Components of IceCube DAQ in use at Pole, Jan-Feb. 2005

Edited by J. Jacobsen, jacobsen@npxdesigns.com

V1.6
02/03/05

Section One: Status/Overview

Project Name	**Runs on**	**Release/tag**	**Stable/released**	**Contact**	**Remarks**
DOM-MB	DOMs	308	yes	Jones	default configuration
DOM-MB	DOMs	pole-fb-01	yes	Jones/Jacobsen	flasher run configuration
dor-driver		V02-05-04	yes	Jacobsen	
domhub-common		V02-04-00	yes	Hays	
domhub-app	sps-ixhub-cont0x	V02-04-01	yes	Hays	
testdaq-io		V03-08-02	yes	Braun	
testdaq-lightsource		SPS-REL1	yes	Braun	
testdaq-control	sps-stringproc-cont0x	SPS-REL1	yes	Braun	
testdaq-collector	sps-stringproc-cont0x	SPS-REL1	yes	Braun	
testdaq-launch	sps-stringproc-cont0x	**forthcoming**	**no**	Krasberg	Mark's startup scripts
tiny-daq	sps-stringproc-cont0x	pole-daq-100	yes	Jones	
icebucket		V01-07-00	yes	Patton	
splicer		V02-02-01	yes	Patton	
iceboss-services		V01-01-01	yes	Patton	
triggerUtil		**forthcoming**	**no**	Wharton	
icetopTrig		**forthcoming**	**no**	Toale, Divia	Placeholder, not yet used
iniceTrig		V03-03-00	**no**	Toale	
daq-common		V03-01-00	yes	McParland, Patton	
monolith	sps-stringproc-cont0x	V00-01-00	**no**	Wharton, Toale	
daq-dispatch	sps-stringproc-cont0x	V00-01-00	**no**	Patton	
PFSYSTEM	sps-fpmaster,	V00-01-00	**no**	Blaufuss	
	sps-fpclient0x				
(data movement)					
spade	sps-sattx	V00-00-04	yes	Mackenzie	
tapeserver		V00-00-04	yes	Mackenzie	

Section Two: Program Summary Documentation

Still needed: ***Dan, Pat:*** *monolith*
Mark Krasberg: *testdaq launch scripts*

Programs:

```
               PROGRAM NAME: ldall
                  AUTHOR(S): Arthur Jones (ALJones@lbl.gov)
WHAT MACHINES(S) IT RUNS ON: any domhub
               WHAT IT DOES: upload DOM-MB release file (release.hex)
HOW TO INSTALL NEW VERSIONS: install dor-driver
   ADDITIONAL DOCUMENTATION: N/A
            HOW TO START IT: ldall release.hex
WHAT ELSE HAS TO BE RUNNING: dor-driver
              PROCESS NAMES
(what you'd see w/ "ps ax"): ldall, dd, domterm
          INPUT DESCRIPTION: DOM-MB release file “release.hex”
```

```
        OUTPUT DESCRIPTION: uploads “release.hex” onto DOM flash
          COMMON PITFALLS:  very stable

              PROGRAM NAME: moat
                 AUTHOR(S): John Jacobsen, Arthur Jones
WHAT MACHINES(S) IT RUNS ON: any domhub
              WHAT IT DOES: exhaustive test of comms/tcalib
HOW TO INSTALL NEW VERSIONS: install dor-driver
   ADDITIONAL DOCUMENTATION:
http://docushare.icecube.wisc.edu/docushare/dsweb/View/Collection-511
           HOW TO START IT: moat -d 10 -c 600 -t 3600 -r 25500 -g
                            (14 hour test)
WHAT ELSE HAS TO BE RUNNING: dor-driver
             PROCESS NAMES
 (what you'd see w/ "ps ax"): moat, stagedtests.pl, readwrite, tcaltest,
                            run-mjb.sh, ...
         INPUT DESCRIPTION: none
        OUTPUT DESCRIPTION: timestamped directory, symlinked as
                            "latest_moat", containing lots of
                            diagnostics, located wherever you started
                            moat from.
          COMMON PITFALLS:  hardware errors


              PROGRAM NAME: dor-driver
                 AUTHOR(S): John Jacobsen <jacobsen@npxdesigns.com>
WHAT MACHINES(S) IT RUNS ON: any domhub
              WHAT IT DOES: implements DOR communications interface
HOW TO INSTALL NEW VERSIONS: wget <release tarball>
                            tar xvzf <release tarball>
                            cd <release-dir>/driver
                            make
                            su
                            make install; make installfw;
                            /etc/init.d/dhrc restart
   ADDITIONAL DOCUMENTATION:
http://docushare.icecube.wisc.edu/docushare/dsweb/View/Collection-511
           HOW TO START IT: starts on boot
WHAT ELSE HAS TO BE RUNNING: nothing
             PROCESS NAMES
 (what you'd see w/ "ps ax"): none; "lsmod | grep dh" shows it's there
         INPUT DESCRIPTION: domhubapp, test scripts
        OUTPUT DESCRIPTION: "                      "
           COMMON PITFALLS:

              PROGRAM NAME: DOMHub App (TestDAQ version)
                 AUTHOR(S): David Hays <dehays@lbl.gov>
WHAT MACHINES(S) IT RUNS ON: any domhub
              WHAT IT DOES: Responds to RMI method calls from testdaq-control
                            to manage cable power, discover communicating DOMs
                            and configure DOMs. It creates data servers
                            for hit data, monitor data and time calibration data
                            to be consumed by the testdaq-collector client.

                            These functions are performed by interacting with the
                            DOR driver proc and dev file interface.
HOW TO INSTALL NEW VERSIONS: domhub-app.jar file must be placed in the
                            ~testdaq/work/lib directory

        CONFIGURATION FILE: /usr/local/etc/dh.properties
              DEPENDENCIES: domhub-common.jar
```

