

IRMIS

Table of Contents

- IRMIS Overview
- **NEW items (Feb. 2014) are indicated by a blue italic bold font.**
- IRMIS @ SLAC
- UI and DATABASE QUERYING
- IRMIS@SLAC INFORMATION FLOW DIAGRAMS AND LOAD PROCESSES LIST
- OPERATIONAL and SUPPORT DETAILS
 - Emails, in chronological order
 - Daily monitoring of e-mails
 - At SLAC, what the PV crawlers update, in a nutshell
 - Oracle schemas and accounts
 - Other Oracle accounts
 - Database structure
 - Crawler scripts
 - cron jobs
 - PV crawler operation summary
 - LOGFILES, Oracle audit table
 - Crawler cron schedule and error reporting
 - Source directories
 - PV Crawler: what gets crawled and how
 - Removing an IOC from the active crawler list
 - TROUBLESHOOTING: post-crawl emails and data_validation_audit entries: what to worry about, how to bypass or fix for a few...
 - In the event of an error that stops crawling, here are the affected end-users of IRMIS PVs:
 - 3 possible worst case workarounds:
 - crawling a brand new directory structure
- HOW TO CHANGE THE IRMISDB ORACLE PASSWORD IN THE IRMIS GUI
- IRMISDB history data cleanup
- SCHEMA DIAGRAM

IRMIS Overview

"IRMIS is a collaborative effort among several EPICS sites to build a common Relational DataBase schema and a set of tools to populate and search an RDB that contains information about the operational EPICS IOCs installed at that site." IRMIS (the schema, crawler programs and UI) was developed by Don Dohan and Claude Saunders at APS. For general information and distributions see the [IRMIS Overview](#)

For reference and more information on EPICS, here is [EPICS Collaboration main site](#)

NEW items (Feb. 2014) are indicated by a blue italic bold font.

IRMIS @ SLAC

IRMIS is used at SLAC for the following purposes, from several different interfaces:

- Data source for PV names for AIDA (nightly cron jobs)
- Data source for element EPICS device names for LCLS_INFRASTRUCTURE (nightly cron jobs)
- PV list and data (IRMIS gui)
- IOC configuration and parameters (IOC Info APEX: <https://seal.slac.stanford.edu/apex/mccqa/f?p=104:8>)
- IOC and application configuration data (IOC Info jsp app: <https://seal.slac.stanford.edu/IRMISQueries>)
- EPICS camdmp application (APEX app: <https://seal.slac.stanford.edu/apex/mccqa/f?p=103:4>)
- ***CURRENTLY OBSOLETE DUE TO THE NEW ARCHIVER ENGINE*** Archiver PV search application (APEX app: <https://seal.slac.stanford.edu/apex/mccqa/f?p=259:8>)
- Lists of IOCs and their PV populations (web page: http://mccas0.slac.stanford.edu/crawler/ioc_report.html)
- ad hoc queries using TOAD, sqlplus, pl/sql, perl scripts, or other db query tool.
- IRMIS crawler logs, with duplicate PV reports: <http://www.slac.stanford.edu/grp/lcls/controls/sysGroup/report>

Elements of the IRMIS database that have been adopted and modified for the controls software group at SLAC:

- The Oracle PV schema
- PV, ALH crawlers
- the IRMIS gui
- Other elements of the collaboration IRMIS installation include cabling, device and application schemas. We are not populating these now...but may in the future.

Elements modified or created at SLAC:

- IOC boot syntax adaptations in the PV crawler and IOC Info crawler (adopted from SNS) and schema

- Special application-oriented tables in the IRMIS schema, and their population scripts (for example for EPICS camdmp and the archiver PV viewer app)
- ***Crawl of IOC.dbDumpRecord, which will slowly supercede crawling of st.cmd files.***
- Addition of config file tree structure crawling in the ALH crawler
- New crawlers for Channel Watcher and Channel Archiver
- New APEX guis
- *New PV Client viewer addition to the IRMIS Desktop - THERE BUT NO LONGER FUNCTIONAL, AND DATA OUT OF DATE*

[#top](#)

UI and DATABASE QUERYING

IRMIS GUI

This is a java UI for the IRMIS Oracle database, developed by Claude Saunders of the EPICS collaboration, can be invoked in 3 ways:

1. **from the lclshome edm display:**
click the "IRMIS..." button
2. **from a Solaris or Linux workstation:**
run this script:
irmisUI

The gui paradigm is a set of "document types"; click the File/New Document menu for the list. Right now there are only 2 available for use at SLAC:

- idt::pv – Search for lists of PVs and IOCs. This is the most useful interface, and it comes up upon application startup.

Query results can be saved to an ascii file for further processing.

IOC Parameters APEX application

This is an APEX application showing IOC configuration data, and various operational parameter snapshots (obtained live nightly using caget).

<https://seal.slac.stanford.edu/apex/mccqa/f?p=104:8>

EPICS camdmp APEX application

Various reports listing PVs and their module and channel connections.

<https://seal.slac.stanford.edu/apex/mccqa/f?p=103:4>

IOC list report (run nightly)

This html report is created nightly by the LCLS PV Crawler

http://mccas0.slac.stanford.edu/crawler/ioc_report.html

PV Crawler logs (which include the duplicate PV lists)

<http://www.slac.stanford.edu/grp/lcls/controls/sysGroup/report/>

a subset here: <http://www.slac.stanford.edu/cgi-bin/lwgate/CONTROLS-SOFTWARE-REPORTS/archives>

SQL querying

A view has been created to ease sql querying for PV lists. This view combines data from the IOC_BOOT, IOC, REC and REC_TYPE tables. It selects currently loaded PVs, where IOC_BOOT.CURRENT_LOAD = 1, with the latest IOC boot date captured.

