS 7), support for S7-1500 PLCs, CS-Studio tools adaptations for TCR and industrial systems operation.

- Mar-2019 – Version 6.1, upgrade of PostgreSQL (v10) and Eclipse (v4.7.3), further extensions for TCR deployments, programs handling in SDD, 1st stable OPC-UA version.

- Oct 2020 – Version 6.2, full integration of OPC UA.
2 CODAC CORE SYSTEM

2.1 Overview

CODAC designates the central control system responsible for operating ITER.

The different plant systems that constitute the ITER device will each be driven by local instrumentation and control, designated as a Plant System I&C.

A plant system I&C contains local controllers that are either:

- standardized programmable logic controllers, designated as PLCs or Slow Controllers,
- standardized rack mounted computers which may control PCI/PCIe/cRIO input/outputs and are designated as PCFs (plant controller, fast) or Fast Controllers,
- non-standardized controllers, designated as POCs (Plant Other Controllers).

Each plant system I&C also includes exactly one Plant System Host (PSH), supplied by the ITER Organization (IO), which implements standard functions not requiring plant system specific software nor any signal inputs/outputs.

The plant system I&C physical architecture is illustrated in Figure 2-1.

![Figure 2-1 – Plant System I&C Physical Architecture](image)

During development and testing, the CODAC central infrastructure, including servers and operator terminals, is replaced with a computer designated as Mini-CODAC, which implements a subset of the central CODAC functions.

CODAC Core System is a software package that is distributed by the ITER Organization for the development of the plant system I&C. It includes the software for the different types of
I&C computers and 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.

CODAC Core System comes bundled together with the Red Hat Enterprise Linux (RHEL) Operating System, preconfigured for the CODAC Core System.

The CODAC Core System software is based on EPICS [RD2], an open-source distributed control system software platform that is widely used in experimental physics. The EPICS framework and utilities are developed and maintained by its user community, which ITER is a part of.

### 2.2 Release Scheme

All CODAC Core System releases are labelled with a sequence of 3 numbers (the CODAC Core System version identifier) that identify the major branch, the minor variant, starting from 0 for each major version, and the maintenance iteration, starting from 0 for each minor release. For example, CCS version 4.3.1 is a maintenance release for the 4.3 version while CCS version 5.1 is a minor release of version 5.

Since 2017, one version per year is planned, with a release date in February. In addition, maintenance releases are issued when cumulated bug fixes need to be distributed to many users.

Until 2016, there have been two official CODAC Core System releases every year: release “A” in February (major or minor version), introducing major changes and new features, and release “B” in June or July (minor version), improving and extending the “A” release.

Patches to fix known issues are produced and distributed to affected users, on demand.

Preliminary versions (alpha and beta releases) are produced before any official version. They are distributed to developers and testers and shall not be used for I&C development.

Two or more different minor versions can be installed simultaneously on a computer, provided their major version is the same, but only one minor version can be active at any time. This feature is provided to facilitate migration to the latest version.

*It is strongly recommended to migrate to the latest version of CODAC Core System.* Tools are provided for upgrading systems and for migrating applications. The support team can also assist users to facilitate such a migration.

With each release, the list of enhancements and bug fixes is recorded. This list is published in the release notes and the details can be obtained from CODAC support. Generally, there is no retrofitting of bug fixes to previously released versions, but patches can be provided, if required.
3 USERS, DISTRIBUTION, SUPPORT AND TRAINING

3.1 ITER Users

Users who contribute to the development of the ITER I&C System, e.g., ITER Domestic Agencies, research institutes associated with them or industry partners working for ITER through contracts, may become registered CODAC Core System users, or in short, registered users.

The procedure for becoming a registered user is described in the ITER CODAC public web pages [RD4] 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 Responsible Officer.

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

3.2 CODAC Core System Distribution

CODAC Core System is available for distribution to all registered users through a software distribution server hosted at IO premises. That server allows access to all the available software in the form of installable software packages (Linux Red Hat RPM packages using YUM as a package manager). Each computer, on which the software shall be installed, must first become a registered system on the distribution server. The installation procedure is fully automated and described in the CCS Installation Manual and installation website, available to all registered users.

Different profiles or roles of CODAC Core System are provided in order to allow configuring the different systems according to their use: mini-CODAC, PSH, PCF or CODAC terminal. These profiles are each available in two variants, one for development that includes the development tools and one for operation that is stripped down to only the components required for operation.

