
bossdata Documentation

Release 0.3.0

David Kirkby

Dec 19, 2017

Contents

1	Installation	3
1.1	Requirements	3
1.2	Optional Dependencies	4
1.3	Quick Demonstration	4
2	Overview of SDSS Spectroscopic Data	5
3	Executable scripts	7
3.1	bossquery	7
3.2	bossfetch	8
3.3	bossplot	9
3.4	bossraw	10
4	Configuration	13
4.1	SEQUELS Data	14
4.2	SDSS-I/II Spectra	14
4.3	eBOSS Proprietary Data	14
5	API Usage	15
6	Examples	17
7	Using bossdata at NERSC	19
8	Contributing	21
8.1	Types of Contributions	21
8.2	Get Started!	22
8.3	Pull Request Guidelines	23
8.4	Version Update Checklist	23
8.5	New External Dependency Checklist	24
9	History	25
9.1	0.3.1 (unreleased)	25
9.2	0.3.0 (2017-12-17)	25
9.3	0.2.8 (2015-10-19)	25
9.4	0.2.7 (2015-09-28)	25
9.5	0.2.6 (2015-08-05)	26
9.6	0.2.5 (2015-07-06)	26

9.7	0.2.4 (2015-06-29)	26
9.8	0.2.3 (2015-06-22)	26
9.9	0.2.2 (2015-06-15)	27
9.10	0.2.1 (2015-06-13)	27
9.11	0.2.0 (2015-06-09)	27
9.12	0.1.0 (2015-05-24)	27
10	Full desisurvey API Reference	29
	Python Module Index	99

A python package for working with [spectroscopic data](#) from the [Sloan Digital Sky Survey](#). Bossdata is free software (MIT license) [hosted on github](#) and released via [pypi](#).

CHAPTER 1

Installation

To install, use the command line:

```
% pip install bosssdata
```

To upgrade to the latest version:

```
% pip install bosssdata --upgrade
```

1.1 Requirements

The following additional packages are used by bosssdata and will be installed automatically by pip, if necessary:

- requests
- progressbar
- astropy
- fitsio
- numpy
- pydl

1.1.1 Numpy Performance Issue

Note that some operations run much slower (but still correctly) with numpy versions 1.10.0 and 1.10.1 so these should be avoided if possible. See [here](#) for details. To determine which version of numpy you are using:

```
import numpy
print numpy.version.version
```

The best solution is to use version 10.0.2 or later. If this is not possible, revert to numpy 1.9.3 and astropy 1.0.4. For example, with conda:

```
conda install numpy=1.9.3
conda install astropy=1.0.4
```

1.2 Optional Dependencies

The following packages are optional and enable additional functionality. They will not be automatically installed by pip, but will be used when available.

- matplotlib (used by the bossdata.plot module and bossplot script)

1.3 Quick Demonstration

If you have `matplotlib` installed, you can quickly test that everything is working with:

```
bossplot
```

This should download a small data file for a single spectrum and plot the data in a window. Close the plot window to exit. For more information on `bossplot` and other available command-line scripts, see [Executable scripts](#).

Overview of SDSS Spectroscopic Data

This package is primarily intended for working with data from the [SDSS-III BOSS](#) survey, but can also be used to access older data from SDSS-I/II and newer data from the [SEQUELS](#) ancillary program and the [SDSS-IV eBOSS](#) survey (see [Configuration](#) for details).

BOSS data consists of [spectroscopic observations](#) of astrophysical [targets](#). An observation is identified by a triplet of numbers (PLATE,MJD,FIBER). Most BOSS targets only have a single observation. Each observation consists of several 15-minute exposures using red and blue cameras with overlapping wavelength coverage that are combined to give a single co-added spectrum.

The table below summarizes the different files produced by the [spectroscopic pipeline](#) containing the individual and combined exposures contributing to each observation. Files contain from 1 to 1000 spectra, with some duplication between files. Each file provides wavelength, flux, inverse variance, [mask bits](#) and subtracted sky for each of its spectra.

Type	Size	#Tgts	#Exp	Coadd?	Calib?	Datamodel	Bosssdata Class
lite	0.2Mb	1	0	Y	Y	lite	bosssdata.spec.SpecFile
spec	1.7Mb	1	ALL	Y	Y	spec	bosssdata.spec.SpecFile
plate	110Mb	1000	0	Y	Y	plate	bosssdata.plate.PlateFile
cframe	70Mb	500	1	N	Y	cframe	bosssdata.plate.FrameFile
frame	30Mb	500	1	N	N	frame	bosssdata.plate.FrameFile

The following examples show how the same combined spectrum can be [plotted](#) from lite files and plate files:

```
bossplot --plate 6641 --mjd 56383 --fiber 30
bossplot --plate 6641 --mjd 56383 --fiber 30 --platefile
```

Individual exposures can also be plotted using either spec files, cframe files or frame files:

```
bossplot --plate 6641 --mjd 56383 --fiber 30 --exposure 0
bossplot --plate 6641 --mjd 56383 --fiber 30 --exposure 2 --cframe
bossplot --plate 6641 --mjd 56383 --fiber 30 --exposure 2 --frame
```

Note that the indexing of exposures is different for spec files, which only index exposures used in the final coadd, and (c)frame files which index all available exposures. The indices used in the example all refer to exposure number

00158842, which can be verified by adding the `--verbose` option to these commands. The difference between the `cframe` and `frame` files is that the `frame` gives fluxes in units of flat-fielded detected electrons, before the step of calibrating fluxes using standard stars.

The following per-exposure calibration data products can also be accessed using the `ftype` parameter to `bossdata.plate.Plan.get_exposure_name()` and `bossdata.spec.Exposures.get_exposure_name()`. These files are in 1-1 correspondence with the `sp(C)Frame` files.

Size	Type	Datamodel	Description
5Mb	science	spFluxcalib	Flux calibration vectors derived from standard stars
1Mb	science	spFluxcorr	Flux correction vectors for a science exposure
6Mb	arc	spFlat	Results derived from an arc calibration exposure
4Mb	flat	spFlat	Results derived from an flat calibration exposure

The definitive reference for how these calibration data are created and used is the [IDL pipeline code](#).

CHAPTER 3

Executable scripts

For complete documentation on the command-line options of any script use the `--help` option, for example:

```
bossquery --help
```

You will normally want to configure `bossdata` by setting some *environment variables*.

3.1 bossquery

Query the meta data for BOSS observations. For example (the initial setup might take a few minutes if you have never used this command before):

```
bossquery --what PLATE,MJD,FIBER,PLUG_RA,PLUG_DEC,Z --where 'OBJTYPE="QSO"' --sort Z -  
↪-save qso.dat
```

The `--save` option supports *many different output formats* that are automatically selected based on the file extension. In addition, this program automatically maps the `.dat` and `.txt` extensions to the *ascii* format.

The `--what`, `--where` and `--sort` options all use SQL syntax (these are in fact substituted into a SQL string).

- `--what` takes a comma separated list of column names (like SQL SELECT) and defaults to PLATE,MJD,FIBER:

```
--what PLATE,MJD,FIBER,PLUG_RA,PLUG_DEC,Z
```

- `--where` takes a SQL 'WHERE' string:

```
--where '(OBJTYPE="QSO" and Z > 0.1) or CLASS="QSO"'
```

- `--sort` takes a list of columns with optional DESC keyword following columns to reverse their order (a la SQL ORDER BY):

```
--sort 'CLASS, Z DESC'
```

This command uses an sqlite3 database of metadata that will be created if necessary. By default, the “lite” version database will be used, which provides faster queries and a smaller database file. However, the full [spAll data model](#) is also available with the `-full` option (resulting in slower queries and a larger database file). The “lite” and “full” databases are separate files based on different downloads. Once either has been created the first time, it will be immediately available for future queries. Note that it can take a while to create the initial database file: allow about 30 minutes for either version. Once the database has been created, you can safely delete the downloaded source file if you are short on disk space.

The columns in the lite database are a subset of those in the full database but the values are not numerically identical between them because they are truncated in the text file used to generate the lite database. However, the level of these truncation errors should be insignificant for any science applications.

There are some minor inconsistencies between the data models of the lite and full versions of the meta data provided by BOSS. In particular, the lite format uses the name *FIBER* while the full version uses *FIBERID*. We resolve this by consistently using the shorter form *FIBER* in both SQL databases. Also, the full format includes columns that are themselves arrays. One of these, *MODELFLUX(5)*, is included in the lite format using names *MODELFLUX0...MODELFLUX4*. We normalize the mapping of array columns to scalar SQL columns using the syntax *COLNAME_I* for element [i] of a 1D array and *COLNAME_I_J* for element [i,j] of a 2D array, with indices starting from zero. This means, for example, that *MODELFLUX(5)* values are consistently named *MODELFLUX_0...MODELFLUX_4* in both SQL databases.

In the case where a query is made without specifying `-full` but the lite database file is not present, an attempt will be made to use the full database. If neither DB files are present the same logic is applied to the catalog files. If present, the lite catalog file will be parsed and the lite DB created; if that is not present, the full catalog file will be parsed and the full DB created. Only after exhausting these options will a download (of the lite DB) file be attempted.

Note that specifying `-full` will only (and always) use the full DB or catalog file.

The `-quasar-catalog` option can be used to query the [BOSS quasar catalog](#) instead of spAll. By default, the current version of the catalog will be used; use the `-quasar-catalog-name` option to specify an earlier version.

The `--platelist` option can be used to query the [BOSS plate list database](#) instead of spAll.

3.2 bossfetch

Fetch BOSS data files containing the spectra of specified observations and mirror them locally. For example:

```
bossfetch --verbose qso.dat
```

Fetches files will be placed under `$BOSS_LOCAL_ROOT` with paths that exactly match the URLs they are downloaded from with the prefix substitution:

```
$BOSS_DATA_URL => $BOSS_LOCAL_ROOT
```

For example, with the default configuration given above, the file at:

```
http://dr12.sdss3.org/sas/dr12/boss/spectro/redux/v5_7_0/spectra/lite/3586/spec-3586-  
↪55181-0190.fits
```

would be downloaded to:

```
$BOSS_LOCAL_ROOT/sas/dr12/boss/spectro/redux/v5_7_0/spectra/lite/3586/spec-3586-55181-  
↪0190.fits
```

By default, the “lite” format of each spectrum data file is downloaded, which is sufficient for many purposes and significantly (about 8x) smaller. The “lite” format contains HDUs 0-3 of the [full spectrum data file](#) and does not include the spectra of individual exposures. To download the full files instead, use the `--full` option. Both types

of files can co-exist in your local mirror. You can also load the plate `spFrame` or flux-calibrated `spCFrame` files using the `--frame` or `--cframe` options, respectively. These files contain a half plate of spectra for a single band (blue/red) and exposure. Finally, you can load the `spPlate` files containing combined spectra for a whole plate using the `--platefile` option. See the [Overview of SDSS Spectroscopic Data](#) for details.

The `--verbose` option displays a progress bar showing the fraction of files already locally available. Any files that were previously fetched will not be downloaded again so it is safe and efficient to run `bossfetch` for overlapping lists of observations. Note that the progress bar may appear to update unevenly if some files are already mirrored and others need to be downloaded.

Each data file download is streamed to a temporary files with `.downloading` appended to their name then renamed to remove this extension after the download completes normally. If a download is interrupted or fails for some reason, the partially downloaded file will remain in the local mirror. Re-running a `bossfetch` command will automatically re-download any partially downloaded file.

By default, downloading is split between two parallel subprocesses but you can change this with the `--nproc` option. For downloading “lite” files, using more than 2 subprocesses will probably not improve the overall performance.

If you want to transfer large amounts of files, you should consider using [globus](#). To prepare a *globus* bulk data transfer file list, use the `-globus` option to specify the remote/local endpoint pair `remote#endpoint:local#endpoint`. Note that the `-save` option must also be used to specify an output filename. SDSS endpoints are documented at [here](#).

For example, to transfer files from `lbnl#sdss3` to `local#endpoint`:

```
bossfetch qso.dat --globus lbnl#sdss3:username#endpoint --save globus-xfer.dat
ssh username@cli.globusonline.org transfer -s 1 < globus-xfer.dat
```

3.3 bossplot

Plot the spectrum of a single BOSS observation, identified by its PLATE, MJD of the observation, and the FIBER that was assigned to the target whose spectrum you want to plot. For example (these are the defaults if you omit any parameters):

```
bossplot --plate 6641 --mjd 56383 --fiber 30
```

This should open a new window containing the plot that you will need to close in order to exit the program. To also save your plot, add the `--save-plot` option with a filename that has a standard graphics format extension (`pdf`, `png`, ...). If you omit the filename, `--save-plot` uses the name `bossplot-{plate}-{mjd}-{fiber}.png`. To save plots directly without displaying them, also use the `--no-display` option.

You can also save the data shown in a plot using `--save-data` with an optional filename (the default is `bossplot-{plate}-{mjd}-{fiber}.dat`). Data is saved using the [ascii.basic](#) format and only wavelengths with valid data are included in the output.

Use `--wlen-range [MIN:MAX]` to specify a wavelength range over which to plot (x-axis), overriding the default, auto-detected range. Similarly, `--flux-range [MIN:MAX]` and `--wdisp-range [MIN:MAX]` work for the flux (left y-axis) and dispersion (right y-axis). MIN and MAX can be either blank (which means use the default value), an absolute value (1000), or a percentage (10%), and percentages and absolute values may be mixed. Working examples:

```
--wlen-range [:7500]
--wlen-range [10%:90%]
--wlen-range [10%:8000]
```

Note that a percentage value between 0-100% is interpreted as a percentile for vertical (flux, wdisp) axes. In all other cases, percentage values specify a limit value equal to a fraction of the full range `[lo:hi]`:

```
limit = lo + fraction*(hi - lo)
```

and can be < 0% or > 100% to include padding. Another visual option `--scatter` will give a scatter plot of the flux rather than the flux 1-sigma error band.

Plots include a label `PLATE-MJD-FIBER` by default (or `PLATE-MJD-FIBER-EXPID` for a single exposure). Add the option `--label-pos <VALIGN>-<HALIGN>` option to change its position, with `<VALIGN>` = `top`, `center`, `bottom` and `<HALIGN>` = `left`, `center`, `right`. Use `--label-pos none` to remove the label. Use `--no-grid` to remove the default wavelength grid lines.

Several options are available to see data beyond just object flux. Use `--show-sky` to show the subtracted sky (modeled) flux, `--add-sky` to show the total of object flux and modeled sky flux, `--show-mask` to show grayed regions where data has been masked out because it is deemed invalid, and `--show-dispersion` to show wavelength dispersion.

You will sometimes want to see data that would normally be masked as invalid. To include pixels with a particular [mask bit](#) set, use the `--allow-mask` option, e.g.:

```
bossplot --allow-mask 'BRIGHTSKY|SCATTEREDLIGHT'
```

Note that multiple flags can be combined using the logical-or symbol `|`, but this requires quoting as shown above. To show all data, including any invalid pixels, use the `--show-invalid` option.

The `bossplot` command will automatically download the appropriate data file if necessary. This is ‘conservative’: if an existing local file can be used to satisfy a request, no new files will be downloaded.

Spectra can be plotted from different data files. By default the `spec-lite` data file is used for a coadd or the `spec` file for an individual exposure. Use the `--frame` or `--cframe` options to plot a single-exposure spectrum from a `plate spFrame` file or its flux-calibrated equivalent `spCFrame` file. Use the `--platefile` option to plot the combined spectrum from an `spPlate` file. See the [Overview of SDSS Spectroscopic Data](#) for details.

To plot a single exposure, use the `--exposure` option to specify the sequence number (0,1,...) of the desired exposure. You can also set the `--band` option either `blue` or `red` to plot a single camera’s data, or `both` to superimpose the overlapping data from both cameras. Note that when displaying data from a co-added data product (`spec`, `speclite`, `spPlate`), the exposure sequence number only indexes exposures that were actually used in the final co-added spectrum. However, the `spFrame` and `spCFrame` data products include all exposures used as input to the co-add (based on a [bosssdata.plate.Plan](#)) so, in cases where not all exposures are used, the `--exposure` option indexes a larger list of science exposures. Use the `--verbose` option to display information about the available exposures in either case.

This script uses the [matplotlib](#) python library, which is not required for the `bosssdata` package and therefore not automatically installed, but is included in scientific python distributions like [anaconda](#).

3.4 bossraw

Assemble the raw CDD exposures used from one camera for a coadd into a single multi-extension FITS file. For example:

```
bossraw --plate 6641 --mjd 56383 --camera b1 --verbose
```

The MJD argument is optional in the common case that there is only one possible value. The output will be saved to a file `{plate}-{mjd}-{camera}.fits` by default, or you can specify a file name with the `--save` argument.

The saved raw data is bias subtracted by default (or use `--no-bias-subtraction`) and consists of only the [data regions](#) of each CCD quadrant. See [bosssdata.raw.RawImageFile.get_data\(\)](#) for details on the remaining optional arguments.

See the [sdR datamodel](#) for more information about raw data.

To view the images stored in the output file, open it in [DS9](#) using the *File / Open As / Multiple Extension Frames...* menu item.

CHAPTER 4

Configuration

You will normally want to establish your local configuration and specify which remote data you want to work with using some environment variables:

- **BOSS_LOCAL_ROOT**: The top-level directory where all downloaded data files will be locally mirrored. Make sure there is enough space here for the files you plan to use locally. You might want to exclude this directory from your backups since it can get large and is already backed up remotely.
- **BOSS_DATA_URL**: The top-level URL for downloading data, possibly including account information for accessing proprietary data.
- **BOSS_SAS_PATH**: The top-level path of the data you want to work with, which will normally begin with “/sas”.
- **BOSS_REDUX_VERSION**: The pipeline reconstruction version that you want to work with.

If any of these variables is not specified, defaults appropriate for access the public [Data Release 12](#) will be used and any downloaded data will be saved to a temporary local directory. At a minimum, you should normally specify a permanent location for storing local data by setting the **BOSS_LOCAL_ROOT** environment variable.

The default settings of the other environment variables are equivalent to (in bash):

```
export BOSS_DATA_URL=http://dr12.sdss3.org
export BOSS_SAS_PATH=/sas/dr12/boss
export BOSS_REDUX_VERSION=v5_7_0
```

However these variables are set, the following unix shell command should always print a valid URL that displays a directory listing in any browser:

```
echo $BOSS_DATA_URL/$BOSS_SAS_PATH/boss/spectro/redux/$BOSS_REDUX_VERSION/
```

You can optionally define one more environment variable **BOSS_SPECLOG** to locate a local checkout of the `speclog` svn product. This is only required if you need to access the full plug maps (including non-science fibers) and prefer to use an environment variable instead of passing a path argument. See the [read_plug_map\(\)](#) documentation for details.

If you are running on a system where the full dataset is available directly via the file system (e.g., at NERSC), use a `$BOSS_DATA_URL` that starts with the `file://` URI to indicate that data does not need to be transferred via network to your `$BOSS_LOCAL_ROOT`. In this case, the local root will still be used for the sqlite files created by the *meta module*. For details on using bossdata at NERSC, see [this guide](#).

The sections below describe how to access sources of data other than the default public DR12 release.

4.1 SEQUELS Data

Quoting from [here](#):

For BOSS, the main galaxy clustering survey is entirely contained in v5_7_0. After the main survey was finished, ancillary programs continued — these were processed as v5_7_2, which is the same code but a different processing version number to keep the datasets distinct. The SEQUELS ancillary program has plates in both v5_7_0 and v5_7_2.

To access SEQUELS data processed as v5_7_2, use:

```
export BOSS_SAS_PATH=/sas/dr12/boss
export BOSS_REDUX_VERSION=v5_7_2
export BOSS_DATA_URL=http://dr12.sdss3.org
```

4.2 SDSS-I/II Spectra

Some spectra from plates 0266 - 3006 are included in the public DR12 release and available under pipeline reduction versions 26, 103 and 104. To access version 26, for example, use:

```
export BOSS_SAS_PATH=/sas/dr12/sdss
export BOSS_REDUX_VERSION=26
export BOSS_DATA_URL=http://dr12.sdss3.org
```

4.3 eBOSS Proprietary Data

Proprietary data from the [eBOSS survey](#) is password protected but still accessible via bossdata. Contact the authors for details if you are an SDSS-IV collaborator.

To use the `bossdata` package in your own python projects, you will normally start with:

```
import bossdata
finder = bossdata.path.Finder()
mirror = bossdata.remote.Manager()
```

This code will use the environment variables `$BOSS_SAS_PATH`, `$BOSS_REDUX_VERSION`, `$BOSS_DATA_URL` and `$BOSS_LOCAL_ROOT` to configure your access to SDSS data files (see [Configuration](#) for details.) The `finder` and `mirror` objects can be used together to access locally mirrored copies of BOSS data files. For example:

```
remote_path = finder.get_spec_path(plate=4567, mjd=55589, fiber=88, lite=True)
local_path = mirror.get(remote_path)
```

Refer to the [API documentation](#) for details on using the `bossdata.path` and `bossdata.remote` modules.

Certain data files have a helper class for accessing their contents:

- spec,spec-lite: `bossdata.spec.SpecFile`
- plate: `bossdata.plate.PlateFile`
- plan: `bossdata.plate.Plan`
- frame,cframe: `bossdata.plate.FrameFile`

For example, to open the spec-lite file used in the example above, use:

```
spec = bossdata.spec.SpecFile(local_path)
```

The pattern for accessing large metadata files is somewhat different, and handled by the `bossdata.meta.Database` class.

CHAPTER 6

Examples

The following IPython notebooks demonstrate using the API to work with BOSS data:

- [How to plot the spatial distributions of quasar metadata](#)
- [How to use the speclite package to stack sky and quasar spectra](#)
- [Demonstration of `bossdata.plot` functions](#)

Using bosssdata at NERSC

Use the following commands to install bosssdata at NERSC, based on the DESI conda environment:

```
# Use the DESI conda environment
source /project/projectdirs/desi/software/desi_environment.sh

# Setup a scratch area where additional packages can be installed
cd $SCRATCH
mkdir -p desi/lib/python3.5/site-packages desi/bin desi/code
export PYTHONPATH=$SCRATCH/desi/lib/python3.5/site-packages:$PYTHONPATH
export PATH=$SCRATCH/desi/bin:$PATH

# Install bosssdata into this scratch area
cd $SCRATCH/desi/code
git clone https://github.com/dkirkby/bosssdata
cd bosssdata
python setup.py develop --prefix $SCRATCH/desi

# Create a directory for sqlite databases created by the meta module.
mkdir -p $SCRATCH/bosssdata
```

The commands are run once, for the initial bosssdata installation. The following commands must be run each time you login to select this environment:

```
# Use the DESI conda environment
source /project/projectdirs/desi/software/desi_environment.sh

# Use additional packages from the scratch area
export PYTHONPATH=$SCRATCH/desi/lib/python3.5/site-packages:$PYTHONPATH
export PATH=$SCRATCH/desi/bin:$PATH

# Configure bosssdata
export BOSS_LOCAL_ROOT=$SCRATCH/bosssdata
export BOSS_DATA_URL=file:///global/projecta/projectdirs/sdss/www
export BOSS_SAS_PATH=/sas/dr12/boss
export BOSS_REDUX_VERSION=v5_7_0
```

The environment variable settings above are for the public DR12 BOSS data release, but can be adjusted for other releases as described [here](#). The key point is that your `$BOSS_DATA_URL` should start with `file:///` to indicate that all data files are available locally and so do not need to be downloaded via the network. In this case, `$BOSS_LOCAL_ROOT` is still used for the sqlite databases used by the *meta module*.

To test your setup, try some of the *command line tools*.

To update your version of bossdata to the latest master branch, use:

```
cd $SCRATCH/desi/code/bossdata
git pull
python setup.py develop --prefix $SCRATCH/desi
```


Contributions are welcome, and they are greatly appreciated! Every little bit helps, and credit will always be given. You can contribute in many ways:

8.1 Types of Contributions

8.1.1 Report Issues

Report issues on our [issues page](#). First check that if your issue is already addressed. If so, feel free to join its conversation and add any relevant information from your experience. If this is a new issue, click the `New Issue` button to describe it, including:

- The type of data you are trying to access (BOSS, SEQUELS, ...)
- Any details about your local setup that might be helpful in troubleshooting.
- Detailed steps to reproduce the bug.

8.1.2 Propose a New Feature

You can also open a new issue to propose a new feature:

- Explain in detail how it would work.
- Keep the scope as narrow as possible, to make it easier to implement.
- Remember that this is a volunteer-driven project, and that contributions are welcome :)

8.1.3 Work on Issues

Look through the [open issues](#) for areas where we currently need help from developers like you. If you find an issue that you are willing to contribute to, start by joining its conversation and tell us about your ideas.

8.1.4 Write Documentation

bosssdata could always use more documentation, whether as part of the official bosssdata docs, in docstrings, or even on the web in blog posts, articles, and such.

We use the [sphinx napoleon extension](#) and write google-style docstrings. Some helpful tips:

- Use ``text <http://url>`_` to embed external links (don't forget the space!)
- Add `.. _scriptname:` before the heading for new scripts in *bin/scripts.rst*. You can refer to these from other markup as `:ref:`scriptname``.
- Refer to another markup document *docs/otherdoc.rst* as `:doc:`/otherdoc``.
- Add cross references to locally defined API entities using:
 - `classes :class:`bosssdata.module.Class``
 - `methods :meth:`bosssdata.module.Class.method``
 - `functions :func:`bosssdata.module.func``
- You can override the default link text by changing `:role:`target`` to `:role:`text <target>``.

8.2 Get Started!

Ready to contribute? Here's how to set up bosssdata for local development.

1. Fork the bosssdata repo on GitHub.
2. Clone your fork locally:

```
git clone git@github.com:your_name_here/bosssdata.git
```

3. Install your local copy for local development:

```
cd bosssdata/  
python setup.py develop --user
```

To later revert back to a system-installed version of the package, un-install your development install using:

```
python setup.py develop --user --uninstall
```

4. Create a branch for local development:

```
git checkout -b '#nnn'  
git push -u origin '#nnn'
```

where `nnn` is the number of the issue you are working on (quotes are required because of the `#` symbol in the branch name). Now you can make your changes locally.

5. When you're done making changes, check that your changes pass flake8 and the unit tests:

```
flake8 --doctests --exclude bosssdata/bits.py --max-line-length 95 bosssdata  
py.test --doctest-modules --verbose bosssdata
```

Note that `--doctest-modules` will require that all external modules imported from our modules are installed, so omit that option if you only want to run the unit tests. If you don't already have flake8, you can pip install it.

6. Commit your changes and push your branch to GitHub:

```
git add .
git commit -m "Your detailed description of your changes."
git push origin '#nnn'
```

7. Submit a pull request.

8.3 Pull Request Guidelines

Before you submit a pull request, check that it meets these guidelines:

1. The pull request should include tests, if appropriate.
2. If the pull request adds functionality, the docs should be updated. Put your new functionality into a function with a docstring, and add the feature to the list in HISTORY.rst.
3. The pull request should work for Python 2.6 and 2.7. Check https://travis-ci.org/dkirkby/bossdata/pull_requests and make sure that the tests pass for all supported Python versions.

8.4 Version Update Checklist

1. Start a new release candidate branch, e.g:

```
git checkout -b 0.2.1rc
git push -u origin 0.2.1rc
```

2. Update the version in setup.py
3. Update the `__version__` in `__init__.py`
4. Add a brief description of the changes to HISTORY.rst and update the What's New section of DESCRIPTION.rst (which is what pypi will display for this release). You can get a list of merges to master since the last tagged release using:

```
git log --oneline --merges `git describe --tags --abbrev=0`..HEAD
```

5. Push changes to github, which will trigger a Travis integration test of the release-candidate branch.
6. Create a pull request on github for this branch and ask someone else to review it and give feedback.
7. Merge the pull request.
8. Update local master and tag the new version, e.g:

```
git fetch
git checkout master
git pull
git tag 0.2.1
git push --tags
git branch -d 0.2.1rc
```

9. Submit the changes to pypi:

```
python setup.py sdist bdist_wheel upload
```

10. Update the version in setup.py and `__version__` in `__init__.py` to indicate that master is under development, e.g. to 0.2.2dev.