- CURR_PVS view has all currently loaded PVs

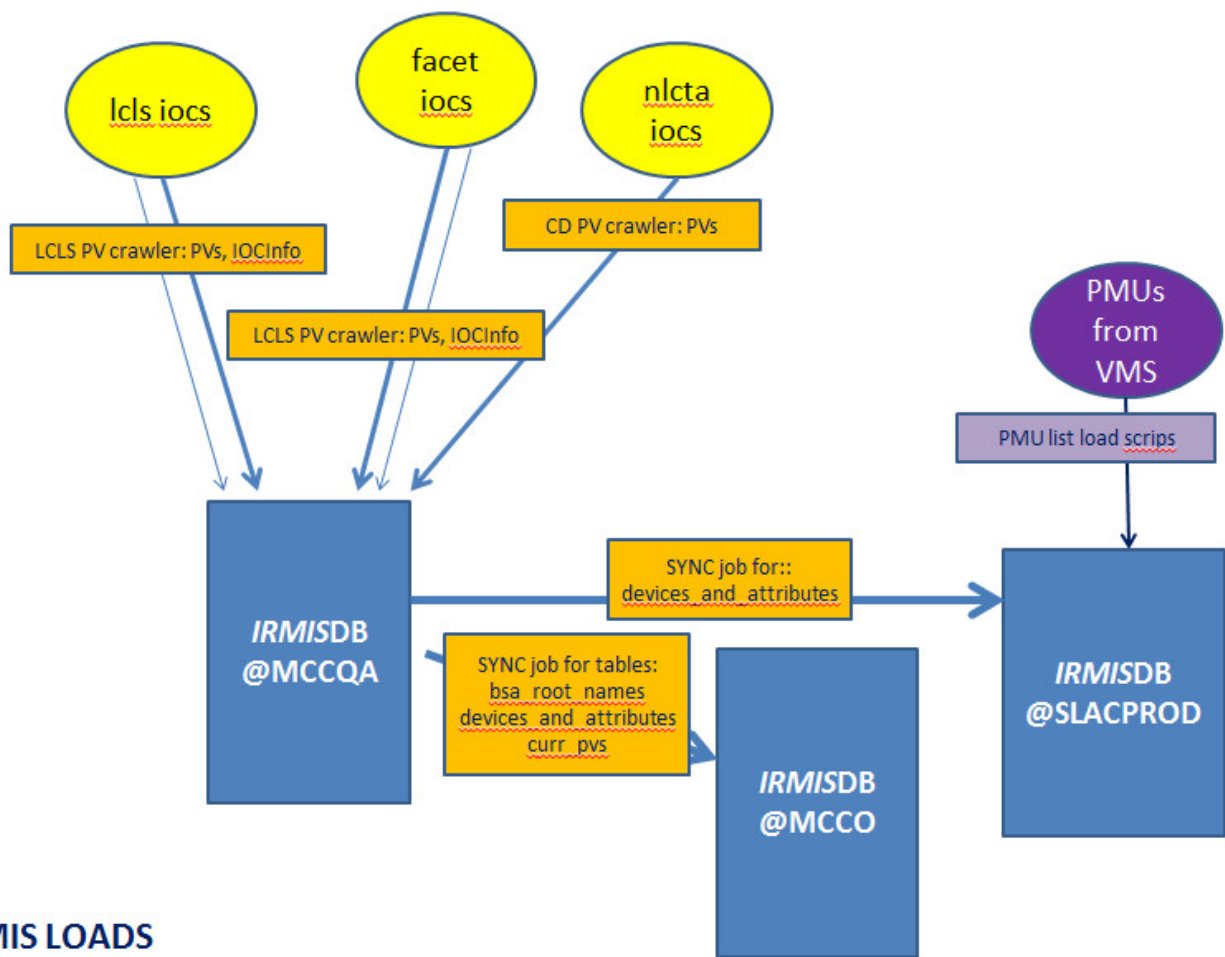
Discontinued: IOC Info query application

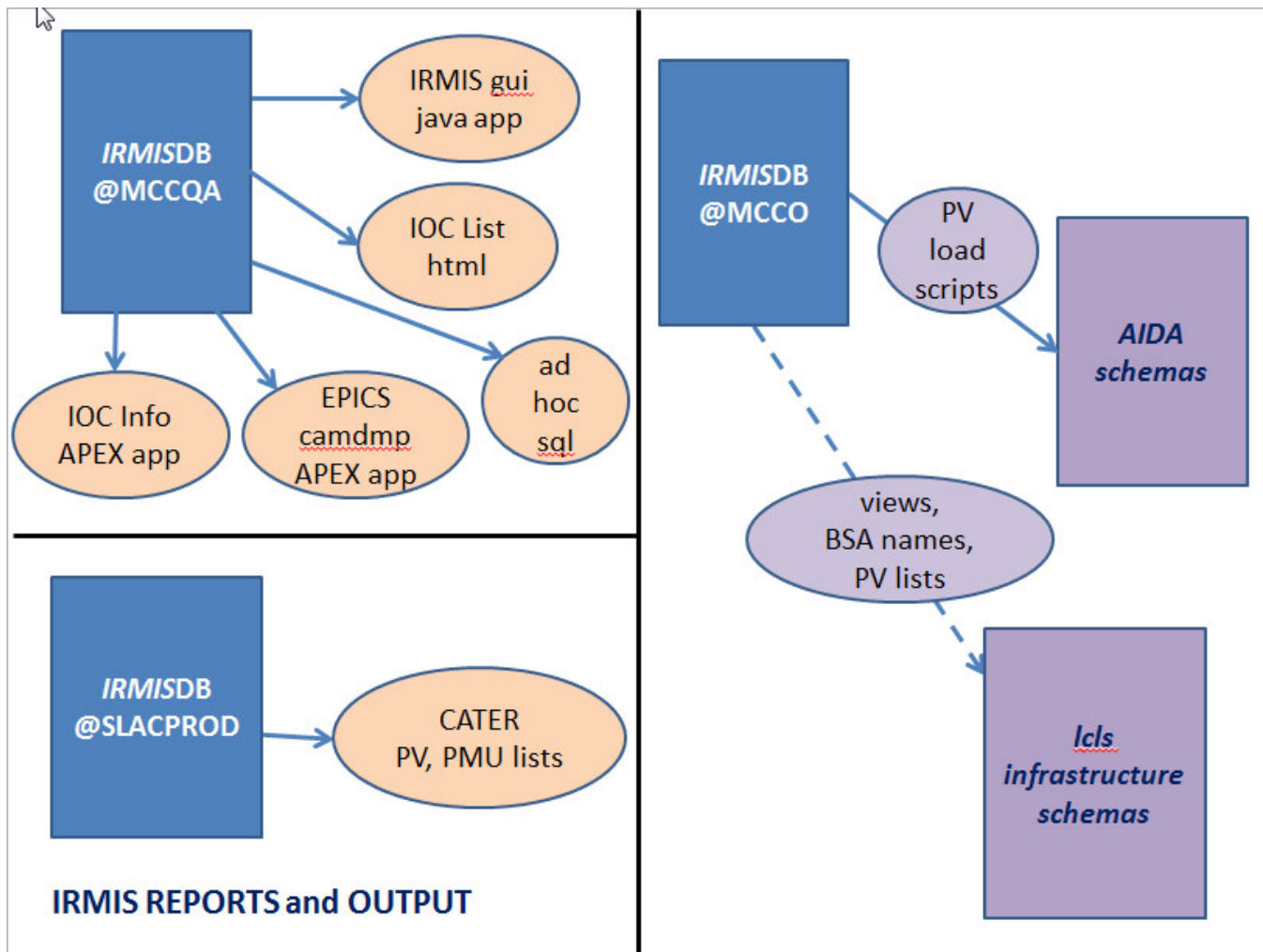
This is a jsp web application containing data about IOCs and applications, and their configurations. It is no longer maintained.

