artdaq - Idea #5961

Coordinate the requested changes to art for an online environment

04/18/2014 09:59 AM - Kurt Biery

Status:	Closed	**Start date:**	04/18/2014
Priority:	Normal	**Due date:**	
Assignee:		**% Done:**	0%
Category:	Needed Enhancements	**Estimated time:**	100.00 hours
Target version:			
Experiment:			

Description

There are a few changes that have been requested of art to make its use in an online environment more in line with behavior that people would like to see in a DAQ system.

These issues include:

- not creating a new disk file until a new run or subrun has started (see Issue #3984)
- supporting the automatic switching from one disk file to another when a configurable size or time limit is reached (e.g. Issue #3189)
- only calling analyzer module endrun and beginrun methods when runs end/begin (currently these are called when files are closed and opened, i.e. when subruns and and begin)

The purpose of this Issue is not to do the work. Relevant Issues in the art project should be created for that, and some have already been created. The purpose of this Issue is to work with the art team to describe the issues and answer questions (as needed) and integrate the art changes into artdaq (when they become available).

Related issues:

Related to ds50daq - Bug #4789: Why is MonitorSer fitting histograms in the m...	**Assigned**	**10/11/2013**	**10/18/2013**

History

#1 - 04/21/2014 10:03 AM - Kurt Biery

- Estimated time changed from 80.00 h to 100.00 h

Another issue to be added to the list: during our discussions about how to avoid problems when an online monitoring module throws an exception, it was suggested that it might be possible to configure art to suppress exceptions from specified modules (and leave the default behavior for other modules. We should look into this.

#2 - 04/21/2014 02:20 PM - Kurt Biery

In the interest of completeness, I'm including notes from Jim from a meeting last June. They may be out-of-date in some cases, but they may also provide useful background material in some cases.

```
Darkside-50 DAQ needs from art:
```

Since art is a fundamental feature within artdaq for handling the data processing, it is receiving update requests to further serve the needs of the DS-50 DAQ community. Kurt and I had a meeting with Alessandro this morning to talk through the request list and better understand the requirements (and shape them) and the relative priorities. These are listed in the priority order that Alessandro assigned. For each I will make a statement about the issue that was observed, followed by a brief statement of what I think the actual requirement is, and finish with the discussion points, where we can possibly go from here, the amount of effort needed, and the next steps needed to move forward.

Action items:

```
meet next week Friday - Marc will need to take Jim's place (requires reading and commenting)
    complete the initial investigations next week if possible and accept or reject feature descriptions (Chris
 and Marc)
    figure out how to get the end of July items done on the 8th or 9th (include the four of us)
```

Issue 4031 (part 1):
Statement - The files coming out of art are too big and as a result there is an increased risk of loosing data or having corrupt data when DAQ system problems occur and files are not closed cleanly. This happens because there is no file rollover rule when size limits are exceeded and art requires a clean file close for the file to be useful.
Requirement - Need a low overhead procedure for closing out files and starting we ones. Low overhead means short time (<100ms?), and no data readout stopping (>100ms of buffering).
Notes - They can have very very long runs. Pause/resume causes subrun changes, and therefore can be used, but the the overhead is too high and

too many things are affected by this action.
Right now a subrun change starts a new set of files, this can be directly initiated by the event builder through the event store interface. A solution is to put a watchdog task into place in the event builder that periodically issues what we would call a “periodic checkpoint subrun”. If configured to run every 1/2 hour (wall clock time), then new files would be started at this frequency.
Since this checkout subrun can be initiated at any time, timeout and long transition times can trigger it in anticipation of a failure.
If this is accepted, then it moves the issue into artdaq, or perhaps ds50daq.
Question - what is the time and space overhead of a subrun change instigated from the event builder?
Due date - end of July
Effort - one week (but not really an art issue as specified above)

Issue 4031(part 2):
Statement - The output files from art are badly named when there is a file change. In addition, the naming scheme is not helpful when trying to integrate higher level systems that move completed files out of the system.
Requirement - Want control over the file naming from the parameter sets and this includes, through file system inspection, one to know if a file is being written to, completed successfully, or completed with exceptions or errors.
Notes - They want to specify a prefix for file names as a pattern, and have the system use metadata to complete the pattern. The format of the file name would be <prefix>_<status>.root. Where <prefix> is supplied in the parameter set file and <status> is populated and managed by the framework. The standard <prefix> would be “$RUN_$SUBRUN_$TIMESTAMP”. The framework supplied <status> would take on one of the value “active”, “complete”, “exception”. The framework would move the file to a new name with a new status each time it changed state. The “file close” callback should be able to handle the rename.
Effort - unknown, need short investigate of a day or so. The outcome will be whether or not we can or should do the user-specified file name pattern and the difficulty of doing the file rename (what information about exceptions needs to be visible).
Due Date - I have no date recorded.

