

Migrating ASP to S3DF

Deadlines

- **November 1, 2023:** rhel6-64 logins disabled.
- **March 31, 2024:** LSF license expires.

Resources

- Migration scripts, etc., are maintained in the [asp_migration](#) github repo.
- Locations of copies of pre-built Fermi software releases at S3DF: `/sdf/data/fermi/a/ground/releases`
- [s3df-migration slack channel](#) in the Fermi-LAT workspace.
- [Get Started on S3DF - Cheat Sheet](#)
- [Migrating L1Proc to S3DF](#)
- [ASP Installation Instructions](#)
- [Using RHEL6 Singularity Container](#)
- Brian's instructions for converting scriptlets to batch jobs (NB: the code block below has been corrected, but not checked. Here's the link to [bvan's original slack message](#)):

1. Convert your scriptlet to a job, include requisite batchOptions
2. Include `#!/usr/bin/env python3` at the top of your file so it's executable with python3 (on the batch nodes). Ensure your code is compatible with python3. You may need to wrap numeric variables with `int` or `float` where necessary.
3. Emulate locals that are already set. Add a `pipeline` object that's added to the scriptlet. Only two methods are supported, `setVariable`, `createStream`. Include shell variables as python variables by updating `locals()`. For the most part, you should be able to just add this to the top of your scriptlet definition:

```
#!/usr/bin/env python3
import os
class Pipeline:
    def setVariable(self, key, value):
        with open(os.environ["PIPELINE_SUMMARY"], "a") as f:
            f.write("Pipeline.%s: %s\n" % (key, value))
    def createStream(self, task, stream, args):
        with open(os.environ["PIPELINE_SUMMARY"], "a") as f:
            f.write("PipelineCreateStream.%s.%d: %s\n" % (task, int(stream), args))
pipeline = Pipeline()
locals().update({k:v for k,v in os.environ.items() if "." not in k})
```

To Do List

- Update ASP CVS repository with most recent modifications. ✓
- Convert the [ASP](#) and [ASP-scons](#) CVS areas to git repos. ✓
 - Move these repos to the [fermi-lat github org](#).
- Write up [current installation instructions](#) on Confluence. ✓
- S3DF installation of GPLtools: `/sdf/data/fermi/a/ground/PipelineConfig/GPLtools/prod`. Modifications are needed. See below. ✓
- Singularity container (generated by Wei) to use with `/sdf/data/fermi/a/ground/releases/volume02/ScienceTools-09-32-05/` `/sdf/group/fermi/sw/containers/fermi-rhel6.20230922.sif`. This contains the 64-bit BLAS and LAPACK libraries needed by numpy. ✓
- Example ASP P-II task on S3DF using legacy Fermi software releases, including Oracle db queries: [ASP_db_query_test](#) ✓
- Example ASP P-II task that uses the dataCatalog and GPLtools staging code: [ASP_datacat_test](#) ✓
 - Modify [GPLtools](#) to use paths to xrootd locations on S3DF ✓
- Prototype P-II task to demo registering outputs to datacatalog ✓
- Prototype P-II task that creates substreams ✓
 - NB: Creating substreams seem to need to be done from a python3 script included as a `<job>` via a CDATA section replacing an explicit executable attribute. Would be good to be able to import custom python code in that CDATA section. ⚠
 - Supposedly, using the `pipeline` object within scriptlets works now (We'll see...) ⚠
- Code to replace scriptlets in order to run datacat and pipeline commands in batch jobs ✓
 - This can be done via code like [this](#).
- Ensure schema locations in XML pipeline defs use "https":

```
<pipeline xmlns="http://glast-ground.slac.stanford.edu/pipeline"
  xmlns:xs="http://www.w3.org/2001/XMLSchema-instance"
  xs:schemaLocation="http://glast-ground.slac.stanford.edu/pipeline https://glast-ground.slac.stanford.edu/Pipeline-II/schemas/2.2/pipeline.xsd">
```

- File paths used by ASP that need S3DF equivalents:
 - `/afs/slac/g/glast/ground/links/data/ASP/GCN_Archive` (`GCN_Notice_processor.py`)
 - `/afs/slac/g/glast/ground/links/data/ASP/scratch` (`makeDrpLcTables.py`)
 - `/afs/slac/g/glast/isoc/lightOps/rhel5_gcc41/ISOC_PROD/bin/isoc` (`makeDrpLcTables_part2.py`, `fastCopy.py` for `FastCopy`)
 - `/afs/slac/g/glast/ground/links/data/ASP/Results` (`aspLauncher.py`)

- /afs/slac/g/glast/ground/ASP/catalogs (launchStreams.py)
 - /afs/slac/g/glast/ground (DbEntry.py)
- Building ASP against ST-00-32-05 in container. ST-10-01-01 is available at /sdf/data/fermi/a/ground/releases/volume09/ScienceTools-10-01-01/
 - ✓
 - It would be good to be able to build against ST-10-01-01 since the bash wrapper scripts generated by our SConstruct files unfortunately write explicit paths rather than using env vars so that all of the wrapper scripts need to be updated to use the SDF3 paths. However, the SCons build code for ST is extremely obscure, and it's not at all clear how the value provided via the `--with-GLAST-EXT` command line option is passed (or if it is in fact passed at all) to the generated bash wrappers.
- Sort out batch options to use in P-II task xml definitions, e.g., `--account fermi`.
- S3DF installation of Oracle drivers. On rhel6-64/centos7, the required distribution is in /afs/slac.stanford.edu/package/oracle/11.1.0/amd64_linux26/11.1.0. An rsync'd copy on /sdf seems to work. ✓
- Replace trscron job for sending Monitored Source light curve data to GSFC.
- Update ASP P-II tasks to run on s3df, using dev db tables.
 - Update paths to pipeline commands, etc.
- Remove all GRB-related executables from P-II tasks.

Won't Do List

- Python 3 conversion of ASP code
- Updating ASP to use modern FermiTools distributions
- Rebuilding, i.e., with SCons, the ASP distribution no new ASP release tags.

Non-Pipeline-II Services

- procmail-triggered processing of incoming GCN Notices
- [cron job](#) to ingest GCN notices into ASP db tables
- trscron job to prepare Monitored Source List light curves and sent them to GSFC using FastCopy
 - Executable: /afs/slac/g/glast/ground/links/data/ASP/makeDrpLcTables.sh