<https://seal.slac.stanford.edu/IRMISQueries/>

[#top](#)

IRMIS@SLAC INFORMATION FLOW DIAGRAMS AND LOAD PROCESSES LIST





IRMIS LOAD PROCESSES IN REVERSE CHRONOLOGICAL ORDER (snip from Data Validation daily e-mail)

CONTROLS_GLOBAL.DATA_VALIDATION_AUDIT ROWS FOR THE LAST CRAWLER RUNS				
PROCESS	STAGE	STATUS	MESSAGE	TIME
ALL_DATA_UPDATE	FINISH	SUCCESS	All is well	02-21-2014 01:30
FACET_DATA_VALIDATION	FINISH	SUCCESS		02-21-2014 01:30
FACET_DATA_VALIDATION	START			02-21-2014 01:30
DATA_VALIDATION	FINISH	SUCCESS		02-21-2014 01:30
DATA_VALIDATION	TEST			02-21-2014 01:30
CD PV Crawler	FINISH	SUCCESS	No problems found	02-21-2014 01:30
CD PV Crawler	START			02-21-2014 01:30
LCLS PV Crawler	FINISH	SUCCESS	WARNING: Crawler messages - see latest logfiles in /u1/lcls/tools/crawler/pv and status e-mails	02-20-2014 21:36
LCLS PV Crawler	START			02-20-2014 21:00
ALL_DATA_UPDATE	START			02-20-2014 21:00
FACET PV Crawler	FINISH	SUCCESS	WARNING: Crawler messages - see latest logfiles in /u1/facet/tools/crawler/pv and status e-mails	02-20-2014 20:00
FACET PV Crawler	START			02-20-2014 20:00

12 rows selected.

[#top](#)

OPERATIONAL and SUPPORT DETAILS

Each job and process sends a status e-mail to a selected list of recipients.

Emails, in chronological order

1. **FACET PV Crawler errors and warnings:** output from the FACET crawler (to: Shashi, Jingchen, Poonam, Judy) **note: warnings are not serious, just fyi. In general.**
2. **LCLS PV Crawler errors and warnings:** output from the LCLS crawler (to: controls-software-reports, Shashi, Jingchen, Poonam, Judy) **note: warnings are not serious, just fyi. In general.**
3. **MCCQA SLAC CD PV Crawler errors and warnings:** output from the CD (NLCTA) crawler (to: Shashi, Jingchen, Poonam, Judy)
4. **<timestamp> IRMIS data validation <SUCCESSFUL or FAILED>:** shows whether the data validation succeeded or not, and the status of each job in the sequence in reverse cron order, and logfiles to refer to in case of issues (to: controls-software-reports, Shashi, Jingchen, Poonam, Judy)
5. **IRMIS data sync to MCCO completed:** Only received if the data validation was successful, and the sync to MCCO completed. (to: Shashi, Jingchen, Poonam, Judy)
6. **DUPLICATE PVs FOUND in FACET and LCLS:** Only received if duplicate PVs were found. (to: controls-software-reports, Ernest, Shashi, Jingchen, Mike Z., Murali)

**** Only LCLS data validation will affect the sync to MCCO. FACET, and CD e-mails are fyi only.**

Daily monitoring of e-mails

1. Check the "IRMIS data validation" e-mail title: SUCCESS or FAILED? If it's SUCCESSful, and there is a subsequent "IRMIS data sync to MCCO completed", then all is well.
2. If the data validation FAILED:
 - i. What does the message for the DATA_VALIDATION step say? The errors found will be listed. (e.g. "PV Count drop > 5000", and/or "LCLS Crawler did not complete", etc.)
 - ii. Refer to the <acc> PV Crawler errors and warnings e-mails. These show the changes in PV counts by IOC (useful in locating which IOC's PVs count dropped for example)
 - iii. If needed, go to the logfiles: see "IRMIS data validation" email for the log directory for each crawler. Also, see table below for logfiles to refer to.

At SLAC, what the PV crawlers update, in a nutshell

- all IRMIS tables in MCCQA. All IRMIS UIs query data on MCCQA: PV and IOC data.
- 3 production tables on MCCO, which are used by other system – AIDA, BSA applications

Oracle schemas and accounts

The IRMIS database schema is installed in 4 SLAC Oracle instances:

- **MCCQA** IRMISDB schema: **Production for all gui's and applications.** Contains data populated nightly by perl crawler scripts from production IOC configuration files. Data validation is done following each load.
- **MCCO** IRMISDB schema: contains production data, currently for 3 tables only: bsa_root_names, devices_and_attributes, curr_pvs. MCCQA data is copied to MCCO once it has been validated. So MCCO is as close as possible to pristine data at all times.
- **SLACDEV** IRMISDB schema: sandbox testing and staging new features before release to production on MCCQA (see accounts below)
- **MCCODEV** IRMISDB schema: can be used for staging a release. Currently, the schema and data are out of date. Refresh from prod, then apply SLACDEV changes to use for staging.
- **SLACPROD** IRMISDB schema: this is **obsolete**; however, it is referenced in some scripts or other and has to be kept around until the scripts are pointed elsewhere. See Poonam, Elie, Mike Z. for details.