The documentation (listed in chapter 8) is included in the software distribution with any supporting sample code.

3.3 User Support

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

The requests are sent to codac-support@iter.org. A dedicated team address those, records requests and actions in an issue tracking system, which is monitored by the different IO ROs that are contributing to the support activities and validating the actions.

Since the release of version 6.0, support for versions 5.x is maintained, but the support for versions 4.x will be limited to migration assistance. No regular support is provided for versions older than 4.x.
3.4 Training

Hands-on workshops for ITER I&C developers (4 days) are organized at IO Headquarters on a regular basis, and occasionally at Domestic Agency premises according to needs.

On-line training lessons (presentations and demonstrations) are also available for ITER users through the ITER On-line training centre site [RD5].
4 System Components

4.1 RHEL

Red Hat Enterprise Linux (RHEL) is the standard operating system for CODAC Core System, used on mini-CODAC, PSHs, Fast Controllers, CODAC Terminals and CODAC Servers. All of these systems use a version of RHEL that is officially supported by Red Hat.

The operating system is included in the software distribution, so that any CCS installation will first install the RHEL operating system on the target computer.

The only supported architecture is Linux-x86_64, corresponding to 64-bit architecture. 32-bit architecture is not supported.

The following table shows which RHEL version is used for each major version of CCS.

<table>
<thead>
<tr>
<th>CCS major version</th>
<th>RHEL version</th>
</tr>
</thead>
<tbody>
<tr>
<td>6.x</td>
<td>7.4</td>
</tr>
<tr>
<td>5.x</td>
<td>6.3</td>
</tr>
<tr>
<td>4.x</td>
<td>6.1</td>
</tr>
<tr>
<td>3.x</td>
<td>5.5</td>
</tr>
<tr>
<td>2.x</td>
<td>5.3</td>
</tr>
</tbody>
</table>

For real time dependent applications on Fast Controllers, the MRG-Realtime (MRG-R) kernel extension is also provided on an as-needed basis.

4.2 EPICS

EPICS Base is included in the CCS distribution and is required for all system profiles. The EPICS toolkit is the base for control processes running on Fast Controllers and PSHs, and the EPICS communication protocols, Channel Access (CA) and pvAccess (PVA), are the standard communication protocols to access plant system I&C data over the Plant Operation Network (PON).

Up until CCS version 4.x, the version of EPICS Base is 3.14.

In CCS version 5.x, the version of EPICS Base is 3.15.

In CCS version 6.x, EPICS 7 is used, which comprises EPICS Base 3.16 and the so-called PVA modules, which include the pvData and pvAccess libraries to handle structured data over the network.

The following major EPICS components are included:

- Sequencer, for the execution of state machine programs written using State Notation Language (SNL) in PSHs, Fast Controllers and mini-CODAC systems.
- IOC log, for error and trace logging.
- Autosave, for automatically saving and reloading the values of configuration variables to and from the disk where persistence is required.
- Channel Access Gateway, for allowing Channel Access clients to access servers on a different sub-network.
- Multi-Core Utilities, for configuring how EPICS threads are handled by the different CPU cores, isolating them, and tweaking priorities.
- Java implementations of Channel Access and pvAccess, for Java applications that are CODAC clients.
- Python wrappers for Channel Access and pvAccess, for Python applications that are CODAC clients.
- IOC Monitor / System Monitor, for monitoring IOC and system (host) health parameters as PVs.
- Visual Database Configuration Tool, for configuring EPICS databases using a GUI.
- ASYN, a generic driver framework for developing EPICS Device Support.
- Stream Device, a generic Device Support module that allows writing stream based data exchange protocols between EPICS records and hardware.
- Area Detector, a generic Device Support module used for image acquisition devices.
- OPC UA Device Support, a Device Support module allowing IOCs to interface OPC UA servers.

In addition, Device Support modules for all supported hardware from the Fast and Slow Controller catalogues are provided.

4.3 Other System Components

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

- A PostgreSQL relational database, containing the following databases:
  - SDD, for storing plant system I&C data
  - Alarm, for storing alarm server configuration and alarm history
  - Archive, for storing archive engine configuration and archived data
  - Logbook, for storing events.

- The Eclipse platform used for specific editors, SDD Editor and Maven Editor, as well as for the Control System Studio tools. Plug-ins for development with C/C++, Java, Python, and Subversion are also included.

