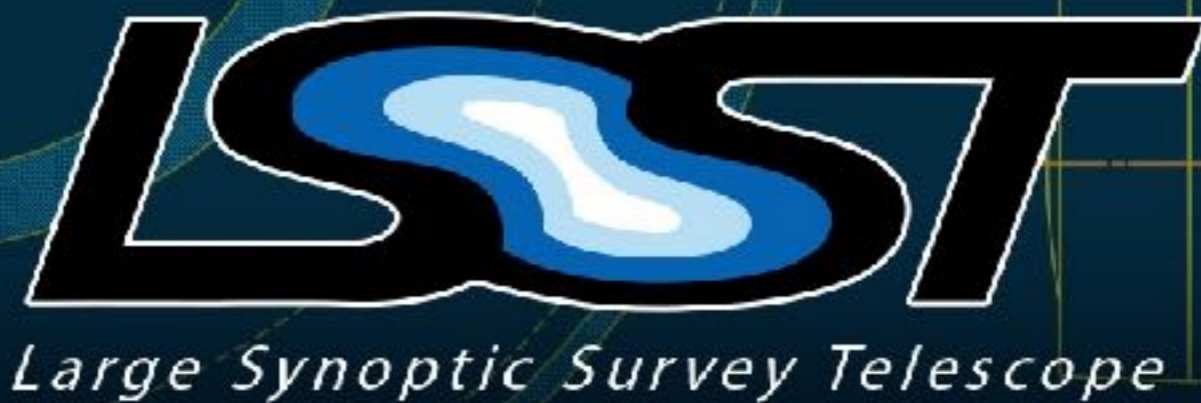


SQuaRE's Documentation Efforts

A (super) brief overview

Frossie Economou • frossie@lsst.org



The “User Guide”

SQuaRE inherited the “User Guide” and the “Developer Guide” from R&D, a set of pages in Confluence. This was essentially pipelines documentation. My main concerns were:

- Docs were “far from the code”, became quickly obsolete
- Was distinct from the codebase tooling (git), making release management inefficient (eg can't branch docs along with code)
- Was not parsable so we could not eg. CI examples for accuracy
- There was a gap between low-level code documentation and the user guide that presented a barrier to users of the pipelines so we needed to incentivize creating that content
- Lots of useful snippets put in Confluence and then forgotten - wikis are notoriously poor platform for software documentation
- Primary content also exists in (now defunct) TRAC

Square's plan



What was the roadmap

1. Hire someone who can refresh the toolchain, develop content and pivot into community management -> jsick ✓
2. Get software docs off Confluence into git and put a publishing layer in front -> pipelines.lsst.io, developer.lsst.io ✓
3. Create git-based workflow for "useful snippets" -> technotes ✓
4. Create publication platform (can't use OtS) -> LTD ✓
5. Create templates for in-code documentation that guide developers to capturing the "what does this do and why" layer (cf. supertask) -> "API docs", *in progress - next 2 slides*
6. Create curated landing page / indexer -> DocHub
7. Set up CI of examples
8. Move to higher levels of content (eg. video tutorials) and community outreach

pipelines.lsst.io homepage



processing type

LSST Science Pipelines

Version DM-6199

On this page
Getting Started
Processing data
Frameworks
Python modules

Cite the Science Pipelines

Discuss community.lsst.org

More LSST docs lsst.io

Multi-epoch processing

In depth

Tutorials

Command line tasks

module type

LSST Science Pipelines

Version DM-6199

On this page
In depth
Tasks
Python API reference
C++ API reference
Packaging

Cite this page

Discuss community.lsst.org

More LSST docs lsst.io

Isst.afw.table – Table data structures

See also

In depth

Tasks

Python API reference

C++ API reference

Packaging

Isst.afw.table is part of the afw EUPS package. Fork on GitHub.

Top-level packages

Isst.afw.table is included when you install these:

- lsst_apps
- lsst_distrib
- lsst_obs
- lsst_ci

Dependencies

Isst.afw.table is used by these packages:

Users

Isst.afw.table is used by these packages:

more details:
dmtn-030.lsst.io

task type

LSST Science Pipelines

Version DM-6199

On this page
Configuration
Entrypoint
Butler inputs
Butler outputs
Examples
Debugging variables
Algorithm

Cite this page

Discuss community.lsst.org

More LSST docs lsst.io

ProcessCcdTask

ProcessCcdTask is part of the lsst.pipe.tasks module.