Other Oracle accounts

- IRMIS_RO – read-only account (not used much yet – but available)
- IOC_MGMT – created for earlier IOC info project with a member of the EPICS group which is not active at the moment- new ioc info work is being done using IRMISDB

For passwords see Poonam Pandey or Elie Grunhaus (or use getPwd)

As of September 15, all **crawler-related shell scripts and perl scripts use Oracle Wallet** to get the latest Oracle password. Oracle passwords must be changed every 6 months, and is coordinated and implemented by Poonam Pandey.



Note

The IRMIS GUI uses hardcoded passwords. This must be changed “manually” at every password change cycle. See below for detailed procedure.

Database structure

see [schema diagram](#) below. (this diagram excludes the EPICS camdmp structure, which is documented separately here: <url will be supplied>)

Crawler scripts

The PV crawler is run once for each IOC boot directory structure. The LCLS and FACET PV crawlers run the same code, that in the LCLS directory structure. The LCLS crawler code has been modified to be production-specific; it is a different version than the SLAC PV crawler.

The SLAC PV crawler (runSLACPVCrawler.csh) runs the crawler a couple of times to accommodate the various CD IOC directory structures.

NEW: As of mid-January 2014, the LCLS PV crawler now has a function to crawler \$IOC_DATA/<ioc>/IOC.dbDumpRecord files, which contain an IOC's record names and a subset of configured fields. This file is waaaaaay simpler to crawler than any ioc's st.cmd syntax. If IOC.dbDumpRecord is found, it will be crawled. If the file is not found, the old crawler code will operate on the ioc's st.cmd file. At the same time, the FACET PV crawler scripts were modified to share the LCLS crawler code, so IOC.dbDumpRecord files are also crawled for FACET.

As IOCs are rebooted during maintenance and upgrades, IOC.dbDumpRecord will eventually become the only files crawled in the system, resulting in easier maintenance.

cron jobs

- **LCLS side:** laci on lcls-daemon2: runLCLSPVcrawlerLx.bash: crawls LCLS PVs and creates lcls-specific tables (bsa_root_names, devices_and_attributes).
- **LCLS side:** laci on lcls-daemon2: caget4curr_ioc_device.bash: does cagets to populate curr_ioc_devices for the IOC Info APEX app. Run separately from the crawlers because cagets can hang unexpectedly – they are best done in an isolated script!
- **CD side:** laci on lcls-prod01: runAllCDCrawlers.csh: runs CD PV crawler, data validation, and sync to MCCO. *Client crawlers are no longer run! alh and car data sources have changed and the code has not been updated.*
- **FACET side:** flaci on facet-daemon1: runFACETPVcrawlerLx.bash: crawls FACET PVs.

PV crawler operation summary

For the location of the crawler scripts, see [Source code directories](#) below.
Basic steps as called by cron scripts are:

- run FACET pv crawler to populate MCCQA tables
- run LCLS pv crawlers to populate MCCQA tables
- run Data Validation for PV data in MCCQA
- if Data Validation returns SUCCESS, run synchronization of MCCQA data to selected (3 only at the moment) MCCO tables.
- run caget4curr_ioc_device to populate caget columns of curr_ioc_device

**** For PV Crawlers: the crawler group for any given IOC is determined by its row in the IOC table. The system column refers to the boot group for the IOC, as shown below.**

LOGFILES, Oracle audit table

Log filenames are created by appending a timestamp to the root name shown in the tables below.

The major steps in the crawler jobs write entries into the Oracle CONTROLS_GLOBAL.DATA_VALIDATION_AUDIT table. Each entry has these attributes:

- Instance
 - Schema
 - Process
 - Stage
 - Status
 - Message
 - TOD (time of day)
- (see below for details on querying this table)

Descriptions of the MAIN scripts (there other subsidiary scripts as well):

these are all ultimately invoked from the cron jobs shown above; the cron scripts call the others.

BLUE script names are on the CD side

GREEN script names are on the LCLS side

PURPLE script names are on the FACET side

Unknown macro: {td}	
Unknown macro: {td}	
Unknown macro: {td}	
Unknown macro: {td}	script name
Unknown macro: {td}	descr
Unknown macro: {td}	runAllCDCrawlers.csh
Unknown macro: {td}	irmis/cd_script

runs

- SLAC PV crawler
 - data validation for all current crawl data (LCLS and CD)
 - sync to MCCO
- Logfile: /nfs/slac/g/cd/log/irmis/pv/ CDCrawlerAll.log

Unknown macro: {td}	
Unknown macro: {td}	runSLACPVCrawler.csh
Unknown macro: {td}	irmis/cd_script
Unknown macro: {td}	

run by runAllCDCrawlers.csh: crawls NLCTA IOC's (previously handled PEP'II IOC's)
The PV crawler is run 4 times within this script to accommodate the various boot directory structures:

system	IOC's crawled	log files
NLCTA	IOC's booted from \$CD_IOC on gtw00. Boot dirs mirrored to /nfs/mcdfs0/u1/pepii/mirror/cd/ioc for crawling	/nfs/slac/g/cd/log/irmis/pv/pv_crawlerLOG_ALL.* /tmp/pvCrawler.log
TR01	TR01 only (old boot structure)	/nfs/slac/g/cd/log/irmis/pv/pv_crawlerLOG_TR01.* /tmp/pvCrawler.log

Unknown macro: {td}
Unknown macro: {td}
Unknown macro: {td}

runLCLSPVCrawlerLx.bash
is/script/

crawls LCLS IOC's

system	IOC's crawled	log files
LCLS	LCLS PRODUCTION IOC's. all IOC dirs in \$IOC /usr/local/lcls/epics/iocCommon	/u1/lcls/tools/crawler/pv/pv_crawlerLOG_LCLS. timestamp /tmp/ LCLSPVCrawl.log (log in as laci@lcls-daemon2)