11. Reset the What 's New section of `DESCRIPTION.rst` and add a new entry at the bottom of `HISTORY.rst`, e.g:

```
0.2.2 (unreleased)
-----

* No changes yet.
```

12. Update master so that new topic branches will include these changes, e.g:

```
git add setup.py bossdata/__init__.py HISTORY.rst DESCRIPTION.rst
git commit -m 'Start development on version 0.2.2'
git push
```

8.5 New External Dependency Checklist

These steps are not required for modules that are included with the python standard library.

1. Add to `MOCK_MODULES` in `docs/conf.py`.
2. Add the actual version being used to `requirements.txt`
3. Add to the `requirements` list in `setup.py`
4. Mention in `docs/installation.rst`

9.1 0.3.1 (unreleased)

- Fix issues #108 #109 and merges #110 #120
- Remove requirements.txt, add docs/rtd-pip-requirements, update .travis.yaml
- Add support for running at sites where data is locally visible (e.g., nersc)
- Add support for reading spArc and spFlat calibration files.

9.2 0.3.0 (2017-12-17)

- Update for python 3.x

9.3 0.2.8 (2015-10-19)

- Fix issues #29 #103 and addresses #106
- Add support for reading raw image data and plug map files.
- Add warnings about not using numpy 1.10.0 or 1.10.1.

9.4 0.2.7 (2015-09-28)

- Fix issues #92 #94 #96 #97 #100
- Add support for reading per-exposure flux calibration and correction vectors.
- Add plot functions for per-fiber data vs fiber number or focal-plane position.

- Add a `plug_map` attribute to `spPlate`, `spFrame`, `spCFrame`.
- `FrameFile` infers the spectrograph index and whether flux calibration has been applied.
- `bossdata` infers MJD when possible.
- `bossplot` option “`–camera`” renamed to “`–band`”.

9.5 0.2.6 (2015-08-05)

- Fix issues #67 #74 #86
- The `camera` arg to `SpecFile.get_valid_data` (and related methods) should now be `b1`, `b2`, `r1`, `r2` instead of `blue` or `red`.
- New options for the `get_valid_data` methods: `use_ivar`, `use_loglam`, `fiducial_grid`.

9.6 0.2.5 (2015-07-06)

- Fix issues #27 #28 #63 #64 #68
- New command-line options include:
- `bossplot`: `–platefile`, `–flux-range`, `–wlen-range`, `–wdisp-range`, `–label-pos`, `–no-grid`, `–show-invalid`
- `bossfetch`: `–platefile`
- Adds support for `spPlate` files and `platelist` metadata.
- Adds command-line options to customize `bossplot` axes, add labels and grids, and display invalid data.
- General documentation cleanup.
- Better error handling in `bossplot`.

9.7 0.2.4 (2015-06-29)

- Fix issues #11 #36 #41 #43 #45 #50
- New command-line options include:
- `bossfetch`: `–plate-name`, `–mjd-name`, `–fiber-name`
- `bosscatalog`: `–quasar-catalog`, `–quasar-catalog-name`
- The main new functionality is support for querying the quasar catalog, using different data sources, and built-in defaults for any of the four environment variables that is not set.

9.8 0.2.3 (2015-06-22)

- Fix issues #2 #10 #16 #18 #19 #21 #24
- New command-line options include:
- `bossfetch`: `–globus`, `–dry-run`
- `bossplot`: `–save-data`

- bossquery: -sort
- The main new library functionality is support for using wavelengths and dispersions encoded as “trace sets” in spFrame files via `bossdata.plate.TraceSet`.

9.9 0.2.2 (2015-06-15)

- Really fix issues #9 #13.
- Add support for finding and fetching spFrame and spCFrame files (#17).

9.10 0.2.1 (2015-06-13)

- Fix issues #9 #12 #13

9.11 0.2.0 (2015-06-09)

- Fix issues #3 #5 #6
- Add support for accessing subtracted sky flux to the *spec* module and *bossplot* script.
- This version breaks backwards compatibility with 0.1.0 since the previous `$BOSS_SAS_ROOT` environment variable is now named `$BOSS_SAS_PATH` and has the instrument name (usually *boss*) appended.
- bash users can update by replacing `export BOSS_SAS_ROOT=/sas/dr12` with `export BOSS_SAS_PATH=/sas/dr12/boss` in their `.bashrc` file.

9.12 0.1.0 (2015-05-24)

- First release on PyPI.

Define bit masks used in BOSS data and support symbolic operations on masks.

The SDSS bitmasks are documented at <http://www.sdss3.org/dr10/algorithms/bitmasks.php>. The authoritative definition of the bit masks is the file <http://www.sdss3.org/svn/repo/idlutils/trunk/data/sdss/sdssMaskbits.par>. A copy of this file is included in this package's top-level directory and was used to automatically generate the bitmask definitions in this file with the `extract_sdss_bitmasks()` function.

```
class bossdata.bits.ANCILLARY_TARGET1
    BOSS survey target flags for ancillary programs

    AMC
        int – (1<<0) defined in blake_boss_v2.descr

    RQSS_SF
        int – (1<<32) defined in rqss090630.descr

    RED_KG
        int – (1<<48) defined in redkg.descr

    BRIGHTERM
        int – (1<<45) defined in sd3targets_final.descr

    CXORED
        int – (1<<60) defined in brandt-xray.par

    QSO_GRI
        int – (1<<29) defined in sdss3_fan.descr

    BLAZXRVAR
        int – (1<<10) defined in brandtxmm-andersonblazar-merged.descr

    WHITEDWARF_NEW
        int – (1<<42) defined in WDV5_eisenste_fixed.descr

    SPEC_SN
        int – (1<<40) defined in ancillary_supernova_hosts_v5.descr
```

BLAZXRSAM

int – (1<<9) defined in brandtxmm-andersonblazar-merged.descr

CXOGRIZ

int – (1<<59) defined in brandt-xray.par

BLAZGVAR

int – (1<<6) defined in brandtxmm-andersonblazar-merged.descr

MTEMP

int – (1<<63) defined in blake-transient-v3.fits

SN_GAL3

int – (1<<38) defined in ancillary_supernova_hosts_v5.descr

SN_GAL1

int – (1<<36) defined in ancillary_supernova_hosts_v5.descr

BRIGHTERL

int – (1<<44) defined in sd3targets_final.descr

FAINTERM

int – (1<<47) defined in sd3targets_final.descr

ODDBAL

int – (1<<17) defined in master-BAL-targets.descr

FAINTERL

int – (1<<46) defined in sd3targets_final.descr

RVTEST

int – (1<<49) defined in redkg.descr

RQSS_STMC

int – (1<<35) defined in rqss090630.descr

VARBAL

int – (1<<20) defined in master-BAL-targets.descr

PREVBAL

int – (1<<19) defined in master-BAL-targets.descr

RQSS_STM

int – (1<<34) defined in rqss090630.descr

ELG

int – (1<<61) defined in kneib-cfht-elg.fits

BLAZGRFLAT

int – (1<<50) defined in anderson-blazar.par

XMMGRIZ

int – (1<<12) defined in brandtxmm-andersonblazar-merged.descr

QSO_RADIO

int – (1<<25) defined in qsoals_v2.descr

LBQSBAL

int – (1<<16) defined in master-BAL-targets.descr

QSO_RADIO_AAL

int – (1<<26) defined in qsoals_v2.descr

QSO_AAL

int – (1<<22) defined in qsoals_v2.descr

OTBAL

int – (1<<18) defined in master-BAL-targets.descr

BLAZR

int – (1<<7) defined in brandtxmm-andersonblazar-merged.descr

WHITEDWARF_SDSS

int – (1<<43) defined in WDV5_eisenste_fixed.descr

FBQSBAL

int – (1<<15) defined in master-BAL-targets.descr

BLAZGXR

int – (1<<54) defined in anderson-blazar.par

SPOKE

int – (1<<41) defined in BOSS_slowpokes_v2.descr

BLAZGX

int – (1<<52) defined in anderson-blazar.par

RQSS_SFC

int – (1<<33) defined in rqss090630.descr

LOW_MET

int – (1<<4) defined in blake_boss_v2.descr

SN_GAL2

int – (1<<37) defined in ancillary_supernova_hosts_v5.descr

QSO_NOAALS

int – (1<<28) defined in qsoals_v2.descr

BRIGHTGAL

int – (1<<21) defined in bright_gal_v3.descr

VARs

int – (1<<5) defined in blake_boss_v2.descr

HPM

int – (1<<3) defined in blake_boss_v2.descr

XMMRED

int – (1<<14) defined in brandtxmm-andersonblazar-merged.descr

BLUE_RADIO

int – (1<<56) defined in tremonti-blue-radio.fits.gz

QSO_HIZ

int – (1<<30) defined in sdss3_fan.descr

QSO_AALS

int – (1<<23) defined in qsoals_v2.descr

SN_LOC

int – (1<<39) defined in ancillary_supernova_hosts_v5.descr

XMMHR

int – (1<<13) defined in brandtxmm-andersonblazar-merged.descr

QSO_RIZ

int – (1<<31) defined in sdss3_fan.descr

BLAZXR

int – (1<<8) defined in brandtxmm-andersonblazar-merged.descr

QSO_RADIO_IAL

int – (1<<27) defined in qsoals_v2.descr

FLARE1

int – (1<<1) defined in blake_boss_v2.descr

CHANDRAV1

int – (1<<57) defined in haggard-sf-accrete.fits

XMMBRIGHT

int – (1<<11) defined in brandtxmm-andersonblazar-merged.descr

BLAZGRQSO

int – (1<<51) defined in anderson-blazar.par

GAL_NEAR_QSO

int – (1<<62) defined in weiner-qso-sightline.fits

CXOBRIGHT

int – (1<<58) defined in brandt-xray.par

BLAZGXQSO

int – (1<<53) defined in anderson-blazar.par

FLARE2

int – (1<<2) defined in blake_boss_v2.descr

QSO_IAL

int – (1<<24) defined in qsoals_v2.descr

class bossdata.bits.**ANCILLARY_TARGET2**

additional BOSS survey target flags for ancillary programs

QSO_STD

int – (1<<20) defined in margala.descr

HIZ_LRG

int – (1<<21) defined in newman.descr

_2MASSFILL

int – (1<<51) defined in rocksi_ges_segue.descr

GES

int – (1<<35) defined in rockosi_ges_segue.descr

QSO_WISE_SUPP

int – (1<<9) defined in BOSS_QSO_targets_July_WISE.descr

SPOKE2

int – (1<<17) defined in dhital.descr

KOEKAP_STAR

int – (1<<42) defined in knapp_kappaori.descr

KOE2068BSTAR

int – (1<<46) defined in knapp_ngc2068.descr

SPIDERS_PILOT
int – (1<<25) defined in GreenMerloni_MD01.descr

_25ORI_WISE_W3
int – (1<<41) defined in knapp_25ori.descr

TDSS_PILOT_PM
int – (1<<28) defined in TDSS_SPIDERS_MD03.descr

COROTGESAPOG
int – (1<<48) defined in rocksi_ges_segue.descr

XMM_PRIME
int – (1<<32) defined in georgekaksi_xmmxll.descr

KQSO_BOSS
int – (1<<2) defined in mcmahon-ukidss.fits

QSO_WISE_FULL_SKY
int – (1<<10) defined in none

LBG
int – (1<<57) LBG described in QSO_DEEP_LBG.descr

PTF_GAL
int – (1<<19) defined in kasliwal.descr

HIZQSO82
int – (1<<0) defined in mcgreer-hizqso.fits

SEGUE1
int – (1<<36) defined in rockosi_ges_segue.descr

KOE2023_STAR
int – (1<<43) defined in knapp_ngc2023.descr

ELAIS_N1_JVLA
int – (1<<62) LOFAR-selected target

HIZQSOIR
int – (1<<1) defined in mcgreer-hizqso.fits

QSO_XD_KDE_PAIR
int – (1<<15) defined in myers.descr

ELAIS_N1_GMRT_TAYLOR
int – (1<<61) LOFAR-selected target

XMM_SECOND
int – (1<<33) defined in georgekaksi_xmmxll.descr

KOEKAPBSTAR
int – (1<<47) defined in knapp_kappaori.descr

SEQUELS_ELG_LOWP
int – (1<<39) defined in sequels_elg.descr

DISKEMITTER_REPEAT
int – (1<<13) defined in shen.descr

ELAIS_N1_GMRT_GARN
int – (1<<60) LOFAR-selected target

QSO_EBOSS_W3_ADM
int – (1<<31) defined in myers_eboss_qso_w3.descr

RM_TILE1
int – (1<<54) reverberation mapping, high priority

ELAIS_N1_FIRST
int – (1<<59) LOFAR-selected target

SEQUELS_ELG
int – (1<<34) defined in sequels_elg.descr

QSO_SUPPZ
int – (1<<7) defined in qso_suppz.descr

WISE_COMPLETE
int – (1<<23) defined in weiner_wise.descr

RM_TILE2
int – (1<<55) reverberation mapping, low priority

TDSS_PILOT
int – (1<<24) defined in GreenMerloni_MD01.descr

QSO_VAR_FPG
int – (1<<4) defined in nathalie-ancillary3.par

_25ORI_WISE
int – (1<<40) defined in knapp_25ori.descr

CLUSTER_MEMBER
int – (1<<16) defined in finoguenov_auxBOSS.descr

KOE2023BSTAR
int – (1<<45) defined in knapp_ngc2023.descr

SEGUE2
int – (1<<37) defined in rockosi_ges_segue.descr

QSO_VAR_SDSS
int – (1<<8) defined in VARQSO.descr

STRIPE82BCG
int – (1<<6) defined in alexie-bcgs.fits

TAU_STAR
int – (1<<52) defined in knapp_taurus.descr

QSO_DEEP
int – (1<<56) DEEP QSO described in QSO_DEEP_LBG.descr

XMMSDSS
int – (1<<11) defined in georgakakis.descr

RADIO_2LOBE_QSO
int – (1<<5) defined in kimball-radio-2lobe-qso.fits.gz

KOE2068_STAR
int – (1<<44) defined in knapp_ngc2068.descr

ELAIS_N1_LOFAR
int – (1<<58) LOFAR-selected target

WISE_BOSS_QSO*int* – (1<<14) defined in ross_wisebossqso.descr**TDSS_PILOT_SNHOST***int* – (1<<29) defined in TDSS_SPIDERS_MD03.descr**IAMASERS***int* – (1<<12) defined in zaw.descr**APOGEE***int* – (1<<50) defined in rocksi_ges_segue.descr**QSO_VAR_LF***int* – (1<<27) defined in palanque_str82.descr**FAINT_ELG***int* – (1<<18) defined in comparat.descr**TDSS_SPIDERS_PILOT***int* – (1<<26) defined in GreenMerloni_MD01.descr**FAINT_HIZ_LRG***int* – (1<<30) defined in newman_lrg_w3.descr**SEQUELS_TARGET***int* – (1<<53) any target in SEQUELS darktime program**COROTGES***int* – (1<<49) defined in rocksi_ges_segue.descr**QSO_VAR***int* – (1<<3) defined in butler-variable.fits.gz**SDSSFILLER***int* – (1<<38) defined in rockosi_ges_segue.descr**LRG_ROUND3***int* – (1<<22) defined in newman.descr**class** bossdata.bits.**APOGEE2_TARGET1**

APOGEE2 primary target bits

APOGEE2_ONEBIN_GT_0_5*int* – (1<<0) Selected in single (J-Ks)_o > 0.5 color bin**APOGEE2_DSPH_CANDIDATE***int* – (1<<21) Selected as potential dSph member (non Sgr) (based on photometry)**APOGEE2_SGR_DSPH***int* – (1<<26) Selected as confirmed Sgr core/stream member**APOGEE2_DSPH_MEMBER***int* – (1<<20) Selected as confirmed dSph member (non Sgr)**APOGEE2_MANGA_LED***int* – (1<<15) Star on a shared MaNGA-led design**APOGEE2_RRLYR***int* – (1<<24) Selected as a bulge RR Lyrae star**APOGEE2_MEDIUM***int* – (1<<12) Selected as part of a medium cohort

APOGEE2_WASH_GIANT

int – (1<<7) Selected as Wash+DDO51 photometric giant

APOGEE2_TWOBIN_GT_0_8

int – (1<<2) Selected in red (J-Ks)_o > 0.8 color bin

APOGEE2_SCI_CLUSTER

int – (1<<9) Science cluster candidate member

APOGEE2_STREAM_CANDIDATE

int – (1<<19) Selected as potential halo tidal stream member (based on photometry)

APOGEE2_WASH_NOCLASS

int – (1<<17) Selected because it has no W+D classification

APOGEE2_LONG

int – (1<<13) Selected as part of a long cohort

APOGEE2_NO_DERED

int – (1<<6) Selected with no dereddening

APOGEE2_ONEBIN_GT_0_3

int – (1<<16) Selected in single (J-Ks)_o > 0.3 color bin

APOGEE2_MAGCLOUD_MEMBER

int – (1<<22) Selected as confirmed Mag Cloud member

APOGEE2_TWOBIN_0_5_TO_0_8

int – (1<<1) Selected in blue 0.5 < (J-Ks)_o < 0.8 color bin

APOGEE2_FAINT_EXTRA

int – (1<<29) Faint star (fainter than cohort limit; not required to reach survey S/N requirement)

APOGEE2_WASH_DWARF

int – (1<<8) Selected as Wash+DDO51 photometric dwarf

APOGEE2_NORMAL_SAMPLE

int – (1<<14) Selected as part of the random sample

APOGEE2_APOKASC

int – (1<<30) Selected as part of the APOKASC program (incl. seismic/gyro targets and others)

APOGEE2_IRAC_DERED

int – (1<<3) Selected with RJCE-IRAC dereddening

APOGEE2_STREAM_MEMBER

int – (1<<18) Selected as confirmed halo tidal stream member

APOGEE2_APOKASC_GIANT

int – (1<<27) Selected as part of APOKASC giant sample

APOGEE2_MAGCLOUD_CANDIDATE

int – (1<<23) Selected as potential Mag Cloud member (based on photometry)

APOGEE2_SFD_DERED

int – (1<<5) Selected with SFD_EBV dereddening

APOGEE2_SHORT

int – (1<<11) Selected as part of a short cohort

APOGEE2_WISE_DERED

int – (1<<4) Selected with RJCE-WISE dereddening

APOGEE2_APOKASC_DWARF

int – (1<<28) Selected as part of APOKASC dwarf sample

class bossdata.bits.APOGEE2_TARGET2

APOGEE2 secondary target bits

APOGEE2_1M_TARGET

int – (1<<22) Selected as a 1-m target

APOGEE2_ARGOS_OVERLAP

int – (1<<15) Overlap with ARGOS

APOGEE2_TELLURIC

int – (1<<9) Telluric calibrator target

APOGEE2_OBJECT

int – (1<<30) This object is an APOGEE-2 target

APOGEE2_INTERNAL_CALIB

int – (1<<6) Internal survey calibration target (observed in at least 2 of: APOGEE-1, -2N, -2S)

APOGEE2_RAVE_OVERLAP

int – (1<<18) Overlap with RAVE

APOGEE2_GAIA_OVERLAP

int – (1<<16) Overlap with Gaia

APOGEE2_GALAH_OVERLAP

int – (1<<17) Overlap with GALAH

APOGEE2_GES_OVERLAP

int – (1<<14) Overlap with Gaia-ESO

APOGEE2_CALIB_CLUSTER

int – (1<<10) Selected as calibration cluster member

APOGEE2_STANDARD_STAR

int – (1<<2) Stellar parameters/abundance standard

APOGEE2_SKY

int – (1<<4) Sky fiber

APOGEE2_EXTERNAL_CALIB

int – (1<<5) External survey calibration target (generic flag; others below dedicated to specific surveys)

APOGEE2_LITERATURE_CALIB

int – (1<<13) Overlap with high-resolution literature studies

APOGEE2_RV_STANDARD

int – (1<<3) Stellar RV standard

class bossdata.bits.APOGEE2_TARGET3

APOGEE2 trinary target bits

APOGEE2_ANCILLARY

int – (1<<8) Selected as an ancillary target

APOGEE2_SUBSTELLAR_COMPANIONS

int – (1<<4) Selected as part of the substellar companion search

APOGEE2_EB

int – (1<<1) Selected as part of the EB program

APOGEE2_MDWARF

int – (1<<3) Selected as part of the M dwarf study

APOGEE2_YOUNG_CLUSTER

int – (1<<5) Selected as part of the young cluster study (IN-SYNC)

APOGEE2_KOI

int – (1<<0) Selected as part of the long cadence KOI study

APOGEE2_MASSIVE_STAR

int – (1<<9) Selected as part of the Massive Star program

APOGEE2_KOI_CONTROL

int – (1<<2) Selected as part of the long cadence KOI control sample

class bosssdata.bits.APOGEE_ASPCAPFLAG

APOGEE ASPCAP mask bits

ROTATION_BAD

int – (1<<26) Spectrum has broad lines, with possible bad effects: FWHM of cross-correlation of spectrum with best RV template relative to auto-correlation of template > 2 (BAD)

STAR_BAD

int – (1<<23) BAD overall for star: set if any of TEFF, LOGG, CHI2, COLOTE, ROTATION, SN error are set, or any parameter is near grid edge (GRIDEDGE_BAD is set in any PARAMFLAG)

TEFF_WARN

int – (1<<0) WARNING on effective temperature (see PARAMFLAG[0] for details)

NFE_WARN

int – (1<<6) WARNING on [N/Fe] (see PARAMFLAG[6] for details)

SN_WARN

int – (1<<11) S/N<70 (WARN)

COLOTE_WARN

int – (1<<9) effective temperature more than 500K from photometric temperature for dereddened color (WARN)

CHI2_WARN

int – (1<<8) high χ^2 (> 2*median at ASPCAP temperature (WARN)

COLOTE_BAD

int – (1<<25) effective temperature more than 1000K from photometric temperature for dereddened color (BAD)

METALS_BAD

int – (1<<19) BAD metals (see PARAMFLAG[3] for details)

METALS_WARN

int – (1<<3) WARNING on metals (see PARAMFLAG[3] for details)

TEFF_BAD

int – (1<<16) BAD effective temperature (see PARAMFLAG[0] for details)

CHI2_BAD

int – (1<<24) high χ^2 (> 5*median at ASPCAP temperature (BAD)

ALPHAFe_BAD

int – (1<<20) BAD [alpha/Fe] (see PARAMFLAG[4] for details)

CFE_BAD

int – (1<<21) BAD [C/Fe] (see PARAMFLAG[5] for details)

VMICRO_WARN

int – (1<<2) WARNING on vmicro (see PARAMFLAG[2] for details)

STAR_WARN

int – (1<<7) WARNING overall for star: set if any of TEFF, LOGG, CHI2, COLORTE, ROTATION, SN warn are set

ALPHAFe_WARN

int – (1<<4) WARNING on [alpha/Fe] (see PARAMFLAG[4] for details)

ROTATION_WARN

int – (1<<10) Spectrum has broad lines, with possible bad effects: FWHM of cross-correlation of spectrum with best RV template relative to auto-correlation of template > 1.5 (WARN)

LOGG_BAD

int – (1<<17) BAD log g (see PARAMFLAG[1] for details)

CFE_WARN

int – (1<<5) WARNING on [C/Fe] (see PARAMFLAG[5] for details)

SN_BAD

int – (1<<27) S/N<50 (BAD)

NO_ASPCAP_RESULT

int – (1<<31) No result

NFe_BAD

int – (1<<22) BAD [N/Fe] (see PARAMFLAG[6] for details)

VMICRO_BAD

int – (1<<18) BAD vmicro (see PARAMFLAG[2] for details)

LOGG_WARN

int – (1<<1) WARNING on log g (see PARAMFLAG[1] for details)

class bosssdata.bits.APOGEE_EXTRATARG

APOGEE pixel level mask bits

DUPLICATE

int – (1<<4) Duplicate observation of star

TELLURIC

int – (1<<2) Telluric target

NOT_MAIN

int – (1<<0) Not main survey target

COMMISSIONING

int – (1<<1) Commissioning data

APO1M

int – (1<<3) APO1M + APOGEE observation

class bosssdata.bits.APOGEE_PARAMFLAG

APOGEE parameter mask bits (set for each stellar parameter in ASPCAP fit)

OTHER_WARN

int – (1<<10) Other warning condition

GRIDEDGE_WARN

int – (1<<8) Parameter within 1/2 grid spacing of grid edge

GRIDEDGE_BAD

int – (1<<0) Parameter within 1/8 grid spacing of grid edge

CALRANGE_WARN

int – (1<<9) Parameter in possibly unreliable range of calibration determination

CALRANGE_BAD

int – (1<<1) Parameter outside valid range of calibration determination

PARAM_FIXED

int – (1<<16) Parameter set at fixed value, not fit

OTHER_BAD

int – (1<<2) Other error condition

class bosssdata.bits.**APOGEE_PIXMASK**

APOGEE extra targeting bits

LITTROW_GHOST

int – (1<<8) Pixel falls in Littrow ghost, may be affected

UNFIXABLE

int – (1<<3) Pixel marked as unfixable in ap3d

BADPIX

int – (1<<0) Pixel marked as BAD in bad pixel mask

SIG_TELLURIC

int – (1<<13) Pixel falls near telluric line that has significant absorption

SATPIX

int – (1<<2) Pixel marked as saturated in ap3d

CRPIX

int – (1<<1) Pixel marked as cosmic ray in ap3d

BADDARK

int – (1<<4) Pixel marked as bad as determined from dark frame

BADERR

int – (1<<6) Pixel set to have very high error (not used)

PERSIST_LOW

int – (1<<11) Pixel falls in low persistence region, may be affected

PERSIST_MED

int – (1<<10) Pixel falls in medium persistence region, may be affected

PERSIST_HIGH

int – (1<<9) Pixel falls in high persistence region, may be affected

BADFLAT

int – (1<<5) Pixel marked as bad as determined from flat frame

NOSKY

int – (1<<7) No sky available for this pixel from sky fibers

SIG_SKYLINE

int – (1<<12) Pixel falls near sky line that has significant flux compared with object

class bosssdata.bits.**APOGEE_STARFLAG**

APOGEE star-level mask bits

VERY_BRIGHT_NEIGHBOR

int – (1<<3) Star has neighbor more than 100 times brighter: BAD

BAD_PIXELS

int – (1<<0) Spectrum has many bad pixels (>40%): BAD

SUSPECT_BROAD_LINES

int – (1<<17) WARNING: cross-correlation peak with template significantly broader than autocorrelation of template

PERSIST_JUMP_NEG

int – (1<<13) Spectrum show obvious negative jump in blue chip: WARN

PERSIST_JUMP_POS

int – (1<<12) Spectrum show obvious positive jump in blue chip: WARN

SUSPECT_RV_COMBINATION

int – (1<<16) WARNING: RVs from synthetic template differ significantly from those from combined template

LOW_SNR

int – (1<<4) Spectrum has low S/N ($S/N < 5$): BAD

COMMISSIONING

int – (1<<1) Commissioning data (MJD<55761), non-standard configuration, poor LSF: WARN

PERSIST_MED

int – (1<<10) Spectrum has significant number (>20%) of pixels in medium persistence region: WARN

BRIGHT_NEIGHBOR

int – (1<<2) Star has neighbor more than 10 times brighter: WARN

PERSIST_HIGH

int – (1<<9) Spectrum has significant number (>20%) of pixels in high persistence region: WARN