- Apache ActiveMQ as a JMS implementation for Control System Studio.
5 DEVELOPMENT TOOLS

5.1 EPICS Records

EPICS includes a framework for developing control processes, called IOCs (Input Output Controllers). Each IOC process is built around a database of records that belong to a set of pre-defined record types.

The EPICS build tools have not been modified, but have been encapsulated into ITER specific commands (cf. ITER Maven Tools, below). The development of EPICS configuration files should be done with the SDD tools in order to satisfy the requirements for integration. This includes enforcing of naming conventions and conventions on files that imported EPICS software shall satisfy.

5.2 Self-Description Data

5.2.1 The Self-Description Data Concept

Self-Description Data (SDD) is an ITER concept covering the static data that configures 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 an IO database upon software delivery. During development, it is stored in a local database, the SDD database.

The SDD Toolkit has been developed by IO to allow the user to configure the plant system I&C. It includes:

- The SDD Editor to define the plant system interface, the I&C components, the interfaced signals and to configure variables, alarms, archiving, etc.
  The SDD editor is an Eclipse RCP application integrated with the other development tools and operating on a local SDD database.
- The SDD Translator to convert the SDD data into the required configuration data for mini-CODAC, PSH and Fast Controllers and to generate the required STEP 7 files for interfacing PLCs.
- The SDD Sync tool to save and load SDD data to/from XML files.
- The SDD Parser to parse user-provided or user-modified configuration files, e.g., EPICS database definitions, and retrofit the changes into the SDD database.
5.2.2 SDD Content

Information about the following elements can be included in the SDD of an I&C project:

- Plant Systems (PBS…)
- Cubicles (location…)
- Chassis (location…)
- Control units (type, network interfaces…)
- I/O modules (type…)
- Components (type…)
- Signals associated with components (direction, type, logic, range…)
- Control groups (CBS…)
- CODAC Servers (network interfaces…)
- Plant System I&Cs (CBS…)
- Plant System Operating State machine (PSOS) states and transitions, and how they are mapped to those of the Common Operating State machine (COS)
- I&C programs (deployment target, variables interfaced, I/O modules controlled…)
- I&C functions (CBS…)
- Variables, including:
  - Tier (Conventional, Interlock, Safety)
  - Network (PON, SDN, DAN)
  - Signal mapping
  - EPICS/SDN/DAN attributes (if PON/SDN/DAN variable respectively)
  - Alarm attributes
5.3 The Development Workflow

The development workflow, as illustrated in Figure 5-2, consists of the following steps:

1) Define all configuration with SDD tools, including:
   - the variables with their EPICS details for building the IOC process
   - the programs
   - the functions (CBS nodes for variables, OPIs, alarms, etc.)
   - the configuration for alarms, archiving.
2) Generate the configuration files.
3) Create the initial version of the software unit.
4) Develop any additional software for the project as required, such as:
   - the code for the C/C++ programs
   - state machines in SNL or other scripts
   - operator Interfaces
   - the code for the PLCs.
5) Build the software unit, creating the binary RPM packages to be deployed on the different target computers.

5.4 ITER Maven Tools

The development workflow is supported by the SDD tools and by commands implemented using the Apache Maven build tool.
These commands implement all the steps for building and testing the software unit. They also allow saving, restoring and synchronizing the project files with a Subversion (SVN) source repository (cf. next section).

A graphical tool, the Maven Editor, provides the user with a graphical HMI for executing these commands – either as a stand-alone tool or by means of a dedicated view in the SDD Editor.

As illustrated in Figure 5-3, from the Maven Editor interfaces the user can build the software, run the executables for test purposes on local or remote hosts and build the RPM packages for deployment.

Figure 5-3 – Maven Editor

5.5 Source Code Repository

ITER provides a Subversion (SVN) server as a central service to all ITER I&C developers for maintaining any software product for I&C. The nodes in the central repository are created by IO for receiving the software associated with a PA or contract. The suppliers are in charge of delivering and tagging the I&C software.

The I&C software shall be delivered in source format for allowing it to be re-built using appropriate procedures before deployment.

The main commands for exchanging source files with the central repository are:

- **Import** – takes a local directory that is not synchronized with any SVN location, and performs an initial import to a specified SVN location.
- **Checkout** – the reverse of import. Takes a directory on the central SVN repository and makes a copy on the local system.
- **Update** – checks the central SVN repository and, if changes have been made, it updates the local copy with those changes.
- **Commit** – performs the reverse operation as update. Changes made on the local copy are pushed as a new revision to the central SVN repository.

