SuperTask Conversion Kick-Off Meeting Notes, 2018-06-13

SuperTask Introduction

- Steps involved in launching/running SuperTasks
- SuperTask API walkthrough (https://github.com/lsst/pipe_supertask/blob/master/python/lsst/pipe/supertask/superTask.py)

Potential Prerequisites

- ⚠ RFC-352 - Jira project doesn't exist or you don't have permission to view it.
- Command-line driver for single-SuperTask execution (single Quantum? multiple Quantum?)
  - single-SuperTask QuantumGraph generation (placeholder for preflight)
  - single-node QuantumGraph executor (extract from pipe.supertask.CmdLineFwk, polish)
- ci_hsc refactoring?
- LoadReferenceObjects refactoring
- Exposure ID handling (in Gen3 Butler, in Tasks that make SourceCatalog IdFactory)
- Call butler.get/put on DatasetRefs directly
- Need to polish passing (output dataset only?) DataUnits to run() from runQuantum()
- Syntactic sugar for ConfigFields of DatasetConfigs
- Make "catalog.schema" datasets writeable with no units even when "catalog" does have units.

Proposed Conversion Pattern

- Move all regular Task logic into a mix-in class with a `run` method that conforms to RFC-352.
- New concrete CmdLineTask subclasses CmdLineTask base class and mix-in, implementing only runDataRef and CmdLineTask introspection methods.
- New concrete SuperTask subclasses SuperTask base class and mix-in, implementing only runQuantum and SuperTask introspection methods.
- Can use gen3-middleware branch (see Gen3 Middleware Software Stack) for now, but need to get `daf_butler` and `pipe_supertask` in lsst-distrib before conversion begins in earnest.

Other Considerations

- Gen3 team will also need help with obs_* related work (YAML Camera persistence, afw::table::io Detector persistence, obs_* rewrites), and some of you may be needed more for that.