Unknown macro: {td}
Unknown macro: {td}
Unknown macro: {td}

runFACETPVCrawlerLx.bash
is/script/ runs LCLS crawler code

crawls FACET IOC's

system	IOC's crawled	log files
FACET	FACET PRODUCTION IOC's all IOC dirs in facet \$IOC /usr/local/facet/epics /iocCommon	/u1/facettools/crawler/pv/pv_crawlerLOG_LCLS. timestamp /tmp/FACETPVCrawl.log (log in as flaci@facet-daemon1)

Unknown macro: {td}
Unknown macro: {td}
Unknown macro: {td}
Unknown macro: {tr}

run_find_devices.bash
is/script/
for LCLS and FACET only, populate the devices_and_attributes table, a list of device names and attributes based on the LCLS
or PV DEV:AREA:UNIT:ATTRIBUTE, DEV:AREA:UNIT is the device, ATTRIBUTE is the attribute.

Unknown macro: {td}
Unknown macro: {td}

run_load_bsa_root_names.bash
is/script/

loads bsa_root_names table by running stored procedure LOAD_BSA_ROOT_NAMES. LCLS and FACET names.

Unknown macro: {td}
Unknown macro: {td}
Unknown macro: {td}
Unknown macro: {td}

ioc_report.bash
is/script/
ioc_report-facet.bash
is/script/

runs at the end of the LCLS PV crawl, which is last, creates the web ioc report:

http://www.slacsfed2.slu/grp/cd/soft/database/reports/ioc_report.html

Unknown macro: {td}
Unknown macro: {td}
Unknown macro: {td}
Unknown macro: {td}

updateMaterializedViews.csh
irmis/cd_script
refreshes materialized view from curr_pvs

Unknown macro: {td}

findDupePVs.bash
is/script/

findDupePVs-all.bash

Unknown macro: {td}

finds duplicate PVs for reporting to e-mail
system as a parameter)

Unknown macro: {td}

refresh_curr_ioc_device.bash
is/script/

Unknown macro: {td}

refreshes the curr_ioc_device table with currently booted info from ioc_device (speeds up query in the Archiver PV APEX app).
device.bash below.

Unknown macro: {td}

find_LCLSpv_count_changes.bash
is/script/

is/script/

Value of α
0.0000
0.0001
0.0002
0.0003
0.0004
0.0005
0.0006
0.0007
0.0008
0.0009
0.0010
0.0011
0.0012
0.0013
0.0014
0.0015
0.0016
0.0017
0.0018
0.0019
0.0020
0.0021
0.0022
0.0023
0.0024
0.0025
0.0026
0.0027
0.0028
0.0029
0.0030
0.0031
0.0032
0.0033
0.0034
0.0035
0.0036
0.0037
0.0038
0.0039
0.0040
0.0041
0.0042
0.0043
0.0044
0.0045
0.0046
0.0047
0.0048
0.0049
0.0050
0.0051
0.0052
0.0053
0.0054
0.0055
0.0056
0.0057
0.0058
0.0059
0.0060
0.0061
0.0062
0.0063
0.0064
0.0065
0.0066
0.0067
0.0068
0.0069
0.0070
0.0071
0.0072
0.0073
0.0074
0.0075
0.0076
0.0077
0.0078
0.0079
0.0080
0.0081
0.0082
0.0083
0.0084
0.0085
0.0086
0.0087
0.0088
0.0089
0.0090
0.0091
0.0092
0.0093
0.0094
0.0095
0.0096
0.0097
0.0098
0.0099
0.0100
0.0101
0.0102
0.0103
0.0104
0.0105
0.0106
0.0107
0.0108
0.0109
0.0110
0.0111
0.0112
0.0113
0.0114
0.0115
0.0116
0.0117
0.0118
0.0119
0.0120
0.0121
0.0122
0.0123
0.0124
0.0125
0.0126
0.0127
0.0128
0.0129
0.0130
0.0131
0.0132
0.0133
0.0134
0.0135
0.0136
0.0137
0.0138
0.0139
0.0140
0.0141
0.0142
0.0143
0.0144
0.0145
0.0146
0.0147
0.0148
0.0149
0.0150
0.0151
0.0152
0.0153
0.0154
0.0155
0.0156
0.0157
0.0158
0.0159
0.0160
0.0161
0.0162
0.0163
0.0164
0.0165
0.0166
0.0167
0.0168
0.0169
0.0170
0.0171
0.0172
0.0173
0.0174
0.0175
0.0176
0.0177
0.0178
0.0179
0.0180
0.0181
0.0182
0.0183
0.0184
0.0185
0.0186
0.0187
0.0188
0.0189
0.0190
0.0191
0.0192
0.0193
0.0194
0.0195
0.0196
0.0197
0.0198
0.0199
0.0200
0.0201
0.0202
0.0203
0.0204
0.0205
0.0206
0.0207
0.0208
0.0209
0.0210
0.0211
0.0212
0.0213</

```
populate_io_counters_and_fields.bash
```

```
run_parse_camac_io.bash
```

[illegible]

Figure 1

[In /usr/local/csr/bin/mis/script/](#)

in /afs/slac/g/cd/soft/tools/irmis/cd_script

.../script/

- Unbalanced means: (2x)

fundata.validation@ac.esh

Thelotrema masonii (Ach.)