All these commands are integrated into Maven Editor so that they can be easily executed with any I&C software unit.
6 CODAC SERVICES

6.1 Mini-CODAC

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

- A local CODAC Server implementing the following CODAC services:
  - operator interface,
  - alarm handling,
  - data archiving,
  - electronic logbook,
  - error & trace logging
- A development workstation with the SDD database and other shared files or file systems.

Development and testing, including Factory Acceptance Tests (FAT), shall be executed using the mini-CODAC as a replacement for CODAC.

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

6.2 Control System Studio

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

Tools developed in this environment have been adopted for the mini-CODAC services: HMIs, alarms and archiving.

6.3 Operator Interface Development

The display manager BOY, developed with the CS-Studio platform, allows the development of dynamic displays using an interactive editor and the run-time management of such displays. In addition to its integration into the CS-Studio environment, BOY offers a rich set of graphical widgets and dynamic configuration features that can be both extended and customized to satisfy specific requirements.

A dedicated library of industrial symbols, with graphical widgets for display and dynamic attributes, has been added for ITER applications (I&C OPIs).

BOY supports the standardized ITER HMI Design Guide for OPI templates, fonts and colours, as illustrated in Figure 6-1.
CS-Studio is not only used for developing OPIs, but also for running them. Running an OPI means having its elements or widgets animated in real-time with the process data and allowing the operators to interact with the plant system.

OPIs developed for the plant system I&C shall be available for monitoring and controlling the equipment. In addition, some simple tabular PV OPIs are generated for all CBS functions that include all variables of the function. These OPIs can also be used for application testing and verification.

CODAC HMIs consist of a display for each CBS function, a navigation pane to navigate between the CBS functions, an alarm pane displaying any triggered alarms, component faceplates and alarm pages for every alarm enabled variable.

### 6.4 Alarm Handling

**BEAST** is a distributed alarm system consisting of alarm servers that monitor alarms from the EPICS IOC processes via Channel Access and a user interface for viewing the current alarms as well as acknowledging them and browsing the history of alarm messages.

An alarm can also trigger the execution of automated actions, including sending an email or submitting a new logbook entry.

Alarm logs and configurations are stored in the included relational database.
Alarm displays are fully integrated into the CODAC HMI. They are visible to operators in a variety of ways:

- The top right window of the HMI contains the Alarm Pane, where all the triggered alarms, current and unacknowledged are listed. Right-clicking on an alarm in the alarm pane provides quick access for operators to execute several actions, such as:
  - Acknowledging the alarm
  - Opening an alarm page to get detailed information about the alarm
  - Triggering a pre-configured command
  - Opening windows with pre-configured guidance messages.
- Widgets in the mimic linked to a PV may have a current alarm change colour to reflect their alarm severity.
- Nodes on the navigation pane change colour to display the highest unacknowledged alarm severity at their CBS level or below.

Alarm descriptions can also be automatically annunciated by a text-to-speech converter.

### 6.5 Data Archiving

The EPICS data archiver **BEAUTY** is included in the CCS distribution. It allows deploying archive engines that save process value data into a relational database.

A data viewer, the **CS-Studio DataBrowser**, allows retrieving and viewing archived data as well as monitoring of live data. Archived data trends can also be viewed in OPIs using the Archive XY Graph widget.

Data archiving requires the set of variables to archive to be configured with SDD tools.

Archived data is also pushed to data servers to be accessible by means of the Unified Data Access (UDA) retrieval layer.
6.6 Electronic Logbook

An electronic Logbook is also available in the operational configuration of CS-Studio and integrated with the other CS-Studio tools.

Formatted text and screenshots can be included in logbook entries. OPIs, alarm views and data plots can be inserted on demand. Logbook entries are also created from alarms in an automatic manner.

![Figure 3 - Log entry](image-url)

Logbook Entry Example:

- **Date:** May 31, 2013
- **Description:** Failure of the INO Switch.
- **Tags:** N/A
- **Data Browser Plot:** Attached screenshot of the data browser plot.
- **Alarm Details:**
  - **Severity:** High
  - **Message:** Major Switch Alarm
  - **Value:** N/A

Figure 3 - Log entry
7 CONTROLLERS SUPPORT

7.1 System Monitoring

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

Each IOC is built including a monitoring module that provides, by means of Process Variables, information about the IOC process (process ID, start time, port used…) and its resource usage (CPU, memory, internal queue usage, number of remote clients…).

