
PFS Target Database (targetDB)

Release 2021

Masato Onodera

Apr 26, 2022

CONTENTS:

1	Input Target List	1
1.1	Inputs from observers	1
1.2	Inputs from the observatory	3
1.3	File format	4
1.4	Notes on future development	5
2	Indices and tables	7

INPUT TARGET LIST

1.1 Inputs from observers

1.1.1 Proposal-related information

The following information is required for each *proposal*.

name	datatype	description	required	default
proposal_id	str	Proposal ID (e.g., S22A-QN001)	*	

1.1.2 Target-related information

The following information is required for each *target*.

name	datatype	description	required
obj_id	int64	Object ID	*
ra	float	RA (ICRS, degree)	*
dec	float	Dec (ICRS, degree)	*
epoch	str	Epoch (e.g., J2000.0, J2016.0)	
tract	int	Same definition as HSC-SSP	
patch	int	Same definition as HSC-SSP	
input_catalog	str	Input catalog name (e.g., hscssp_pdr3_wide)	
fiber_mag_g	float	g-band magnitude within a fiber (AB mag)	
fiber_mag_r	float	r-band magnitude within a fiber (AB mag)	
fiber_mag_i	float	i-band magnitude within a fiber (AB mag)	
fiber_mag_z	float	z-band magnitude within a fiber (AB mag)	
fiber_mag_y	float	y-band magnitude within a fiber (AB mag)	
fiber_mag_j	float	J band magnitude within a fiber (AB mag)	
psf_mag_g	float	g-band PSF magnitude (AB mag)	
psf_mag_r	float	r-band PSF magnitude (AB mag)	
psf_mag_i	float	i-band PSF magnitude (AB mag)	
psf_mag_z	float	z-band PSF magnitude (AB mag)	
psf_mag_y	float	y-band PSF magnitude (AB mag)	
psf_mag_j	float	J band PSF magnitude (AB mag)	
psf_flux_g	float	g-band PSF flux (nJy)	
psf_flux_r	float	r-band PSF flux (nJy)	

continues on next page

Table 1 – continued from previous page

name	datatype	description	required
psf_flux_i	float	i-band PSF flux (nJy)	
psf_flux_z	float	z-band PSF flux (nJy)	
psf_flux_y	float	y-band PSF flux (nJy)	
psf_flux_j	float	J band PSF flux (nJy)	
priority	float	Priority defined by the observer within the proposal	*
effective_exptime	float	Requested effective exposure time (s)	*
is_medium_resolution	bool	True if the medium resolution mode is requested	
qa_relative_throughput	float	Relative throughput to the reference value requested by the observer	
qa_relative_noise	float	Relative noise to the reference value requested by the observer	
qa_reference_lambda	float	Reference wavelength to evaluate effective exposure time (angstrom or nm?)	

Notes

tract, patch

If the look-up `tract` and `patch` from the coordinates is not expensive, it is possible for the observatory to automatically fill these information without asking inputs from observers.

input_catalog

Currently, the following catalogs are considered, and the list can be easily expanded.

```
input_catalog_id,input_catalog_name,input_catalog_description
0,"simulated","simulated catalog"
1,"gaia_dr1","Gaia Data Release 1"
2,"gaia_dr2","Gaia Data Release 2"
3,"gaia_edr3","Gaia Early Data Release 3"
4,"gaia_dr3","Gaia Data Release 3"
5,"hscssp_pdr1_wide","HSC-SSP Public Data Release 1 (Wide)"
6,"hscssp_pdr1_dud","HSC-SSP Public Data Release 1 (Deep+UltraDeep)"
7,"hscssp_pdr2_wide","HSC-SSP Public Data Release 2 (Wide)"
8,"hscssp_pdr2_dud","HSC-SSP Public Data Release 2 (Deep+UltraDeep)"
9,"hscssp_pdr3_wide","HSC-SSP Public Data Release 3 (Wide)"
10,"hscssp_pdr3_dud","HSC-SSP Public Data Release 3 (Deep+UltraDeep)"
11,"hscssp_pdr4_wide","HSC-SSP Public Data Release 4 (Wide)"
12,"hscssp_pdr4_dud","HSC-SSP Public Data Release 4 (Deep+UltraDeep)"
```

For individual proposals, either assigning a new `input_catalog_id` (e.g., s22a-qn0001_00001 with `input_catalog_id=10001`) or allow them to use pre-assigned `input_catalog` should work.