[illegible]

```
firmis/cd_script
```

Unknown macro: {td}

Definition 1.1.1 Let $f: A \rightarrow B$ be a function. Then

```
#!/script/
```

layers

[#top](#)

time	script	cron owner/host
8 pm	runFACETPVCrawlerLx.bash	flaci/facet-daemon1
9 pm	runLCLSPVCrawlerLx.bash	laci/lcls-daemon2
1:30 am	runAllCDCrawlers.csh	laci/lcls-prod01
4 am	caget4curr_ioc_device.bash	laci/lcls-daemon2

Source directories

description	cvs root	production directory tree root	details
IRMIS software	<p>SLAC code has diverged from the original collaboration version, and LCLS IRMIS code has diverged from the main SLAC code (i.e. we have 2 different version of the IRMIS PV crawler.)</p> <p>CD: /afs/slac/package/epics/tools/irmisV2_SLAC http://www.slac.stanford.edu/cgi-wrap/cvsweb/tools/irmisV2_SLAC/?cvsroot=SLAC-EPICS-Releases</p> <p>LCLS:</p> <p>Please note different CVS location: the production directory for the LCLS crawlers is /usr/local/lcls/package/irmis/irmisV2_SLAC/, the LCLS "package" dir</p>	<p>CD: /afs/slac /package/epics /tools /irmisV2_SLAC</p> <p>LCLS: /usr/local/lcls /package/irmis /irmisV2_SLAC</p>	<ul style="list-style-type: none"> db/src/crawlers contains crawler source code. SLAC-specific crawlers

	<p>structure is NOT in CVSed. So, to modify crawler code, you can cvs co the code, modify, test, copy the mods straight into production, then CVS them.</p> <p>CVS location is in tools/irmis/crawler_code_CVS http://www.slac.stanford.edu/cgi-wrap/cvsweb/tools/irmis/crawler_code_CVS/?cvsroot=LCLS</p>		<p>are in directories named *SLAC</p> <ul style="list-style-type: none"> db/src/crawlers/pv contains PV crawler source code db.properties file points to the database instance and username apps/src contains UI source code. apps/build.xml is the ant build file README shows how to build the UI app using ant
CD scripts	<p>For ease and clarity, the CD scripts are also in the LCLS CVS repository under tools/irmis/cd_script, cd_utils, cd_config http://www.slac.stanford.edu/cgi-wrap/cvsweb/tools/irmis/crawler_code_CVS/?cvsroot=LCLS</p>	/afs/slac/g/cd/tools/irmis	<ul style="list-style-type: none"> cd_script contains most scripts cd_utils has a few scripts cd_config contains client crawler directory lists
LCLS scripts	<p>/afs/slac/g/lcls/cvs http://www.slac.stanford.edu/cgi-wrap/cvsweb/tools/irmis/?cvsroot=LCLS</p>	/usr/local/lcls/tools/irmis	<ul style="list-style-type: none"> script contains the LCLS pv crawler run scripts, etc. util has some subsidiary scripts
FACET scripts	<p>These scripts share the LCLS repository (different names so they don't collide with LCLS scripts) http://www.slac.stanford.edu/cgi-wrap/cvsweb/tools/irmis/crawler_code_CVS/?cvsroot=LCLS</p>	/usr/local/facet/tools/irmis	<ul style="list-style-type: none"> script contains the FACET pv crawler run scripts, etc. util has some subsidiary scripts

PV Crawler: what gets crawled and how

- runSLACPVCrawler.csh, runLCLSPVCrawlerLx.bash and runFACETPVCrawlerLx.bash set up for and run the IRMIS pv crawler multiple times to hit all the boot structures and crawl groups. Environment variables set in pvCrawlerSetup.bash, pvCrawlerSetup-facet.bash and pvCrawlerSetup.csh, point the crawler to IOC boot directories, log directories, etc. Throughout operation, errors and warnings are written to log files.
- The IOC table in the IRMIS schema contains the master list of IOCs. The SYSTEM column designates which crawler group the IOC belongs to.
- An IOC will be hit by the pv crawler if its ACTIVE column is 1 in the IOC table: ACTIVE is set to 0 and 1 by the crawler scripts, depending on SYSTEM, to control what is crawled. i.e. LCLS IOCs are set to active by the LCLS crawler job. NLCTA IOCs are set to active by the CD crawler job.
- An IOC will be crawled if it has been rebooted since the last crawler run: if its LCLS/FACET STARTTOD or CD TIMEOF BOOT (boot time) PV has changed since the last crawl. Otherwise it will be skipped.
- Crawler results will be saved to the DB if at least boot file has changed, as determined by mod date and file size.
- Crawling specific files can be triggered by changing the mod date (e.g. touch).
- In the pv_crawler.pl script, there's a mechanism for forcing the crawling of subsets of IOCs (see the pv_crawler.pl code)
- IOCs without a STARTTOD/TIMEOFBOOT PV will be crawled every time. (altho PV data is only written when IOC .db files have changed)

[#top](#)

Removing an IOC from the active crawler list

Sometimes an IOC becomes obsolete. Although IOCs are automatically added to the IRMISDB.IOC table when the crawler encounters a new \$IOC /<ioc> directory, IOCs are not automatically removed from the system. Typically, an IOC engineer will request that her IOC be removed from the crawler (and hence all current PV and IOC lists, and reports).

To remove an IOC from the active crawler list:

Update the IRMISDB.IOC table in MCCQA to set the SYSTEM value to something that is not understood by the crawler:

```
update ioc  
  
set system = 'LCLS-NOCRWL'  
  
where ioc_nm = 'iocname';
```

or

```
update ioc  
  
set system = 'FACET-NOCRWL'  
  
where ioc_nm = 'iocname';
```

TROUBLESHOOTING: post-crawl emails and data_validation_audit entries: what to worry about, how to bypass or fix for a few...



Please Note

Once a problem is corrected, the next night's successful crawler jobs will fix up the data.

- The DATA_VALIDATION FINISH step MSG field in the CONTROLS_GLOBAL.DATA_VALIDATION_AUDIT table will show the list of errors found by the data validation proc:
 - Log into Oracle on MCCQA as IRMISDB
 - use TOAD or
 - from the Linux command line:
 - source /usr/local/lcls/epics/setup/oracleSetup.bash
 - export TWO_TASK=MCCQA
 - sqlplus irmisdb/ getPwd irmisdb`

select * from controls_global.data_validation_audit where schema_nm='IRMISDB' order by tod desc;

This will show you (most recent first) the data validation entries, with any error messages.

