

DEVICE CONTROL DATABASE TOOL

DCDB

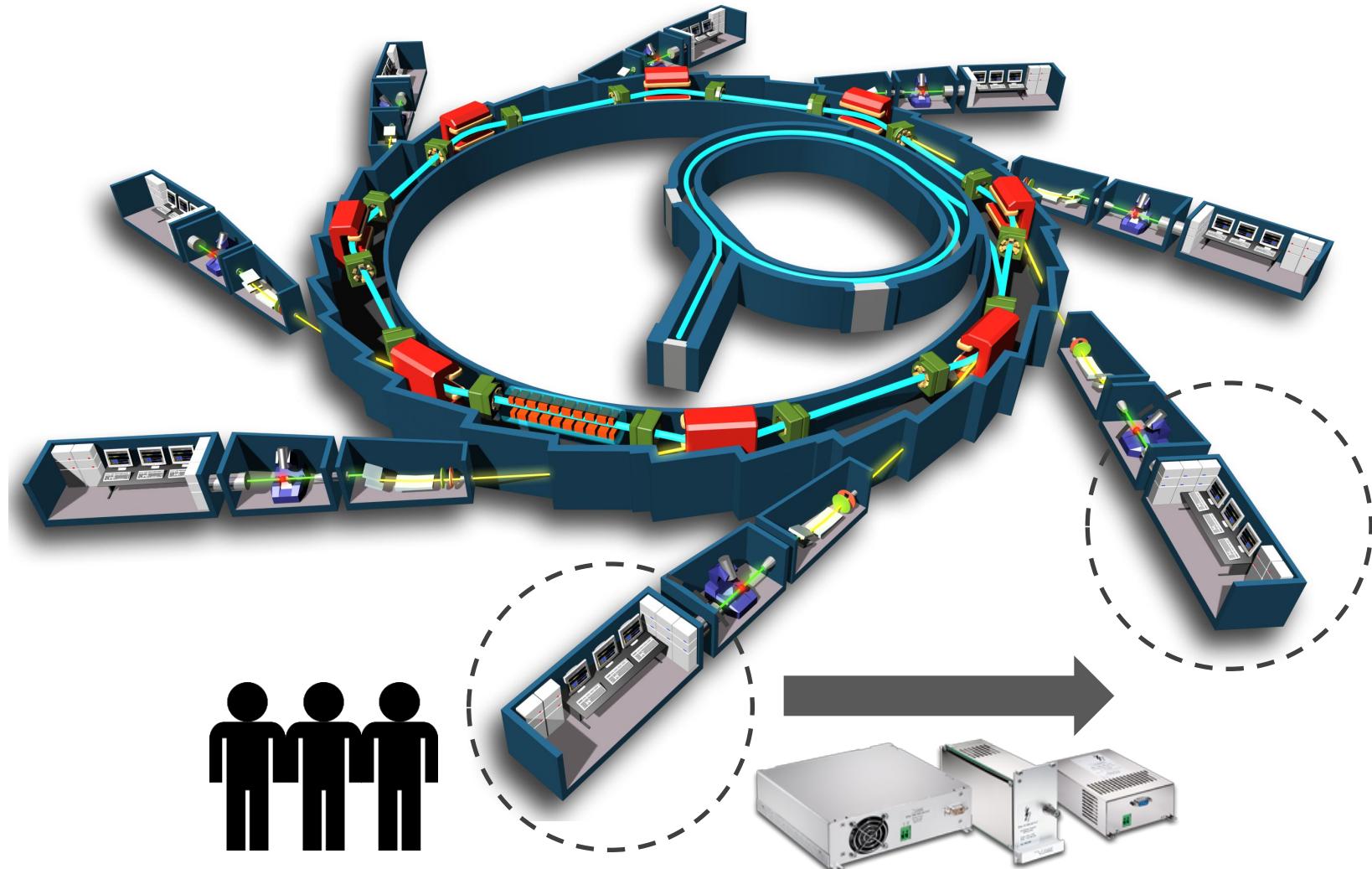


Your **TRUSTED** Control System Partner

Contents



- Intro/features
- DCBD tool architecture
- Software used
 - Dynamic library loader
 - EPICS Device support modules
 - DCDB-server
 - DCDB-client
- RPM packages & Build server



Your **TRUSTED** Control System Partner

Problems

- Changing parameters
 - IP address, port number, HW parameters, macros, etc.
- Need to recompile IOCs