PERSIST_LOW

int – (1<<11) Spectrum has significant number (>20%) of pixels in low persistence region: WARN

class bosssdata.bits.**APOGEE_TARGET1**

APOGEE primary target bits

APOGEE_MEDIUM

int – (1<<1) Selected in medium bin of cohort

APOGEE_SERENDIPITOUS

int – (1<<15) Serendipitously interesting target to reobserve

APOGEE_INTERMEDIATE

int – (1<<12) Intermediate cohort target

APOGEE_SGR_DSPH

int – (1<<26) Sagittarius dwarf spheroidal member

APOGEE_HIRES

int – (1<<20) Star with optical hi-res spectra (ancillary)

APOGEE_EXTENDED

int – (1<<10) Extended object

APOGEE_WISE_DERED

int – (1<<4) Selected using RJCE-WISE dereddening

APOGEE_SFD_DERED

int – (1<<5) Selected using SFD E(B-V) dereddening

APOGEE_CHECKED

int – (1<<31) This target has been checked

APOGEE_MDWARF

int – (1<<19) M dwarfs selected for RV program (ancillary)

APOGEE_KEPLER_SEISMO

int – (1<<27) Kepler asteroseismology program target

APOGEE_KEPLER_EB

int – (1<<23) Eclipsing binary from Kepler (ancillary)

APOGEE_MASSIVE_STAR

int – (1<<25) Selected as massive star (ancillary)

APOGEE_FAINT_EXTRA

int – (1<<29) Selected as faint target for low target-density field

APOGEE_IRAC_DERED

int – (1<<3) Selected using RJCE-IRAC dereddening

APOGEE_OLD_STAR

int – (1<<21) Selected as old star (ancillary)

APOGEE_BRIGHT

int – (1<<2) Selected in bright bin of cohort

APOGEE_DISK_RED_GIANT

int – (1<<22) Disk red giant (ancillary)

APOGEE_DO_NOT_OBSERVE

int – (1<<14) Do not observe (again)

APOGEE_SHORT

int – (1<<11) Short cohort target

APOGEE_M31_CLUSTER

int – (1<<18) M31 cluster target (ancillary)

APOGEE_GC_PAL1

int – (1<<24) Star in globular cluster (ancillary)

APOGEE_WASH_GIANT

int – (1<<7) Selected as giant in Washington photometry

APOGEE_ANCILLARY

int – (1<<17) An ancillary program

APOGEE_SCI_CLUSTER

int – (1<<9) Probable cluster member

APOGEE_SEGUE_OVERLAP

int – (1<<30) Selected because of overlap with SEGUE survey

APOGEE_WASH_DWARF

int – (1<<8) Selected as dwarf in Washington photometry

APOGEE_NO_DERED

int – (1<<6) Selected using no dereddening

APOGEE_FAINT

int – (1<<0) Selected in faint bin of cohort

APOGEE_FIRST_LIGHT

int – (1<<16) First list plate target

APOGEE_LONG

int – (1<<13) Long cohort target

APOGEE_KEPLER_HOST

int – (1<<28) Kepler planet-host program target

class bossdata.bits.APOGEE_TARGET2

APOGEE secondary target bits

SKY_BAD

int – (1<<5) Selected as sky but identified as bad (via visual exam or observation)

APOGEE_TELLURIC_BAD

int – (1<<8) Selected as telluric standard but identified as bad (via SIMBAD or observation)

APOGEE_GC_SUPER_GIANT

int – (1<<12) Probable supergiant in Galactic Center

LIGHT_TRAP

int – (1<<0) Light trap

APOGEE_KEPLER_COOLDWARF

int – (1<<16) Kepler cool dwarf/subgiant (ancillary)

APOGEE_FLUX_STANDARD

int – (1<<1) Flux standard

APOGEE_TELLURIC

int – (1<<9) Hot (telluric) standard

APOGEE_EMBEDDEDCLUSTER_STAR

int – (1<<13) Young embedded clusters (ancillary)

APOGEE_EMISSION_STAR

int – (1<<15) Emission-line star (ancillary)

SKY

int – (1<<4) Sky

APOGEE_GC_GIANT

int – (1<<11) Probable giant in Galactic Center

APOGEE_STANDARD_STAR

int – (1<<2) Stellar abundance, parameters standard

APOGEE_CHECKED

int – (1<<31) This target has been checked

GUIDE_STAR

int – (1<<6) Guide star

APOGEE_MIRCLUSTER_STAR

int – (1<<17) Candidate MIR-detected cluster member (ancillary)

APOGEE_LONGBAR

int – (1<<14) Probable RC star in long bar (ancillary)

APOGEE_CALIB_CLUSTER

int – (1<<10) Known calibration cluster member

BUNDLE_HOLE

int – (1<<7) Bundle hole

APOGEE_RV_STANDARD

int – (1<<3) Radial velocity standard

class bosssdata.bits.BOSSTILE_STATUS

BOSS tiling code status bits

KNOWN_OBJECT

int – (1<<16) galaxy has known redshift

OUT_OF_BOUNDS

int – (1<<13) outside bounds for this sort of target (for restricted QSO geometry, e.g.)

NAKED

int – (1<<1) not in area covered by tiles

REPEAT

int – (1<<10) included on more than one tile

MIDLEVEL_PRIORITY

int – (1<<23) targets (from ancillary list) tiled between gals and ancillaries

BLUEFIBER

int – (1<<8) allocate this object a blue fiber

DUPLICATE

int – (1<<17) trimmed as a duplicate object (only checked if not trimmed for any other reason)

DECOLLIDED

int – (1<<3) in the decollided set of high priority

PREVIOUS_CHUNK

int – (1<<15) included because not tiled in previous overlapping chunk

TOOBRIGHT

int – (1<<7) fibermag too bright

TILED

int – (1<<0) assigned a fiber

FILLER

int – (1<<11) was a filler (not normal repeat)

POSSIBLE_KNOCKOUT

int – (1<<5) knocked out of at least one tile by BOSSTARGET

BAD_CALIB_STATUS

int – (1<<14) bad CALIB_STATUS

TOOFAINT

int – (1<<20) trimmed because it was fainter than the ifiber2mag limit

ANCILLARY

int – (1<<4) in the lower priority, ancillary set

NOT_TILED_TARGET

int – (1<<12) though in input file, not a tiled target

BOSSTARGET

int – (1<<2) in the high priority set of targets

SUPPLEMENTARY

int – (1<<21) supplementary targets tiles after the ancillaries (subset of ancillaries)

ANCILLARY_ROUND2

int – (1<<22) new ancillaries added June 2012 (tiled after old ancillaries)

DUPLICATE_TILED

int – (1<<19) trimmed as a duplicate object, and its primary was tiled

DUPLICATE_PRIMARY

int – (1<<18) has associated duplicate object that were trimmed (but this one is kept)

IGNORE_PRIORITY

int – (1<<6) priority exceeds max (ANCILLARY only)

CENTERPOST

int – (1<<9) 92 arcsec collision with center post

class bossdata.bits.BOSS_TARGET1

BOSS survey primary target selection flags

QSO_CORE_ED

int – (1<<42) Extreme Deconvolution in Core

GAL_LOZ

int – (1<<0) low-z lrgs

GAL_IFIBER2_FAINT

int – (1<<8) ifiber2 > 21.5, extinction corrected. Used after Nov 2010

STD_FSTAR

int – (1<<20) standard f-stars

QSO_CORE

int – (1<<10) restrictive qso selection: commissioning only

QSO_CORE_MAIN

int – (1<<40) Main survey core sample

QSO_KDE

int – (1<<19) selected by kde+chi2

GAL_CMASS_SPARSE

int – (1<<3) GAL_CMASS_COMM & (!GAL_CMASS) & (i < 19.9) sparsely sampled

GAL_CMASS_ALL

int – (1<<7) GAL_CMASS and the entire sparsely sampled region

GAL_LODPERP_DEPRECATED

int – (1<<5) (DEPRECATED) Same as hiz but between dperp00 and dperp0

STD_QSO

int – (1<<22) qso

QSO_FIRST_BOSS

int – (1<<18) FIRST radio match

QSO_KDE_COADD

int – (1<<16) kde targets from the stripe82 coadd

QSO_NN

int – (1<<14) Neural Net that match to sweeps/pass cuts

TEMPLATE_STAR_PHOTO

int – (1<<34) stellar templates

GAL_CMASS

int – (1<<1) dperp > 0.55, color-mag cut

STD_WD

int – (1<<21) white dwarfs

TEMPLATE_QSO_SDSS1

int – (1<<33) QSO templates

QSO_KNOWN_LOHIZ

int – (1<<13) known qso outside of miz range. never target

QSO_CORE_LIKE

int – (1<<43) Likelihood that make it into core

QSO_BONUS

int – (1<<11) permissive qso selection: commissioning only

QSO_UKIDSS

int – (1<<15) UKIDSS stars that match sweeps/pass flag cuts

QSO_KNOWN_MIDZ

int – (1<<12) known qso between [2.2,9.99]

TEMPLATE_GAL_PHOTO

int – (1<<32) galaxy templates

GAL_CMASS_COMM

int – (1<<2) dperp > 0.55, commissioning color-mag cut

QSO_LIKE

int – (1<<17) likelihood method

QSO_KNOWN_SUPPZ

int – (1<<44) known qso between [1.8,2.15]

SDSS_KNOWN

int – (1<<6) Matches a known SDSS spectra

QSO_BONUS_MAIN

int – (1<<41) Main survey bonus sample

TEMPLATE_STAR_SPECTRO

int – (1<<35) stellar templates (spectroscopically known)

class bosssdata.bits.CALIB_STATUS

Calibration status for an SDSS image

UNPHOT_EXTRAP_CLEAR

int – (1<<2) Extrapolate the solution from the clear part of a night (that was ubercalibrated) to the cloudy part

PHOTOMETRIC

int – (1<<0) Photometric observations

PS1_LOW_RMS

int – (1<<14) Low RMS in comparison with PS1

PT_CLOUDY

int – (1<<9) PT calibration for cloudy data

UNPHOT_OVERLAP

int – (1<<1) Unphotometric observations, calibrated based on overlaps with clear, ubercalibrated data; this is done on a field-by-field basis

PS1_CONTRAIL

int – (1<<7) Comparison to PS1 reveals possible contrail

PT_CLEAR

int – (1<<8) PT calibration for clear data

NO_UBERCAL

int – (1<<11) not uber-calibrated

ASTROMBAD

int – (1<<12) catastrophically bad astrometry

PS1_PCOMP_MODEL

int – (1<<13) Enough information for PS1-based principal component flat model

PS1_UNPHOT

int – (1<<6) Comparison to PS1 reveals unphotometric conditions

UNPHOT_DISJOINT

int – (1<<4) Data is disjoint from the rest of the survey (even though conditions may be photometric), the calibration is suspect

UNPHOT_EXTRAP_CLOUDY

int – (1<<3) The solution here is based on fitting the a-term to cloudy data.

DEFAULT

int – (1<<10) a default calibration used

INCREMENT_CALIB

int – (1<<5) Incrementally calibrated by considering overlaps with ubercalibrated data

class bossdata.bits.**EBOSS_TARGET0**

targeting bitmask for SEQUELS (eBOSS precursor)

DO_NOT_OBSERVE

int – (1<<0) Don't put a fiber on this object

SEQUELS_COLLIDED

int – (1<<41) Collided galaxies from BOSS

QSO_EBOSS_FIRST

int – (1<<14) Objects with FIRST radio matches

LRG_RIW

int – (1<<2) LRG selection in r/i/W plan with (i-z) cut

QSO_KNOWN

int – (1<<18) Known QSOs from previous surveys

SPIDERS_ERASS_CLUS

int – (1<<23) ERASS Cluster sources

QSO_EBOSS_KDE

int – (1<<13) KDE-selected QSOs (sequels only)

TDSS_FES_MGII

int – (1<<34) TDSS Few epoch spectroscopy

TDSS_A

int – (1<<30) Main PanSTARRS selection for TDSS

QSO_PTF

int – (1<<11) QSOs with variability in PTF imaging

QSO_EBOSS_CORE

int – (1<<10) QSOs in XDQSOz+WISE selection for clustering

LRG_IZW

int – (1<<1) LRG selection in i/z/W plane

TDSS_FES_DE

int – (1<<31) TDSS Few epoch spectroscopy

SPIDERS_RASS_CLUS

int – (1<<21) RASS Cluster sources

TDSS_FES_DWARFC

int – (1<<32) TDSS Few epoch spectroscopy

DR9_CALIB_TARGET

int – (1<<19) Target found in DR9-calibrated imaging

QSO_SDSS_TARGET

int – (1<<17) Known TARGETS from SDSS with spectra

SEQUELS_PTF_VARIABLE

int – (1<<40) Variability objects from PTF

SPIDERS_ERASS_AGN

int – (1<<22) ERASS AGN sources

QSO_BAD_BOSS

int – (1<<15) QSOs from BOSS with bad spectra

SPIDERS_RASS_AGN

int – (1<<20) RASS AGN sources

TDSS_FES_NQHISN

int – (1<<33) TDSS Few epoch spectroscopy

QSO_REOBS

int – (1<<12) QSOs from BOSS to be reobserved

QSO_BOSS_TARGET

int – (1<<16) Known TARGETS from BOSS with spectra

TDSS_FES_VARBAL

int – (1<<35) TDSS Few epoch spectroscopy

class bossdata.bits.**EBOSS_TARGET1**

targeting bitmask for eBOSS

DO_NOT_OBSERVE

int – (1<<0) Don't put a fiber on this object

TDSS_TARGET

int – (1<<30) Target for TDSS (subclass found in eboss_target2)

QSO1_EBOSS_KDE

int – (1<<13) KDE-selected QSOs (sequels only)

LRG1_IDROP

int – (1<<2) LRG selection in r/i/W plan with (i-z) cut

QSO1_BAD_BOSS

int – (1<<15) QSOs from BOSS with bad spectra

SPIDERS_TARGET

int – (1<<31) Target for SPIDERS (subclass found in eboss_target2)

QSO_SDSS_TARGET

int – (1<<17) Known TARGETS from SDSS with spectra

QSO1_VAR_S82

int – (1<<9) Variability-selected QSOs in the repeated Stripe 82 imaging

QSO1_EBOSS_CORE

int – (1<<10) QSOs in XDQSOz+WISE selection for clustering

QSO_KNOWN

int – (1<<18) Known QSOs from previous surveys

QSO1_REOBS

int – (1<<12) QSOs from BOSS to be reobserved

ELG_TEST1

int – (1<<40) Test targets for ELG selection

QSO1_EBOSS_FIRST

int – (1<<14) Objects with FIRST radio matches

STD_WD

int – (1<<51) white dwarfs

LRG1_WISE

int – (1<<1) LRG selection in i/z/W plane

QSO1_PTF

int – (1<<11) QSOs with variability in PTF imaging

QSO_BOSS_TARGET

int – (1<<16) Known TARGETS from BOSS with spectra

LRG_KNOWN

int – (1<<3) LRG selection in r/i/W plan with (i-z) cut

STD_QSO

int – (1<<52) qso

STD_FSTAR

int – (1<<50) standard f-stars

class bossdata.bits.**EBOSS_TARGET2**

targeting bitmask for eBOSS

SPIDERS_XCLASS_CLUS

int – (1<<5) XMM serendipitous clusters

SPIDERS_XMMSL_AGN

int – (1<<4) XMM Slew survey

TDSS_FES_HYPSTAR

int – (1<<28) TDSS Few epoch spectroscopy

ELG_DESI_TEST1

int – (1<<42) DESI selection for test1 ELG plates

SPIDERS_ERASS_CLUS

int – (1<<3) ERASS Cluster sources

TDSS_B

int – (1<<26) Main TDSS SES version B

TDSS_A

int – (1<<20) Main PanSTARRS selection for TDSS

TDSS_FES_DWARFC

int – (1<<22) TDSS Few epoch spectroscopy

ELG_GRIW_TEST1

int – (1<<46) WISE selection for test1 ELG plates

SPIDERS_ERASS_AGN

int – (1<<2) ERASS AGN sources

SPIDERS_RASS_CLUS

int – (1<<1) RASS Cluster sources

ELG_SDSS_TEST1

int – (1<<43) SDSS-only selection for test1 ELG plates

TDSS_FES_WDDM

int – (1<<29) TDSS Few epoch spectroscopy

ELG_SCUSS_TEST1

int – (1<<40) SCUSS selection for test1 ELG plates

TDSS_FES_MGII

int – (1<<24) TDSS Few epoch spectroscopy

ELG_UGRIZW_TEST1

int – (1<<44) WISE selection for test1 ELG plates

TDSS_FES_DE

int – (1<<21) TDSS Few epoch spectroscopy

ELG_UGRIZWbright_TEST1

int – (1<<45) WISE selection for test1 ELG plates

TDSS_FES_HYPQSO

int – (1<<27) TDSS Few epoch spectroscopy

TDSS_FES_ACTSTAR

int – (1<<30) TDSS Few epoch spectroscopy

SPIDERS_RASS_AGN

int – (1<<0) RASS AGN sources

TDSS_FES_NQHISN

int – (1<<23) TDSS Few epoch spectroscopy

TDSS_CP

int – (1<<31) TDSS in common with CORE/PTF

TDSS_FES_VARBAL

int – (1<<25) TDSS Few epoch spectroscopy

ELG_DES_TEST1

int – (1<<41) DES selection for test1 ELG plates

class bosssdata.bits.FLUXMATCH_STATUS

Flags from flux-based matching to SDSS photometry

PARENT_FLUXMATCH

int – (1<<4) overlapping parent has no children, so used it

FIBER_FLUXMATCH

int – (1<<1) flagged due to fiberflux/aperflux issue

ORIGINAL_FLUXMATCH

int – (1<<0) used the original positional match (which exists)

NONMATCH_FLUXMATCH

int – (1<<2) flagged due to non-match

NOPARENT_FLUXMATCH

int – (1<<3) no overlapping parent in primary field

BRIGHTEST_FLUXMATCH

int – (1<<5) picked the brightest child

class bosssdata.bits.**IMAGE_STATUS**

Sky and instrument conditions of SDSS image

DEAD_CCD

int – (1<<8) CCD bad (unphotometric)

BAD_FOCUS

int – (1<<5) Focus bad (set score=0)

CLOUDY

int – (1<<1) Cloudy skies (unphotometric)

CLEAR

int – (1<<0) Clear skies

SHUTTERS

int – (1<<6) Shutter out of place (set score=0)

UNKNOWN

int – (1<<2) Sky conditions unknown (unphotometric)

BAD_ROTATOR

int – (1<<3) Rotator problems (set score=0)

BAD_ASTROM

int – (1<<4) Astrometry problems (set score=0)

NOISY_CCD

int – (1<<9) CCD noisy (unphotometric)

FF_PETALS

int – (1<<7) Flat-field petals out of place (unphotometric)

class bosssdata.bits.**MANGA_DAPQUAL**

Mask bits for MaNGA DAP quality flags

VALIDFILE

int – (1<<0) File is valid

class bosssdata.bits.**MANGA_DRP2PIXMASK**

Mask bits per fiber or pixel for 2d MaNGA spectra.

SMEARHIGHSN

int – (1<<11) S/N sufficient for full smear fit

SMEARIMAGE

int – (1<<10) Smear available for red and blue cameras

BADARC

int – (1<<3) Bad arc solution

BADTRACE

int – (1<<1) Bad trace

NOPLUG

int – (1<<0) Fiber not listed in plugmap file

LOWFLAT

int – (1<<18) Flat field less than 0.5

SCATTEREDLIGHT

int – (1<<21) Scattered light significant

CROSSTALK

int – (1<<22) Cross-talk significant

PARTIALREJECT

int – (1<<20) Some pixels rejected in extraction model fit

REDMONSTER

int – (1<<29) Contiguous region of bad χ^2 in sky residuals (with threshold of relative $\chi^2 > 3$).

FULLREJECT

int – (1<<19) Pixel fully rejected in extraction model fit (INVVAR=0)

SMEARMEDSN

int – (1<<12) S/N only sufficient for scaled median fit

BADFLAT

int – (1<<2) Low counts in fiberflat

LARGESHIFT

int – (1<<6) Large spatial shift between flat and object position

BRIGHTSKY

int – (1<<24) Sky level > flux + 10*(flux_err) AND sky > 1.25 * median(sky,99 pixels)

BADSKYCHI

int – (1<<28) Relative $\chi^2 > 3$ in sky residuals at this wavelength

NODATA

int – (1<<25) No data available in combine B-spline (INVVAR=0)

BADFLUXFACTOR

int – (1<<27) Low flux-calibration or flux-correction factor

NOSKY

int – (1<<23) Sky level unknown at this wavelength (INVVAR=0)

BADPIX

int – (1<<15) Pixel flagged in badpix reference file.

NEARBADPIXEL

int – (1<<17) Bad pixel within 3 pixels of trace.

COSMIC

int – (1<<16) Pixel flagged as cosmic ray.

DEADFIBER

int – (1<<13) Broken fiber according to metrology files

MANYBADCOLUMNS

int – (1<<4) More than 10% of pixels are bad columns

NEARWHOPPER

int – (1<<8) Within 2 fibers of a whopping fiber (exclusive)

COMBINEREJ

int – (1<<26) Rejected in combine B-spline

MANYREJECTED

int – (1<<5) More than 10% of pixels are rejected in extraction

_3DREJECT

int – (1<<30) Used in RSS file, indicates should be rejected when making 3D cube

WHOPPER

int – (1<<9) Whopping fiber, with a very bright source.

BADSKYFIBER

int – (1<<7) Sky fiber shows extreme residuals

class bossdata.bits.**MANGA_DRP2QUAL**

Mask bits for MaNGA DRP-2d quality flags

SKYSUBBAD

int – (1<<10) Bad sky subtraction

BADIFU

int – (1<<4) One or more IFUs missing/bad in this frame

RAMPAGINGBUNNY

int – (1<<9) Rampaging dust bunnies in IFU flats

LOWEXPTIME

int – (1<<3) Exposure time less than 10 minutes

EXTRACTBRIGHT

int – (1<<2) Extracted spectra abnormally bright

ARCFOCUS

int – (1<<8) Bad focus on arc frames

EXTRACTBAD

int – (1<<1) Many bad values in extracted frame

BADDITHER

int – (1<<7) Bad dither location information

SKYSUBFAIL

int – (1<<11) Failed sky subtraction

FULLCLOUD

int – (1<<12) Completely cloudy exposure

VALIDFILE

int – (1<<0) File is valid

SCATFAIL

int – (1<<6) Failure to correct high scattered light levels

HIGHSCAT

int – (1<<5) High scattered light levels

class bossdata.bits.**MANGA_DRP3PIXMASK**

Mask bits per spaxel for a MaNGA data cube.

LOWCOV

int – (1<<1) Low coverage depth in cube

NOCOV

int – (1<<0) No coverage in cube

FORESTAR

int – (1<<3) Foreground star

DEADFIBER

int – (1<<2) Major contributing fiber is dead

DONOTUSE

int – (1<<10) Do not use this spaxel for science

class bossdata.bits.**MANGA_DRP3QUAL**

Mask bits for MaNGA DRP-3d quality flags

SKYSUBBAD

int – (1<<2) Bad sky subtraction in one or more frames

BADDEPTH

int – (1<<1) IFU does not reach target depth

VARIABLELSF

int – (1<<5) LSF varies signif. between component spectra

BADOMEGA

int – (1<<6) Omega greater than threshold in one or more sets

CRITICAL

int – (1<<30) Critical failure in one or more frames

BADASTROM

int – (1<<4) Bad astrometry in one or more frames

BADSET

int – (1<<7) One or more sets are bad

VALIDFILE

int – (1<<0) File is valid

HIGHSCAT

int – (1<<3) High scattered light in one or more frames

BADFLUX

int – (1<<8) Bad flux calibration

class bossdata.bits.**MANGA_TARGET1**

Mask bits identifying galaxy samples.

SECONDARY_COM

int – (1<<2) March 2014 commissioning

COLOR_ENHANCED_v1_1_0

int – (1<<6) First tag, August 2014 plates

COLOR_ENHANCED_COM

int – (1<<3) March 2014 commissioning

ANCILLARY

int – (1<<14) Ancillary program targets

PRIMARY_COM2

int – (1<<7) July 2014 commissioning

PRIMARY_PLUS_COM

int – (1<<1) March 2014 commissioning

SECONDARY_v1_1_0

int – (1<<5) First tag, August 2014 plates

SECONDARY_v1_2_0

int – (1<<11)

PRIMARY_v1_2_0

int – (1<<10)

SECONDARY_COM2

int – (1<<8) July 2014 commissioning

PRIMARY_v1_1_0

int – (1<<4) First tag, August 2014 plates

COLOR_ENHANCED_v1_2_0

int – (1<<12)

FILLER

int – (1<<13) Filler targets

NONE

int – (1<<0)

COLOR_ENHANCED_COM2

int – (1<<9) July 2014 commissioning

class bosssdata.bits.**MANGA_TARGET2**

Mask bits identifying non-galaxy samples.

STELLIB_KNOWN_COM

int – (1<<4) Commissioning selection of known parameter stars

STD_APASS_COM

int – (1<<25) Commissioning selection of stds using APASS photometry

STD_STD_COM

int – (1<<22)

STELLIB_2MASS_COM

int – (1<<3) Commissioning selection using 2MASS photometry

SKY

int – (1<<1)

STELLIB_COM_mar2015

int – (1<<5) Commissioning selection in March 2015

STD_WD

int – (1<<24)

STELLIB_SDSS_COM

int – (1<<2) Commissioning selection using SDSS photometry

STD_WD_COM

int – (1<<21)

STD_FSTAR_COM

int – (1<<20)

STD_FSTAR

int – (1<<23)

NONE

int – (1<<0)

class bosssdata.bits.**MANGA_TARGET3**

Mask bits identifying ancillary samples.

PAIR_RECENTER

int – (1<<8)

DEEP_COMA

int – (1<<19)

VOID

int – (1<<5)

EDGE_ON_WINDS

int – (1<<6)

LETTERS

int – (1<<11)

AGN_PALOMAR

int – (1<<4)

AGN_OIII

int – (1<<2)

DWARF

int – (1<<14)

PAIR_2IFU

int – (1<<10)

RADIO_JETS

int – (1<<15)

PAIR_ENLARGE

int – (1<<7)

ANGST

int – (1<<18)

AGN_BAT

int – (1<<1)

MWA

int – (1<<13)

BCG

int – (1<<17)

DISKMASS
int – (1<<16)

MASSIVE
int – (1<<12)

PAIR_SIM
int – (1<<9)

AGN_WISE
int – (1<<3)

NONE
int – (1<<0)

class bosssdata.bits.**M_EYEBALL**

Eyeball flags for mergers in VAGC

DONE
int – (1<<0)

QUESTIONABLE
int – (1<<2)

MIXED_REDBLUE
int – (1<<11)

RING
int – (1<<6)

DURING
int – (1<<14)

BEFORE
int – (1<<13)

REPEAT
int – (1<<12)

TIDAL_TAILS
int – (1<<4)

NOT_MERGER
int – (1<<1)

ALL_RED
int – (1<<9)

ALL_BLUE
int – (1<<10)

MAJOR
int – (1<<7)

AFTER
int – (1<<15)

SHELLS
int – (1<<5)

DRY
int – (1<<3)

MULTIPLE

int – (1<<8)

class bosssdata.bits.OBJECT1

Object flags from photo reductions for SDSS (first 32)

MOVED

int – (1<<31) The object appears to have moved during the exposure. Such objects are candidates to be deblended as moving objects.

NODEBLEND

int – (1<<6) Although this object was marked as a blend, no deblending was attempted.

SATUR

int – (1<<18) The object contains saturated pixels; INTERP is also set.

BINNED4

int – (1<<30) The object was detected in a 4x4 binned image. The objects detected in the 2x2 binned image are not removed before doing this.

CHILD

int – (1<<4) Object is a child, created by the deblender.

BADSKY

int – (1<<22) The estimated sky level is so bad that the central value of the radial profile is crazily negative; this is usually the result of the subtraction of the wings of bright stars failing.