You can also see a complete listing of data_validation_audit entries in reverse chron order by using AIDA (but you will have to pick out the IRMISDB lines):

- Launch AIDA web <https://mccas1.slac.stanford.edu/aidaweb>
- In the query line, enter LCLS/DBValidationStatus
 - ** Please focus only on entries where schema_nm is IRMISDB. This report shows entries for ALL of our database operations; sometimes entries from different systems are interleaved.*

For a "good" set of IRMISDB entries, scroll down and see those from: 9/22 9 pm, continuing into 9/23, which completed successfully. The steps in ascending order are:

FACET PV Crawler start
 FACET PV Crawler finish
 ALL_DATA_UPDATE start
 LCLS PV Crawler start
 LCLS PV Crawler finish
 CD PV Crawler start
 CD PV Crawler finish
 DATA_VALIDATION start
 DATA_VALIDATION finish
 FACET DATA_VALIDATION start
 FACET DATA_VALIDATION finish
 ALL_DATA_UPDATE finish

then there are multiple steps for the sync to MCCO, labelled REFRESH_MCCO_IRMIS_TABLES



Note

REFRESH_MCCO_IRMIS_TABLES will not kick off unless ALL_DATA_UPDATE finishes with a status of SUCCESS.

ONLY VALIDATION OF LCLS STEPS AND SUCCESS WILL AFFECT THE SYNC TO MCCO. FACET, and CD e-mails are FYI ONLY.

some specific error checks:

- **comparison of PV counts: pre and post crawl**
 An error is flagged if the total PV count has dropped more than 5000, because this may indicate that some IOCs have been skipped that were previously being crawled. This would happen when a change occurs in the IOC boot files or IOC boot structure that the crawler doesn't understand. For example, if the crawler encounters new IOC startup file syntax that it doesn't understand, but that doesn't actually crash the crawler program.
 The logfile and e-mail message will tell you which IOCs are affected. The first thing to do is check with the responsible IOC engineer(s). It's possible that the PV count drop is "real", i.e. the engineer(s) in question intentionally removed a large block of PVs from the system.
- **If the drop is intentional:** You will need to update the database manually to enable the crawler to proceed: the count difference is discovered by comparing the row count of the newly populated curr_pvs view with the row count of the materialized view curr_pvs_mv, which was updated on the previous day. If the PV count has dropped > 5000 PVs, the synchronization is cautious - it prevents updating good data in MCCO until the reason for the drop is known.
 The step to say "it's ok, the drop was intentional or at least non-destructive - synchronization is now ok" is to go ahead and update the materialized view with current data. Then the next time the crawler runs, the counts will be closer (barring some other problem) and the synchronization can proceed.
 So, to enable the crawlers proceed, you need to update the materialized view with data in the current curr_pvs view, like this:
 log in as laci on lcls-prod01 and run
 /afs/slac/g/cd/tools/irmis/cd_script/updateMaterializedViews.csh
 The next time the crawler runs (that night), the data validation will be making the comparison with correct current data, and you should be good to go.
- ◦ **If the drop is NOT intentional:** Check the log for that IOC (search for "Processing iocname") and/or ask the IOC engineer to check for changes in the IOC boot directories, and IOC boot files for:
 - a syntax error in the boot file or directory
 - new IOC st.cmd syntax has change to something that the crawler hasn't learned yet. May require a PVCrawlerParser.pm modification to teach IRMIS a new syntax. This is rare but does occur. It's possible the IOC engineer can switch back to an older-style syntax, add quotes around a string, etc. to temporarily adjust the situation.
- ◦ **comparison of processed IOC counts in 2 log files**
 If the numbers don't match, the PV Crawler terminated early, before it processed all the IOCs. This usually happens when the crawler encounters new IOC startup file syntax that crashes the crawler perl program. Go to the logfile to determine the cause. This is rare, but usually requires a PVCrawlerParser.pm modification. As a temporary fix, update the IOC table and for the offending IOC(s), set SYSTEM to something other than LCLS or NLCTA (e.g. LCLS-TEMP-NOCRAWL) so it/they will not be crawled next time.
- ◦ **bsa_root_names row count checks:** check for problem accessing the table, or 0 or otherwise unexpected row counts for expected bsa types
- ◦ **devices_and_attributes row count checks:** problem accessing the table, or row count 0, of devices_and_attributes row count differs from distinct rows in the curr_pvs view.
- ◦ **If there is a discrepancy:**
 - A possible candidate is PVs with the same name in both FACET and LCLS, or between more than 1 IOC in LCLS. Since devices_and_attributes has unique PV names, while in curr_pvs PV name+IOC name, names duplicated in curr_pvs will only show up once in devices_and_attributes, hence a count difference. Here is a useful select statement to find the duplicates:
 select distinct (rec_nm) from curr_pvs where system='LCLS'
 minus
 select distinct(rec_nm) from devices_and_attributes where system='LCLS'
 - The fix will have to occur in the implicated IOC application...and the next crawl following the fix will resolved the problem.
 - For a shortterm mitigation, edit IRMISDataValidation.pl and comment out the check_DEVICES_AND_ATTRIBUTES step.
- ◦ **check for successful completion of all crawler steps:** check to make sure all crawler steps enter a start and a completion row in controls_global.data_validation_audit.
some circumstances where steps are missing:

- step launched but didn't finish: check the status of processes launched by the cron job using `ps --ef | grep`. An example: when perl dbi was hanging due to the 199-day-Linux-server-uptime bug. Several LCLS PV crawler jobs had launched, but had hung in the db_connect statement, and had to be killed from the Linux command line.
- step launched and finished, but the completed step was never written: the getPwd problems cause this symptom. See entries starting 9/23 9 pm for an illustration.
- step never launched: is the script available? Is the server up? Is crontab/trsrontab configured correctly? Are there permission problems? etc.
- other mysteries: figure out where the job in question stopped, using `ps --ef`, logfiles, etc...

Logfile messages sent in e-mail:

- - **Duplicate PV report** (lcls only near the top of the message)
Forward these to the relevant IOC Engineer.
 - **List of IOCs with PV counts that changed during the current crawl** (lcls only near the top of the message) – helpful for figuring out a big PV count drop
 - **IOCS not found in IRMIS, added to IOC table** (lcls only) – nice to know which new IOCs were just added.
If an IOC directory in \$IOC is not in the IRMIS IOC database table yet, IRMIS will automatically add it, and produce a message which you'll see here.
 - **PV Crawler log messages**
 - Don't worry about "expecting 3 arguments to nfsMount"!
 - Quickly check "Could not locate" - usually an IOC dir that isn't quite in production yet.
 - You can send a message to the IOC engineer if you see "unreachable" - it could be a typo, or it could be intentional.
 - A "Could not locate" an st.cmd file will create "Parse error: 2" messages, which can be ignored if the IOC isn't in production.