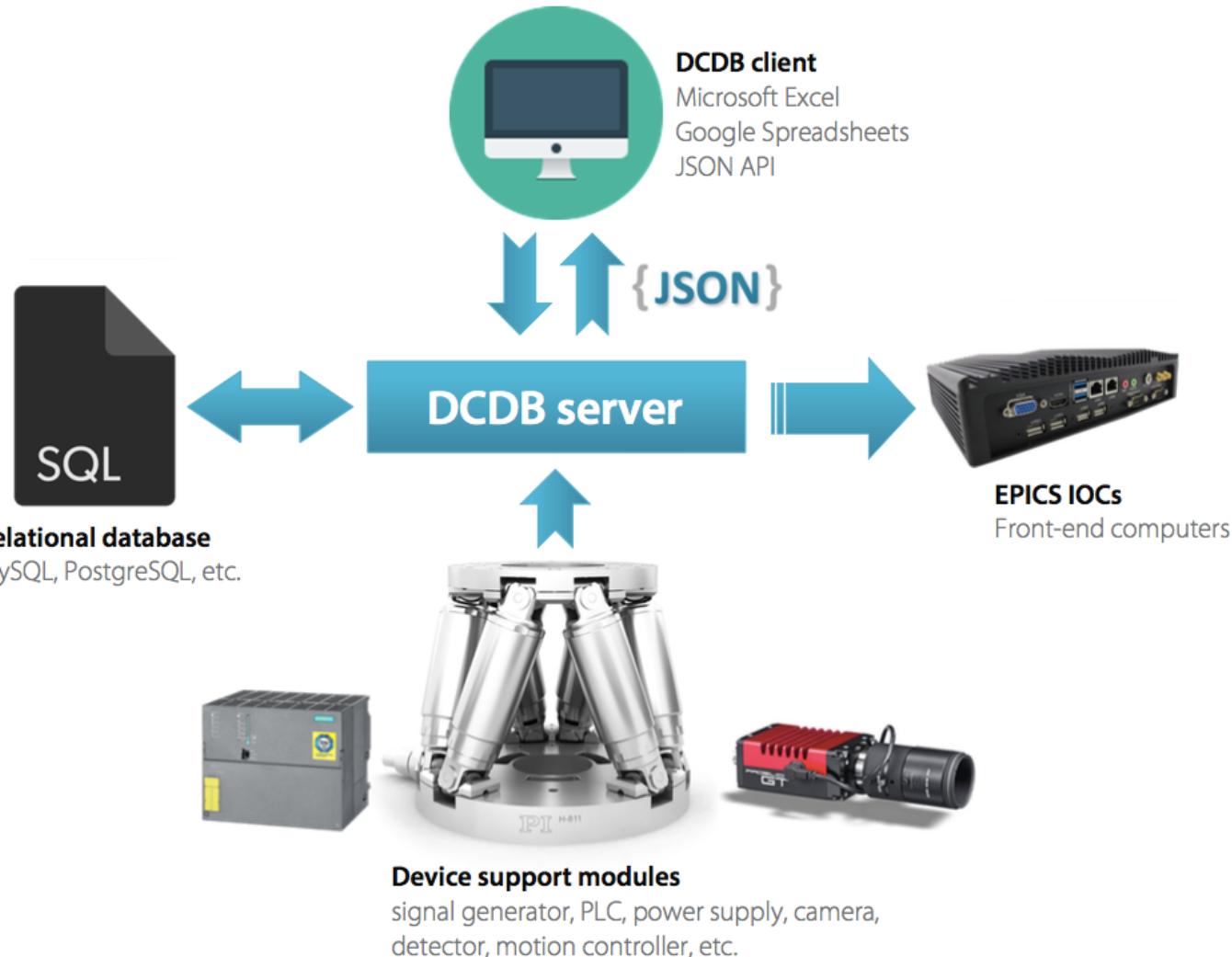


Features



- *DCDB tool* is:
 - a control system **configuration tool**, which provides an easy-to-use interface for quick configuration of the whole accelerator in just a few clicks.
- **Features:**
 - Generation and deployment of **IOC** startup scripts
 - Change IOC parameters on the fly
 - Start/stop/restart IOC (utilizing ***dlloader*** by Dirk Zimoch of **PSI**)
 - **Siemens S7PLC, StreamDevice** support
 - Google Spreadsheets client
 - Compliant with CODAC v4 (ESS, ITER, ELI-NP)
 - Chinese localization 设备控制数据库配置软件