Environment tool dependencies (These are assumed to be available from the domhub system setup):
/usr/local/bin/domserv
commons-logging.jar, log4j.jar

ADDITIONAL DOCUMENTATION: DOM Hub Application Developers Guide:
http://docushare.icecube.wisc.edu/docushare/dsweb/Get/Document-7746/DHAppDevGuide.doc

HOW TO START IT: Normally started using Mark Krasberg's 'ready' script.
(~testdaq/bin/ready)
Actual command line with the necessary jars on CLASSPATH: java icecube.daq.domhub.DOMHub

WHAT ELSE HAS TO BE RUNNING: rmiregistry (If it is not already started, use the command 'rmiregistry &'. Also see COMMON PITFALL regarding CLASSPATH necessary when starting rmiregistry.)

PROCESS NAMES
(what you'd see w/ "ps ax"): rmiregistry
java icecube.daq.domhub.DOMHub /usr/local/etc/dh.properties
/usr/local/bin/domserv -dh

COMMON PITFALLS: RMI Exceptions :--
- Check that domhub-app.jar and domhub-common.jar were on CLASSPATH when rmiregistry was started.
- Check that the rmiHost value if the fully qualified name of the host on which DOMHub App is running.

Further troubleshooting, send the domhubapp.log file (or preferably, the log section for the period for which a problem was encountered) to dehays@lbl.gov

PROGRAM NAME: **testdaq-control**
AUTHOR(S): Jim Braun
WHAT MACHINE(S) IT RUNS ON: sps-stringproc01
WHAT IT DOES: **Manages data transfer between testdaq-collector and domhubs; supplies configuration information to DOMs through configuration input file**
HOW TO INSTALL NEW VERSIONS: testdaq-control.jar and testdaq-control-test.jar files must be placed in the ~testdaq/work/lib/ directory
ADDITIONAL DOCUMENTATION:
http://docushare.icecube.wisc.edu/docushare/dsweb/Get/Document-8910/tdaquserguide.ps
HOW TO START IT: java icecube.testdaq.control.RevTestDAQControlCommandLine
{config file} {output dir} {data description}
{-d domhub1 hostname} {-d domhub2 hostname} .....
WHAT ELSE HAS TO BE RUNNING: domhub-app must be running on all specified domhubs
PROCESS NAMES: java
INPUT DESCRIPTION: config file: Test description file suppling test configuration information and specific configuration information for each DOM
output dir: Directory to which data will be written
data description: Short description of data that will be taken
OUTPUT DESCRIPTION: N/A
COMMON PITFALLS: rmiregistry must be running on sps-stringproc01