MANYR50

int – (1<<13) More than one radius was found to contain 50% of the Petrosian flux. (For this to happen part of the radial profile must be negative).

BRIGHT

int – (1<<1) Indicates that the object was detected as a bright object. Since these are typically remeasured as faint objects, most users can ignore BRIGHT objects.

CR

int – (1<<12) Object contains at least one pixel which was contaminated by a cosmic ray. The OBJECT1_INTERP flag is also set. This flag does not mean that this object is a cosmic ray; rather it means that a cosmic ray has been removed.

MANYPETRO

int – (1<<9) Object has more than one possible Petrosian radius.

BINNED1

int – (1<<28) The object was detected in an unbinned image.

DEBLENDED_AS_PSF

int – (1<<25) When deblending an object, in this band this child was treated as a PSF.

ELLIPFAINT

int – (1<<27) No isophotal fits were performed.

PEAKCENTER

int – (1<<5) Given center is position of peak pixel, as attempts to determine a better centroid failed.

BINNED2

int – (1<<29) The object was detected in a 2x2 binned image after all unbinned detections have been replaced by the background level.

EDGE

int – (1<<2) Object is too close to edge of frame in this band.

NOPROFILE

int – (1<<7) Frames couldn't extract a radial profile.

SUBTRACTED

int – (1<<20) Object (presumably a star) had wings subtracted.

NOPETRO_BIG

int – (1<<10) The Petrosian ratio has not fallen to the value at which the Petrosian radius is defined at the outermost point of the extracted radial profile. NOPETRO is set, and the Petrosian radius is set to the outermost point in the profile.

NOSTOKES

int – (1<<21) Object has no measured Stokes parameters.

NOPETRO

int – (1<<8) No Petrosian radius or other Petrosian quantities could be measured.

MANYR90

int – (1<<14) More than one radius was found to contain 90% of the Petrosian flux. (For this to happen part of the radial profile must be negative).

CANONICAL_CENTER

int – (1<<0) The quantities (psf counts, model fits and likelihoods) that are usually determined at an object's center as determined band-by-band were in fact determined at the canonical center (suitably transformed). This is due to the object being too close to the edge to extract a profile at the local center, and OBJECT1_EDGE is also set.

BLENDED

int – (1<<3) Object was determined to be a blend. The flag is set if: more than one peak is detected within an object in a single band together; distinct peaks are found when merging different colours of one object together; or distinct peaks result when merging different objects together.

PETROFAINT

int – (1<<23) At least one candidate Petrosian radius occurred at an unacceptably low surface brightness.

INTERP

int – (1<<17) The object contains interpolated pixels (e.g. cosmic rays or bad columns).

BAD_RADIAL

int – (1<<15) Measured profile includes points with a S/N ≤ 0 . In practice this flag is essentially meaningless.

TOO_LARGE

int – (1<<24) The object is (as it says) too large. Either the object is still detectable at the outermost point of the extracted radial profile (a radius of approximately 260 arcsec), or when attempting to deblend an object, at least one child is larger than half a frame (in either row or column).

DEBLEND_TOO_MANY_PEAKS

int – (1<<11) The object had the OBJECT1_DEBLEND flag set, but it contained too many candidate children to be fully deblended. This flag is only set in the parent, i.e. the object with too many peaks.

INCOMPLETE_PROFILE

int – (1<<16) A circle, centered on the object, of radius the canonical Petrosian radius extends beyond the edge of the frame. The radial profile is still measured from those parts of the object that do lie on the frame.

DEBLEND_PRUNED

int – (1<<26) When solving for the weights to be assigned to each child the deblender encountered a nearly singular matrix, and therefore deleted at least one of them.

NOTCHECKED

int – (1<<19) Object includes pixels that were not checked for peaks, for example the unsmoothed edges of frames, and the cores of subtracted or saturated stars.

class bossdata.bits.OBJECT2

Object flags from photo reductions for SDSS (second 32)

STATIONARY

int – (1<<4) A moving objects velocity is consistent with zero

INTERP_CENTER

int – (1<<12) An object's center is very close to at least one interpolated pixel.

BAD_MOVING_FIT

int – (1<<3) The fit to the object as a moving object is too bad to be believed.

TOO_FEW_DETECTIONS

int – (1<<2) The object has the MOVED flag set, but has too few detection to be deblended as moving.

MAYBE_EGHOST

int – (1<<25) Object appears in the right place to be an electronics ghost.

BAD_MOVING_FIT_CHILD

int – (1<<9) A putative moving child's velocity fit was too poor, so it was discarded, and the parent was not deblended as moving

SATUR_CENTER

int – (1<<11) An object's center is very close to at least one saturated pixel; the object may well be causing the saturation.

MAYBE_CR

int – (1<<24) This object may be a cosmic ray. This bit can get set in the cores of bright stars, and is quite likely to be set for the cores of saturated stars.

PEAKS_TOO_CLOSE

int – (1<<5) Peaks in object were too close (set only in parent objects).

DEBLENDED_AS_MOVING

int – (1<<0) The object has the MOVED flag set, and was deblended on the assumption that it was moving.

TOO_FEW_GOOD_DETECTIONS

int – (1<<16) A child of this object had too few good detections to be deblended as moving.

CANONICAL_BAND

int – (1<<20) This band was the canonical band. This is the band used to measure the Petrosian radius used to calculate the Petrosian counts in each band, and to define the model used to calculate model colors; it has no effect upon the coordinate system used for the OBJC center.

DEBLENDED_AT_EDGE

int – (1<<13) An object so close to the edge of the frame that it would not ordinarily be deblended has been deblended anyway. Only set for objects large enough to be EDGE in all fields/strips.

BAD_COUNTS_ERROR

int – (1<<8) An object containing interpolated pixels had too few good pixels to form a reliable estimate of its error

AMOMENT_SHIFT

int – (1<<22) Object's center moved too far while determining adaptive moments. In this case, the M_e1 and M_e2 give the (row, column) shift, not the object's shape.

CENTER_OFF_AIMAGE

int – (1<<17) At least one peak’s center lay off the atlas image in some band. This can happen when the object’s being deblended as moving, or if the astrometry is badly confused.

BINNED_CENTER

int – (1<<6) When centroiding the object the object’s size is larger than the (PSF) filter used to smooth the image.

BRIGHTEST_GALAXY_CHILD

int – (1<<19) This is the brightest child galaxy in a blend.

LOCAL_EDGE

int – (1<<7) The object’s center in some band was too close to the edge of the frame to extract a profile.

NOTCHECKED_CENTER

int – (1<<26) Center of object lies in a NOTCHECKED region. The object is almost certainly bogus.

AMOMENT_MAXITER

int – (1<<23) Too many iterations while determining adaptive moments.

DEBLEND_PEEPHOLE

int – (1<<28) The deblend was modified by the optimizer

SPARE3

int – (1<<29)

SPARE1

int – (1<<31)

DEBLEND_NOPEAK

int – (1<<14) A child had no detected peak in a given band, but we centroided it anyway and set the BINNED1

PSF_FLUX_INTERP

int – (1<<15) The fraction of light actually detected (as opposed to guessed at by the interpolator) was less than some number (currently 80%) of the total.

DEBLEND_DEGENERATE

int – (1<<18) At least one potential child has been pruned because its template was too similar to some other child’s template.

SPARE2

int – (1<<30)

HAS_SATUR_DN

int – (1<<27) This object is saturated in this band and the bleed trail doesn’t touch the edge of the frame, we we’ve made an attempt to add up all the flux in the bleed trails, and to include it in the object’s photometry.

NODEBLEND_MOVING

int – (1<<1) The object has the MOVED flag set, but was not deblended as a moving object.

DEBLEND_UNASSIGNED_FLUX

int – (1<<10) After deblending, the fraction of flux assigned to none of the children was too large (this flux is then shared out as described elsewhere).

AMOMENT_UNWEIGHTED

int – (1<<21) ‘Adaptive’ moments are actually unweighted.

class bosssdata.bits.Q_EYEBALL

Quality eyeball flags from VAGC

PLANETARY_NEBULA

int – (1<<24)

DOUBLE_STAR
int – (1<<6)

DONE
int – (1<<0)

IN_HUGE_OBJECT
int – (1<<16)

USE_CHILD_IMAGE
int – (1<<27)

BAD_DEBLEND
int – (1<<4)

IS_STAR
int – (1<<22)

NEGATIVE_QSO_FIT
int – (1<<19)

OTHER
int – (1<<1)

POSSIBLE_LENS
int – (1<<21)

BAD_Z
int – (1<<12)

STAR_ON_GALAXY
int – (1<<17)

USE_PARENT
int – (1<<15)

NEED_BIGGER_IMAGE
int – (1<<3)

BAD_SPECTRUM
int – (1<<20)

HII
int – (1<<7)

SATELLITE
int – (1<<10)

FLECK
int – (1<<5)

EDGE
int – (1<<9)

INTERNAL_REFLECTION
int – (1<<13)

UNCLASSIFIABLE
int – (1<<2)

PLANE
int – (1<<11)

BAD_SPEC_CLASS*int* – (1<<14)**GOOD_Z***int* – (1<<26)**QSO_ON_GALAXY***int* – (1<<18)**USE_CHILD_SPECTRUM***int* – (1<<28)**DOUBLE_Z***int* – (1<<23)**BAD_PARENT_CENTER***int* – (1<<25)**USE_ANYWAY***int* – (1<<8)**class bossdata.bits.RESOLVE_STATUS**

Resolve status for an SDSS catalog entry. Only one of bits RUN_PRIMARY, RUN_RAMP, RUN_OVERLAPONLY, RUN_IGNORE, and RUN_DUPLICATE can be set. RUN_EDGE can be set for any object. To get a unique set of objects across the whole survey, search for objects with SURVEY_PRIMARY set. To get a unique set of objects within a run, search for objects with RUN_PRIMARY set.

SURVEY_BADFIELD*int* – (1<<11) In field with score=0**RUN_OVERLAPONLY***int* – (1<<2) only appears in the overlap between two fields**RUN_DUPLICATE***int* – (1<<5) duplicate measurement of same pixels in two different fields**SURVEY_BEST***int* – (1<<9) Best observation within the full survey, but it does not appear in the primary observation of this part of the sky**SURVEY_SECONDARY***int* – (1<<10) Repeat (independent) observation of an object that has a different primary or best observation**SURVEY_EDGE***int* – (1<<12) Not kept as secondary because it is RUN_RAMP or RUN_EDGE object**SURVEY_PRIMARY***int* – (1<<8) Primary observation within the full survey, where it appears in the primary observation of this part of the sky**RUN_PRIMARY***int* – (1<<0) primary within the objects own run (but not necessarily for the survey as a whole)**RUN_EDGE***int* – (1<<4) near lowest or highest column**RUN_IGNORE***int* – (1<<3) bright or parent object that should be ignored**RUN_RAMP***int* – (1<<1) in what would be the overlap area of a field, but with no neighboring field

```
class bosssdata.bits.SEGUE1_TARGET
    SEGUE-1 primary target bits

    SEG1LOW_AGB
        int – (1<<27) low latitude selection of AGB stars

    SEGUE1_KG
        int – (1<<14) K-giants (l and red)

    SEG1LOW_KG
        int – (1<<10) low latitude selection of K-giant stars

    SEGUE1_BD
        int – (1<<21) brown dwarfs

    SEGUE1_MAN
        int – (1<<24) manual selection

    SEGUE1_MSWD
        int – (1<<12) main-sequence, white dwarf pair

    SEGUE1_SDM
        int – (1<<22) M sub-dwarfs

    SEGUE1_CHECKED
        int – (1<<31) was a checked object

    SEGUE1_WD
        int – (1<<19) white dwarf

    SEG1LOW_TO
        int – (1<<11) low latitude selection of bluetip stars

    SEGUE1_FG
        int – (1<<9) F and G stars, based on g-r color (0.2<g-r<0.48 and 14<g<20.2)

    SEGUE1_MPMSTO
        int – (1<<20) metal-poor main sequence turn-off

    SEGUE1_BHB
        int – (1<<13) blue horizontal branch star

    SEGUE1_GD
        int – (1<<18) G-dwarf

    SEGUE1_AGB
        int – (1<<23) asympototic giant branch stars

    SEGUE1_KD
        int – (1<<15) K-dwarfs

    SEGUE1_LM
        int – (1<<16) low metallicity star

    SEGUE1_CWD
        int – (1<<17) cool white dwarf

class bosssdata.bits.SEGUE1_TARGET2
    SEGUE-1 secondary target bits

    SEGUE1_QA
        int – (1<<3) QA Duplicate Observations (unused)
```

REDDEN_STD

int – (1<<1) reddening standard star

SPECTROPHOTO_STD

int – (1<<5) spectrophotometry standard (typically an F-star)

SEGUE1_TEST

int – (1<<31) SEGUE-1 test target

SEGUE1_SCIENCE

int – (1<<30) SEGUE-1 science target

SKY

int – (1<<4) sky target

class bosssdata.bits.SEGUE2_TARGET1

SEGUE-2 primary target bits

SEGUE2_HVS

int – (1<<5) hyper velocity candidate

SEGUE2_MII

int – (1<<7) M giant

SEGUE2_HHV

int – (1<<8) High-velocity halo star candidate

SEGUE2_CHECKED

int – (1<<31) was a checked object

SEGUE2_REDKG

int – (1<<1) Red K-giant stars

SEGUE2_BHB

int – (1<<13) Blue horizontal branch star

SEGUE2_LKG

int – (1<<2) K-giant star identified by l-color

SEGUE2_MSTO

int – (1<<0) Main-sequence turnoff

SEGUE2_PMKG

int – (1<<3) K-giant star identified by proper motions

SEGUE2_LM

int – (1<<4) Low metallicity

SEGUE2_CWD

int – (1<<17) Cool white dwarf

SEGUE2_XDM

int – (1<<6) extreme sdM star

class bosssdata.bits.SEGUE2_TARGET2

SEGUE-2 secondary target bits

QUALITY_HOLE

int – (1<<8) quality hole

SEGUE2_TEST

int – (1<<2) test target

LIGHT_TRAP

int – (1<<0) light trap hole

SEGUE2_CLUSTER

int – (1<<10) SEGUE-2 stellar cluster target

SKY

int – (1<<4) empty area for sky-subtraction

SEGUE2_CHECKED

int – (1<<31) was a checked object

SEGUE2_REDDENING

int – (1<<1) reddening standard

SEGUE2_QA

int – (1<<3) repeat target across plates

SEGUE2_SPECPHOTO

int – (1<<5) spectrophotometric star

HOT_STD

int – (1<<9) hot standard

GUIDE_STAR

int – (1<<6) guide star

BUNDLE_HOLE

int – (1<<7) bundle hole

SEGUE2_STETSON

int – (1<<11) Stetson standard target

class bossdata.bits.**SPECIAL_TARGET1**

SDSS special program target bits

PREMARVELS

int – (1<<19) pre-MARVELS stellar target

HIPM

int – (1<<34) high proper motion

DISKSTAR

int – (1<<4) thin/thick disk star

FAINT_QSO

int – (1<<26) faint QSO in south

LOWZ_LOVEDAY

int – (1<<38) low redshift galaxy selected by Loveday

APBIAS

int – (1<<0) aperture bias target

SOUTHERN_EXTENDED

int – (1<<20) simple extension of southern targets

FAINT_LRG

int – (1<<25) faint LRG in south

ALLPSF

int – (1<<31) i<19.1 point sources

COMMISSIONING_STAR

int – (1<<3) star in commissioning

STRAIGHT_RADIO

int – (1<<28) straight double-lobed radio source

FSTAR

int – (1<<5) F-stars

ORION_MSTAR_EARLY

int – (1<<13) Early-type M-star (M0-3) in Orion

PREBOSS_QSO

int – (1<<17) QSO for pre-BOSS observations

SPECIAL_FILLER

int – (1<<15) filler from completeTile, check primtarget for details

BENT_RADIO

int – (1<<27) bent double-lobed radio source

SOUTHERN_COMPLETE

int – (1<<21) completion in south of main targets

U_EXTRA2

int – (1<<24) extra u-band target

PHOTOZ_GALAXY

int – (1<<16) test galaxy for photometric redshifts

TAURUS_GALAXY

int – (1<<36) galaxy on taurus or reddening plate

VARIABLE_HIPRI

int – (1<<29) high priority variable

QSO_M31

int – (1<<2) QSO in M31

DEEP_GALAXY_RED_II

int – (1<<9) deep LRG

VARIABLE_LOPRI

int – (1<<30) low priority variable

ORION_MSTAR_LATE

int – (1<<14) Late-type M-star (M4-) in Orion

HYADES_MSTAR

int – (1<<6) M-star in Hyades

DEEP_GALAXY_RED

int – (1<<8) deep LRG

LOWZ_ANNIS

int – (1<<1) low-redshift cluster galaxy

MSTURNOFF

int – (1<<11) main sequence turnoff

BCG

int – (1<<10) brightest cluster galaxy

TAURUS_STAR

int – (1<<35) star on taurus or reddening plate

U_EXTRA

int – (1<<23) extra u-band target

U_PRIORITY

int – (1<<22) priority u-band target

ALLPSF_STELLAR

int – (1<<33) i<19.1 point sources on stellar locus

LOWZ_GALAXY

int – (1<<7) low-redshift galaxy

PERSEUS

int – (1<<37) galaxy in perseus-pisces

ORION_BD

int – (1<<12) Brown dwarf in Orion

PREBOSS_LRG

int – (1<<18) QSO for pre-BOSS observations

ALLPSF_NONSTELLAR

int – (1<<32) i<19.1 point sources off stellar locus

class bosssdata.bits.SPPIXMASK

Mask bits for an SDSS spectrum. 0-15 refer to each fiber, 16-31 refer to each pixel in a spectrum.

SMEARHIGHSN

int – (1<<11) S/N sufficient for full smear fit

SMEARIMAGE

int – (1<<10) Smear available for red and blue cameras

NOSKY

int – (1<<22) Sky level unknown at this wavelength (INVVAR=0)

BADTRACE

int – (1<<1) Bad trace from routine TRACE320CRUDE

NOPLUG

int – (1<<0) Fiber not listed in plugmap file

LOWFLAT

int – (1<<17) Flat field less than 0.5

SCATTEREDLIGHT

int – (1<<20) Scattered light significant

CROSSTALK

int – (1<<21) Cross-talk significant

PARTIALREJECT

int – (1<<19) Some pixels rejected in extraction

REDMONSTER

int – (1<<28) Contiguous region of bad χ^2 in sky residuals (with threshold of relative $\chi^2 > 3$).

FULLREJECT

int – (1<<18) Pixel fully rejected in extraction (INVVAR=0)

SMEARMEDSN

int – (1<<12) S/N only sufficient for scaled median fit

BADFLAT

int – (1<<2) Low counts in fiberflat

LARGESHIFT

int – (1<<6) Large spatial shift between flat and object position

BRIGHTSKY

int – (1<<23) Sky level > flux + 10*(flux_err) AND sky > 1.25 * median(sky,99 pixels)

BADSKYCHI

int – (1<<27) Relative $\chi^2 > 3$ in sky residuals at this wavelength

NODATA

int – (1<<24) No data available in combine B-spline (INVVAR=0)

BADFLUXFACTOR

int – (1<<26) Low flux-calibration or flux-correction factor

BADARC

int – (1<<3) Bad arc solution

NEARBADPIXEL

int – (1<<16) Bad pixel within 3 pixels of trace.

MANYBADCOLUMNS

int – (1<<4) More than 10% of pixels are bad columns

NEARWHOPPER

int – (1<<8) Within 2 fibers of a whopping fiber (exclusive)

COMBINEREJ

int – (1<<25) Rejected in combine B-spline

MANYREJECTED

int – (1<<5) More than 10% of pixels are rejected in extraction

WHOPPER

int – (1<<9) Whopping fiber, with a very bright source.

BADSKYFIBER

int – (1<<7) Sky fiber shows extreme residuals

class bosssdata.bits.**TARGET**

Primary target mask bits in SDSS-I, -II (for LEGACY_TARGET1 or PRIMTARGET).

SERENDIP_FIRST

int – (1<<21) coincident with FIRST sources but fainter than the equivalent in quasar target selection (also includes non-PSF sources)

SERENDIP_DISTANT

int – (1<<23) lying outside the stellar locus in color space

ROSAT_E

int – (1<<27) ROSAT All-Sky Survey match, but too faint or too bright for SDSS spectroscopy

SOUTHERN_SURVEY

int – (1<<31) Set in primtarget if this is a special program target

STAR_BROWN_DWARF

int – (1<<15) brown dwarfs (note this sample is tiled)

QSO_MAG_OUTLIER

int – (1<<25) Stellar outlier; too faint or too bright to be targeted

QSO_REJECT

int – (1<<29) Object in explicitly excluded region of color space, therefore not targeted at QSO

QSO_FIRST_CAP

int – (1<<3) FIRST source with stellar colors at high Galactic latitude

GALAXY_RED

int – (1<<5) Luminous Red Galaxy target (any criteria)

STAR_WHITE_DWARF

int – (1<<19) hot white dwarfs

STAR_PN

int – (1<<28) central stars of planetary nebulae

ROSAT_D

int – (1<<12) ROSAT All-Sky Survey match, are otherwise bright enough for SDSS spectroscopy

STAR_CARBON

int – (1<<14) dwarf and giant carbon stars

GALAXY_BRIGHT_CORE

int – (1<<8) Galaxy targets who fail all the surface brightness selection limits but have r-band fiber magnitudes brighter than 19

SERENDIP_BLUE

int – (1<<20) lying outside the stellar locus in color space

QSO_SKIRT

int – (1<<2) ugri-selected quasar at low Galactic latitude

GALAXY

int – (1<<6) Main sample galaxy

SERENDIP_RED

int – (1<<22) lying outside the stellar locus in color space

GALAXY_BIG

int – (1<<7) Low-surface brightness main sample galaxy ($\mu_{50} > 23$ in r-band)

SERENDIP_MANUAL

int – (1<<24) manual serendipity flag

STAR_CATY_VAR

int – (1<<17) cataclysmic variables

ROSAT_C

int – (1<<11) ROSAT All-Sky Survey match, fall in a broad intermediate category that includes stars that are bright, moderately blue, or both

GALAXY_RED_II

int – (1<<26) Luminous Red Galaxy target (Cut II criteria)

STAR_BHB

int – (1<<13) blue horizontal-branch stars

ROSAT_B

int – (1<<10) ROSAT All-Sky Survey match, have SDSS colors of AGNs or quasars

QSO_FIRST_SKIRT

int – (1<<4) FIRST source with stellar colors at low Galactic latitude

QSO_HIZ

int – (1<<0) High-redshift (griz) QSO target

STAR_SUB_DWARF

int – (1<<16) low-luminosity subdwarfs

QSO_CAP

int – (1<<1) ugri-selected quasar at high Galactic latitude

ROSAT_A

int – (1<<9) ROSAT All-Sky Survey match, also a radio source

STAR_RED_DWARF

int – (1<<18) red dwarfs

class bossdata.bits.**TTARGET**

Secondary target mask bits in SDSS-I, -II (for LEGACY_TARGET2, SPECIAL_TARGET2 or SECTARGET).

REDDEN_STD

int – (1<<1) reddening standard star

SPECTROPHOTO_STD

int – (1<<5) spectrophotometry standard (typically an F-star)

QUALITY_HOLE

int – (1<<8) hole drilled for plate shop quality measurements

BUNDLE_HOLE

int – (1<<7) fiber bundle hole

LIGHT_TRAP

int – (1<<0) hole drilled for bright star, to avoid scattered light

HOT_STD

int – (1<<9) hot standard star

SOUTHERN_SURVEY

int – (1<<31) a segue or southern survey target

GUIDE_STAR

int – (1<<6) guide star hole

SKY

int – (1<<4) sky target

TEST_TARGET

int – (1<<2) a test target

QA

int – (1<<3) quality assurance target

class bossdata.bits.**T_EYEBALL**

Type eyeball flags from VAGC

PITCH_2

int – (1<<10)

DUST_LANE

int – (1<<17)

DUST_ASYMMETRY

int – (1<<24)

DONE
int – (1<<0)

PITCH_0
int – (1<<8) tightly wound

RING
int – (1<<19)

SPIRAL_STRUCTURE
int – (1<<16)

ELLIPTICAL
int – (1<<3)

TIDAL_TAILS
int – (1<<20)

OTHER
int – (1<<1)

DISK
int – (1<<4)

MERGER
int – (1<<26)

SHELLS
int – (1<<21)

PITCH_3
int – (1<<11)

PITCH_1
int – (1<<9)

UNUSED_0
int – (1<<6)

OUTFLOW
int – (1<<27)

BLUE_CORE
int – (1<<22)

NEAR_NEIGHBORS
int – (1<<25)

PITCH_4
int – (1<<12) openly wound

UNCLASSIFIABLE
int – (1<<2)

ASYMMETRIC
int – (1<<14)

S0
int – (1<<7)

IRREGULAR
int – (1<<5)

HII_REGIONS
int – (1<<15)

BAR
int – (1<<18)

PSF
int – (1<<13)

WARPED_DISK
int – (1<<23)

class bossdata.bits.**VAGC_SELECT**
Selection flags for Main VAGC sample

PLATEHOLE
int – (1<<1) selected because near a hole on an SDSS plate

TILED
int – (1<<0) selected because near a tiled target

MAIN
int – (1<<2) selected according to slightly adjusted Main sample criteria

class bossdata.bits.**ZWARNING**
Warnings for SDSS spectra.

UNPLUGGED
int – (1<<7) the fiber was unplugged, so no spectrum obtained

NODATA
int – (1<<9) No data for this fiber, e.g. because spectrograph was broken during this exposure (ivar=0 for all pixels)

MANY_OUTLIERS
int – (1<<4) fraction of points more than 5 sigma away from best model is too large (>0.05)

NEGATIVE_MODEL
int – (1<<3) synthetic spectrum is negative (only set for stars and QSOs)

SMALL_DELTA_CHI2
int – (1<<2) chi-squared of best fit is too close to that of second best (<0.01 in reduced chi-squared)

NEGATIVE_EMISSION
int – (1<<6) a QSO line exhibits negative emission, triggered only in QSO spectra, if C_IV, C_III, Mg_II, H_beta, or H_alpha has LINEAREA + 3 * LINEAREA_ERR < 0

LITTLE_COVERAGE
int – (1<<1) too little wavelength coverage (WCOVERAGE < 0.18)

SKY
int – (1<<0) sky fiber

Z_FITLIMIT
int – (1<<5) chi-squared minimum at edge of the redshift fitting range (Z_ERR set to -1)

BAD_TARGET
int – (1<<8) catastrophically bad targeting data (e.g. ASTROMBAD in CALIB_STATUS)

bossdata.bits.**bitmask_from_text** (*mask*, *text*)
Initialize a bitmask from text.

Builds an integer value from text containing bit names that should be set. The complement of `decode_bitmask()`. For example:

```
>>> COLORS = define_bitmask('COLORS', 'Primary colors', RED=0, BLUE=1, GREEN=4)
>>> '{0:b}'.format(bitmask_from_text(COLORS, 'GREEN|BLUE'))
'10010'
```

Parameters

- **mask** – A bitmask type, normally created with `create_bitmask()`, that defines the symbolic bit names that are allowed.
- **text** – A list of bit names separated by '|'.

Returns Integer with bits set for each bit name appearing in the text.

Return type int

Raises `ValueError` – invalid text specification.

`bosssdata.bits.decode_bitmask(mask, value, strict=True)`

Decode a integer value into its symbolic bit names.