DCDB architecture



Software



- MySQL database
- Python backend (flask-restful, sqlalchemy, paramiko)
- Microsoft Excel front-end (C# .NET)
 - Google spreadsheets (JavaScript)
- ESS CODAC 4.1, ITER CODAC 4.3
 - Also tested on Ubuntu 14, Mac OSX 10.9, CentOS 6, Windows
- procServ*** (developed by Ralph Lange)
- dlloader*** (developed by Dirk Zimoch, PSI)
 - customized by CSL

Dynamic library loader



- *dLLoader* is an EPICS-based tool that allows you to load EPICS device support libraries by just adding its' definitions in the startup script
- Features:
 - load device support without the need to compile IOCs
 - Just issue *require <name>,<ver>* in the epics shell
 - integrated in CODAC 4.x
 - supports procServ
 - comes in the form of an IOC (*dlloader-procServ*, *dlloader-screen*) or library (not tied to IOC: just start softloc, load *dlloader* using *dlload* and start loading your device support modules)

Device support modules



- Support modules consist of:
 - a library file (`lib/linux-x86_64/xxx.so`)
 - a database definition file (`dbd/xxx.dbd`)
 - db/substitution files (`db/xxx.db`)
 - `init.cmd`
 - `init-pre.cmd`
 - `init-post.cmd`

Device support modules



init.cmd

```
require BeamPositionMonitor
```

init-pre.cmd

```
# @field DAQ_X1
# @type LINK
# @restrictions DataAcquisition
# Signal from data acquisition module (left).
```

```
# @field DEFAULT_ALARM
# @type ALARM
# Default alarm.
```

```
dbLoadRecords "db/BPM.db", "MODULE_INSTANCE_NAME=${MODULE_INSTANCE_NAME},
DAQ_X1=${DAQ_X1}, DAQ_Y1=${DAQ_Y1}, DAQ_X2=${DAQ_X2}, DAQ_Y2=${DAQ_Y2}"
```

init-post.cmd

```
dbpf ${DAQ_X1}.FLNK ${MODULE_INSTANCE_NAME}:CALC
dbpf ${DAQ_Y1}.FLNK ${MODULE_INSTANCE_NAME}:CALC
dbpf ${DAQ_X2}.FLNK ${MODULE_INSTANCE_NAME}:CALC
dbpf ${DAQ_Y2}.FLNK ${MODULE_INSTANCE_NAME}:CALC
```

Device support modules



- Create support modules (using dlloader epics template):

```
bled@bled:~$ mvn newunit -Dunit=m-BeamPositionMonitor
bled@bled:~$ cd m-BeamPositionMonitor
bled@bled:~$ mvn newdlloader
bled@bled:~$ mvn clean compile test package install
```

- Files to deploy:

```
.
├── db
│   └── BeamPositionMonitor.db
├── dbd
│   └── BeamPositionMonitor.dbd
├── init.cmd
├── init-post.cmd
├── init-pre.cmd
└── lib
    └── linux-x86_64
        ├── libBeamPositionMonitor.a
        └── libBeamPositionMonitor.so
```

- Register support module using DCDB import tool:

```
bled@bled:~$ dcdb
Usage: dcdb [--pom=] [--pre=] [--db=] [-v] [--help] [--version] [[-delete]]
```

