

Operational image visualization

From the main agenda page:

- Firefly has been used in support of early LATISS operations, and has thrown up some problems; no doubt [Robert Lupton](#) can expand on those.
- What is DM's response?
- Should consider:
 - future Firefly development plans ([Gregory Dubois-Felsmann](#))
 - the report of the Image Display WG (DMTN-126, [Yusra AlSayyad](#))
 - the possibility of including Ginga and/or JS9 in the Nublado environment.
 - DM-PORTAL milestone in 2021 is to restart portal development - the portal is a little different to this perhaps (firefly blends these) - we should discuss.
 - If we restart end 2021 early in 2021 should we do a technology survey possibly with EPO. When DM-PORTAL was added no prior package for exploration was added (nor milestone)

[Gregory Dubois-Felsmann](#):

Regarding Firefly:

- Given past project decisions, we (IPAC) weren't expecting Firefly to be part of the operational environment for AuxTel. Note that IPAC was asked not to work directly on control-room support on DM funds as far back as the DM replan.
- We were not consulted for setup instructions, do not have any access to the summit systems, and, I believe due to the understandable priority order of work being done in the initial observing runs, have not been able to get any information on the operational setup that was used. We were also not able to schedule any time with the summit team for tests of new versions.
- I'm personally happy to provide support, and have been educating myself on some of the relevant code. In addition, DM provides a fraction of an FTE of funding to IPAC for maintenance. The latter time is most efficiently spent when we have concrete bug reports and/or developer access to see a misbehavior in context. The principal IPAC developer providing support is [Trey Roby](#); we have access to other people in the IPAC Firefly group but their time has to be arbitrated with other local release deadlines, etc.
- Rubin/LSST is not providing any support for new-feature development.
 - Near-term priorities at IPAC are directing most Firefly effort this year to support of archive improvements such as greater richness in the display of spectral data. 2021-era priorities, including Science Platform-flavored work, are more consistent with interactive / Python / developer-flavored image interactions.
 - With the latter in mind, as well as with the desire to use Firefly in support of SPHEREx pipeline development over the next 3 years, I'm willing to do a small amount of new-feature work in the Python interface on my own time, and am planning to work with [Robert Lupton](#) on improving `afw.display`'s UI-driven callback support to expose the greater richness of the Firefly UI.
 - If the Rubin/LSST project (whether DM, commissioning, or operations) were to decide to commission specific work that would require additional IPAC staff time, it would take some time to free up developer cycles from other current tasks (e.g., one of the key developers of recent years currently does not work on Firefly at all). It would be significantly easier to accommodate this on a fiscal-year boundary.
- The AuxTel team set up a Firefly server on the summit, behind a VPN for which we do not have credentials, not using the configuration normally used within the LSP. We do not know how it was set up, nor what physical resources are available to it. The AuxTel team were using it from an instance of JupyterLab which was brought up ad-hoc, not using Nublado tooling.
- Firefly was used by the AuxTel team in at least two ways:
 - Interactive image display in the "usual" way, driven from `afw.display` calls in notebooks; and
 - Establishment of an automated quasi-real-time new-image display, where `afw.display` calls were made in a loop, sending partially-processed images to the summit Firefly server for display. This is a mode I would like to support but has not been tested offline (see above). By its nature it is sensitive to the accumulation of leaks, orphaned resources, cached image file space, etc.; these were impossible for us to monitor because of the lack of access.
- A variety of issues were raised, notably:
 1. Initial attempts to use Firefly from Python outside Nublado were complicated by the team members passing around recipes that contained inappropriate or outdated content, in order to work around the absence of behind-the-scenes "it just works" behavior in the Nublado environment. I believe these have largely been addressed once we were consulted on the appropriate recipes. Most notably: an LSST-requested feature allowing multiple users to view the same display from different browsers was accidentally being used all the time because of an inappropriate recipe that was propagated. Better documentation would still help; if non-Nublado interaction will continue we should write a short "getting started" guide.
 2. Some instability of the Firefly server was seen, with, for instance, several restarts being needed on at least one night. We suspect that this may be due to it being given inadequate resources compared to the load being placed on it, particularly by the quasi-real-time image monitor. We made a number of attempts to ask for additional information and/or access, without success.
 3. Some outright bugs in the visualization and UI were noticed. Two of these have been fixed and are awaiting release (IPAC tickets [FIREF LY-476 "Recenter-image-to-selected-area function does not function as expected"](#), and [FIREFLY-477 "Flux readout should not display 'NaN' when mouse pointer is outside the image frame"](#)).
 4. Some familiar UI requests reappeared, especially regarding improving the UX for zooming on a non-centered point (a "zoom to selected area" function is available in the UI and partially addresses the need, but we agree that something like a keyboard-based "zoom at location of mouse" function would be useful and relatively easy to implement), and regarding the ability to change the stretch rapidly (this is architecturally harder but it would at least be useful to define a precise request).
 - a. As has been said before, interaction-latency concerns that are associated with the client-server model of Firefly are somewhat "baked-in". Some are more easily addressed than others, and over time we have been migrating more and more computation to the browser, so there is a path to improvement, but this is certainly new-development-type effort.
 5. For the quasi-real-time monitor, [Merlin Fisher-Levine](#) wanted the ability to apply labels over an image display, to display data acquisition metadata. We discovered that neither the `afw.display` support for labels nor the DS9-based "text region" behavior that Firefly supports (and which `afw.display` invokes) are really suitable for this (because of a combination of issues related to text alignment -

only center-alignment is supported - and scaling to browser windows of different sizes). We are willing to consider some work in this area as below new-feature threshold. I have at least one concrete alternative solution to offer Merlin that should work with current Firefly capabilities.

6. [Robert Lupton](#) had difficulties getting interactive callbacks from UI actions working, and believes that this is a non-negotiable condition for proceeding further with Firefly. This turned out to be in part due to poor documentation, which I consider obviously maintenance work and will improve shortly. I was able to verify that the lower-level Python callback model in `firefly_client` does work, as previously demonstrated, both in standalone and Nublado modes, but we are still looking at some anomalous behavior Robert sees.
 - a. The integration of Firefly's interaction UI model with `afw.display`'s was never completed. Firefly has a much greater richness of potential interaction compared to DS9; Robert and I are thinking about how to extend `afw.display` to make it easier to integrate the two. Again, I consider completing the Firefly part of this work as clearly maintenance and will do it in conjunction with [Trey Roby](#) and Robert.
7. I'm tracking a variety of issues on this page: [Firefly and Portal issues arising in Commissioning](#)
 - I expect to complete relevant documentation work, provide an alternative labeling model, and sort out the callbacks in time for the next observing run. I believe this should be done regardless of what decision is made on how to proceed.
 - Work that is currently being done on moving AuxTel operations to Gen3, and on producing ObsCore data from Gen3, will integrate well with existing Firefly capabilities for the display of collections of image data based on ObsCore metadata. This would allow putting up a display of, say, "data taken last night" in Firefly, driven from a notebook, with minimal development effort. It does not require running an ObsTAP service (though that would also work).
 - We cannot help with operational stability without being involved with the deployment and having access to the systems and logs. It would be very helpful indeed if we could schedule a time to work on this when the AuxTel team is not under pressure to actually observe.