PROGRAM NAME: **testdaq-collector**
AUTHOR(S): Jim Braun
WHAT MACHINE(S) IT RUNS ON: sps-stringproc01
WHAT IT DOES: **Applies time calibration to DOM data stream and writes DOM data to file through configuration commands from testdaq-control**
HOW TO INSTALL NEW VERSIONS: testdaq-collector.jar file must be placed in the ~testdaq/work/lib/ directory
ADDITIONAL DOCUMENTATION:

```
http://docushare.icecube.wisc.edu/docushare/dsweb/Get/Document-8910/tdaquserguide.ps
            HOW TO START IT: N/A
WHAT ELSE HAS TO BE RUNNING: N/A
              PROCESS NAMES: java
          INPUT DESCRIPTION: N/A
         OUTPUT DESCRIPTION: Raw testdaq-format DOM data file
            COMMON PITFALLS: Rmiregistry must be running on sps-stringproc01


               PROGRAM NAME: domcap-setup
                  AUTHOR(S): Arthur Jones <aljones@lbl.gov>
WHAT MACHINES(S) IT RUNS ON: any unix box, most likely sps-stringprocXX
               WHAT IT DOES: creates named pipes (fifos) for engdemux
HOW TO INSTALL NEW VERSIONS: make install from tiny-daq
   ADDITIONAL DOCUMENTATION: not yet
            HOW TO START IT: domcap-setup
WHAT ELSE HAS TO BE RUNNING: nothing
              PROCESS NAMES: domcap-setup, mkfifo
(what you'd see w/ "ps ax"):
          INPUT DESCRIPTION: none
         OUTPUT DESCRIPTION: side effect: /tmp/domcap/domXXX fifos created...
            COMMON PITFALLS: /tmp/domcap not owned by user.

               PROGRAM NAME: engdemux
                  AUTHOR(S): Arthur Jones <aljones@lbl.gov>
WHAT MACHINES(S) IT RUNS ON: any unix box, most likely sps-stringprocXX
               WHAT IT DOES: demuxes the stream of hits from testdaq
HOW TO INSTALL NEW VERSIONS: make install from tiny-daq
   ADDITIONAL DOCUMENTATION: not yet
            HOW TO START IT: engdemux file.hits (file.hits can be a pipe)
WHAT ELSE HAS TO BE RUNNING: nothing
              PROCESS NAMES: engdemux
(what you'd see w/ "ps ax"):
          INPUT DESCRIPTION: testdaq hits file
         OUTPUT DESCRIPTION: set of fifos in /
            COMMON PITFALLS: /tmp/domcap/domXXX not created yet with
                             domcap-setup script.

               PROGRAM NAME: engsort
                  AUTHOR(S): Arthur Jones <aljones@lbl.gov>
WHAT MACHINES(S) IT RUNS ON: any unix box, most likely sps-stringprocXX
               WHAT IT DOES: sorts demuxed data from testdaq
HOW TO INSTALL NEW VERSIONS: make install from tiny-daq
   ADDITIONAL DOCUMENTATION: not yet
            HOW TO START IT: engsort
WHAT ELSE HAS TO BE RUNNING: engdemux
              PROCESS NAMES: engsort, engdemux
(what you'd see w/ "ps ax"):
          INPUT DESCRIPTION: demuxed data pipes filled by engdemux
         OUTPUT DESCRIPTION: sorted testdaq hits file
            COMMON PITFALLS: /tmp/domcap/domXXX not created yet with
                             domcap-setup script.

               PROGRAM NAME: multtrig
                  AUTHOR(S): Arthur Jones <aljones@lbl.gov>
WHAT MACHINES(S) IT RUNS ON: any unix box, most likely sps-stringprocXX
               WHAT IT DOES: simple multiplicity trigger
HOW TO INSTALL NEW VERSIONS: make install in tiny-daq
   ADDITIONAL DOCUMENTATION: not yet
            HOW TO START IT: multtrig file.hits file.idx
WHAT ELSE HAS TO BE RUNNING: engsort
```

PROCESS NAMES: engsort, multtrig
(what you'd see w/ "ps ax"):
INPUT DESCRIPTION: sorted testdaq hits data (wrapped eng format)
OUTPUT DESCRIPTION: wrapped eng format and ascii index file
COMMON PITFALLS: times are in ns, not us.

PROGRAM NAME: **PFSYSTEM** (Consists of two basic pieces, a server process which connects to DaqDispatch for input (Master) and a client software which runs an instance of IceTray (from offline software suite)

AUTHOR(S): Erik Blaufuss <blaufuss@icecube.umd.edu>
Ping Huang <huang@icecube.umd.edu>

WHAT MACHINES(S) IT RUNS ON: Server process runs on sps-fpmaster. Client processes run on sps-fpslave01,02

WHAT IT DOES: **Processing and Filtering (PnF)**. The PnF system is designed to be the place where online reconstructions are performed and filter takes place. Currently, with the single string, and the lack of Pole-qualified reconstruction software, the PnF system is running, and simply passing all data through (filtering with all data set to KEEP). Currently, it's output is just a wrapper/header ontop of EventBuilder output format from Monolith.