Use this function to convert a bitmask value into a list of symbolic bit names, for example:

```
>>> COLORS = define_bitmask('COLORS', 'Primary colors', RED=0, BLUE=1, GREEN=4)
>>> decode_bitmask(COLORS, COLORS.RED | COLORS.BLUE)
('RED', 'BLUE')
```

For pretty printing, try:

```
>>> print(' | '.join(decode_bitmask(COLORS, COLORS.RED | COLORS.BLUE)))
RED | BLUE
```

Parameters

- **mask** – A bitmask type, normally created with `create_bitmask()`, that defines the symbolic bit names to use for the decoding.
- **value** (*int*) – The integral value to decode.
- **strict** (*bool*) – If set, then all bits set in value must be defined in the bitmask type definition.

Returns

A tuple of symbolic bit names in order of increasing bit offset. If **strict** is `False`, then any bits without corresponding symbolic names will appear as `'1<n'` for offset `n`.

Return type tuple

Raises

- `AttributeError` – mask does not have the attributes necessary to define a bitmask.
- `ValueError` – value has a bit set that has no symbolic name defined and **strict** is `True`.

`bosssdata.bits.define_bitmask(mask_name, mask_description, **bits)`

Define a new type for a bitmask with specified symbolic bit names.

After defining a bitmask type, its bit names are accessible as class-level attributes of the returned type and can be used as integer values, for example:

```
>>> COLORS = define_bitmask('COLORS', 'Primary Colors', RED=0, BLUE=1, GREEN=4)
>>> COLORS.BLUE
2
>>> '{0:b}'.format(COLORS.RED | COLORS.GREEN)
'10001'
```

The `decode_bitmask()` function is useful for converting an integral value back to a list of symbolic bit names.

Parameters

- **mask_name** (*str*) – The type name for this mask. By convention, this name is upper case and matches the name assigned to this function’s return value, as in the examples above.
- **mask_description** (*str*) – A description of this bit mask that will be available as the docstring of the new defined type.
- **bits** (*dict*) – A dictionary of name,definition pairs that define the mapping from symbolic bit names to bit offsets and optional comments. Although this argument can be passed as a dictionary, the dictionary is usually implicitly defined by the argument list, as in the examples above. By convention, bit names are all upper case. Each bit definition can either be specified as an integer offset ≥ 0 or else an (offset,description) tuple.

Returns

A new type with the specified name that has class-level attributes for each named bit (see the examples above). The type also defines a reverse map that is normally accessed via `decode_bitmask()`.

Return type

Raises

- `TypeError` – missing name and/or description args.
- `ValueError` – bit definition is invalid or an offset is repeated.

`bossdata.bits.extract_sdss_bitmasks` (*filename*='sdssMaskbits.par', *indent*='')
Scan the parfile defining SDSS bitmasks and print code to define these types for `bossdata.bits`.

This function is intended to be run by hand with the output pasted into this module, to bootstrap or update the official SDSS bitmask definitions defined here. The generated code is printed directly to the standard output. This function should normally be run from the package top-level directory as:

```
python bossdata/bits.py > bitdefs.py
```

and will read `sdssMaskBits.par` from the same directory. The contents of `bitdefs.py` is then pasted directly into this file, replacing any previous pasted version.

Parameters

- **filename** (*str*) – Path of the parfile to read.
- **indent** (*str*) – Indentation to use in the generated output.

Raises `RuntimeError` – Parse error while reading the input file.

`bossdata.bits.summarize_bitmask_values` (*mask*, *values*, *strict*=`True`)
Summarize an array of bitmask values.

Parameters

- **mask** – A bitmask type, normally created with `create_bitmask()`, that defines the symbolic bit names to summarize.

- **values** (*numpy.ndarray*) – An array of values that will be decoded and summarized.

Returns

A dictionary with bit names as keys and the number of values in which each bit is set as values. Any bit that is never set will not appear in the list of keys.

Return type dict

Support for querying the metadata associated with BOSS observations.

class `bossdata.meta.Database` (*finder=None, mirror=None, lite=True, quasar_catalog=False, quasar_catalog_name=None, platelist=False, verbose=False*)

Initialize a searchable database of BOSS observation metadata.

Parameters

- **finder** (`bossdata.path.Finder`) – Object used to find the names of BOSS data files. If not specified, the default Finder constructor is used.
- **mirror** (`bossdata.remote.Manager`) – Object used to interact with the local mirror of BOSS data. If not specified, the default Manager constructor is used.
- **lite** (*bool*) – Use the “lite” metadata format, which is considerably faster but only provides a subset of the most commonly accessed fields. Ignored if either `quasar_catalog` or `platelist` is True.
- **quasar_catalog** (*bool*) – Initialize database using the BOSS quasar catalog instead of `spAll`.
- **quasar_catalog_name** (*str*) – The name of the BOSS quasar catalog to use, or use the default if this is None.
- **platelist** (*bool*) – Initialize the database use the platelist catalog instead of `spAll`.

prepare_columns (*column_names*)

Validate column names and lookup their types.

Parameters **column_names** (*str*) – Comma-separated list of column names or the special value ‘*’ to indicate all available columns.

Returns

Tuple (names,dtypes) of lists of column names and corresponding numpy data types.
Use `zip()` to convert the return value into a recarray dtype.

Return type tuple

Raises `ValueError` – Invalid column name.

select_all (*what='*', where=None, sort=None, max_rows=100000*)

Fetch all results of an SQL select query.

Since this method loads all the results into memory, it is not suitable for queries that are expected to return a large number of rows. Instead, use `select_each()` for large queries.

Parameters

- **what** (*str*) – Comma separated list of column names to return or ‘*’ to return all columns.
- **where** (*str*) – SQL selection clause or None for no filtering. Reserved column names such as PRIMARY must be escaped with backticks in this clause.
- **max_rows** (*int*) – Maximum number of rows that will be returned.

Returns

Table of results with column names matching those in the database, and column types inferred automatically. Returns None if no rows are selected.

Return type `astropy.table.Table`

Raises `RuntimeError` – failed to execute query.

`select_each` (*what*='*', *where*=None)

Iterate over the results of an SQL select query.

This method is normally used as an iterator, e.g.

for row in select(...): # each row is a tuple of values ...

Since this method does not load all the results of a large query into memory, it is suitable for queries that are expected to return a large number of rows. For smaller queries, the `select_all()` method might be more convenient.

Parameters

- **what** (*str*) – Comma separated list of column names to return or '*' to return all columns.
- **where** (*str*) – SQL selection clause or None for no filtering. Reserved column names such as PRIMARY must be escaped with backticks in this clause.

Raises `sqlite3.OperationalError` – failed to execute query.

`bosssdata.meta.create_meta_full` (*catalog_path*, *db_path*, *verbose*=True, *primary_key*='(PLATE, MJD, FIBER)')

Create the “full” meta database from a locally mirrored catalog file.

The created database renames FIBERID to FIBER and has a composite primary index on the (PLATE,MJD,FIBER) columns. Sub-array columns are also unrolled: see `sql_create_table()` for details. The conversion takes about 24 minutes on a laptop with sufficient memory (~4 Gb). During the conversion, the file being written has the extension *.building* appended, then this extension is removed (and the file is made read only) once the conversion successfully completes. This means that if the conversion is interrupted for any reason, it will be restarted the next time this function is called and you are unlikely to end up with an invalid database file.

Parameters

- **catalog_path** (*str*) – Absolute local path of the “full” catalog file, which is expected to be a FITS file.
- **db_path** (*str*) – Local path where the corresponding sqlite3 database will be written.

`bosssdata.meta.create_meta_lite` (*sp_all_path*, *db_path*, *verbose*=True)

Create the “lite” meta database from a locally mirrored spAll file.

The created database has a composite primary index on the (PLATE,MJD,FIBER) columns and the input columns MODELFLUX0.4 are renamed MODELFLUX_0.4 to be consistent with their names in the full database after sub-array un-rolling.

The DR12 spAll lite file is ~115Mb and converts to a ~470Mb SQL database file. The conversion takes about 3 minutes on a laptop with sufficient memory (~4 Gb). During the conversion, the file being written has the extension *.building* appended, then this extension is removed (and the file is made read only) once the conversion successfully completes. This means that if the conversion is interrupted for any reason, it will be restarted the next time this function is called and you are unlikely to end up with an invalid database file.

Parameters

- **sp_all_path** (*str*) – Absolute local path of the “lite” spAll file, which is expected to be a gzipped ASCII data file.

- **db_path** (*str*) – Local path where the corresponding sqlite3 database will be written.

`bosssdata.meta.get_plate_mjd_list(plate, finder=None, mirror=None)`

Return the list of MJD values when a plate was observed.

Uses a query of the `platelist`, so this file will be automatically downloaded if necessary. Only MJD values for which the observation data quality is marked “good” will be returned.

Parameters

- **plate** (*int*) – Plate number.
- **finder** (`bosssdata.path.Finder`) – Object used to find the names of BOSS data files. If not specified, the default Finder constructor is used.
- **mirror** (`bosssdata.remote.Manager`) – Object used to interact with the local mirror of BOSS data. If not specified, the default Manager constructor is used.

Returns

A list of MJD values when this plate was observed. The list will be empty if this plate has never been observed.

Return type

list

`bosssdata.meta.sql_create_table(table_name, recarray_dtype, renaming_rules={}, primary_key=None)`

Prepare an SQL statement to create a database for a numpy structured array.

Any columns in the structured array data type that are themselves arrays will be unrolled to a list of scalar columns with names `COLNAME_I` for element [i] of a 1D array and `COLNAME_I_J` for element [i,j] of a 2D array, etc, with indices I,J,... starting from zero.

Parameters

- **table_name** (*str*) – Name to give the new table.
- **recarray_dtype** – Numpy structured array data type that defines the columns to create.
- **renaming_rules** (*dict*) – Dictionary of rules for renaming columns. There are no explicit checks that these rules do not create duplicate column names or that all rules are applied.
- **primary_key** (*str*) – Column name(s) to use as the primary key, after apply renaming rules. No index is created if this argument is None.

Returns

Tuple (sql,num_cols) where sql is an executable SQL statement to create the database and `num_cols` is the number of columns created.

Return type

tuple

Raises `ValueError` – Cannot map data type to SQL.

Generate paths to BOSS data files.

The path module provides convenience methods for building the paths of frequently used data files. Most scripts will create a single `Finder` object using the default constructor for this purpose:

```
import bosssdata
finder = bosssdata.path.Finder()
```

This finder object is normally configured by the `$BOSS_SAS_PATH` and `$BOSS_REDUX_VERSION` environment variables and no other modules uses these variables, except through a `Finder` object. These parameters can also

be set by *Finder* constructor arguments. When neither the environment variables nor the constructor arguments are set, defaults appropriate for the most recent public data release (DR12) are used.

Finder objects never interact with any local or remote filesystems: use the *bossdata.remote* module to download data files and access them locally. See *API Usage* for recommendations on using the *bossdata.path* and *bossdata.remote* modules together.

class *bossdata.path.Finder* (*sas_path=None, redux_version=None, verbose=True*)

Initialize a path finder object.

When the constructor is called with no arguments, it will raise a *ValueError* if either *BOSS_SAS_PATH* or *BOSS_REDUX_VERSION* is not set.

Parameters

- **sas_path** (*str*) – Location of the SAS root path to use, e.g., */sas/dr12*. Will use the value of the *BOSS_SAS_PATH* environment variable if this is not set.
- **redux_version** (*str*) – String tag specifying the BOSS spectro reduction version to use, e.g., *v5_7_0*. Will use the value of the *BOSS_REDUX_VERSION* environment variable if this is not set.

Raises *ValueError* – No SAS root or redux version specified on the command line or via environment variables.

get_plate_path (*plate, filename=None*)

Get the path to the specified plate directory or file.

The returned path contains files that include all targets on the plate. Use the *get_spec_path()* method for the path of a single spectrum file.

This method only performs minimal checks that the requested plate number is valid.

Parameters

- **plate** (*int*) – Plate number, which must be positive.
- **filename** (*str*) – Name of a file within the plate directory to append to the returned path.

Returns Full path to the specified plate directory or file within this directory.

Return type *str*

Raises *ValueError* – Invalid plate number must be > 0 .

get_plate_plan_path (*plate, mjd, combined=True*)

Get the path to the specified plate plan file.

A combined plan may span several nearby MJDs, in which case the last MJD is the one used to identify the plan.

Parameters

- **plate** (*int*) – Plate number, which must be positive.
- **mjd** (*int*) – Modified Julian date of the observation, which must be > 45000 .
- **combined** (*bool*) – Specifies the combined plan, which spans all MJDs associated with a coadd, but does not include calibration frames (arcs, flats) for a specific MJD.

Returns Full path to the requested plan file.

Return type *str*

Raises *ValueError* – Invalid plate or mjd inputs.

get_plate_spec_path (*plate*, *mjd*)

Get the path to the file containing combined spectra for a whole plate.

Combined spectra for all exposures of a plate are packaged in [spPlate files](#). As of DR12, these files are about 110Mb for 1000 spectra.

Parameters

- **plate** (*int*) – Plate number, which must be positive.
- **mjd** (*int*) – Modified Julian date of the observation, which must be > 45000.

Returns Full path to the requested plan file.

Return type str

Raises `ValueError` – Invalid plate or mjd inputs.

get_platelist_path ()

Get the location of the platelist summary file.

The [platelist](#) contains one row per observation (PLATE-MJD), unlike most other sources of metadata which contain one row per target (PLATE-MJD-FIBER).

get_quasar_catalog_path (*catalog_name=None*)

Get the location of the quasar catalog file.

The [quasar catalog](#) is documented at <http://www.sdss.org/dr12/algorithms/boss-dr12-quasar-catalog/>. As of DR12, the file size is about 513Mb.

Parameters **catalog_name** (*str*) – BOSS quasar catalog name. Will use the `get_default_quasar_catalog_name()` method if this is not set.

get_raw_path (*mjd*, *camera*, *exposure*)

Get the location of the raw data for the specified camera and exposure.

The DR12 data model for BOSS raw data in the sdR format is at http://data.sdss3.org/datamodel/files/BOSS_SPECTRO_DATA/MJD/sdR.html

Parameters

- **mjd** (*int*) – Modified Julian date of the observation, which must be > 45000.
- **camara** (*str*) – One of b1, b2, r1, r2.
- **exposure** (*int*) – Exposure number to return.

Raises `ValueError` – Invalid camera or exposure number.

get_sp_all_path (*lite=True*)

Get the location of the metadata summary file.

The [spAll](#) file provides extensive metadata for all survey targets as a FITS file. There is also a smaller “lite” version containing a subset of this metadata in compressed text format. As of DR12, the full file size is about 10Gb and the lite file is about 115Mb.

Parameters **lite** (*bool*) – Specifies the “lite” version which contains all rows but only the most commonly used subset of columns. The lite version is a compressed (.gz) text data file, while the full version is a FITS file.

get_spec_path (*plate*, *mjd*, *fiber*, *lite=True*, *mock=False*, *compressed=False*)

Get the location of the spectrum file for the specified observation.

The DR12 data model for the returned files is at http://dr12.sdss3.org/datamodel/files/BOSS_SPECTRO_REDUX/RUN2D/spectra/PLATE4/spec.html but only HDUs 0-3 are included in the (default) lite format. Each lite (full) file is approximately 0.2Mb (1.7Mb) in size.

Use the `get_plate_path()` method for the path to files that include all targets on a plate.

This method only performs minimal checks that the requested plate-mjd-fiber are valid.

Parameters

- **plate** (*int*) – Plate number, which must be positive.
- **mjd** (*int*) – Modified Julian date of the observation, which must be > 45000.
- **fiber** (*int*) – Fiber number of the target on this plate, which must be in the range 1-1000 (or 1-640 for plate < 3510).
- **lite** (*bool*) – Specifies the “lite” version which contains only HDUs 0-3, so no per-exposure data is included.

Returns Full path to the spectrum file for the specified observation.

Return type str

Raises `ValueError` – Invalid plate, mjd or fiber inputs.

Access BOSS plate data products.

class `bossdata.plate.FrameFile` (*path, index=None, calibrated=None*)

A BOSS frame file containing a single exposure of one spectrograph (half plate).

This class supports both types of frame data files: the uncalibrated `spFrame` and the calibrated `spCFrame`. Use `get_valid_data()` to access this plate’s data and the `plug_map` attribute to access this plate’s `plug map`.

BOSS spectrographs read out 500 fibers each. SDSS-I/II spectrographs (plate < 3510) read out 320 fibers each. The `plate`, `camera` and `exposure_id` attributes provide the basic metadata for this exposure. The complete HDU0 header is available as the `header` attribute.

This class is only intended for reading the BOSS frame file format, so generic operations on spectroscopic data (redshifting, resampling, etc) are intentionally not included here, but are instead provided in the `speclite` package.

Parameters

- **path** (*str*) – Local path of the frame FITS file to use. This should normally be obtained via `Plan.get_exposure_name()` and can be automatically mirrored via `bossdata.remote.Manager.get()` or using the `bossfetch` script. The file is opened in read-only mode so you do not need write privileges.
- **index** (*int*) – Identifies if this is the first (1) or second (2) spectrograph, which determines whether it has spectra for fibers 1-500 (1-320) or 501-1000 (321-640). You should normally obtain this value using `Plan.get_spectrograph_index()`. As of v0.2.7, this argument is optional and will be inferred from the file header when not provided, or checked against the file header when provided.
- **calibrated** (*bool*) – Identifies whether this is a calibrated (`spCFrame`) or un-calibrated (`spFrame`) frame file. As of v0.2.7, this argument is optional and will be inferred from the file header when not provided, or checked against the file header when provided.

get_fiber_offsets (*fiber*)

Convert fiber numbers to array offsets.

Parameters **fibers** (*numpy.ndarray*) – Numpy array of fiber numbers 1-1000 (or 1-640 for plate < 3510). All fibers must be in the appropriate range 1-500 (1-320) or 501-1000 (321-640) for this frame’s spectrograph. Fibers do not need to be sorted and repetitions are ok.

Returns Numpy array of offsets 0-499 (or 0-319 for plate < 3510).

Return type `numpy.ndarray`

Raises `ValueError` – Fiber number is out of the valid range for this spectrograph.

get_pixel_masks (*fibers*)

Get the pixel masks for specified fibers.

The entire mask is returned for each fiber, including any pixels with zero inverse variance.

Parameters **fibers** (`numpy.ndarray`) – Numpy array of fiber numbers 1-1000 (or 1-640 for plate < 3510). All fibers must be in the appropriate range 1-500 (1-320) or 501-1000 (321-640) for this frame’s spectrograph. Fibers do not need to be sorted and repetitions are ok.

Returns

Integer numpy array of shape (nfibers,npixels) where (i,j) encodes the mask bits defined in `bosssdata.bits.SPPIXMASK` (see also http://www.sdss3.org/dr10/algorithms/bitmask_sppixmask.php) for pixel-j of the fiber with index `fibers[i]`.

Return type `numpy.ndarray`

get_valid_data (*fibers*, *pixel_quality_mask=None*, *include_wdisp=False*, *include_sky=False*, *use_ivar=False*, *use_loglam=False*)

Get the valid for the specified fibers.

Parameters

- **fibers** (`numpy.ndarray`) – Numpy array of fiber numbers 1-1000 (or 1-640 for plate < 3510). All fibers must be in the appropriate range 1-500 (1-320) or 501-1000 (321-640) for this frame’s spectrograph. Fibers do not need to be sorted and repetitions are ok.
- **pixel_quality_mask** (*int*) – An integer value interpreted as a bit pattern using the bits defined in `bosssdata.bits.SPPIXMASK` (see also http://www.sdss3.org/dr10/algorithms/bitmask_sppixmask.php). Any bits set in this mask are considered harmless and the corresponding spectrum pixels are assumed to contain valid data.
- **include_wdisp** – Include a wavelength dispersion column in the returned data.
- **include_sky** – Include a sky flux column in the returned data.
- **use_ivar** – Replace `dflux` with `ivar` (inverse variance) in the returned data.
- **use_loglam** – Replace `wavelength` with `loglam(log10(wavelength))` in the returned data.

Returns

Masked array of shape (nfibers,npixels). Pixels with no valid data are included but masked. The record for each pixel has at least the following named fields: `wavelength` in Angstroms, `flux` and `dflux` in $1e-17$ ergs/s/cm²/Angstrom. Wavelength values are strictly increasing and `dflux` is calculated as `ivar**-0.5` for pixels with valid data. Optional fields are `wdisp` in constant-log10-lambda pixels and `sky` in $1e-17$ ergs/s/cm²/Angstrom. The `wavelength` (or `loglam`) field is never masked and all other fields are masked when `ivar` is zero or a pipeline flag is set (and not allowed by `pixel_quality_mask`).

Return type `numpy.ma.MaskedArray`

class `bosssdata.plate.Plan` (*path*)

The plan file for configuring the BOSS pipeline to combine exposures of a single plate.

Combined [plan files](#) are small text files that list the per-spectrograph (b1,b2,r1,r2) exposures used as input to a single coadd. Use the `exposure_table` attribute to access this information. Note that `bosssdata.spec`.

SpecFile has a similar *exposures* attribute which only includes exposures actually used in the final co-add, so is generally a subset of the planned exposures.

Parameters *path* (*str*) – The local path to a plan file.

get_exposure_name (*sequence_number*, *band*, *fiber*, *ftype*=*'spCFrame'*)

Get the file name of a single science exposure data product.

Use the exposure name to locate FITS data files associated with individual exposures. The supported file types are: *spCFrame*, *spFrame*, *spFluxcalib* and *spFluxcorr*. Note that this method returns *None* when the requested exposure is not present in the plan, so the return value should always be checked.

Parameters

- **sequence_number** (*int*) – Science exposure sequence number, counting from zero. Must be less than our *num_science_exposures* attribute.
- **fiber** (*int*) – Fiber number to identify which spectrograph to use, which must be in the range 1-1000 (or 1-640 for plate < 3510).
- **band** (*str*) – Must be 'blue' or 'red'.
- **fctype** (*str*) – Type of exposure file whose name to return. Must be one of *spCFrame*, *spFrame*, *spFluxcalib*, *spFluxcorr*. An *spCFrame* is assumed to be uncompressed, and all other files are assumed to be compressed.

Returns

Exposure name of the form [ftype]-[cc]-[eeeeeeee].[ext] where [cc] identifies the spectrograph (one of b1,r1,b2,r2) and [eeeeeeee] is the zero-padded exposure number. The extension [ext] is "fits" for *spCFrame* files and "fits.gz" for all other file types. Returns *None* if the name is unknown for this band and fiber combination.

Return type *str*

Raises *ValueError* – one of the inputs is invalid.

get_spectrograph_index (*fiber*)

Get the spectrograph index 1,2 for the specified fiber.

Parameters **fiber** (*int*) – Fiber number to identify which spectrograph to use, which must be in the range 1-1000 (or 1-640 for plate < 3510).

Returns

Value of 1 if fiber is read out by the first spectrograph 1-500 (1-320), or else 2 for the second spectrograph.

Return type *int*

Raises *ValueError* – fiber is outside the allowed range 1-1000 (1-640) for this plate.

class *bossdata.plate.PlateFile* (*path*)

A BOSS plate file containing combined exposures for a whole plate.

This class provides an interface to the *spPlate* data product, containing all co-added spectra for a single observation. To instead read individual co-added spectra, use *bossdata.spec.SpecFile*. To access individual exposures of a half-plate use *FrameFile*.

Use *get_valid_data()* to access this plate's data, or the *exposures* attribute for a list of exposures used in the coadd. The *num_exposures* attribute gives the number of science exposures used for this target's co-added spectrum (counting a blue+red pair as one exposure). The *plug_map* attribute records this plate's *plug map*.

This class is only intended for reading the BOSS plate file format, so generic operations on spectroscopic data (redshifting, resampling, etc) are intentionally not included here, but are instead provided in the [speclite](#) package.

Parameters `path` (*str*) – Local path of the plate FITS file to use. This should normally be obtained via `bossdata.path.Finder.get_plate_spec_path()` and can be automatically mirrored via `bossdata.remote.Manager.get()` or using the `bossfetch` script. The file is opened in read-only mode so you do not need write privileges.

get_fiber_offsets (*fiber*)

Convert fiber numbers to array offsets.

Parameters `fibers` (*numpy.ndarray*) – Numpy array of fiber numbers 1-1000 (or 1-640 for plate < 3510). Fibers do not need to be sorted and repetitions are ok.

Returns Numpy array of offsets 0-999.

Return type `numpy.ndarray`

Raises `ValueError` – Fiber number is out of the valid range for this plate.

get_pixel_masks (*fibers*)

Get the pixel masks for specified fibers.

The entire mask is returned for each fiber, including any pixels with zero inverse variance. Returns the 'and_mask' and ignores the 'or_mask'.

Parameters `fibers` (*numpy.ndarray*) – Numpy array of fiber numbers 1-1000 (or 1-640 for plate < 3510). Fibers do not need to be sorted and repetitions are ok.

Returns

Integer numpy array of shape (nfibers,npixels) where (i,j) encodes the mask bits defined in `bossdata.bits.SPPIXMASK` (see also http://www.sdss3.org/dr10/algorithms/bitmask_sppixmask.php) for pixel-j of the fiber with index `fibers[i]`.

Return type `numpy.ndarray`

get_valid_data (*fibers*, *pixel_quality_mask=None*, *include_wdisp=False*, *include_sky=False*, *use_ivar=False*, *use_loglam=False*, *fiducial_grid=False*)

Get the valid for the specified fibers.

Parameters

- **fibers** (*numpy.ndarray*) – Numpy array of fiber numbers 1-1000 (or 1-640 for plate < 3510). Fibers do not need to be sorted and repetitions are ok.
- **pixel_quality_mask** (*int*) – An integer value interpreted as a bit pattern using the bits defined in `bossdata.bits.SPPIXMASK` (see also http://www.sdss3.org/dr10/algorithms/bitmask_sppixmask.php). Any bits set in this mask are considered harmless and the corresponding spectrum pixels are assumed to contain valid data. This mask is applied to the AND of the masks for each individual exposure. No mask is applied if this value is None.
- **include_wdisp** – Include a wavelength dispersion column in the returned data.
- **include_sky** – Include a sky flux column in the returned data.
- **use_ivar** – Replace `dflux` with `ivar` (inverse variance) in the returned data.
- **use_loglam** – Replace wavelength with `loglam(log10(wavelength))` in the returned data.
- **fiducial_grid** – Return co-added data using the `fiducial wavelength grid`. If False, the returned array uses the native grid of the `SpecFile`, which generally trims

pixels on both ends that have zero inverse variance. Set this value True to ensure that all co-added spectra use aligned wavelength grids when this matters.

Returns

Masked array of shape (nfibers,npixels). Pixels with no valid data are included but masked. The record for each pixel has at least the following named fields: wavelength in Angstroms (or loglam), flux and dflux in $1e-17$ ergs/s/cm2/Angstrom (or flux and ivar). Wavelength values are strictly increasing and dflux is calculated as $ivar^{*-0.5}$ for pixels with valid data. Optional fields are wdisp in constant-log10-lambda pixels and sky in $1e-17$ ergs/s/cm2/Angstrom. The wavelength (or loglam) field is never masked and all other fields are masked when ivar is zero or a pipeline flag is set (and not allowed by pixel_quality_mask).

Return type numpy.ma.MaskedArray

class bossdata.plate.TraceSet (hdu)

A set of interpolating functions along each trace of a half plate.

TraceSets use the terminology that x is the pixel coordinate along the nominal wavelength direction and y is some quantity to be interpolated as a function of x. This implementation is based on the original SDSS IDL code: <https://trac.sdss3.org/browser/repo/idlutils/trunk/pro/trace/traceset2xy.pro>

Note that red and blue CCDs are handled differently, as described [here](#):

The plan is to switch from 1-phase to 2-phase readout on the red CCDs in summer 2010. This will effectively make the pixels more uniform, and the flat-fields much better.

A problem introduced will be that the central two rows will each be taller by 1/6 pix. That will flat-field, but there will be a discontinuity of 1/3 pix across this point. Technically, the PSF will also be different for those pixels, and the resulting resolution function.