In addition, for each PSH and controller, a dedicated IOC process is built to provide information about the computer (time accuracy, boot time, software version) and its resources (CPU load, memory usage, file descriptors, status of I/O boards). A set of Process Variables provides the user with an interface to this 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.

The monitoring of the cubicle environment (e.g., temperature, door status…), implemented by dedicated PLCs in standard cubicles, is also interfaced via such variables.

7.2 Support for PLCs

CODAC Slow Controllers are standardized as Siemens S7 PLCs. The recommended components are described in the ITER catalogue for I&C products - Slow controllers PLC[RD7]. Siemens S7 PLCs do not run CODAC Core System, as they use their own system environment, Step 7. The tool used to program these PLCs should be either Siemens Simatic Manager or TIA Portal. These tools are not included in the CODAC Core System distribution.

Communication with the PLCs is implemented by an EPICS IOC process deployed by default on the PSH, alternatively on a fast controller.

The communication is 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 to an EPICS process variable in the PSH and to a program variable in the PLC.

CODAC PLCs must synchronize their system time using NTP and use the PSH as a time source.

7.2.1 Integration using SPSS

When using this option, PLC S7 programs must be built based on a predefined program provided with the CCS distribution called Standard PLC Software Structure (SPSS). The SPSS essentially establishes the communication channels with a PLC IOC and provides a program structure on which the final program should be built. The SDD tools generate the variables exchanged with the EPICS IOC in dedicated data blocks that are handled by the SPSS-based programs.
Automatically generated data are also inserted in the transmitted frames (header, trailer, length, application version number, timestamp, heartbeat) in order to detect communication errors and data inconsistencies. This data is also generated by SDD Translator.

Redundant PLCs (used for example in interlock systems) are supported. This includes the transparent connection to the active master CPU and the reconnection to the new master in the case that a switch is triggered because of a failure.

### 7.2.2 Integration using OPC UA

When using this option, an OPC UA server must be running on top of the Slow Controller. On the S7-1500 series, this can be enabled in the TIA Portal software after a specific OPC UA server software license has been acquired for the PLC.

Under OPC UA, variables on the PLC are accessible as OPC UA nodes through a unique node ID, a combination of a string identifier and a namespace number. The OPC UA representation of PLC variables is visible and configurable in TIA Portal; the node IDs have to be entered in SDD to correctly create the EPICS IOC configuration representing the PLC variables.

From the Siemens S7 PLCs in the ITER Slow Controller catalogue, only the models of the 1500 series are providing an embedded OPC UA server. Redundant PLCs (used for example in interlock systems) are not supported.

### 7.3 Timing and Synchronization

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

PXI boards as well as Ethernet interfaces compliant with the PTP protocol are supported for Fast Controllers.

The timing and synchronization board for Fast Controllers is **NI PXI-6683H**. Support for the older version, **NI PXI-6682**, is now stopped.

The software support includes Linux drivers, EPICS drivers, SDD integration, test programs and documentation.

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 1 µs.
- Generation of events and clock signals using the precise time distributed by the TCN.
- Time-stamping incoming signals.

On the ITER TCN network, the accuracy of the board time will be 50 ns RMS.

The TCN library provides C/C++ developers with functions for waiting for an ITER time or registering actions synchronized with ITER time.

### 7.4 Support for I/O Boards

The hardware that is recommended by IO for Fast Controllers is published in the ITER Catalog of I&C products – Fast Controllers [RD8]. This catalog is a satellite document of the PCDH [RD1]. The scope of I/O board software support is limited to the boards that are defined in this catalog and are adopted for plant system I&C. The software support includes Linux drivers, EPICS drivers, SDD integration, test programs and documentation.
All National Instruments modules referenced in this document are supported.

These are:

- A multi-function data acquisition board, **NI PXI-6259** with the 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 (16 bits), 1.25 MS/s for 4 channels, 2.86 MS/s for 1 channel.
  - 48 bi-directional Digital channels
  - 2 counters (32 bits).

- A digital I/O board, **NI PXI-6528**, with the following channels:
  - 24 optically isolated input channels
  - 24 solid-state relay output channels.

- The **NI FlexRIO PXIe-796xR** bundles, composed of an FPGA board with an adapter module for analog I/O, digital I/O or camera interface.