HOW TO INSTALL NEW VERSIONS: New versions are built on sps-access within the SPOLE-SURFACE meta-project. System admins have install scripts to propogate installed versions to appropriate locations.

ADDITIONAL DOCUMENTATION: Forthcoming.

HOW TO START IT: pfclient-main processes are started automatically at boot time, and if they exit/disconnect from the server, they restart and attempt to reconnect automatically. No user intervention should be required to start/stop them.

Master (server process) is configured and controlled via the JMX console on sps-fpmaster (setup via ssh tunnelling).

In JMX Agent View on sps-fpmaster, the icecube.pnf subsection has three entries:

```
  configure- for configuring pnf (Default values at startup
             should be correct for operation on sps-*)
   control - for stopping/starting process
   monitor - checking data throughput, status, etc.
```

Additional information on current status of Master server process can be found in its run log: sps-fpmaster:*/usr/local/icecube/pfcommon/logs/** (look for latest)

WHAT ELSE HAS TO BE RUNNING: Master requires DAQ-Dispath to be running to serve as input. If not running, it will keep trying to connect for some period of time before giving up.

PROCESS NAMES
(what you'd see w/ "ps ax"): on sps-fpmaster: /usr/local/icecube/pfcommon/bin/Master
on sps-fpslave0?: ./bin/pfclient-main

INPUT DESCRIPTION: Socket connection to DAQ-Dispatch. Build events from DAQ Event Builder (or monolith) served up in chunks over socket.

OUTPUT DESCRIPTION: Files (eventually to be transfered over satellite/tape connection). A small header with filtering/reconstruction information on top event builder data format.

COMMON PITFALLS: Currently, biggest problem is instability of DAQ-Dispatch. If it crashes while the connection is open, then Master process could temporarily hang.

PROGRAM NAME: **Daq-Dispatch**
AUTHOR(S): Simon Patton <sjpatton@lbl.gov>
WHAT MACHINES(S) IT RUNS ON: sps-evbuilder
WHAT IT DOES: **Reads events from files** (or evbuilder when it is present) a**nd makes them available, via a network socket, to PnF.**
HOW TO INSTALL NEW VERSIONS: "stop" the iceboss0 system service on the node, install the new dd-control.sar, then "start" the iceboss0 system service.
ADDITIONAL DOCUMENTATION: In progress
HOW TO START IT: Access up the "control" MBean (see below) and invoke the "startup" operation.
WHAT ELSE HAS TO BE RUNNING: Nothing.
PROCESS NAMES (what you'd see w/ "ps ax"): java -server -Djava.security.manager=default [...] However the above only checks that JBoss is running, the following three MBeans should be available on the jmx-console when daq-dispatch is running. (All are currently in the "icecube.dispatch" domain)

acme-aspect=configuration,component=daqdispatch,daqdispatch=FakeClient
acme-aspect=control,component=daqdispatch,daqdispatch=Delivery
acme-aspect=monitor,component=daqdispatch,daqdispatch=Delivery

INPUT DESCRIPTION: Input is done via the "configuration" MBean, and can only be done when daq-dispatch is not in the running state. (The running state is available in the "control" MBean.

In the "configuration" MBean the only attribute which is important is the "source" attribute. This contains an XML element that describes the configuration of the data source and should look like the following.

```
<source>
      <watch iterator="icecube.control.dispatch.testdaq.TestDAQFileIterator">
            <param type="java.lang.String">
/data/exp/IceCube/2005/TestDAQ/00/00/SPS-DAQ-01_run0000000_PedestalPattern-ATWD0-
TurnOffPowerManagement/SPS-DAQ-01_run000000_PedestalPattern-
ATWD0TurnOffPowerManagement.events
            </param>
            <param type="java.lang.String">
                 events
            </param>
            <param type="java.lang.String">
                 events.ready
            </param>
      </watch>
</source>
```

Items to note are: - The first parameter should be the last file PnF successfully processed. (This will eventually be automated).

- In deployment the file name will have a different name for the 'steering file', check with Mark K. on what this should be.
- The second and third parameters are the file extension and semaphore files respectively. The ones shown here were used during development, but deployment may have different values.

OUTPUT DESCRIPTION: All output is sent to the JBoss log file.

COMMON PITFALLS: The current implementation does not delete data files once PnF has completely finished with them. Therefore the data disk filling up can be an issue.

The "watching" mechanism expects the data directory tree to be complete, with no extraneous or empty directories.
If either of these conditions occur then data forwarding may stall. Cleaning up the directory tree and restarting the service (remembering to reset the last run processed by PnF) should fix this.