Parameters hdu – fitsio HDU containing the trace set data as a binary table.

Raises ValueError – Unable to initialize a trace set with this HDU.

get_y (xpos=None, ignore_jump=False)

Evaluate the interpolating function for each trace.

Parameters

- **xpos** (numpy.ndarray) – Numpy array of shape (ntrace,nx) with x-pixel coordinates along each trace where y(x) should be evaluated. For BOSS, ntrace = 500 and for SDSS-I/II (plate < 3510), ntrace = 320. The value of ntrace is available as self.ntrace. If this argument is not set, self.default_xpos will be used, which consists of num_fibers identical traces with x-pixel coordinates at each integer pixel value covering the full allowed range.
- **ignore_jump** (bool) – Include a jump when this is set and this is a 2-phase readout. There is probably no good reason to set this False, but it is included for compatibility with the original IDL code.

Returns

Numpy array y with shape (ntrace,nx) that matches the input xpos or else the default self.default_xpos. ypos[[i,x]] gives the value of the interpolated y(x) with x equal to xpos[[i,x]].

Return type `numpy.ndarray`

`bosssdata.plate.get_num_fibers(plate)`

Return the number of fiber holes for a given plate number.

Plate numbers 3510 or larger are (e)BOSS plates with 1000 fibers. Smaller plate numbers are assumed to be SDSS-I/II with 640 fibers.

Parameters `plate` (*int*) – Plate number.

Returns The value 640 or 1000.

Return type `int`

Support for plotting BOSS spectroscopic data in different formats.

These functions use the optional matplotlib dependency so will raise an `ImportError` if this is not installed. Functions do not create figures or call `matplotlib.pyplot.show()` before exiting, to provide the maximum flexibility. To display a single plot, you can use the following wrapper:

```
plt.figure(figsize=(11,8.5))
# ... call one of the plot functions ...
plt.tight_layout()
plt.show()
```

See the [Examples](#) for details.

`bosssdata.plot.by_fiber(data, mask=None, subsets={}, percentile_cut=0.0, plot_label=None, data_label=None, default_options={'s': 60, 'lw': 0.5, 'marker': 'o'})`

Plot per-fiber data values in fiber order.

This is a useful plot to show any dependence of the data value on a fiber's position on the CCD and slithead. Both spectrographs are superimposed on the same plot. The points for each fiber are color-coded according to their associated data value using the same scheme as `focal_plane()`.

Parameters

- **data** (*numpy.ndarray*) – A 1D array of data values to plot, where the array index matches the fiber number and all fibers are included.
- **mask** (*numpy.ndarray*) – An optional 1D array of boolean values with True values used to mask out values in the data array. Masked values will not be plotted and will not be used to calculate the plot data range.
- **subsets** (*dict*) – A dictionary of fiber subsets that will be separately identified in the plot. Each dictionary must define values for two keys: 'options' and 'fibers'. The options are a dictionary of arguments passed to `matplotlib.pyplot.scatter()` and used to style the subset. The fibers value is used to index the data array to pick out the subset's data values.
- **percentile_cut** (*float*) – Data will be clipped to this percentile value on both sides of its distribution. Use a value of zero (the default) for no clipping.
- **plot_label** (*str*) – A label identifying this plot that will be displayed in the top-left corner.
- **data_label** (*str*) – A label identifying the data values that will be used to label the y axis.
- **default_options** (*dict*) – A dictionary of options passed to `matplotlib.pyplot.scatter()` that is used to draw data points. Options in a subset dictionary override any values here. Fibers not in any subset are drawn using these default options.

```

bosssdata.plot.focal_plane(xfocal, yfocal, data, mask=None, subsets={}, background=None,
                           numbered=None, percentile_cut=0.0, mesh_refinement=0,
                           plot_label=None, data_label=None, show_background_mesh=False,
                           number_color='red', default_options={'edgecolors': 'k', 's': 60, 'lw':
                           0.5, 'marker': 'o'}, rmax=350.0)

```

Plot per-fiber data values using focal-plane positions.

This is a useful plot to show any dependence of the data value on a fiber's position in the focal plane. The points for each fiber are color-coded according to their associated data value using the same scheme as `by_fiber()`.

Parameters

- **xfocal** (*numpy.ndarray*) – A 1D array of x focal-plane positions, where the array index matches the fiber number and all fibers are included.
- **yfocal** (*numpy.ndarray*) – A 1D array of y focal-plane positions, where the array index matches the fiber number and all fibers are included.
- **data** (*numpy.ndarray*) – A 1D array of data values to plot, where the array index matches the fiber number and all fibers are included.
- **mask** (*numpy.ndarray*) – An optional 1D array of boolean values with True values used to mask out values in the data array. Masked values will not be plotted and will not be used to calculate the plot data range.
- **subsets** (*dict*) – A dictionary of fiber subsets that will be separately identified in the plot. Each dictionary must define values for two keys: 'options' and 'fibers'. The options are a dictionary of arguments passed to `matplotlib.pyplot.scatter()` and used to style the subset. The fibers value is used to index the data array to pick out the subset's data values.
- **background** (*numpy.ndarray*) – An optional subset of fibers whose data values are used to fill the background using interpolation. The resulting background fill will only cover the convex hull of the subset, where interpolation is possible.
- **numbered** (*numpy.ndarray*) – An optional subset of fibers that will be numbered in the generated plot.
- **percentile_cut** (*float*) – Data will be clipped to this percentile value on both sides of its distribution. Use a value of zero (the default) for no clipping.
- **mesh_refinement** (*int*) – Smoothness of background fill interpolation to use. A value of zero (the default) corresponds to linear interpolation.
- **plot_label** (*str*) – A label identifying this plot that will be displayed in the top-left corner.
- **data_label** (*str*) – A label identifying the data values that will be used to label the y axis.
- **show_background_mesh** (*bool*) – Draw the triangulation used for the background fill when this is True.
- **number_color** (*str*) – Matplotlib color used to draw fiber numbers.
- **default_options** (*dict*) – A dictionary of options passed to `matplotlib.pyplot.scatter()` that is used to draw data points. Options in a subset dictionary override any values here. Fibers not in any subset are drawn using these default options.

Work with raw spectroscopic science and calibration images.

```
class bosssdata.raw.RawImageFile(path)
```

Wrapper for a raw science or calibration sdR image format.

Each camera (b1,b2,r1,r2) has its own raw image file for a given exposure ID. See the [sdR datamodel](#) for details. The FITS file is opened, read, and closed by the constructor.

The raw files used in a co-add can be located and opened using `bossdata.spec.SpecFile.get_raw_image()`.

Parameters `path` (*str*) – Local path of the FITS image file to use.

header

dict – Dictionary of FITS header keyword-value pairs.

plate

int – Plate number used to record this raw data.

exposure_id

int – Exposure ID for this raw data.

camera

str – One of b1, b2, r1, r2.

flavor

str – One of science, arc, flat.

data

numpy.ndarray – 2D array of raw pixel values.

get_amplifier_bias (*amplifier_index*, *percentile_cut=1*)

Estimate the amplifier bias.

Estimate the bias in one amplifier (quadrant) using the truncated mean of pixel values in its *overscan region*.

Parameters

- **amplifier_index** (*int*) – Amplifier's are indexed 0-3.
- **percentile_cut** (*float*) – Percentage of outliers to ignore on both sides of the distribution.

Returns Estimated bias.

Return type float

get_amplifier_region (*amplifier_index*, *region_type*)

Get overscan and data regions for one amplifier.

Region definitions are taken from the [sdR datamodel](#). Each amplifier reads out one quadrant of the sensor. Amplifiers 0-3 have outside corner pixel indices [0, 0], [0, -1], [-1, 0], [-1, -1], respectively. Results are only valid for MJD >= 55113 (9-Oct-2009).

Parameters

- **amplifier_index** (*int*) – Amplifier's are indexed 0-3.
- **region_type** (*str*) – One of data, overscan.

Returns

Tuple (rows, cols) of pixel slices that define the requested region. The return value can be used to create a view of our raw data as `self.data[rows, cols]`.

Return type tuple

get_data (*bias_subtracted=True*, *percentile_cut=1*, *bias_point=100*)

Get the data region of this raw image.

The data region is the union of the four *amplifier data regions*.

Parameters

- **bias_subtracted** (*bool*) – Subtract bias from each amplifier quadrant, estimated using `get_amplifier_bias()` (rounded to the nearest integer value).
- **percentile_cut** (*float*) – Percentage of outliers to ignore on both sides of the distribution for estimating bias (when `bias_subtracted` is True).
- **bias_point** (*int*) – When `bias_subtracted` is True, raw pixel values will be offset so that the bias value is shifted to the specified value. Must be between 0 - 0xffff and smaller than the estimated bias value.

Returns

2D array of pixel values as unsigned 16-bit integers. The return value is a copy (rather than a view) of the raw data array in the file. When `bias_subtracted` is True, saturated values (0xffff) will remain saturated, and values that underflow (<0) after bias subtraction will be set to zero.

Return type `numpy.ndarray`

read_plug_map (*speclog_path=None*)

Read the plug map associated with this plate and exposure.

Plug maps are not stored under the same SAS tree as other data products so reading a plug map requires that a separate directory containing plug map files is already available on the local disk. To download the set of plug map files used by the pipeline, use the SDSS public SVN repository to make a local checkout of the `speclog` product:

```
svn co https://svn.sdss.org/public/data/sdss/speclog/trunk speclog
```

This will take a while to run (about 15 minutes) and will copy about 25 Gb of data into the newly created `speclog` sub-directory. Pass the full name of this directory to this method.

You can create your `speclog` directory anywhere, but you should avoid putting it under `$BOSS_SAS_PATH` in your `$BOSS_LOCAL_ROOT` since plug map files do not originate from SAS. Note that ~11Gb of the files in `speclog` are *guidermon* files and not very useful. You cannot avoid downloading them with the `svn` checkout, but you can delete them after the checkout using:

```
cd speclog
find . -name 'guiderMon-*.par' -delete
```

The plug maps are *yanny* parameter files. See the *plug map datamodel* for details.

Parameters **speclog_path** (*str*) – The local path to a directory containing `plPlugMapM` files organized in subdirectories by observation MJD. If `None` is specified, the value of the `BOSS_SPECLOG` environment variable will be used, if available.

Returns

Object wrapper around the plug map yanny file. The `PLUGMAPOBJ` attribute contains the table of plugging info.

Return type `pydl.pydlutils.yanny.yanny`

Download BOSS data files from a remote server.

The remote module is responsible for downloading data files into a local filesystem using a directory layout that mirrors the remote data source. Most scripts will create a single *Manager* object using the default constructor for this purpose:

```
import bosssdata
mirror = bosssdata.remote.Manager()
```

This mirror object is normally configured by the `$BOSS_DATA_URL` and `$BOSS_LOCAL_ROOT` environment variables and no other modules uses these variables, except through a `Manager` object. These parameters can also be set by `Manager` constructor arguments. When neither the environment variables nor the constructor arguments are set, a default data URL appropriate for the most recent public data release (DR12) is used, and a temporary directory is created and used for the local root.

`Manager` objects have no knowledge of how data files are organized or named: use the `bosssdata.path` module to build the paths of frequently used data files. See *API Usage* for recommendations on using the `bosssdata.path` and `bosssdata.remote` modules together.

class `bosssdata.remote.Manager` (`data_url=None`, `local_root=None`, `verbose=True`)

Manage downloads of BOSS data via HTTP.

The default mapping from remote to local filenames is to mirror the remote file hierarchy on the local disk. The normal mode of operation is to establish the local root for the mirror using the `BOSS_LOCAL_ROOT` environment variable. When the constructor is called with no arguments, it will raise a `ValueError` if either `BOSS_DATA_URL` or `BOSS_LOCAL_ROOT` is not set.

Parameters

- **data_url** (*str*) – Base URL of all BOSS data files. A trailing `/` on the URL is optional. If this arg is `None`, then the value of the `BOSS_DATA_URL` environment variable will be used instead.
- **local_root** (*str*) – Local path to use as the root of the locally mirrored file hierarchy. If this arg is `None`, then the value of the `BOSS_LOCAL_ROOT` environment variable, if any, will be used instead. If a value is provided, it should identify an existing writeable directory.

Raises `ValueError` – No such directory `local_root` or missing `data_url`.

download (*remote_path*, *local_path*, *chunk_size=4096*, *progress_min_size=10*)

Download a single BOSS data file.

Downloads are streamed so that the memory requirements are independent of the file size. During the download, the file is written to its final location but with `‘.downloading’` appended to the file name. This means that any download that is interrupted or fails will normally not lead to an incomplete file being returned by a subsequent call to `get()`. Instead, the file will be re-downloaded. There is no facility for resuming a previous partial download. After a successful download, the file is renamed to its final location and has its permission bits set to read only (to prevent accidental modifications of files that are supposed to exactly mirror the remote file system).

Parameters

- **remote_path** (*str*) – The full path to the remote file relative to the remote server root, which should normally be obtained using `bosssdata.path` methods.
- **local_path** (*str*) – The (absolute or relative) path of the local file to write.
- **chunk_size** (*int*) – Size of data chunks to use for the streaming download. Larger sizes will potentially download faster but also require more memory.
- **progress_min_size** (*int*) – Display a text progress bar for any downloads whose size in Mb exceeds this value. No progress bar will ever be shown if this value is `None`.

Returns Absolute local path of the downloaded file.

Return type `str`

Raises

- `ValueError` – `local_path` directory does not exist.
- `RuntimeError` – HTTP request returned an error status.

get (*remote_path*, *progress_min_size=10*, *auto_download=True*, *local_paths=None*)

Get a local file that mirrors a remote file, downloading the file if necessary.

Parameters

- **remote_path** (*str*, *list*) – This arg will normally be a single string but can optionally be a list of strings for some advanced functionality. Strings give the full path to a remote file and should normally be obtained using `bosssdata.path` methods. When passing an iterable, the first item specifies the desired file and subsequent items specify acceptable substitutes. If the desired file is not already available locally but at least one substitute file is locally available, this method immediately returns the first substitute without downloading the desired file. If no substitute is available, the desired file is downloaded and returned.
- **progress_min_size** (*int*) – Display a text progress bar for any downloads whose size in Mb exceeds this value. No progress bar will ever be shown if this value is `None`.
- **auto_download** (*bool*) – Automatically download the file to the local mirror if necessary. If this is not set and the file is not already mirrored, then a `RuntimeError` occurs.
- **local_paths** (*list*) – When this arg is not `None`, the local paths corresponding to each input remote path are stored to this arg, resulting in a list of the same size as the input `remote_path` (or length 1 if `remote_path` is a single string). This enables the following pattern for detecting when a substitution has occurred:

```
mirror = bosssdata.remote.Manager()
remote_paths = [the_preferred_path, a_backup_path]
local_paths = []
local_path = mirror.get(remote_paths, local_paths=local_paths)
if local_path != local_paths[0]:
    print('substituted {} for {}'.format(local_path, local_
    ↪paths[0]))
```

Returns Absolute local path of the local file that mirrors the remote file.

Return type `str`

Raises `RuntimeError` – File is not already mirrored and `auto_download` is `False`.

local_path (*remote_path*, *suffix=None*, *new_suffix=None*)

Get the local path corresponding to a remote path.

Does not check that the file or its parent directory exists. Use `get()` to ensure that the file exists, downloading it if necessary.

Parameters

- **remote_path** (*str*) – The full path to the remote file relative to the remote server root, which should normally be obtained using `bosssdata.path` methods.
- **suffix** (*str*) – The expected suffix of the returned local path. A `RuntimeError` is raised when the local path does not have this suffix according to `str.endswith()`, unless this parameter is `None`.
- **new_suffix** (*str*) – Replace suffix with this value. No change is performed when this parameter is `None`, and `suffix` must also be set with this parameter is not `None`.

Returns

Absolute local path of the local file that mirrors the remote file, with a possible suffix replacement.

Return type str

Raises

- `ValueError` – The `new_suffix` parameter is set but `suffix` is `None`.
- `RuntimeError` – The local path does not have the expected suffix.

Access spectroscopic data for a single BOSS target.

class `bosssdata.spec.Exposures` (*header*)

Table of exposure info extracted from FITS header keywords.

Parse the NEXP and EXPIDnn keywords that are present in the header of HDU0 in `spPlate` and `spec` FITS files.

The constructor initializes the `table` attribute with column names `offset`, `camera`, `science`, `flat` and `arc`, and creates one row for each keyword EXPIDnn, where `offset` equals the keyword sequence number `nn`, `camera` is one of `b1`, `b2`, `r1`, `r2`, and the remaining columns record the science and calibration exposure numbers.

Use `get_info()` to retrieve the `n`-th exposure for a particular camera (`b1`, `b2`, `r1`, `r2`). Note that when this class is initialized from a `spec` file header, it will only describe the two cameras of a single spectrograph (`b1+r1` or `b2+r2`). The `num_by_camera` attribute is a dictionary of ints indexed by camera that records the number of science exposures available for that camera.

Parameters `header` (*dict*) – dictionary of FITS header keyword, value pairs.

Returns:

get_exposure_name (*exposure_index*, *camera*, *ftype*=`'spCFrame'`)

Get the file name of a single science or calibration exposure data product.

Use the exposure name to locate FITS data files associated with individual exposures. The supported file types are: `spCFrame`, `spFrame`, `spFluxcalib`, `spFluxcorr`, `spArc`, `spFlat`. This method is analogous to `bosssdata.plate.Plan.get_exposure_name()`, but operates for a single target and only knows about exposures actually used in the final co-add (including the associated arc and flat exposures).

Parameters

- **exposure_index** (*int*) – The sequence number for the requested camera exposure, in the range 0 - (`num_exposures[camera]-1`).
- **camera** (*str*) – One of `b1`, `b2`, `r1`, `r2`.
- **ftype** (*str*) – Type of exposure file whose name to return. Must be one of `spCFrame`, `spFrame`, `spFluxcalib`, `spFluxcorr`, `spArc`, `spFlat`. An `spCFrame` is assumed to be uncompressed, and all other files are assumed to be compressed. When a calibration is requested (`spArc`, `spFlat`) results from the calibration exposure used to analyze the specified science exposure is returned.

Returns

Exposure name of the form `[ftype]-[cc]-[eeeeeeee].[ext]` where `[cc]` identifies the camera (one of `b1`, `r1`, `b2`, `r2`) and `[eeeeeeee]` is the zero-padded arc/flat/science exposure number. The extension `[ext]` is “fits” for `spCFrame` files and “fits.gz” for all other file types.

Return type str

Raises `ValueError` – one of the inputs is invalid.

get_info (*exposure_index*, *camera*)

Get information about a single camera exposure.

Parameters

- **exposure_index** (*int*) – The sequence number for the requested camera exposure, in the range 0 - (*num_exposures[camera]-1*).
- **camera** (*str*) – One of b1,b2,r1,r2.

Returns A structured array with information about the requested exposure, corresponding to one row of our `table` attribute.

Raises

- `ValueError` – Invalid `exposure_index` or `camera`.
- `RuntimeError` – Exposure not present.

get_raw_image (*plate, mjd, exposure_index, camera, flavor='science', finder=None, mirror=None*)
Get the raw image file associated with an exposure.

Parameters

- **plate** (*int*) – Plate number, which must be positive.
- **mjd** (*int*) – Modified Julian date of the observation, which must be > 45000.
- **exposure_index** (*int*) – The sequence number for the requested camera exposure, in the range 0 - (*num_exposures[camera]-1*).
- **camera** (*str*) – One of b1,b2,r1,r2.
- **flavor** (*str*) – One of science, arc, flat.
- **finder** (`bossdata.path.Finder`) – Object used to find the names of BOSS data files. If not specified, the default Finder constructor is used.
- **mirror** (`bossdata.remote.Manager`) – Object used to interact with the local mirror of BOSS data. If not specified, the default Manager constructor is used.

Returns requested raw image file.

Return type `bossdata.raw.RawImageFile`

Raises `ValueError` – one of the inputs is invalid.

class `bossdata.spec.SpecFile` (*path*)

A BOSS spec file containing summary data for a single target.

A `spec` file contains co-added spectra for a single target of an observation. This class supports the full version described in the data model as well as a `lite` version that does not contain the per-exposure HDUs with indices >= 4. Use the `lite` attribute to detect which version an object represents.

To read all co-added spectra of an observation use `bossdata.plate.PlateFile`. Individual exposures of a half-plate can be read using `bossdata.plate.FrameFile`.

The `plate`, `mjd` and `fiber` attributes specify the target observation. The `info` attribute contains this target's row from `spAll` as a structured numpy array, so its metadata can be accessed as `info['OBJTYPE']`, etc.

Use `get_valid_data()` to access this target's spectra, or the `exposures` attribute for a list of exposures used in the coadd (see `bossdata.plate.Plan` for alternative information about the exposures used in a coadd.) The `num_exposures` attribute gives the number of science exposures used for this target's co-added spectrum (counting a blue+red pair as one exposure). Use `get_exposure_name()` to locate files associated with the individual exposures used for this co-added spectrum.

This class is only intended for reading the BOSS spec file format, so generic operations on spectroscopic data (redshifting, resampling, etc) are intentionally not included here, but are instead provided in the `speclite` package.

Parameters `path` (*str*) – Local path of the spec FITS file to use. This should normally be obtained via `bosssdata.path.Finder.get_spec_path()` and can be automatically mirrored via `bosssdata.remote.Manager.get()` or using the `bossfetch` script. The file is opened in read-only mode so you do not need write privileges.

get_exposure_hdu (*exposure_index*, *camera*)

Lookup the HDU for one exposure.

This method will not work on “lite” files, which do not include individual exposures.

Parameters

- **exposure_index** (*int*) – Individual exposure to use, specified as a sequence number starting from zero, for the first exposure, and increasing up to `self.num_exposures-1`.
- **camera** (*str*) – Which camera to use. Must be one of b1,b2,r1,r2.

Returns The HDU containing data for the requested exposure.

Return type `hdu`

Raises `RuntimeError` – individual exposures not available in lite file.

get_exposure_name (*sequence_number*, *band*, *fctype*='spCFrame')

Get the file name of a single science exposure data product.

Use the exposure name to locate FITS data files associated with individual exposures. The supported file types are: `spCFrame`, `spFrame`, `spFluxcalib` and `spFluxcorr`. This method is analogous to `bosssdata.plate.Plan.get_exposure_name()`, but operates for a single target and only knows about exposures actually used in the final co-add.

Parameters

- **sequence_number** (*int*) – Science exposure sequence number, counting from zero. Must be less than our `num_exposures` attribute.
- **band** (*str*) – Must be ‘blue’ or ‘red’.
- **fctype** (*str*) – Type of exposure file whose name to return. Must be one of `spCFrame`, `spFrame`, `spFluxcalib`, `spFluxcorr`. An `spCFrame` is assumed to be uncompressed, and all other files are assumed to be compressed.

Returns

Exposure name of the form [fctype]-[cc]-[eeeeeeee].[ext] where [cc] identifies the camera (one of b1,r1,b2,r2) and [eeeeeeee] is the zero-padded exposure number. The extension [ext] is “fits” for `spCFrame` files and “fits.gz” for all other file types.

Return type `str`

Raises `ValueError` – one of the inputs is invalid.

get_pixel_mask (*exposure_index*=None, *camera*=None)

Get the pixel mask for a specified exposure or the combined coadd.

Returns the *and_mask* for coadded spectra. The entire mask is returned, including any pixels with zero inverse variance.

Parameters

- **exposure_index** (*int*) – Individual exposure to use, specified as a sequence number starting from zero, for the first exposure, and increasing up to `self.num_exposures-1`. Uses the co-added spectrum when the value is None.

- **camera** (*str*) – Which camera to use. Must be either ‘b1’, ‘b2’ (blue) or ‘r1’, ‘r2’ (red) unless `exposure_index` is `None`, in which case this argument is ignored.

Returns

Array of integers, one per pixel, encoding the mask bits defined in `bossdata.bits.SPPIXMASK` (see also http://www.sdss3.org/dr10/algorithms/bitmask_sppixmask.php).

Return type `numpy.ndarray`

get_raw_image (*sequence_number*, *band*, *flavor*=‘science’, *finder*=`None`, *mirror*=`None`)

Get a raw image file associated with one of this coadd’s exposures.

Parameters

- **sequence_number** (*int*) – The sequence number for the requested camera exposure, in the range 0 - (`num_exposures[camera]-1`).
- **band** (*str*) – Must be ‘blue’ or ‘red’.
- **flavor** (*str*) – One of science, arc, flat.
- **finder** (`bossdata.path.Finder`) – Object used to find the names of BOSS data files. If not specified, the default Finder constructor is used.
- **mirror** (`bossdata.remote.Manager`) – Object used to interact with the local mirror of BOSS data. If not specified, the default Manager constructor is used.

Returns requested raw image file.

Return type `bossdata.raw.RawImageFile`

Raises `ValueError` – one of the inputs is invalid.

get_valid_data (*exposure_index*=`None`, *camera*=`None`, *pixel_quality_mask*=`None`, *include_wdisp*=`False`, *include_sky*=`False`, *use_ivar*=`False`, *use_loglam*=`False`, *fiducial_grid*=`False`)

Get the valid data for a specified exposure or the combined coadd.

You will probably find yourself using this idiom often:

```
data = spec.get_valid_data(...)
wlen, flux, dflux = data['wavelength'][:,], data['flux'][:,], data['dflux'][:,]
```

Parameters

- **exposure_index** (*int*) – Individual exposure to use, specified as a sequence number starting from zero, for the first exposure, and increasing up to `self.num_exposures-1`. Uses the co-added spectrum when the value is `None`.
- **camera** (*str*) – Which camera to use. Must be either ‘b1’, ‘b2’ (blue) or ‘r1’, ‘r2’ (red) unless `exposure_index` is `None`, in which case this argument is ignored.
- **pixel_quality_mask** (*int*) – An integer value interpreted as a bit pattern using the bits defined in `bossdata.bits.SPPIXMASK` (see also http://www.sdss3.org/dr10/algorithms/bitmask_sppixmask.php). Any bits set in this mask are considered harmless and the corresponding spectrum pixels are assumed to contain valid data. When accessing the coadded spectrum, this mask is applied to the AND of the masks for each individual exposure. No mask is applied if this value is `None`.
- **include_wdisp** – Include a wavelength dispersion column in the returned data.
- **include_sky** – Include a sky flux column in the returned data.

- **use_ivar** – Replace `dflux` with `ivar` (inverse variance) in the returned data.
- **use_loglam** – Replace `wavelength` with `loglam` (`log10(wavelength)`) in the returned data.
- **fiducial_grid** – Return co-added data using the `fiducial wavelength grid`. If `False`, the returned array uses the native grid of the `SpecFile`, which generally trims pixels on both ends that have zero inverse variance. Set this value `True` to ensure that all co-added spectra use aligned wavelength grids when this matters.

Returns

Masked array of per-pixel records. Pixels with no valid data are included but masked. The record for each pixel has at least the following named fields: `wavelength` in Angstroms (or `loglam`), `flux` and `dflux` in $1e-17$ ergs/s/cm²/Angstrom (or `flux` and `ivar`). Wavelength values are strictly increasing and `dflux` is calculated as `ivar**-0.5` for pixels with valid data. Optional fields are `wdisp` in constant-log10-lambda pixels and `sky` in $1e-17$ ergs/s/cm²/Angstrom. The `wavelength` (or `loglam`) field is never masked and all other fields are masked when `ivar` is zero or a pipeline flag is set (and not allowed by `pixel_quality_mask`).

Return type `numpy.ma.MaskedArray`

Raises

- `ValueError` – `fiducial grid` is not supported for individual exposures.
- `RuntimeError` – co-added wavelength grid is not aligned with the `fiducial grid`.

`bosssdata.spec.get_fiducial_pixel_index(wavelength)`

Convert a wavelength to a fiducial pixel index.