- The **NI PXIe-6368** board (X-series), which is a high performance (2 MS/s simultaneous) multi-function data acquisition board. The **NI PXIe-6363** board, a lower-cost version with reduced performances (no simultaneous sampling) is also supported from CCS 6.0.

- The **NI CompactRIO** modules described in the Fast Controllers catalog.

7.5 SDN Support

The Synchronous Databus Network (SDN) is the high performance real time network to be used for sharing of real time data across ITER plant systems and with the Plasma Control System (PCS).

The SDN software provides communication services following an anonymous publish-subscribe design pattern. Participants exchange data over logical communication channels, i.e., topics, without the necessity of being aware of other participants.

The data exchanged via SDN is defined using SDD tools as SDN topics. The programs deployed on Fast Controllers can be configured for publishing data on SDN topics or for subscribing to them.

The SDN Application Programming Interface (API) is included in the CCS distribution. It comprises a library for C++ programs that implements all the required services for SDN communications.

An SDN monitoring service can also be started for gathering information on SDN data exchanges.

7.6 DAN Support

The Data Archiving Network (DAN) is the dedicated network for streaming high volume data, such as scientific data derived from diagnostic systems. Data transmitted by the distributed producers are received by a DAN archiver and saved into HDF5 files. A data access library and a set of plotters allow retrieving and visualization the saved data. The data can also be retrieved by means of the Unified Data Access (UDA) layer.
DAN data is defined using SDD tools and processes deployed on Fast Controllers can be configured for publishing on DAN. The distribution includes the DAN Application Programming Interface (API) that allows processes to do that.

A DAN streamer process is provided to be deployed on each Fast Controller for collecting data from the different processes and pushing it over DAN.

### 7.7 Common Operating State Support

COS variables are a mandatory part of the Plant System I&C interface with CODAC. They provide the supervision (SUP) 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. Their implementation is configured by means of a mapping table between the generic COS variables and the Plant System specific variables that the I&C developer should implement. These specific variables can be implemented either on a Fast Controller or on a PLC.

### 7.8 Support for Programs

In addition to EPICS IOCs, real-time control can be implemented by means of C/C++ programs.

These are configured using SDD tools and developed using ad-hoc templates. For each I/O board, a C API is available for direct usage from programs.

C/C++ APIs are provided for

- precise time synchronization and stamping using TCN,
- synchronous data exchange over SDN
- scientific and diagnostic data production over DAN
- error and trace logging.

The interface with CODAC is implemented by means of variables declared with SDD that are generated as EPICS PVs with interface code for the programs.
8 DOCUMENTATION

The following documents are included in the CCS distribution. These are also available from the ITER documentation management system (IDM). Versions are given in the release notes.

Overview

<table>
<thead>
<tr>
<th>Document Code</th>
<th>Title</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ITER_D_34SDZ5</td>
<td>CODAC Core System Overview</td>
<td><em>(This document)</em></td>
</tr>
<tr>
<td>ITER_D_43PSH9</td>
<td>CODAC Core System User Manual</td>
<td></td>
</tr>
</tbody>
</table>

Installation & Support

<table>
<thead>
<tr>
<th>Document Code</th>
<th>Title</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ITER_D_33JNKW</td>
<td>CODAC Core System Installation Manual</td>
<td></td>
</tr>
<tr>
<td>ITER_D_7JCFUD</td>
<td>CODAC Core System Migration Guide</td>
<td></td>
</tr>
<tr>
<td>ITER_D_U2TTSZ</td>
<td>PTPd User Manual</td>
<td></td>
</tr>
</tbody>
</table>

Development & Testing

<table>
<thead>
<tr>
<th>Document Code</th>
<th>Title</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ITER_D_33T8LW</td>
<td>CODAC Core System Application Development Manual</td>
<td></td>
</tr>
<tr>
<td>ITER_D_32Z4W2</td>
<td>SDD Editor User Manual</td>
<td></td>
</tr>
<tr>
<td>ITER_D_46AAXR</td>
<td>SDD Synchronization Guide</td>
<td></td>
</tr>
<tr>
<td>ITER_D_A4WQDZ</td>
<td>How to include a new I/O module in SDD</td>
<td></td>
</tr>
<tr>
<td>ITER_D_7MT2YC</td>
<td>Maven Editor User Guide</td>
<td></td>
</tr>
<tr>
<td>ITER_D_35XFcy</td>
<td>System Health Monitoring Variables</td>
<td></td>
</tr>
<tr>
<td>ITER_D_QEK784</td>
<td>Logging library - Software User Manual</td>
<td></td>
</tr>
</tbody>
</table>