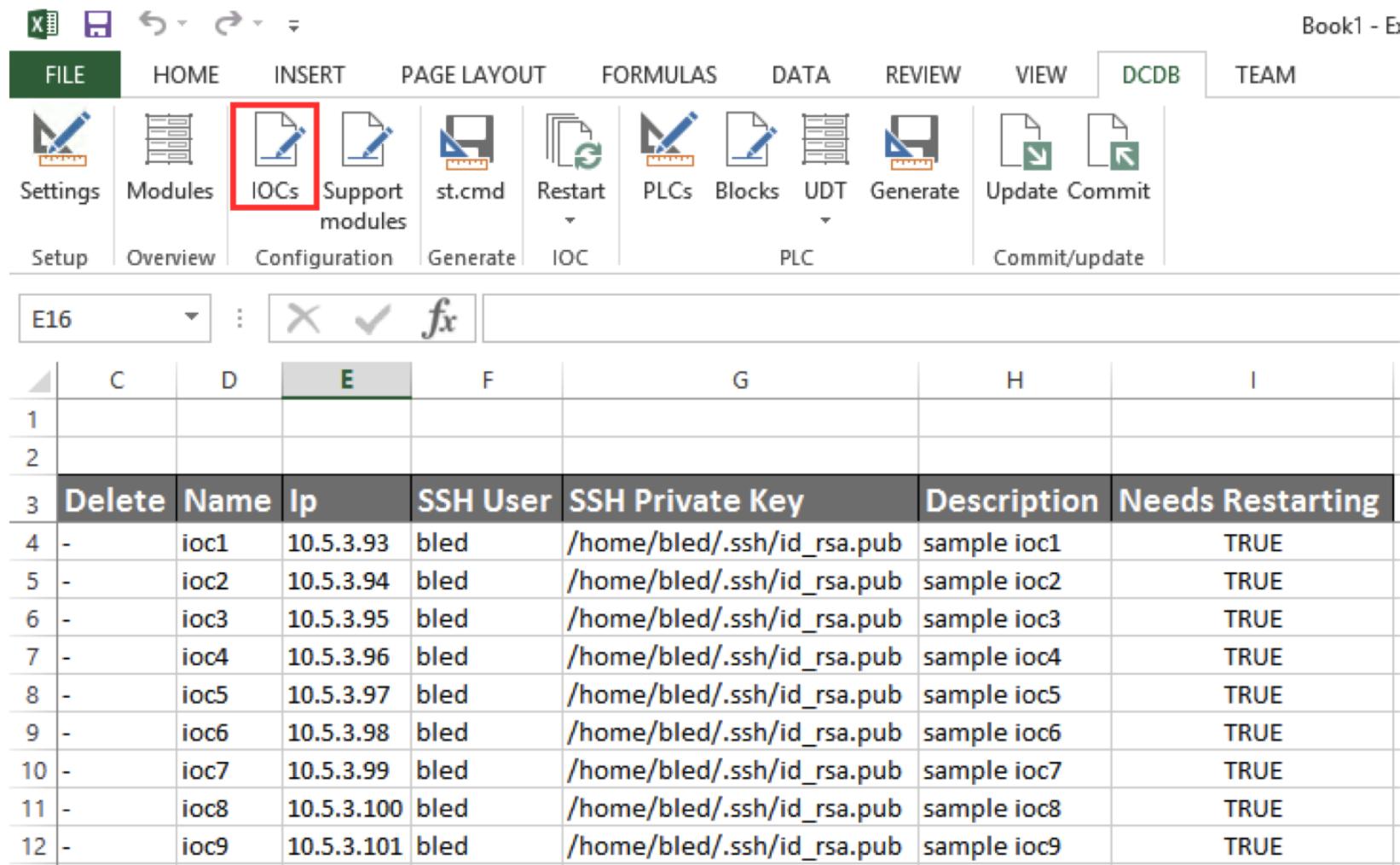
DCDB server

- REST server written in Python
- Uses JSON as the data exchange format
- Uses SSH to deploy configuration onto IOCs
- Deployed as CODAC-service:

```
bled@bled:~$ dcdb-server
Usage: dcdb-server {start|stop|status|restart|fg|log} [--port=5000]
```

Excel client / IOC

IOC configuration



The screenshot shows an Excel spreadsheet titled "Book1 - E". The ribbon at the top has tabs for FILE, HOME, INSERT, PAGE LAYOUT, FORMULAS, DATA, REVIEW, VIEW, DCDB (which is selected), and TEAM. Below the ribbon, there are icons for Settings, Modules, IOCs (which is highlighted with a red box), Support modules, st.cmd, Restart, PLCs, Blocks, UDT, Generate, Update, and Commit. The main area of the spreadsheet displays a table with the following data:

	Delete	Name	IP	SSH User	SSH Private Key	Description	Needs Restarting
3	-	ioc1	10.5.3.93	bled	/home/bled/.ssh/id_rsa.pub	sample ioc1	TRUE
4	-	ioc2	10.5.3.94	bled	/home/bled/.ssh/id_rsa.pub	sample ioc2	TRUE
5	-	ioc3	10.5.3.95	bled	/home/bled/.ssh/id_rsa.pub	sample ioc3	TRUE
6	-	ioc4	10.5.3.96	bled	/home/bled/.ssh/id_rsa.pub	sample ioc4	TRUE
7	-	ioc5	10.5.3.97	bled	/home/bled/.ssh/id_rsa.pub	sample ioc5	TRUE
8	-	ioc6	10.5.3.98	bled	/home/bled/.ssh/id_rsa.pub	sample ioc6	TRUE
9	-	ioc7	10.5.3.99	bled	/home/bled/.ssh/id_rsa.pub	sample ioc7	TRUE
10	-	ioc8	10.5.3.100	bled	/home/bled/.ssh/id_rsa.pub	sample ioc8	TRUE
11	-	ioc9	10.5.3.101	bled	/home/bled/.ssh/id_rsa.pub	sample ioc9	TRUE
12	-						