To bypass crawling specific IOC(s): update the IOC table and for the IOC(s) in question, set SYSTEM to something other than LCLS or NLCTA (e.g. LCLS-TEMP-NOCRAWL) so it/they will not be crawled next time. **NOTE:** if you change the SYSTEM column for an IOC, it will no longer appear in the curr_pvs view.

[#top](#)

In the event of an error that stops crawling, here are the affected end-users of IRMIS PVs:

(usually not an emergency, just decay of the accuracy of the lists over time.)

- **CURR_PVS:** IRMIS PVs create a daily current PV list, a view called curr_pvs. curr_pvs supplies lcls pv names to one of the AIDA names load jobs (LCLS EPICS names). The current PV and IOC lists are also queried by the IRMIS gui, and by web interfaces, and joined with data in lcls_infrastructure by Elie and co. for Greg and co.
- **BSA_ROOT_NAMES:** Qualifying IRMIS PVs populate the bsa_root_names table, which is joined in with Elie's complex views to device data in lcls_infrastructure. For details on the bsa_root_names load, please have a look at the code for the stored procedure which loads it (see one of the tables above)
- **DEVICES_AND_ATTRIBUTES:** PV names are parsed into device names to populate the **devices_and_attributes** table which is used by lcls_infrastructure and associated processing.
- **User interfaces may be affected (but they use MCCQA, so are not affected by failure of the sync to MCCO step):**
 - IRMIS gui
 - the web IOC Report
 - IRMIS iocInfo web and APEX apps
 - EPICS camdmp APEX app
 - Archiver PV search APEX app
 - ad hoc querying

[#top](#)

3 possible worst case workarounds:

1. Synchronization to MCCO that bypasses error checking
 - If you need to run the synchronization to MCCO even though IRMISDataValidation.pl failed (i.e. the LCLS crawler ran fine, but others failed), you can run a special version that bypasses the error checking, and runs the sync no matter what. It's:
`/afs/slac/u/cd/jrock/jrock2/DBTEST/tools/irmis/cd_script/runSync-no-check.csh`
2. Comment out code in IRMISDataValidation.pl
 - If the data validation needs to bypass a step, you can edit IRMISDataValidation.pl (see above tables for location) to remove or change a data validation step and enable the crawler jobs to complete. For example, if a problem with the FACET crawlers causes the sync to MCCO not to run, you may want to simply remove the FACET crawler check from the data validation step.
3. Really worst case! Edit the MCCO tables manually
 - If the PV crawlers will not complete with a sync of good data to MCCO, and you decide to wait til November for me to fix it (this is fine – the PV crawler parser is a complicated piece of code that needs tender loving care and testing!), AND accelerator operations are affected by missing PVs in one of these tables, the tables can be updated manually with names that are needed to operate the machine:
 - aida_names (see Bob and Greg)
 - bsa_root_names (see Elie)
 - devices_and_attributes (see Elie)

[#top](#)

crawling a brand new directory structure

- start off by testing in the SLACDEV instance!! To do this, must check the pv crawler dir out of cvs into a test directory, and modify db.properties to point to SLACDEV instance PLUS cvs checkout crawl scripts and setenv TWO_TASK to SLACDEV for set*IOCsActive scripts.
- all IOC EPICS application directories have to be visible to the crawler.

- \$IOC_DATA has to be visible to the crawler.
- the host running the crawler must be able to "see" Oracle (software and database) and the boot directory structure.
- set up env vars in run*PVCrawler.csh and pvCrawlerSetup.csh.
- if necessary, create crawl group in IOC table, and set*IOCsActive.csh to activate.
- add call to pv_crawler.csh to run with the new env vars...
- hope it works!

[#top](#)

HOW TO CHANGE THE IRMISDB ORACLE PASSWORD IN THE IRMIS GUI

The IRMIS gui uses Hibernate for its database interface. It's possible the Oracle wallet could be superimposed on Hibernate, but the path for doing this is unclear and will take time to work out (maybe worth doing in the future tho!)

Therefore the IRMIS gui Oracle password must be changed manually. Here is the procedure:

1. log into mcclogin
2. cvs co irmis gui code:
 export CVSROOT=/afs/slac/package/epics/slaconly/cvs
 cd to a working directory
 cvs co -d irmis tools/irmisV2_SLAC
3. cd irmis
4. edit site.build.properties and change this line:
 db.trust-read-write.password=*newpw*
5. set java environment to version 1.4 (this may not be needed):
 export JAVA_HOME=/afs/slac/package/java/@sys/jdk1.4
 export JAVA_VER=1.4
 export PATH=\${JAVA_HOME}/bin:\${PATH}
6. cd db
7. ant clean
8. ant deploy
9. cd ..
10. cp db/build/irmisDb.jar apps/deploy
11. cd apps/deploy
12. test the gui:
 java -jar irmis.jar
 The gui should connect to the db.
13. deploy new jars to production:
 scp irmisDb.jar over to /usr/local/lcls/package/irmis/irmisV2_SLAC/apps/deploy/. (this one is used by LCLS and FACET)
 cp irmisDb.jar /afs/slac/package/epics/tools/irmisV2_SLAC/apps/deploy/.

*** Please note, the app directory (parallel to the db directory) is not build-able in ant anymore, because of changes in java version syntax and method calls deep in the gui code. Therefore, please follow the directions above to BUILD AND DEPLOY THE DB JAR ONLY!

[#top](#)

IRMISDB history data cleanup

\$TOOLS/irmis/data_cleanup has scripts and a README for cleaning historical IRMIS data. The procedure has been done once in February, 2012.

[Description and plans](#) and click IRMIS_database_cleanup

[#top](#)

SCHEMA DIAGRAM