See also

Configuration

Entrypoint

Butler inputs

Butler outputs

Examples

Debugging variables

Algorithm

framework type

LSST Science Pipelines

Version DM-6199

On this page
In depth
Tutorials
Modules

Cite this page

Discuss community.lsst.org

More LSST docs lsst.io

Butler framework

In depth

Tutorials

Modules

API type (auto)

Python modules

- lsst.afw
- lsst.afw.dtypes
- lsst.afw.geom
- lsst.afw.image
- lsst.afw.math
- lsst.afw.mask
- lsst.afw.table
- lsst.afw.util
- lsst.afw.versions
- lsst.afw.wcs
- lsst.afw.wcs2
- lsst.afw.wcs3
- lsst.afw.wcs4
- lsst.afw.wcs5
- lsst.afw.wcs6
- lsst.afw.wcs7
- lsst.afw.wcs8
- lsst.afw.wcs9
- lsst.afw.wcs10
- lsst.afw.wcs11
- lsst.afw.wcs12
- lsst.afw.wcs13
- lsst.afw.wcs14
- lsst.afw.wcs15
- lsst.afw.wcs16
- lsst.afw.wcs17
- lsst.afw.wcs18
- lsst.afw.wcs19
- lsst.afw.wcs20
- lsst.afw.wcs21
- lsst.afw.wcs22
- lsst.afw.wcs23
- lsst.afw.wcs24
- lsst.afw.wcs25
- lsst.afw.wcs26
- lsst.afw.wcs27
- lsst.afw.wcs28
- lsst.afw.wcs29
- lsst.afw.wcs30
- lsst.afw.wcs31
- lsst.afw.wcs32
- lsst.afw.wcs33
- lsst.afw.wcs34
- lsst.afw.wcs35
- lsst.afw.wcs36
- lsst.afw.wcs37
- lsst.afw.wcs38
- lsst.afw.wcs39
- lsst.afw.wcs40
- lsst.afw.wcs41
- lsst.afw.wcs42
- lsst.afw.wcs43
- lsst.afw.wcs44
- lsst.afw.wcs45
- lsst.afw.wcs46
- lsst.afw.wcs47
- lsst.afw.wcs48
- lsst.afw.wcs49
- lsst.afw.wcs50
- lsst.afw.wcs51
- lsst.afw.wcs52
- lsst.afw.wcs53
- lsst.afw.wcs54
- lsst.afw.wcs55
- lsst.afw.wcs56
- lsst.afw.wcs57
- lsst.afw.wcs58
- lsst.afw.wcs59
- lsst.afw.wcs60
- lsst.afw.wcs61
- lsst.afw.wcs62
- lsst.afw.wcs63
- lsst.afw.wcs64
- lsst.afw.wcs65
- lsst.afw.wcs66
- lsst.afw.wcs67
- lsst.afw.wcs68
- lsst.afw.wcs69
- lsst.afw.wcs70
- lsst.afw.wcs71
- lsst.afw.wcs72
- lsst.afw.wcs73
- lsst.afw.wcs74
- lsst.afw.wcs75
- lsst.afw.wcs76
- lsst.afw.wcs77
- lsst.afw.wcs78
- lsst.afw.wcs79
- lsst.afw.wcs80
- lsst.afw.wcs81
- lsst.afw.wcs82
- lsst.afw.wcs83
- lsst.afw.wcs84
- lsst.afw.wcs85
- lsst.afw.wcs86
- lsst.afw.wcs87
- lsst.afw.wcs88
- lsst.afw.wcs89
- lsst.afw.wcs90
- lsst.afw.wcs91
- lsst.afw.wcs92
- lsst.afw.wcs93
- lsst.afw.wcs94
- lsst.afw.wcs95
- lsst.afw.wcs96
- lsst.afw.wcs97
- lsst.afw.wcs98
- lsst.afw.wcs99
- lsst.afw.wcs100

pipelines.lsst.io implementation

General sequence and priorities

Documentation engineering (SQuaRE)

1. Build system (integrate with Scons, EUPS, Jenkins, LSST the Docs).
2. Custom reStructuredText directives/roles for auto doc generation.
3. C++ & wrapped Python API reference (collab. with Pybind11 experts).
4. Tutorial/example authoring, run, and testing infrastructure.
5. Citeability (DOIs).
6. Web design & UX refinement.

Content (SQUARE and Science PIPELINES)

1. Author instructions and topic type templates.
2. Python docstrings: semi-automated (?) conversion and improvement.
3. Write pre-mapped topics within the type system:
 - Low-level first: module and task-topics.
 - High-level later: framework and processing topics.
4. Add tutorials and more examples when CI infrastructure is ready.

Meanwhile...



Victims of our own success...

- Publication Board asked for DOIs -> Zenodo ✓
- Project docs needed refreshing -> Did markup transfers of some of the docs on request (eg Word->rST for Idm-135) ✓ but 😡
- Produced a DM documentation and communication policy -> ✓ but 😡
- Nobody wants to give up their preferred platform -> now need complex indexer -> DocHub
- Everybody hates Docushare but it is the project standard and it's lying fallow -> not SQuaRE's remit
- Nobody is keeping key project docs up to date -> not SQuaRE's remit

These are all good and important things **but they are a distraction from the software/user documentation roadmap and have delayed us.**

References



For more information

- [sqr-000: technote platform](#)
- [sqr-006: LSST-the-Docs](#)
- [sqr-011: Communication platforms](#)
- [sqr-013: Dochub design](#)
- [Idm-493: Documentation architecture](#)
- [dmtn-030: Pipeline doc design](#)