Issue 3984:
Statement - If metadata is used to generate a new filename (run, subrun, timestamp), would the information put into the file be consistent with that name, or is the name generated before the metadata is available for the new run/subrun?
Requirement - if the issue regarding file naming is resolved, then the data in the file must be consistent with the given name.
Notes - Maybe the answer is in the rename step from the above file naming requirement. If all this is truly an issue, maybe we generate a temporary name for the “active” stage, and then rename using the final available metadata.
Effort required - not known. short investigation needed and a decision of how to proceed worked out
Due Date - I had none written down.

Issue 4030:
Statement - We need to reconfigure the processing modules within art without stopping the process.
Requirement - The modules that are configured with paths need to accept a new set of parameters. In addition, the should be able to disable/enable any existing module instance or path. Analyzers need to abide by the answer given in upstream filter paths; this is particularly important for prescaled paths.
Notes - This does not include the deleting of module instances or the creation and insertion of modules into paths. It also does not include reordering modules within paths, or adding or subtracting modules from paths.
The examples given as the most important changes that would be given during reconfigure are:

```
stop the file output module from writing data and then start it up again later (module disable could handle th
is)
    change a quality monitoring analyzer module integration threshold (reconfigure)
    change the prescale value for an analyzer (reconfigure, but issues)
    disable an analyzer
```

If the prescaler is the current filter module, then we have a problem, because the event selection feature that Paul was completing is not yet complete, making it more difficult to observe trigger path results from within an analyzer.
We have most of the interfaces and tools in place for this type of reconfiguration. See the redmine issue. There is a need to create a small body of code that uses the “UserInteraction” classes. The code would take in the parameter set describing the change, and then use the interface to push the change into the reconfigure member function of the module. The DS50 analyzers would need to implement “reconfigure”.
We would need to add the disable ability into the worker (one option).
The access to trigger results information is not pretty within analyzers. We can produce as example, or find a way to complete the selector mechanism.
Thoughts - having a decoupled HTTP interface for event delivery would probably have helped here, since the monitoring could be put into the more loosely-couple art processes, meaning that restart would not be an issue because it would be quick and not require DAQ restart of any kind. I guess using the MPI-2 communicator joining features could do something similar.
Due date - end of July
Effort - one week

Issue ????:
Statement - closing file under interrupts and exceptional conditions seems to leave files in a bad state i.e. not closed properly.
Requirement - All the art processes should response properly to the shutdown requests and close the files quickly and properly as a result. The artdaq pieces that send the signals must abide by the shutdown rules. Quickly means when the current event is complete. Shutdown rules are no “kill -9” and giving as adequate grace period for current event completion.
Note - this is complicated by the distributed system aspects. What is required besides the current event completion? A message from the upstream components? Do communications need to occur when a signal arrives at an upstream component? Does the network/MPI output module and service respond properly to the signaling?
Is the divide-by-zero signal still handled properly? This is an ugly thing that requires never returning to normal processing mode before death occurs.
Due date - I have no date.
Effort - investigation and testing needed before this can be determined.

Issue 3982 (in MF):

Statement - a global message facility is needed.
Requirement - Must be able to collect message facility messages in a central place in real-time. The central place is a postgres database.
Notes - There is a proposal for this documented in 3982. This will fulfill the requirement. The key difference is that the back-end database-writing script needs to write to postgres. This is a small change. Alessandro would be happy with writing directly to the postgres database from the MF library. This is a more tightly coupled solution than we are comfortable and limited the ability to make changes. It is also more difficult to program because the manipulation and writing to the database are much easier from Ruby or Python than from C++. The back-end solution also permits multiple writers without application reconfiguration, and allows for offloading the CPU load from the application to a different machine.
Lynn will need to be involved in the configuration of the syslog-ng replacement logger.
Using Qiming's NOvA viewer and analyzer would be very nice here.
Another note - be careful when producing MF messages from the board reader. The metadata fields of the message facility (application, module, function, etc.) can be used to identify instance data. The priority and category should be used for type information only.
Due date - end of summer
Effort - two weeks.

Other things that came up that are not really issues:

```
It looks like the monitoring modules of DS50 can be run in parallel and would benefit from this because the se
rial operation appears to be a bottleneck.
    Should there be an automatic analyzer module deactivator service within art, that watches for real-time vi
olations and performs there actions? (with MF notifications of course)
```

#3 - 04/21/2014 02:26 PM - Kurt Biery

Additional possible enhancements:

- "Implement something to allow end path modules to honor different pre-scaling requirements." [Chris, 21-Jul-2013]
- "Implement the selective disablement of end-path modules on run boundaries." [Chris, 21-Jul-2013] [this may be a duplicate of something that we've already captured]

#4 - 01/13/2017 04:40 PM - Eric Flumerfelt

- Category set to Needed Enhancements

- Target version deleted (576)

#5 - 01/27/2018 12:25 PM - Eric Flumerfelt

- Status changed from New to Closed

I believe that this issue is stale enough to be closed. Further *art* improvements for *artdaq* use will use their Issue Tracker.