Excel client / Modules



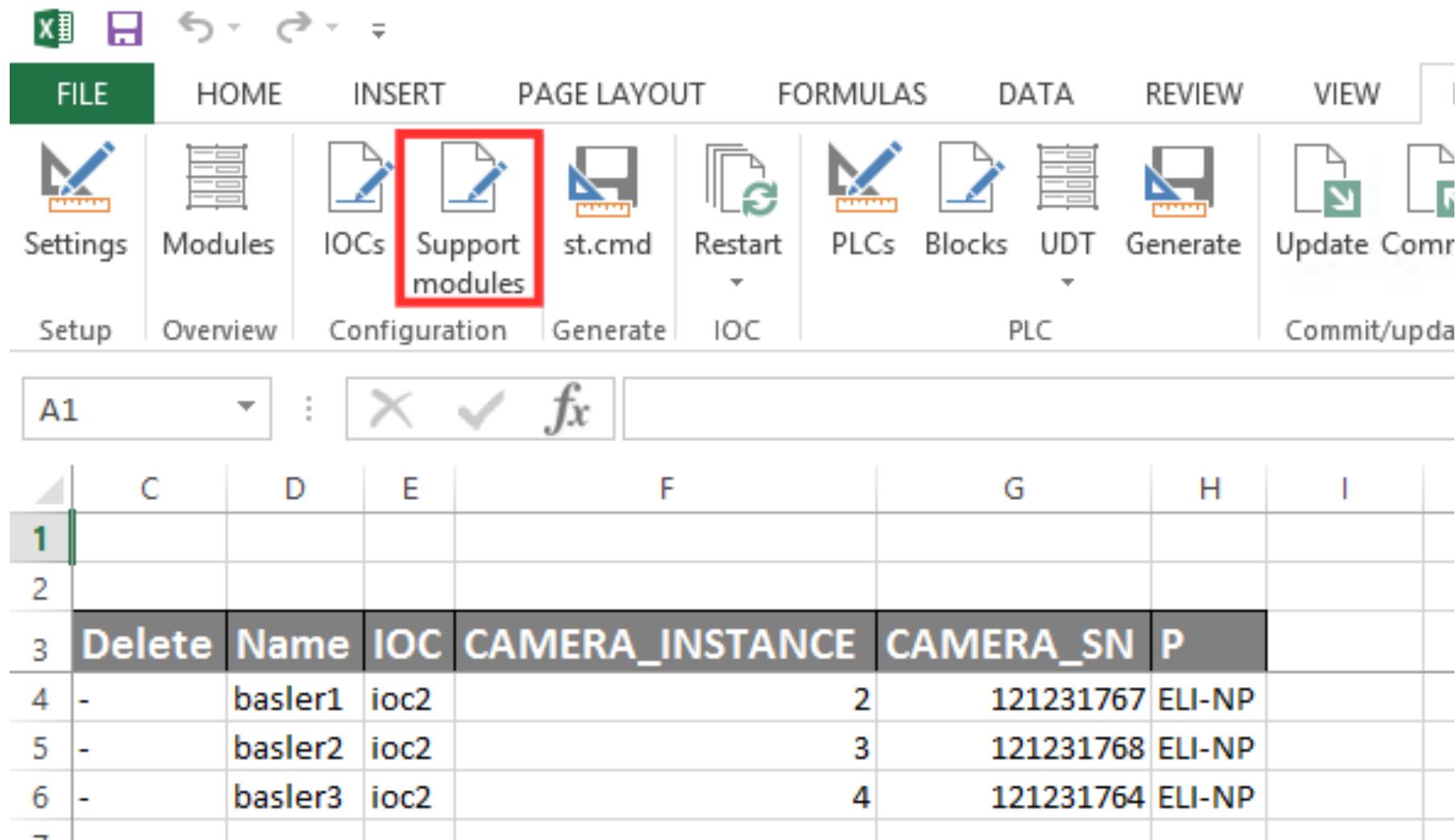
Overview of all support modules:

Screenshot of an Excel spreadsheet titled "Book1 - Excel" showing a list of support modules. The "DCDB" tab is selected in the ribbon. The "Modules" icon in the ribbon is highlighted with a red box.