The fiducial wavelength grid used by all SDSS co-added spectra is logarithmically spaced:

```
wavelength = wavelength0 * 10**(coef * index)
```

The value `coef = 1e-4` is encoded in the FITS HDU headers of SDSS coadded data files with the keyword `CD1_1` (and sometimes also `COEFF1`). The value of `wavelength0` defines `index = 0` and is similarly encoded as `CRVAL1` (and sometimes also `COEFF0`). However, its value is not constant between different SDSS co-added spectra because varying amounts of invalid data are trimmed. This function adopts the constant value 3500.26 Angstrom corresponding to `index = 0`:

```
>>> get_fiducial_pixel_index(3500.26)
0.0
```

Note that the return value is a float so that wavelengths not on the fiducial grid can be converted and detected:

```
>>> get_fiducial_pixel_index(3500.5)
0.29776960129179741
```

The calculation is automatically broadcast over an input wavelength array:

```
>>> wlen = np.arange(4000, 4400, 100)
>>> get_fiducial_pixel_index(wlen)
array([ 579.596863 ,  686.83551692,  791.4898537 ,  893.68150552])
```

Use `fiducial_pixel_index_range` for an index range that covers all SDSS spectra and `fiducial_loglam` to covert integer indices to wavelengths.

Parameters `wavelength` (*float*) – Input wavelength in Angstroms.

Returns

Array of floating-point indices relative to the fiducial wavelength grid.

Return type `numpy.ndarray`

b

- `bossdata`, [29](#)
- `bossdata.bits`, [29](#)
- `bossdata.meta`, [76](#)
- `bossdata.path`, [78](#)
- `bossdata.plate`, [81](#)
- `bossdata.plot`, [86](#)
- `bossdata.raw`, [87](#)
- `bossdata.remote`, [89](#)
- `bossdata.spec`, [92](#)

Symbols

`_25ORI_WISE` (bossdata.bits.ANCILLARY_TARGET2 attribute), 34
`_25ORI_WISE_W3` (bossdata.bits.ANCILLARY_TARGET2 attribute), 33
`_2MASSFILL` (bossdata.bits.ANCILLARY_TARGET2 attribute), 32
`_3DREJECT` (bossdata.bits.MANGA_DRP2PIXMASK attribute), 53

A

`AFTER` (bossdata.bits.M_EYEBALL attribute), 57
`AGN_BAT` (bossdata.bits.MANGA_TARGET3 attribute), 56
`AGN_OIII` (bossdata.bits.MANGA_TARGET3 attribute), 56
`AGN_PALOMAR` (bossdata.bits.MANGA_TARGET3 attribute), 56
`AGN_WISE` (bossdata.bits.MANGA_TARGET3 attribute), 57
`ALL_BLUE` (bossdata.bits.M_EYEBALL attribute), 57
`ALL_RED` (bossdata.bits.M_EYEBALL attribute), 57
`ALLPSF` (bossdata.bits.SPECIAL_TARGET1 attribute), 66
`ALLPSF_NONSTELLAR` (bossdata.bits.SPECIAL_TARGET1 attribute), 68
`ALLPSF_STELLAR` (bossdata.bits.SPECIAL_TARGET1 attribute), 68
`ALPHAFE_BAD` (bossdata.bits.APOGEE_ASPCAPFLAG attribute), 38
`ALPHAFE_WARN` (bossdata.bits.APOGEE_ASPCAPFLAG attribute), 39
`AMC` (bossdata.bits.ANCILLARY_TARGET1 attribute), 29

`AMOMENT_MAXITER` (bossdata.bits.OBJECT2 attribute), 61
`AMOMENT_SHIFT` (bossdata.bits.OBJECT2 attribute), 60
`AMOMENT_UNWEIGHTED` (bossdata.bits.OBJECT2 attribute), 61
`ANCILLARY` (bossdata.bits.BOSSTILE_STATUS attribute), 44
`ANCILLARY` (bossdata.bits.MANGA_TARGET1 attribute), 55
`ANCILLARY_ROUND2` (bossdata.bits.BOSSTILE_STATUS attribute), 44
`ANCILLARY_TARGET1` (class in bossdata.bits), 29
`ANCILLARY_TARGET2` (class in bossdata.bits), 32
`ANGST` (bossdata.bits.MANGA_TARGET3 attribute), 56
`APBIAS` (bossdata.bits.SPECIAL_TARGET1 attribute), 66
`APO1M` (bossdata.bits.APOGEE_EXTRATARG attribute), 39
`APOGEE` (bossdata.bits.ANCILLARY_TARGET2 attribute), 35
`APOGEE2_1M_TARGET` (bossdata.bits.APOGEE2_TARGET2 attribute), 37
`APOGEE2_ANCILLARY` (bossdata.bits.APOGEE2_TARGET3 attribute), 37
`APOGEE2_APOKASC` (bossdata.bits.APOGEE2_TARGET1 attribute), 36
`APOGEE2_APOKASC_DWARF` (bossdata.bits.APOGEE2_TARGET1 attribute), 36
`APOGEE2_APOKASC_GIANT` (bossdata.bits.APOGEE2_TARGET1 attribute), 36
`APOGEE2_ARGOS_OVERLAP` (bossdata.bits.APOGEE2_TARGET2 attribute),

37 APOGEE2_CALIB_CLUSTER data.bits.APOGEE2_TARGET2 (boss-attribute),	38 APOGEE2_MDWARF data.bits.APOGEE2_TARGET3 (boss-attribute),
37 APOGEE2_DSPH_CANDIDATE data.bits.APOGEE2_TARGET1 (boss-attribute),	37 APOGEE2_MEDIUM data.bits.APOGEE2_TARGET1 (boss-attribute),
35 APOGEE2_DSPH_MEMBER data.bits.APOGEE2_TARGET1 (boss-attribute),	35 APOGEE2_NO_DERED data.bits.APOGEE2_TARGET1 (boss-attribute),
35 APOGEE2_EB (bossdata.bits.APOGEE2_TARGET3 attribute), 37	36 APOGEE2_NORMAL_SAMPLE data.bits.APOGEE2_TARGET1 (boss-attribute),
37 APOGEE2_EXTERNAL_CALIB data.bits.APOGEE2_TARGET2 (boss-attribute),	36 APOGEE2_OBJECT data.bits.APOGEE2_TARGET2 (boss-attribute),
37 APOGEE2_FAINT_EXTRA data.bits.APOGEE2_TARGET1 (boss-attribute),	37 APOGEE2_ONEBIN_GT_0_3 data.bits.APOGEE2_TARGET1 (boss-attribute),
36 APOGEE2_GAIA_OVERLAP data.bits.APOGEE2_TARGET2 (boss-attribute),	36 APOGEE2_ONEBIN_GT_0_5 data.bits.APOGEE2_TARGET1 (boss-attribute),
37 APOGEE2_GALAH_OVERLAP data.bits.APOGEE2_TARGET2 (boss-attribute),	35 APOGEE2_RAVE_OVERLAP data.bits.APOGEE2_TARGET2 (boss-attribute),
37 APOGEE2_GES_OVERLAP data.bits.APOGEE2_TARGET2 (boss-attribute),	37 APOGEE2_RRLYR data.bits.APOGEE2_TARGET1 (boss-attribute),
37 APOGEE2_INTERNAL_CALIB data.bits.APOGEE2_TARGET2 (boss-attribute),	35 APOGEE2_RV_STANDARD data.bits.APOGEE2_TARGET2 (boss-attribute),
37 APOGEE2_IRAC_DERED data.bits.APOGEE2_TARGET1 (boss-attribute),	37 APOGEE2_SCI_CLUSTER data.bits.APOGEE2_TARGET1 (boss-attribute),
36 APOGEE2_KOI (bossdata.bits.APOGEE2_TARGET3 attribute), 38	36 APOGEE2_SFD_DERED data.bits.APOGEE2_TARGET1 (boss-attribute),
38 APOGEE2_KOI_CONTROL data.bits.APOGEE2_TARGET3 (boss-attribute),	36 APOGEE2_SGR_DSPH data.bits.APOGEE2_TARGET1 (boss-attribute),
37 APOGEE2_LITERATURE_CALIB data.bits.APOGEE2_TARGET2 (boss-attribute),	35 APOGEE2_SHORT data.bits.APOGEE2_TARGET1 (boss-attribute),
36 APOGEE2_LONG (bossdata.bits.APOGEE2_TARGET1 attribute), 36	36 APOGEE2_SKY (bossdata.bits.APOGEE2_TARGET2 attribute), 37
36 APOGEE2_MAGCLOUD_CANDIDATE data.bits.APOGEE2_TARGET1 (boss-attribute),	37 APOGEE2_STANDARD_STAR data.bits.APOGEE2_TARGET2 (boss-attribute),
36 APOGEE2_MAGCLOUD_MEMBER data.bits.APOGEE2_TARGET1 (boss-attribute),	37 APOGEE2_STREAM_CANDIDATE data.bits.APOGEE2_TARGET1 (boss-attribute),
35 APOGEE2_MANGA_LED data.bits.APOGEE2_TARGET1 (boss-attribute),	36 APOGEE2_STREAM_MEMBER data.bits.APOGEE2_TARGET1 (boss-attribute),
35 APOGEE2_MASSIVE_STAR data.bits.APOGEE2_TARGET3 (boss-attribute),	36

APOGEE2_SUBSTELLAR_COMPANIONS	(boss- data.bits.APOGEE2_TARGET3 attribute), 37	APOGEE_EMISSION_STAR	(boss- data.bits.APOGEE_TARGET2 attribute), 43
APOGEE2_TARGET1 (class in bossdata.bits),	35	APOGEE_EXTENDED	(boss- data.bits.APOGEE_TARGET1 attribute), 41
APOGEE2_TARGET2 (class in bossdata.bits),	37	APOGEE_EXTRATARG (class in bossdata.bits),	39
APOGEE2_TARGET3 (class in bossdata.bits),	37	APOGEE_FAINT (bossdata.bits.APOGEE_TARGET1 attribute),	42
APOGEE2_TELLURIC	(boss- data.bits.APOGEE2_TARGET2 attribute), 37	APOGEE_FAINT_EXTRA	(boss- data.bits.APOGEE_TARGET1 attribute), 42
APOGEE2_TWOBIN_0_5_TO_0_8	(boss- data.bits.APOGEE2_TARGET1 attribute), 36	APOGEE_FIRST_LIGHT	(boss- data.bits.APOGEE_TARGET1 attribute), 42
APOGEE2_TWOBIN_GT_0_8	(boss- data.bits.APOGEE2_TARGET1 attribute), 36	APOGEE_FLUX_STANDARD	(boss- data.bits.APOGEE_TARGET2 attribute), 43
APOGEE2_WASH_DWARF	(boss- data.bits.APOGEE2_TARGET1 attribute), 36	APOGEE_GC_GIANT	(boss- data.bits.APOGEE_TARGET2 attribute), 43
APOGEE2_WASH_GIANT	(boss- data.bits.APOGEE2_TARGET1 attribute), 35	APOGEE_GC_PAL1	(boss- data.bits.APOGEE_TARGET1 attribute), 42
APOGEE2_WASH_NOCLASS	(boss- data.bits.APOGEE2_TARGET1 attribute), 36	APOGEE_GC_SUPER_GIANT	(boss- data.bits.APOGEE_TARGET2 attribute), 43
APOGEE2_WISE_DERED	(boss- data.bits.APOGEE2_TARGET1 attribute), 36	APOGEE_HIRES (bossdata.bits.APOGEE_TARGET1 attribute),	41
APOGEE2_YOUNG_CLUSTER	(boss- data.bits.APOGEE2_TARGET3 attribute), 38	APOGEE_INTERMEDIATE	(boss- data.bits.APOGEE_TARGET1 attribute), 41
APOGEE_ANCILLARY	(boss- data.bits.APOGEE_TARGET1 attribute), 42	APOGEE_IRAC_DERED	(boss- data.bits.APOGEE_TARGET1 attribute), 42
APOGEE_ASPCAPFLAG (class in bossdata.bits),	38	APOGEE_KEPLER_COOLDWARF	(boss- data.bits.APOGEE_TARGET2 attribute), 43
APOGEE_BRIGHT (bossdata.bits.APOGEE_TARGET1 attribute),	42	APOGEE_KEPLER_EB	(boss- data.bits.APOGEE_TARGET1 attribute), 42
APOGEE_CALIB_CLUSTER	(boss- data.bits.APOGEE_TARGET2 attribute), 43	APOGEE_KEPLER_HOST	(boss- data.bits.APOGEE_TARGET1 attribute), 43
APOGEE_CHECKED	(boss- data.bits.APOGEE_TARGET1 attribute), 41	APOGEE_KEPLER_SEISMO	(boss- data.bits.APOGEE_TARGET1 attribute), 42
APOGEE_CHECKED	(boss- data.bits.APOGEE_TARGET2 attribute), 43	APOGEE_LONG (bossdata.bits.APOGEE_TARGET1 attribute),	42
APOGEE_DISK_RED_GIANT	(boss- data.bits.APOGEE_TARGET1 attribute), 42	APOGEE_LONGBAR	(boss- data.bits.APOGEE_TARGET2 attribute), 43
APOGEE_DO_NOT_OBSERVE	(boss- data.bits.APOGEE_TARGET1 attribute), 42	APOGEE_M31_CLUSTER	(boss- data.bits.APOGEE_TARGET1 attribute),
APOGEE_EMBEDDEDCLUSTER_STAR	(boss- data.bits.APOGEE_TARGET2 attribute), 43		

APOGEE_MASSIVE_STAR	(boss-attribute),	APOGEE_TARGET1	attribute),
APOGEE_MDWARF	(boss-attribute),	APOGEE_WASH_GIANT	(boss-attribute),
APOGEE_MEDIUM	(boss-attribute),	APOGEE_WISE_DERED	(boss-attribute),
APOGEE_MIRCLUSTER_STAR	(boss-attribute),	ARCFOCUS	(bosssdata.bits.MANGA_DRP2QUAL attribute),
APOGEE_NO_DERED	(boss-attribute),	ASTROMBAD	(bosssdata.bits.CALIB_STATUS attribute),
APOGEE_OLD_STAR	(boss-attribute),	ASYMMETRIC	(bosssdata.bits.T_EYEBALL attribute),
APOGEE_PARAMFLAG (class in bossdata.bits),	39		
APOGEE_PIXMASK (class in bossdata.bits),	40		
APOGEE_RV_STANDARD	(boss-attribute),		
APOGEE_SCI_CLUSTER	(boss-attribute),		
APOGEE_SEGUE_OVERLAP	(boss-attribute),		
APOGEE_SERENDIPITOUS	(boss-attribute),		
APOGEE_SFD_DERED	(boss-attribute),		
APOGEE_SGR_DSPH	(boss-attribute),		
APOGEE_SHORT	(bosssdata.bits.APOGEE_TARGET1 attribute),		
APOGEE_STANDARD_STAR	(boss-attribute),		
APOGEE_STARFLAG (class in bossdata.bits),	40		
APOGEE_TARGET1 (class in bossdata.bits),	41		
APOGEE_TARGET2 (class in bossdata.bits),	43		
APOGEE_TELLURIC	(boss-attribute),		
APOGEE_TELLURIC_BAD	(boss-attribute),		
APOGEE WASH DWARF	(boss-		

BADDITHER (bossdata.bits.MANGA_DRP2QUAL attribute), 53
 BADERR (bossdata.bits.APOGEE_PIXMASK attribute), 40
 BADFLAT (bossdata.bits.APOGEE_PIXMASK attribute), 40
 BADFLAT (bossdata.bits.MANGA_DRP2PIXMASK attribute), 52
 BADFLAT (bossdata.bits.SPPIXMASK attribute), 69
 BADFLUX (bossdata.bits.MANGA_DRP3QUAL attribute), 54
 BADFLUXFACTOR (bossdata.bits.MANGA_DRP2PIXMASK attribute), 52
 BADFLUXFACTOR (bossdata.bits.SPPIXMASK attribute), 69
 BADIFU (bossdata.bits.MANGA_DRP2QUAL attribute), 53
 BADOMEGA (bossdata.bits.MANGA_DRP3QUAL attribute), 54
 BADPIX (bossdata.bits.APOGEE_PIXMASK attribute), 40
 BADPIX (bossdata.bits.MANGA_DRP2PIXMASK attribute), 52
 BADSET (bossdata.bits.MANGA_DRP3QUAL attribute), 54
 BADSKY (bossdata.bits.OBJECT1 attribute), 58
 BADSKYCHI (bossdata.bits.MANGA_DRP2PIXMASK attribute), 52
 BADSKYCHI (bossdata.bits.SPPIXMASK attribute), 69
 BADSKYFIBER (bossdata.bits.MANGA_DRP2PIXMASK attribute), 53
 BADSKYFIBER (bossdata.bits.SPPIXMASK attribute), 69
 BADTRACE (bossdata.bits.MANGA_DRP2PIXMASK attribute), 52
 BADTRACE (bossdata.bits.SPPIXMASK attribute), 68
 BAR (bossdata.bits.T_EYEBALL attribute), 73
 BCG (bossdata.bits.MANGA_TARGET3 attribute), 56
 BCG (bossdata.bits.SPECIAL_TARGET1 attribute), 67
 BEFORE (bossdata.bits.M_EYEBALL attribute), 57
 BENT_RADIO (bossdata.bits.SPECIAL_TARGET1 attribute), 67
 BINNED1 (bossdata.bits.OBJECT1 attribute), 58
 BINNED2 (bossdata.bits.OBJECT1 attribute), 58
 BINNED4 (bossdata.bits.OBJECT1 attribute), 58
 BINNED_CENTER (bossdata.bits.OBJECT2 attribute), 61
 bitmask_from_text() (in module bossdata.bits), 73
 BLAZGRFLAT (bossdata.bits.ANCILLARY_TARGET1 attribute), 30
 BLAZGRQSO (bossdata.bits.ANCILLARY_TARGET1 attribute), 32
 BLAZGVAR (bossdata.bits.ANCILLARY_TARGET1 attribute), 30
 BLAZGX (bossdata.bits.ANCILLARY_TARGET1 attribute), 31
 BLAZGXQSO (bossdata.bits.ANCILLARY_TARGET1 attribute), 32
 BLAZGXR (bossdata.bits.ANCILLARY_TARGET1 attribute), 31
 BLAZR (bossdata.bits.ANCILLARY_TARGET1 attribute), 31
 BLAZXR (bossdata.bits.ANCILLARY_TARGET1 attribute), 32
 BLAZXRSAM (bossdata.bits.ANCILLARY_TARGET1 attribute), 29
 BLAZXRVAR (bossdata.bits.ANCILLARY_TARGET1 attribute), 29
 BLENDED (bossdata.bits.OBJECT1 attribute), 59
 BLUE_CORE (bossdata.bits.T_EYEBALL attribute), 72
 BLUE_RADIO (bossdata.bits.ANCILLARY_TARGET1 attribute), 31
 BLUEFIBER (bossdata.bits.BOSSTILE_STATUS attribute), 44
 BOSS_TARGET1 (class in bossdata.bits), 45
 bossdata (module), 29
 bossdata.bits (module), 29
 bossdata.meta (module), 76
 bossdata.path (module), 78
 bossdata.plate (module), 81
 bossdata.plot (module), 86
 bossdata.raw (module), 87
 bossdata.remote (module), 89
 bossdata.spec (module), 92
 BOSSTARGET (bossdata.bits.BOSSTILE_STATUS attribute), 44
 BOSSTILE_STATUS (class in bossdata.bits), 44
 BRIGHT (bossdata.bits.OBJECT1 attribute), 58
 BRIGHT_NEIGHBOR (bossdata.bits.APOGEE_STARFLAG attribute), 41
 BRIGHTERL (bossdata.bits.ANCILLARY_TARGET1 attribute), 30
 BRIGHTERM (bossdata.bits.ANCILLARY_TARGET1 attribute), 29
 BRIGHTEST_FLUXMATCH (bossdata.bits.FLUXMATCH_STATUS attribute), 51
 BRIGHTEST_GALAXY_CHILD (bossdata.bits.OBJECT2 attribute), 61
 BRIGHTGAL (bossdata.bits.ANCILLARY_TARGET1 attribute), 31
 BRIGHTSKY (bossdata.bits.MANGA_DRP2PIXMASK attribute), 52
 BRIGHTSKY (bossdata.bits.SPPIXMASK attribute), 69
 BUNDLE_HOLE (bossdata.bits.APOGEE_TARGET2

attribute), 43
 BUNDLE_HOLE (bossdata.bits.SEGUE2_TARGET2 attribute), 66
 BUNDLE_HOLE (bossdata.bits.TTARGET attribute), 71
 by_fiber() (in module bossdata.plot), 86

C

CALIB_STATUS (class in bossdata.bits), 46
 CALRANGE_BAD (bossdata.bits.APOGEE_PARAMFLAG attribute), 40
 CALRANGE_WARN (bossdata.bits.APOGEE_PARAMFLAG attribute), 39
 camera (bossdata.raw.RawImageFile attribute), 88
 CANONICAL_BAND (bossdata.bits.OBJECT2 attribute), 60
 CANONICAL_CENTER (bossdata.bits.OBJECT1 attribute), 59
 CENTER_OFF_AIMAGE (bossdata.bits.OBJECT2 attribute), 60
 CENTERPOST (bossdata.bits.BOSSTILE_STATUS attribute), 45
 CFE_BAD (bossdata.bits.APOGEE_ASPCAPFLAG attribute), 38
 CFE_WARN (bossdata.bits.APOGEE_ASPCAPFLAG attribute), 39
 CHANDRAV1 (bossdata.bits.ANCILLARY_TARGET1 attribute), 32
 CHI2_BAD (bossdata.bits.APOGEE_ASPCAPFLAG attribute), 38
 CHI2_WARN (bossdata.bits.APOGEE_ASPCAPFLAG attribute), 38
 CHILD (bossdata.bits.OBJECT1 attribute), 58
 CLEAR (bossdata.bits.IMAGE_STATUS attribute), 51
 CLOUDY (bossdata.bits.IMAGE_STATUS attribute), 51
 CLUSTER_MEMBER (bossdata.bits.ANCILLARY_TARGET2 attribute), 34
 COLOR_ENHANCED_COM (bossdata.bits.MANGA_TARGET1 attribute), 54
 COLOR_ENHANCED_COM2 (bossdata.bits.MANGA_TARGET1 attribute), 55
 COLOR_ENHANCED_v1_1_0 (bossdata.bits.MANGA_TARGET1 attribute), 54
 COLOR_ENHANCED_v1_2_0 (bossdata.bits.MANGA_TARGET1 attribute), 55
 COLORTE_BAD (bossdata.bits.APOGEE_ASPCAPFLAG attribute), 38

COLORTE_WARN (bossdata.bits.APOGEE_ASPCAPFLAG attribute), 38
 COMBINEREJ (bossdata.bits.MANGA_DRP2PIXMASK attribute), 53
 COMBINEREJ (bossdata.bits.SPPIXMASK attribute), 69
 COMMISSIONING (bossdata.bits.APOGEE_EXTRATARG attribute), 39
 COMMISSIONING (bossdata.bits.APOGEE_STARFLAG attribute), 41
 COMMISSIONING_STAR (bossdata.bits.SPECIAL_TARGET1 attribute), 66
 COROTGES (bossdata.bits.ANCILLARY_TARGET2 attribute), 35
 COROTGESAPOG (bossdata.bits.ANCILLARY_TARGET2 attribute), 33
 COSMIC (bossdata.bits.MANGA_DRP2PIXMASK attribute), 52
 CR (bossdata.bits.OBJECT1 attribute), 58
 create_meta_full() (in module bossdata.meta), 77
 create_meta_lite() (in module bossdata.meta), 77
 CRITICAL (bossdata.bits.MANGA_DRP3QUAL attribute), 54
 CROSSTALK (bossdata.bits.MANGA_DRP2PIXMASK attribute), 52
 CROSSTALK (bossdata.bits.SPPIXMASK attribute), 68
 CRPIX (bossdata.bits.APOGEE_PIXMASK attribute), 40
 CXOBRIGHT (bossdata.bits.ANCILLARY_TARGET1 attribute), 32
 CXOGRIZ (bossdata.bits.ANCILLARY_TARGET1 attribute), 30
 CXORED (bossdata.bits.ANCILLARY_TARGET1 attribute), 29

D

data (bossdata.raw.RawImageFile attribute), 88
 Database (class in bossdata.meta), 76
 DEAD_CCD (bossdata.bits.IMAGE_STATUS attribute), 51
 DEADFIBER (bossdata.bits.MANGA_DRP2PIXMASK attribute), 52
 DEADFIBER (bossdata.bits.MANGA_DRP3PIXMASK attribute), 54
 DEBLEND_DEGENERATE (bossdata.bits.OBJECT2 attribute), 61
 DEBLEND_NOPEAK (bossdata.bits.OBJECT2 attribute), 61

DEBLEND_PEEPHOLE (bossdata.bits.OBJECT2 attribute), 44
 DEBLEND_PRUNED (bossdata.bits.OBJECT1 attribute), 59
 DEBLEND_TOO_MANY_PEAKS (bossdata.bits.OBJECT1 attribute), 59
 DEBLEND_UNASSIGNED_FLUX (bossdata.bits.OBJECT2 attribute), 61
 DEBLENDED_AS_MOVING (bossdata.bits.OBJECT2 attribute), 60
 DEBLENDED_AS_PSF (bossdata.bits.OBJECT1 attribute), 58
 DEBLENDED_AT_EDGE (bossdata.bits.OBJECT2 attribute), 60
 decode_bitmask() (in module bossdata.bits), 74
 DECOLLIDED (bossdata.bits.BOSSTILE_STATUS attribute), 44
 DEEP_COMA (bossdata.bits.MANGA_TARGET3 attribute), 56
 DEEP_GALAXY_RED (bossdata.bits.SPECIAL_TARGET1 attribute), 67
 DEEP_GALAXY_RED_II (bossdata.bits.SPECIAL_TARGET1 attribute), 67
 DEFAULT (bossdata.bits.CALIB_STATUS attribute), 47
 define_bitmask() (in module bossdata.bits), 74
 DISK (bossdata.bits.T_EYEBALL attribute), 72
 DISKEMITTER_REPEAT (bossdata.bits.ANCILLARY_TARGET2 attribute), 33
 DISKMASS (bossdata.bits.MANGA_TARGET3 attribute), 56
 DISKSTAR (bossdata.bits.SPECIAL_TARGET1 attribute), 66
 DO_NOT_OBSERVE (bossdata.bits.EBOSS_TARGET0 attribute), 47
 DO_NOT_OBSERVE (bossdata.bits.EBOSS_TARGET1 attribute), 48
 DONE (bossdata.bits.M_EYEBALL attribute), 57
 DONE (bossdata.bits.Q_EYEBALL attribute), 62
 DONE (bossdata.bits.T_EYEBALL attribute), 71
 DONOTUSE (bossdata.bits.MANGA_DRP3PIXMASK attribute), 54
 DOUBLE_STAR (bossdata.bits.Q_EYEBALL attribute), 61
 DOUBLE_Z (bossdata.bits.Q_EYEBALL attribute), 63
 download() (bossdata.remote.Manager method), 90
 DR9_CALIB_TARGET (bossdata.bits.EBOSS_TARGET0 attribute), 48
 DRY (bossdata.bits.M_EYEBALL attribute), 57
 DUPLICATE (bossdata.bits.APOGEE_EXTRATARG attribute), 39
 DUPLICATE (bossdata.bits.BOSSTILE_STATUS attribute), 44
 DUPLICATE_PRIMARY (bossdata.bits.BOSSTILE_STATUS attribute), 45
 DUPLICATE_TILED (bossdata.bits.BOSSTILE_STATUS attribute), 45
 DURING (bossdata.bits.M_EYEBALL attribute), 57
 DUST_ASYMMETRY (bossdata.bits.T_EYEBALL attribute), 71
 DUST_LANE (bossdata.bits.T_EYEBALL attribute), 71
 DWARF (bossdata.bits.MANGA_TARGET3 attribute), 56