Control System Studio

<table>
<thead>
<tr>
<th>Document Code</th>
<th>Title</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ITER_D_QVBYD8</td>
<td>CODAC Core System CS-Studio User Guide</td>
<td></td>
</tr>
<tr>
<td>ITER_D_3XLESZ</td>
<td>HMI Style Guide and Toolkit</td>
<td></td>
</tr>
<tr>
<td>ITER_D_7367JQ</td>
<td>Operator Interface standardisation - CSS BOY Edition and Runtime</td>
<td></td>
</tr>
<tr>
<td>ITER_D_A69URK</td>
<td>Operator Interface standardisation - CSS BOY Industrial Symbol Library</td>
<td></td>
</tr>
</tbody>
</table>

Slow Controllers (PLC) and OPC UA

<table>
<thead>
<tr>
<th>Document Code</th>
<th>Title</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ITER_D_3QPL4H</td>
<td>PLC Software Engineering Handbook</td>
<td></td>
</tr>
<tr>
<td>ITER_D_PAHXJ</td>
<td>s7PLCA syn EPICS Driver User’s Manual</td>
<td></td>
</tr>
<tr>
<td>ITER_D_2N8C3M</td>
<td>PLC Sample Application Guide</td>
<td></td>
</tr>
</tbody>
</table>
### Fast Controllers – IO

<table>
<thead>
<tr>
<th>Document ID</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ITER_D_2PLQ4P</td>
<td>NI Sync Linux Device Driver User’s Guide</td>
</tr>
<tr>
<td>ITER_D_33Q5TX</td>
<td>NI Sync EPICS Driver User’s Guide</td>
</tr>
<tr>
<td>ITER_D_32GTJY</td>
<td>NI PXI-6259 Linux Driver User’s Guide</td>
</tr>
<tr>
<td>ITER_D_3DEY52</td>
<td>NI PXI-6259 EPICS Driver User’s Guide</td>
</tr>
<tr>
<td>ITER_D_3ZHXQ9</td>
<td>NI PXI-6528 Linux Device Driver User’s Guide</td>
</tr>
<tr>
<td>ITER_D_433VEW</td>
<td>NI PXI-6528 EPICS Driver User’s Guide</td>
</tr>
<tr>
<td>ITER_D_3LTMRE6</td>
<td>NI X-Series Linux Device Driver User’s Guide</td>
</tr>
<tr>
<td>ITER_D_3P4N3R</td>
<td>NI X-Series EPICS Driver User’s Guide</td>
</tr>
<tr>
<td>ITER_D_LW3UFH</td>
<td>NI-RIO Device Driver User Manual</td>
</tr>
<tr>
<td>ITER_D_RAJ9P8</td>
<td>NI-RIO EPICS Device Driver User Manual</td>
</tr>
<tr>
<td>ITER_D_RATM8Z</td>
<td>IRIO Library User’s Manual</td>
</tr>
<tr>
<td>ITER_D_QQMYTY</td>
<td>IRIO Design Rules for LabVIEW for FPGA</td>
</tr>
</tbody>
</table>

### Fast Controllers – HPN

<table>
<thead>
<tr>
<th>Document ID</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ITER_D_N4XTGG</td>
<td>TCN API – Software User Manual</td>
</tr>
<tr>
<td>ITER_D_MUYNT6</td>
<td>TCNd User Manual</td>
</tr>
<tr>
<td>ITER_D_B7SKFU</td>
<td>SDN Software User Manual</td>
</tr>
<tr>
<td>ITER_D_Q6GULS</td>
<td>DAN User Manual</td>
</tr>
<tr>
<td>ITER_D_SD29MG</td>
<td>SDN archiver user manual</td>
</tr>
</tbody>
</table>

### Operational Applications

<table>
<thead>
<tr>
<th>Document ID</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ITER_D_RUZCJ9</td>
<td>User Manual for CODAC Operational Applications</td>
</tr>
<tr>
<td>ITER_D_TPLTKG</td>
<td>UDA User Manual</td>
</tr>
<tr>
<td>ITER_D_RGWSHS</td>
<td>SDN Tools User Manual</td>
</tr>
</tbody>
</table>