	Module	Version	IOC	Instances
4	TimingReceiver	2.1a5	ioc1	1
5	DataAcquisition	1,1	ioc1	4
6	BeamPositionMonitor	1	ioc1	1
7	basler	0.0a1	ioc2	3
8	fug	0.0a1	ioc3	4
9	lambda	0.0a1	ioc4	3
10	sysmon	0.0a1	ioc1	1
11	sysmon	0.0a1	ioc2	1
12	sysmon	0.0a1	ioc3	1
13	iocmon	0.0a1	ioc1	1
14	iocmon	0.0a1	ioc2	1
15	iocmon	0.0a1	ioc3	1

Excel client / Module Instances

- Adding EPICS support module instances



The screenshot shows an Excel spreadsheet with the following data:

	Delete	Name	IOC	CAMERA_INSTANCE	CAMERA_SN	P
4	-	basler1	ioc2		2	121231767 ELI-NP
5	-	basler2	ioc2		3	121231768 ELI-NP
6	-	basler3	ioc2		4	121231764 ELI-NP

Excel client / PLC

Screenshot of an Excel spreadsheet interface for a PLC configuration tool.

The ribbon menu includes:

- FILE
- HOME
- INSERT
- PAGE LAYOUT
- FORMULAS
- DATA
- REVIEW
- VIEW
- DCDB

The PLC tab is selected, showing the following icons:

- Settings
- Modules
- IOCs
- Support modules
- st.cmd
- Restart
- PLCs
- Blocks
- UDT
- Generate (highlighted with a red box)
- Update
- Commit

The main area shows a table with columns: Delete, Object name, IO, and PLC. The PLC column contains values like ioc1, plc2, and pyrobreaker.

A modal dialog box titled "Generate PLC config files" is displayed, prompting the user to "Select IOC:" with a dropdown menu containing "ioc1". It also features "OK" and "Cancel" buttons.

Delete	Object name	IO	PLC
-	valve1	ioc	
-	valve2	ioc	
-	valve3	ioc	
-	pb1	ioc	
-	pb2	ioc	
-	pb3	ioc1	plc2
			pyrobreaker

RPM packages



- ❑ m-python-dcdb-rest (REST server, import tool)
- ❑ m-dotnet-dcdb-ribbon (client in the form of a MS Excel add-on)

- ❑ CODAC support:
 - m-epics-dlloader
 - IOC, library, EPICS templates, stcmdsaver service
 - m-python-modules (vendor python packages)
 - m-common (dlloader support)
 - m-maven-iter-plugin (dlloader support)
 - m-codac-unit-api (dlloader support)

Build server



- ❑ Using Jenkins for automatic builds (nightly)
 - <http://build01.cosylab.com/>

A view dedicated to BleD/DCDB-tool (maintained by pmaslov)

All	ESS Units	ESS-bled (pmaslov)	FAIR	LetrikaSol	Solaris
S	W	Name ↓	Last Success	Last Failure	Last Duration
		m-codac-unit-api	5 days 7 hr - #191	2 mo 4 days - #1	26 sec
		m-common	5 days 7 hr - #514	N/A	9.8 sec
		m-dotnet-bledRibbon	2 days 21 hr - #27	6 days 3 hr - #17	15 sec
		m-epics-dlloader ▾	5 days 5 hr - #751	N/A	13 sec
		m-maven-iter-plugin	5 days 7 hr - #369	N/A	17 sec
		m-python-bledREST	20 days - #342	23 days - #331	35 sec
		m-python-modules	5 days 7 hr - #204	1 mo 20 days - #67	1 min 12 sec

Icon: [S](#) [M](#) [L](#)

Legend [RSS for all](#) [RSS for failures](#) [RSS for just latest builds](#)

Installation

□ Prerequisites:

```
bled@cmp-r1~$ sudo apt-get update
bled@cmp-r1~$ sudo apt-get install python-setuptools python-dev build-essential
bled@cmp-r1~$ sudo easy_install pip
bled@cmp-r1~$ sudo pip install flask-restful sqlalchemy pymysql
bled@cmp-r1~$ sudo pip install python-dateutil paramiko
bled@cmp-r1~$ sudo apt-get install mysql-server mysql-client
```

```
bled@cmp-r1~$ mysql -u root -p
bled@cmp-r1~$ mysql> create user bled@'%' identified by 'bled';
bled@cmp-r1~$ mysql> create user bled@'localhost' identified by 'bled';
bled@cmp-r1~$ mysql> grant all on *.* to bled@'%';
bled@cmp-r1~$ mysql> grant all on *.* to bled@'localhost';
bled@cmp-r1~$ mysql> flush privileges;
bled@cmp-r1~$ sudo service mysqld restart
```

□ Installation:

- Unpack the tarball



New features



FILE HOME INSERT PAGE LAYOUT FORMULAS DATA REVIEW VIEW DCDB

Settings	Modules	IOCs	Support modules	st.cmd	Restart	PLCs	Blocks	UDT	Generate	Update Commit
Setup	Overview	Configuration		Generate	IOC		PLC			Commit/update