1.2 Inputs from the observatory

In the background, the observatory needs to populate the rest of tables such as `proposal_category`, `proposal`, `target_type`, `input_catalog`, and `fluxstd`.

1.2.1 `proposal_category`

Currently, `proposal_category` contains the following information.

```
proposal_category_id,proposal_category_name,proposal_category_description
1,"openuse","Subaru openuse proposal"
2,"keck","Subaru/Keck time exchange proposal"
3,"gemini","Subaru/Gemini time exchange proposal"
4,"uh","University of Hawaii proposal"
```

1.2.2 `target_type`

Currently, `target_type` contains the following entries as defined by the `datamodel`.

```
target_type_id,target_type_name,target_type_description
1,"SCIENCE","the fiber is intended to be on a science target"
2,"SKY","the fiber is intended to be on blank sky, and used for sky subtraction"
3,"FLUXSTD","the fiber is intended to be on a flux standard, and used for flux_
↪calibration"
4,"UNASSIGNED","the fiber is not targeted on anything in particular"
5,"ENGINEERING","the fiber is an engineering fiber"
6,"SUNSS_IMAGING","the fiber goes to the SuNSS imaging leg"
7,"SUNSS_DIFFUSE","the fiber goes to the SuNSS diffuse leg"
```

1.2.3 `proposal`

The `proposal` table's schema is the following.

name	type	primary_key	autoincrement	comment
proposal_id	VAR-CHAR	True	False	Unique identifier for proposal (e.g, S21B-OT06?)
group_id	VAR-CHAR	False	False	Group ID in STARS (e.g., o21195?)
pi_first_name	VAR-CHAR	False	False	PI's first name
pi_last_name	VAR-CHAR	False	False	PI's last name
pi_middle_name	VAR-CHAR	False	False	PI's middle name
rank	FLOAT	False	False	TAC score
grade	VAR-CHAR	False	False	TAC grade (A/B/C/F in the case of HSC queue)
allocated_time	FLOAT	False	False	Total fiberhours allocated by TAC (hour)
proposal_category_id	INTEGER	False	False	
created_at	DATE-TIME	False	False	Creation time [YYYY-MM-DDThh:mm:ss] (UTC)
updated_at	DATE-TIME	False	False	Update time [YYYY-MM-DDThh:mm:ss] (UTC)

1.2.4 Other tables

There are more tables which are still under development such as `sky` and `cluster`.

1.3 File format

As a target list contains proposal-specific and target-specific information, a file format which can handle metadata would be preferable.

A couple of candidates can be recommended.

1. FITS binary table
2. ECSV (Enhanced CSV)

Both can be easily prepared with `Astropy`.

1.3.1 Example

Prepare a list for targets.

```
from astropy.table import Table
tb = Table([[42687868933508256, 42687868933508552],
            [351.32788192091937, 351.2784400775651],
            [0.713249785705446, 0.7191567004328004],
            ["J2000.0", "J2000.0"],
            [9706, 9706],
            [304, 304],
```

(continues on next page)

(continued from previous page)

```

["hscssp_pdr3_dud", "hscssp_pdr3_dud"],
[1.0, 1.0],
[900.0, 900.0]],
names=["obj_id",
        "ra",
        "dec",
        "epoch",
        "tract",
        "patch",
        "input_catalog",
        "priority",
        "effective_exptime"],
meta={"proposal_id": "S22A-QN001"})

```

If you have a similar list as `pandas.DataFrame`, the following should work.

```

tb = Table.from_pandas(df)
tb.meta['proposal_id'] = "S22A-QN001"

```

You can save the object into a file.

```

tb.write('targets_s22a-qn001.fits', format='fits')
tb.write('targets_s22a-qn001.ecsv', format='ascii.ecsv')

```

Reading the data is easy.

```

tb2 = Table.read('targets_s22a-qn001.fits')
tb2 = Table.read('targets_s22a-qn001.ecsv')

print(tb2.meta["proposal_id"])
print(tb2)

```

1.4 Notes on future development

- Currently, only creating entries in `targetDB` is tested. Updating and removing them need to be implemented in the future.
- Also, the script/function to create entries is still as of the 2021 November commissioning. This will be modified accordingly.

INDICES AND TABLES

- `genindex`
- `modindex`
- `search`