DCDB-tool ★

File Edit View Insert Format Data Tools

DCDB Help

Attach to sidebar →

IOC manipulation

Configure IOCs - this will open up a spreadsheet with IOC preferences
 Generate st.cmd - generate and deploy st.cmd for the selected IOC
 Restart IOC - restart a selected IOC
 Restart all IOCs - restart all IOCs with the **Needs restarting** parameter set to true

Support modules

See modules overview - see a list of support modules
 Configure module instances - add/configure/delete support module instances

PLC

Configure PLCs - this will open up a spreadsheet with PLC preferences
 Configure PLC blocks - here you can configure DB blocks for PLC
 Add UDT type - add a new UDT type
 Configure UDT type - add new items to a UDT
 Delete UDT type - removes specific UDT type from DCDB
 Generate PLC - generates PLC configuration files (st.cmd, *.sdf, *.scf)

Comments Share

DCDB preferences

REST service URL:

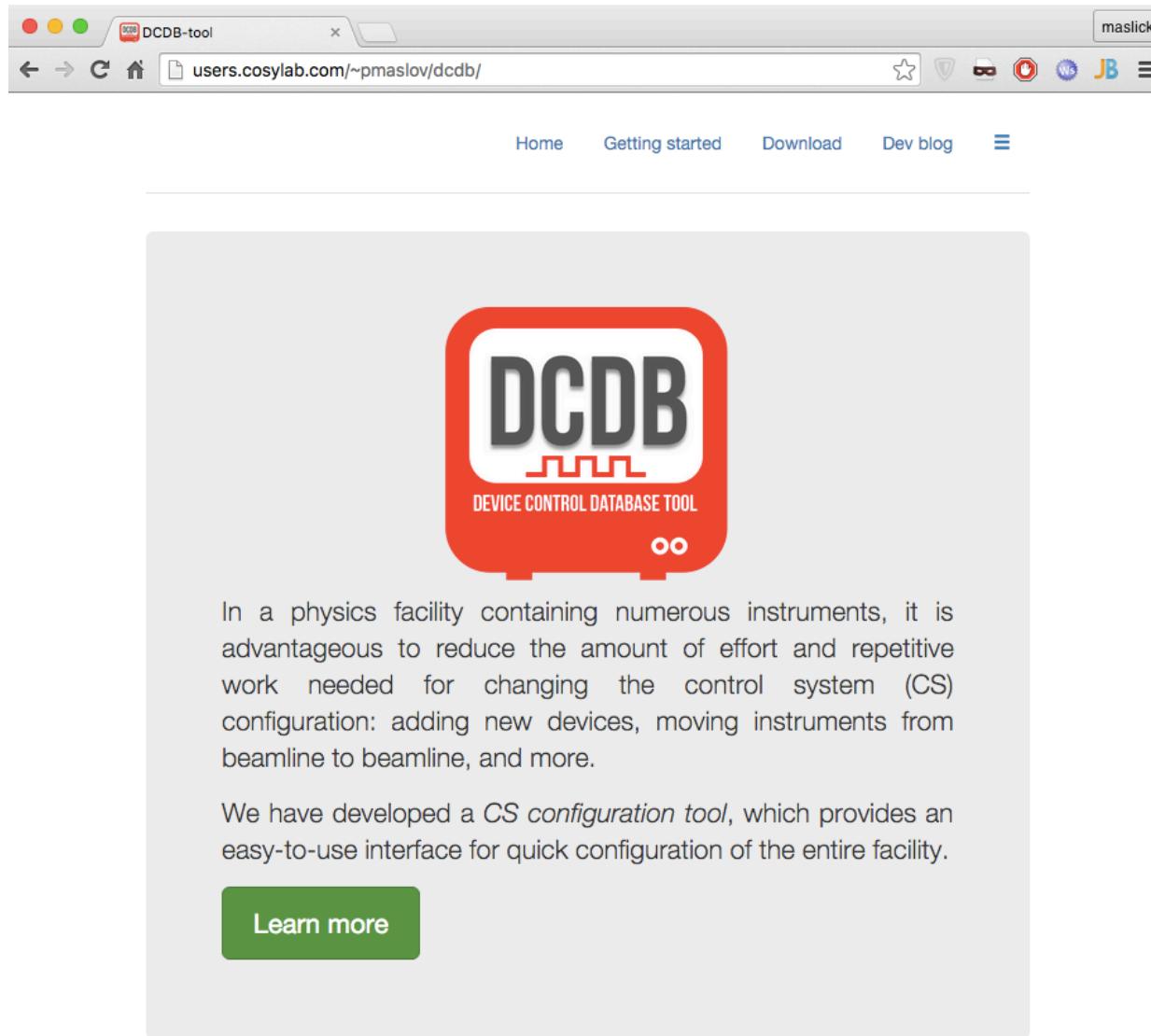
Port number:

Test connectivity

Apply

http://114.215.196.171:5000

Web-page



A screenshot of a web browser window showing the DCDB-tool website. The title bar reads "DCDB-tool". The address bar shows "users.cosylab.com/~pmaslov/dcdb/". The page content includes a navigation bar with links to "Home", "Getting started", "Download", "Dev blog", and a menu icon. Below the navigation is a large red button containing the DCDB logo and text. The main text area describes the tool's purpose in a physics facility and its configuration capabilities. A green "Learn more" button is at the bottom left of the main content area.

The DCDB logo is a red rounded square with a white border. Inside, the letters "DCDB" are in a large, bold, black sans-serif font. Below them is a small red graphic resembling a zigzag or a series of connected lines. At the bottom, the text "DEVICE CONTROL DATABASE TOOL" is written in a smaller, black, sans-serif font. There are two small white circles at the bottom corners of the red square.

In a physics facility containing numerous instruments, it is advantageous to reduce the amount of effort and repetitive work needed for changing the control system (CS) configuration: adding new devices, moving instruments from beamline to beamline, and more.

We have developed a *CS configuration tool*, which provides an easy-to-use interface for quick configuration of the entire facility.

[Learn more](#)

Demo

DCDB-tool demo

{using Docker container}

```
< Hi! This is a docker container. To start DCDB, run ./server.py >
-----
 \   \ \ \_ _/_ 
  \   \ \ \_ 
    (==)\_____
     (_)\ )\ \ \
      ||----w |
      || ||

root@67fb2c70bb13:/home/dcdb$
```

- Start the server `./server.py &`. It will be listening on port `5000`, which maps to `50449` on the docker host [<http://nds-devel.cosylab.com>]
- Check if the server is running by making use of our json API:
`curl http://nds-devel.cosylab.com:50449/excelnew`
You should get something like: `{ "message": "dcdb-server is running", "status": "OK" }`. See the complete API list in the user documentation.
- Install our front-end (Microsoft Excel add-on). One instance of it is running on [10.5.1.102] See more [details](#) on how to connect to this machine using Remote Desktop.
- Launch Excel, navigate to a tab named DCDB in the ribbon, click `Settings` and use `50449` for the port number and <http://nds-devel.cosylab.com> for the REST service URL.

Development blog



The screenshot shows the DCDB website homepage. It features a red header with the text "DEVICE CONTROL DATABASE TOOL" and a large "DCDB" logo with a zigzag base. Below the header are two white icons: a rocket ship and a thumbs-up. To the right are two more icons: a stopwatch and a piggy bank. A navigation bar at the bottom includes links for "About" and "FAQ". A search icon is also present.

DCDB to work in the ESS environment

Now the DCDB tool is compliant with the new ESS EPICS environment. They have a similar approach to using [dynamic library loading](#). The difference between ESS and our setup is **dependency resolution** and **unit development workflow**. The latter is done by simply creating a Makefile, adding source files and running *make install* as described here:

1. Create a new folder and put a Makefile with the following content:

```
include $(EPICS_ENV_PATH)/module.Makefile
LIBVERSION = 0.0a1
PROJECT = basler
DESCRIPTION = basler module
```

2. Create files init-pre.cmd, init-post.cmd and init.cmd. Init.cmd should have this line:

```
require basler, $(EPICS_CURRENT_MODULE_VERSION)
```

Recent Posts

- [DCDB to work in the ESS environment](#)
- [REST/api](#)
- [Developing EPICS device support modules](#)
- [Spreadsheets client finally published!](#)
- [non-CODAC installations](#)

October 2015

M	T	W	T	F	S	S
			1	2	3	4
5	6	7	8	9	10	11
12	13	14	15	16	17	18
19	20	21	22	23	24	25
26	27	28	29	30	31	

Links



- <http://users.cosylab.com/~pmaslov/dcdb>
- <http://users.cosylab.com/~pmaslov/dlloader>



THANK YOU

Pavel Maslov, MSc

COSYLAB

Tel.: +386 406 32 571

Web: www.cosylab.com

Your **TRUSTED** Control System Partner