E

EBOSS_TARGET0 (class in bossdata.bits), 47
 EBOSS_TARGET1 (class in bossdata.bits), 48
 EBOSS_TARGET2 (class in bossdata.bits), 49
 EDGE (bossdata.bits.OBJECT1 attribute), 58
 EDGE (bossdata.bits.Q_EYEBALL attribute), 62
 EDGE_ON_WINDS (bossdata.bits.MANGA_TARGET3 attribute), 56
 ELAIS_N1_FIRST (bossdata.bits.ANCILLARY_TARGET2 attribute), 34
 ELAIS_N1_GMRT_GARN (bossdata.bits.ANCILLARY_TARGET2 attribute), 33
 ELAIS_N1_GMRT_TAYLOR (bossdata.bits.ANCILLARY_TARGET2 attribute), 33
 ELAIS_N1_JVLA (bossdata.bits.ANCILLARY_TARGET2 attribute), 33
 ELAIS_N1_LOFAR (bossdata.bits.ANCILLARY_TARGET2 attribute), 34
 ELG (bossdata.bits.ANCILLARY_TARGET1 attribute), 30
 ELG_DES_TEST1 (bossdata.bits.EBOSS_TARGET2 attribute), 50
 ELG_DESI_TEST1 (bossdata.bits.EBOSS_TARGET2 attribute), 49
 ELG_GRIW_TEST1 (bossdata.bits.EBOSS_TARGET2 attribute), 50
 ELG_SCUSS_TEST1 (bossdata.bits.EBOSS_TARGET2 attribute), 50
 ELG_SDSS_TEST1 (bossdata.bits.EBOSS_TARGET2 attribute), 50
 ELG_TEST1 (bossdata.bits.EBOSS_TARGET1 attribute), 49
 ELG_UGRIZW_TEST1 (bossdata.bits.EBOSS_TARGET2 attribute), 50

ELG_UGRIZWbright_TEST1 (bossdata.bits.EBOSS_TARGET2 attribute), 50
 ELLIPFAINT (bossdata.bits.OBJECT1 attribute), 58
 ELLIPTICAL (bossdata.bits.T_EYEBALL attribute), 72
 exposure_id (bossdata.raw.RawImageFile attribute), 88
 Exposures (class in bossdata.spec), 92
 extract_sdss_bitmasks() (in module bossdata.bits), 75
 EXTRACTBAD (bossdata.bits.MANGA_DRP2QUAL attribute), 53
 EXTRACTBRIGHT (bossdata.bits.MANGA_DRP2QUAL attribute), 53

F

FAINT_ELG (bossdata.bits.ANCILLARY_TARGET2 attribute), 35
 FAINT_HIZ_LRG (bossdata.bits.ANCILLARY_TARGET2 attribute), 35
 FAINT_LRG (bossdata.bits.SPECIAL_TARGET1 attribute), 66
 FAINT_QSO (bossdata.bits.SPECIAL_TARGET1 attribute), 66
 FAINTERL (bossdata.bits.ANCILLARY_TARGET1 attribute), 30
 FAINTERM (bossdata.bits.ANCILLARY_TARGET1 attribute), 30
 FBQSBAL (bossdata.bits.ANCILLARY_TARGET1 attribute), 31
 FF_PETALS (bossdata.bits.IMAGE_STATUS attribute), 51
 FIBER_FLUXMATCH (bossdata.bits.FLUXMATCH_STATUS attribute), 51
 FILLER (bossdata.bits.BOSSTILE_STATUS attribute), 44
 FILLER (bossdata.bits.MANGA_TARGET1 attribute), 55
 Finder (class in bossdata.path), 79
 FLARE1 (bossdata.bits.ANCILLARY_TARGET1 attribute), 32
 FLARE2 (bossdata.bits.ANCILLARY_TARGET1 attribute), 32
 flavor (bossdata.raw.RawImageFile attribute), 88
 FLECK (bossdata.bits.Q_EYEBALL attribute), 62
 FLUXMATCH_STATUS (class in bossdata.bits), 50
 focal_plane() (in module bossdata.plot), 86
 FORESTAR (bossdata.bits.MANGA_DRP3PIXMASK attribute), 54
 FrameFile (class in bossdata.plate), 81
 FSTAR (bossdata.bits.SPECIAL_TARGET1 attribute), 67
 FULLCLOUD (bossdata.bits.MANGA_DRP2QUAL attribute), 53

FULLREJECT (bossdata.bits.MANGA_DRP2PIXMASK attribute), 52
 FULLREJECT (bossdata.bits.SPPIXMASK attribute), 68

G

GAL_CMASS (bossdata.bits.BOSS_TARGET1 attribute), 45
 GAL_CMASS_ALL (bossdata.bits.BOSS_TARGET1 attribute), 45
 GAL_CMASS_COMM (bossdata.bits.BOSS_TARGET1 attribute), 46
 GAL_CMASS_SPARSE (bossdata.bits.BOSS_TARGET1 attribute), 45
 GAL_IFIBER2_FAINT (bossdata.bits.BOSS_TARGET1 attribute), 45
 GAL_LODPERP_DEPRECATED (bossdata.bits.BOSS_TARGET1 attribute), 45
 GAL_LOZ (bossdata.bits.BOSS_TARGET1 attribute), 45
 GAL_NEAR_QSO (bossdata.bits.ANCILLARY_TARGET1 attribute), 32
 GALAXY (bossdata.bits.TARGET attribute), 70
 GALAXY_BIG (bossdata.bits.TARGET attribute), 70
 GALAXY_BRIGHT_CORE (bossdata.bits.TARGET attribute), 70
 GALAXY_RED (bossdata.bits.TARGET attribute), 70
 GALAXY_RED_II (bossdata.bits.TARGET attribute), 70
 GES (bossdata.bits.ANCILLARY_TARGET2 attribute), 32
 get() (bossdata.remote.Manager method), 91
 get_amplifier_bias() (bossdata.raw.RawImageFile method), 88
 get_amplifier_region() (bossdata.raw.RawImageFile method), 88
 get_data() (bossdata.raw.RawImageFile method), 88
 get_exposure_hdu() (bossdata.spec.SpecFile method), 94
 get_exposure_name() (bossdata.plate.Plan method), 83
 get_exposure_name() (bossdata.spec.Exposures method), 92
 get_exposure_name() (bossdata.spec.SpecFile method), 94
 get_fiber_offsets() (bossdata.plate.FrameFile method), 81
 get_fiber_offsets() (bossdata.plate.PlateFile method), 84
 get_fiducial_pixel_index() (in module bossdata.spec), 96
 get_info() (bossdata.spec.Exposures method), 92
 get_num_fibers() (in module bossdata.plate), 86
 get_pixel_mask() (bossdata.spec.SpecFile method), 94
 get_pixel_masks() (bossdata.plate.FrameFile method), 82
 get_pixel_masks() (bossdata.plate.PlateFile method), 84
 get_plate_mjd_list() (in module bossdata.meta), 78
 get_plate_path() (bossdata.path.Finder method), 79
 get_plate_plan_path() (bossdata.path.Finder method), 79
 get_plate_spec_path() (bossdata.path.Finder method), 79

- [get_platelist_path\(\)](#) (bossdata.path.Finder method), [80](#)
[get_quasar_catalog_path\(\)](#) (bossdata.path.Finder method), [80](#)
[get_raw_image\(\)](#) (bossdata.spec.Exposures method), [93](#)
[get_raw_image\(\)](#) (bossdata.spec.SpecFile method), [95](#)
[get_raw_path\(\)](#) (bossdata.path.Finder method), [80](#)
[get_sp_all_path\(\)](#) (bossdata.path.Finder method), [80](#)
[get_spec_path\(\)](#) (bossdata.path.Finder method), [80](#)
[get_spectrograph_index\(\)](#) (bossdata.plate.Plan method), [83](#)
[get_valid_data\(\)](#) (bossdata.plate.FrameFile method), [82](#)
[get_valid_data\(\)](#) (bossdata.plate.PlateFile method), [84](#)
[get_valid_data\(\)](#) (bossdata.spec.SpecFile method), [95](#)
[get_y\(\)](#) (bossdata.plate.TraceSet method), [85](#)
[GOOD_Z](#) (bossdata.bits.Q_EYEBALL attribute), [63](#)
[GRIDEDGE_BAD](#) (bossdata.bits.APOGEE_PARAMFLAG attribute), [39](#)
[GRIDEDGE_WARN](#) (bossdata.bits.APOGEE_PARAMFLAG attribute), [39](#)
[GUIDE_STAR](#) (bossdata.bits.APOGEE_TARGET2 attribute), [43](#)
[GUIDE_STAR](#) (bossdata.bits.SEGUE2_TARGET2 attribute), [66](#)
[GUIDE_STAR](#) (bossdata.bits.TTARGET attribute), [71](#)
- ## H
- [HAS_SATUR_DN](#) (bossdata.bits.OBJECT2 attribute), [61](#)
[header](#) (bossdata.raw.RawImageFile attribute), [88](#)
[HIGHSCAT](#) (bossdata.bits.MANGA_DRP2QUAL attribute), [53](#)
[HIGHSCAT](#) (bossdata.bits.MANGA_DRP3QUAL attribute), [54](#)
[HII](#) (bossdata.bits.Q_EYEBALL attribute), [62](#)
[HII_REGIONS](#) (bossdata.bits.T_EYEBALL attribute), [72](#)
[HIPM](#) (bossdata.bits.SPECIAL_TARGET1 attribute), [66](#)
[HIZ_LRG](#) (bossdata.bits.ANCILLARY_TARGET2 attribute), [32](#)
[HIZQSO82](#) (bossdata.bits.ANCILLARY_TARGET2 attribute), [33](#)
[HIZQSOIR](#) (bossdata.bits.ANCILLARY_TARGET2 attribute), [33](#)
[HOT_STD](#) (bossdata.bits.SEGUE2_TARGET2 attribute), [66](#)
[HOT_STD](#) (bossdata.bits.TTARGET attribute), [71](#)
[HPM](#) (bossdata.bits.ANCILLARY_TARGET1 attribute), [31](#)
[HYADES_MSTAR](#) (bossdata.bits.SPECIAL_TARGET1 attribute), [67](#)
- ## I
- [IAMSASERS](#) (bossdata.bits.ANCILLARY_TARGET2 attribute), [35](#)
[IGNORE_PRIORITY](#) (bossdata.bits.BOSSTILE_STATUS attribute), [45](#)
[IMAGE_STATUS](#) (class in bossdata.bits), [51](#)
[IN_HUGE_OBJECT](#) (bossdata.bits.Q_EYEBALL attribute), [62](#)
[INCOMPLETE_PROFILE](#) (bossdata.bits.OBJECT1 attribute), [59](#)
[INCREMENT_CALIB](#) (bossdata.bits.CALIB_STATUS attribute), [47](#)
[INTERNAL_REFLECTION](#) (bossdata.bits.Q_EYEBALL attribute), [62](#)
[INTERP](#) (bossdata.bits.OBJECT1 attribute), [59](#)
[INTERP_CENTER](#) (bossdata.bits.OBJECT2 attribute), [60](#)
[IRREGULAR](#) (bossdata.bits.T_EYEBALL attribute), [72](#)
[IS_STAR](#) (bossdata.bits.Q_EYEBALL attribute), [62](#)
- ## K
- [KNOWN_OBJECT](#) (bossdata.bits.BOSSTILE_STATUS attribute), [44](#)
[KOE2023_STAR](#) (bossdata.bits.ANCILLARY_TARGET2 attribute), [33](#)
[KOE2023BSTAR](#) (bossdata.bits.ANCILLARY_TARGET2 attribute), [34](#)
[KOE2068_STAR](#) (bossdata.bits.ANCILLARY_TARGET2 attribute), [34](#)
[KOE2068BSTAR](#) (bossdata.bits.ANCILLARY_TARGET2 attribute), [32](#)
[KOEKAP_STAR](#) (bossdata.bits.ANCILLARY_TARGET2 attribute), [32](#)
[KOEKAPBSTAR](#) (bossdata.bits.ANCILLARY_TARGET2 attribute), [33](#)
[KQSO_BOSS](#) (bossdata.bits.ANCILLARY_TARGET2 attribute), [33](#)
- ## L
- [LARGESHIFT](#) (bossdata.bits.MANGA_DRP2PIXMASK attribute), [52](#)
[LARGESHIFT](#) (bossdata.bits.SPPIXMASK attribute), [69](#)
[LBG](#) (bossdata.bits.ANCILLARY_TARGET2 attribute), [33](#)
[LBQSBAL](#) (bossdata.bits.ANCILLARY_TARGET1 attribute), [30](#)
[LETTERS](#) (bossdata.bits.MANGA_TARGET3 attribute), [56](#)

LIGHT_TRAP (bossdata.bits.APOGEE_TARGET2 attribute), 43

LIGHT_TRAP (bossdata.bits.SEGUE2_TARGET2 attribute), 65

LIGHT_TRAP (bossdata.bits.TTARGET attribute), 71

LITTLE_COVERAGE (bossdata.bits.ZWARNING attribute), 73

LITTROW_GHOST (bossdata.bits.APOGEE_PIXMASK attribute), 40

LOCAL_EDGE (bossdata.bits.OBJECT2 attribute), 61

local_path() (bossdata.remote.Manager method), 91

LOGG_BAD (bossdata.bits.APOGEE_ASPCAPFLAG attribute), 39

LOGG_WARN (bossdata.bits.APOGEE_ASPCAPFLAG attribute), 39

LOW_MET (bossdata.bits.ANCILLARY_TARGET1 attribute), 31

LOW_SNR (bossdata.bits.APOGEE_STARFLAG attribute), 41

LOWCOV (bossdata.bits.MANGA_DRP3PIXMASK attribute), 54

LOWEXPTIME (bossdata.bits.MANGA_DRP2QUAL attribute), 53

LOWFLAT (bossdata.bits.MANGA_DRP2PIXMASK attribute), 52

LOWFLAT (bossdata.bits.SPPIXMASK attribute), 68

LOWZ_ANNIS (bossdata.bits.SPECIAL_TARGET1 attribute), 67

LOWZ_GALAXY (bossdata.bits.SPECIAL_TARGET1 attribute), 68

LOWZ_LOVEDAY (bossdata.bits.SPECIAL_TARGET1 attribute), 66

LRG1_IDROP (bossdata.bits.EBOSS_TARGET1 attribute), 48

LRG1_WISE (bossdata.bits.EBOSS_TARGET1 attribute), 49

LRG_IZW (bossdata.bits.EBOSS_TARGET0 attribute), 48

LRG_KNOWN (bossdata.bits.EBOSS_TARGET1 attribute), 49

LRG_RIW (bossdata.bits.EBOSS_TARGET0 attribute), 47

LRG_ROUND3 (bossdata.bits.ANCILLARY_TARGET2 attribute), 35

MANGA_DRP3PIXMASK (class in bossdata.bits), 54

MANGA_DRP3QUAL (class in bossdata.bits), 54

MANGA_TARGET1 (class in bossdata.bits), 54

MANGA_TARGET2 (class in bossdata.bits), 55

MANGA_TARGET3 (class in bossdata.bits), 56

MANY_OUTLIERS (bossdata.bits.ZWARNING attribute), 73

MANYBADCOLUMNS (bossdata.bits.MANGA_DRP2PIXMASK attribute), 53

MANYBADCOLUMNS (bossdata.bits.SPPIXMASK attribute), 69

MANYPETRO (bossdata.bits.OBJECT1 attribute), 58

MANYR50 (bossdata.bits.OBJECT1 attribute), 58

MANYR90 (bossdata.bits.OBJECT1 attribute), 59

MANYREJECTED (bossdata.bits.MANGA_DRP2PIXMASK attribute), 53

MANYREJECTED (bossdata.bits.SPPIXMASK attribute), 69

MASSIVE (bossdata.bits.MANGA_TARGET3 attribute), 57

MAYBE_CR (bossdata.bits.OBJECT2 attribute), 60

MAYBE_EGHOST (bossdata.bits.OBJECT2 attribute), 60

MERGER (bossdata.bits.T_EYEBALL attribute), 72

METALS_BAD (bossdata.bits.APOGEE_ASPCAPFLAG attribute), 38

METALS_WARN (bossdata.bits.APOGEE_ASPCAPFLAG attribute), 38

MIDLEVEL_PRIORITY (bossdata.bits.BOSSTILE_STATUS attribute), 44

MIXED_REDBLUE (bossdata.bits.M_EYEBALL attribute), 57

MOVED (bossdata.bits.OBJECT1 attribute), 58

MSTURNOFF (bossdata.bits.SPECIAL_TARGET1 attribute), 67

MTEMP (bossdata.bits.ANCILLARY_TARGET1 attribute), 30

MULTIPLE (bossdata.bits.M_EYEBALL attribute), 57

MWA (bossdata.bits.MANGA_TARGET3 attribute), 56

N

NAKED (bossdata.bits.BOSSTILE_STATUS attribute), 44

NEAR_NEIGHBORS (bossdata.bits.T_EYEBALL attribute), 72

NEARBADPIXEL (bossdata.bits.MANGA_DRP2PIXMASK attribute), 52

NEARBADPIXEL (bossdata.bits.SPPIXMASK attribute), 69

M

M_EYEBALL (class in bossdata.bits), 57

MAIN (bossdata.bits.VAGC_SELECT attribute), 73

MAJOR (bossdata.bits.M_EYEBALL attribute), 57

Manager (class in bossdata.remote), 90

MANGA_DAPQUAL (class in bossdata.bits), 51

MANGA_DRP2PIXMASK (class in bossdata.bits), 51

MANGA_DRP2QUAL (class in bossdata.bits), 53

- NEARWHOPPER (bossdata.bits.MANGA_DRP2PIXMASK attribute), 53
- NEARWHOPPER (bossdata.bits.SPPIXMASK attribute), 69
- NEED_BIGGER_IMAGE (bossdata.bits.Q_EYEBALL attribute), 62
- NEGATIVE_EMISSION (bossdata.bits.ZWARNING attribute), 73
- NEGATIVE_MODEL (bossdata.bits.ZWARNING attribute), 73
- NEGATIVE_QSO_FIT (bossdata.bits.Q_EYEBALL attribute), 62
- NFE_BAD (bossdata.bits.APOGEE_ASPCAPFLAG attribute), 39
- NFE_WARN (bossdata.bits.APOGEE_ASPCAPFLAG attribute), 38
- NO_ASPCAP_RESULT (bossdata.bits.APOGEE_ASPCAPFLAG attribute), 39
- NO_UBERCAL (bossdata.bits.CALIB_STATUS attribute), 47
- NOCOV (bossdata.bits.MANGA_DRP3PIXMASK attribute), 54
- NODATA (bossdata.bits.MANGA_DRP2PIXMASK attribute), 52
- NODATA (bossdata.bits.SPPIXMASK attribute), 69
- NODATA (bossdata.bits.ZWARNING attribute), 73
- NODEBLEND (bossdata.bits.OBJECT1 attribute), 58
- NODEBLEND_MOVING (bossdata.bits.OBJECT2 attribute), 61
- NOISY_CCD (bossdata.bits.IMAGE_STATUS attribute), 51
- NONE (bossdata.bits.MANGA_TARGET1 attribute), 55
- NONE (bossdata.bits.MANGA_TARGET2 attribute), 56
- NONE (bossdata.bits.MANGA_TARGET3 attribute), 57
- NONMATCH_FLUXMATCH (bossdata.bits.FLUXMATCH_STATUS attribute), 51
- NOPARENT_FLUXMATCH (bossdata.bits.FLUXMATCH_STATUS attribute), 51
- NOPESTRO (bossdata.bits.OBJECT1 attribute), 59
- NOPESTRO_BIG (bossdata.bits.OBJECT1 attribute), 59
- NOPLUG (bossdata.bits.MANGA_DRP2PIXMASK attribute), 52
- NOPLUG (bossdata.bits.SPPIXMASK attribute), 68
- NOPROFILE (bossdata.bits.OBJECT1 attribute), 58
- NOSKY (bossdata.bits.APOGEE_PIXMASK attribute), 40
- NOSKY (bossdata.bits.MANGA_DRP2PIXMASK attribute), 52
- NOSKY (bossdata.bits.SPPIXMASK attribute), 68
- NOSTOKES (bossdata.bits.OBJECT1 attribute), 59
- NOT_MAIN (bossdata.bits.APOGEE_EXTRATARG attribute), 39
- NOT_MERGER (bossdata.bits.M_EYEBALL attribute), 57
- NOT_TILED_TARGET (bossdata.bits.BOSSTILE_STATUS attribute), 44
- NOTCHECKED (bossdata.bits.OBJECT1 attribute), 59
- NOTCHECKED_CENTER (bossdata.bits.OBJECT2 attribute), 61
- ## O
- OBJECT1 (class in bossdata.bits), 58
- OBJECT2 (class in bossdata.bits), 60
- ODDBAL (bossdata.bits.ANCILLARY_TARGET1 attribute), 30
- ORIGINAL_FLUXMATCH (bossdata.bits.FLUXMATCH_STATUS attribute), 51
- ORION_BD (bossdata.bits.SPECIAL_TARGET1 attribute), 68
- ORION_MSTAR_EARLY (bossdata.bits.SPECIAL_TARGET1 attribute), 67
- ORION_MSTAR_LATE (bossdata.bits.SPECIAL_TARGET1 attribute), 67
- OTBAL (bossdata.bits.ANCILLARY_TARGET1 attribute), 31
- OTHER (bossdata.bits.Q_EYEBALL attribute), 62
- OTHER (bossdata.bits.T_EYEBALL attribute), 72
- OTHER_BAD (bossdata.bits.APOGEE_PARAMFLAG attribute), 40
- OTHER_WARN (bossdata.bits.APOGEE_PARAMFLAG attribute), 39
- OUT_OF_BOUNDS (bossdata.bits.BOSSTILE_STATUS attribute), 44
- OUTFLOW (bossdata.bits.T_EYEBALL attribute), 72
- ## P
- PAIR_2IFU (bossdata.bits.MANGA_TARGET3 attribute), 56
- PAIR_ENLARGE (bossdata.bits.MANGA_TARGET3 attribute), 56
- PAIR_RECENTER (bossdata.bits.MANGA_TARGET3 attribute), 56
- PAIR_SIM (bossdata.bits.MANGA_TARGET3 attribute), 57
- PARAM_FIXED (bossdata.bits.APOGEE_PARAMFLAG attribute), 40

PARENT_FLUXMATCH (bossdata.bits.FLUXMATCH_STATUS attribute), 50

PARTIALREJECT (bossdata.bits.MANGA_DRP2PIXMASK attribute), 52

PARTIALREJECT (bossdata.bits.SPPIXMASK attribute), 68

PEAKCENTER (bossdata.bits.OBJECT1 attribute), 58

PEAKS_TOO_CLOSE (bossdata.bits.OBJECT2 attribute), 60

PERSEUS (bossdata.bits.SPECIAL_TARGET1 attribute), 68

PERSIST_HIGH (bossdata.bits.APOGEE_PIXMASK attribute), 40

PERSIST_HIGH (bossdata.bits.APOGEE_STARFLAG attribute), 41

PERSIST_JUMP_NEG (bossdata.bits.APOGEE_STARFLAG attribute), 41

PERSIST_JUMP_POS (bossdata.bits.APOGEE_STARFLAG attribute), 41

PERSIST_LOW (bossdata.bits.APOGEE_PIXMASK attribute), 40

PERSIST_LOW (bossdata.bits.APOGEE_STARFLAG attribute), 41

PERSIST_MED (bossdata.bits.APOGEE_PIXMASK attribute), 40

PERSIST_MED (bossdata.bits.APOGEE_STARFLAG attribute), 41

PETROFAINT (bossdata.bits.OBJECT1 attribute), 59

PHOTOMETRIC (bossdata.bits.CALIB_STATUS attribute), 46

PHOTOZ_GALAXY (bossdata.bits.SPECIAL_TARGET1 attribute), 67

PITCH_0 (bossdata.bits.T_EYEBALL attribute), 72

PITCH_1 (bossdata.bits.T_EYEBALL attribute), 72

PITCH_2 (bossdata.bits.T_EYEBALL attribute), 71

PITCH_3 (bossdata.bits.T_EYEBALL attribute), 72

PITCH_4 (bossdata.bits.T_EYEBALL attribute), 72

Plan (class in bossdata.plate), 82

PLANE (bossdata.bits.Q_EYEBALL attribute), 62

PLANETARY_NEBULA (bossdata.bits.Q_EYEBALL attribute), 61

plate (bossdata.raw.RawImageFile attribute), 88

PlateFile (class in bossdata.plate), 83

PLATEHOLE (bossdata.bits.VAGC_SELECT attribute), 73

POSSIBLE_KNOCKOUT (bossdata.bits.BOSSTILE_STATUS attribute), 44

POSSIBLE_LENS (bossdata.bits.Q_EYEBALL attribute), 62

PREBOSS_LRG (bossdata.bits.SPECIAL_TARGET1 attribute), 68

PREBOSS_QSO (bossdata.bits.SPECIAL_TARGET1 attribute), 67

PREMARVELS (bossdata.bits.SPECIAL_TARGET1 attribute), 66

prepare_columns() (bossdata.meta.Database method), 76

PREVBAL (bossdata.bits.ANCILLARY_TARGET1 attribute), 30

PREVIOUS_CHUNK (bossdata.bits.BOSSTILE_STATUS attribute), 44

PRIMARY_COM2 (bossdata.bits.MANGA_TARGET1 attribute), 55

PRIMARY_PLUS_COM (bossdata.bits.MANGA_TARGET1 attribute), 55

PRIMARY_v1_1_0 (bossdata.bits.MANGA_TARGET1 attribute), 55

PRIMARY_v1_2_0 (bossdata.bits.MANGA_TARGET1 attribute), 55

PS1_CONTRAIL (bossdata.bits.CALIB_STATUS attribute), 46

PS1_LOW_RMS (bossdata.bits.CALIB_STATUS attribute), 46

PS1_PCOMP_MODEL (bossdata.bits.CALIB_STATUS attribute), 47

PS1_UNPHOT (bossdata.bits.CALIB_STATUS attribute), 47

PSF (bossdata.bits.T_EYEBALL attribute), 73

PSF_FLUX_INTERP (bossdata.bits.OBJECT2 attribute), 61

PT_CLEAR (bossdata.bits.CALIB_STATUS attribute), 47

PT_CLOUDY (bossdata.bits.CALIB_STATUS attribute), 46

PTF_GAL (bossdata.bits.ANCILLARY_TARGET2 attribute), 33

Q

Q_EYEBALL (class in bossdata.bits), 61

QA (bossdata.bits.TTARGET attribute), 71

QSO1_BAD_BOSS (bossdata.bits.EBOSS_TARGET1 attribute), 48

QSO1_EBOSS_CORE (bossdata.bits.EBOSS_TARGET1 attribute), 49

QSO1_EBOSS_FIRST (bossdata.bits.EBOSS_TARGET1 attribute), 49

QSO1_EBOSS_KDE (bossdata.bits.EBOSS_TARGET1 attribute), 48

QSO1_PTF (bossdata.bits.EBOSS_TARGET1 attribute), 49

QSO1_REOBS (bossdata.bits.EBOSS_TARGET1 attribute), 49	QSO_KNOWN (bossdata.bits.EBOSS_TARGET0 attribute), 47
QSO1_VAR_S82 (bossdata.bits.EBOSS_TARGET1 attribute), 49	QSO_KNOWN (bossdata.bits.EBOSS_TARGET1 attribute), 49
QSO_AAL (bossdata.bits.ANCILLARY_TARGET1 attribute), 30	QSO_KNOWN_LOHIZ (bossdata.bits.BOSS_TARGET1 attribute), 46
QSO_AALS (bossdata.bits.ANCILLARY_TARGET1 attribute), 31	QSO_KNOWN_MIDZ (bossdata.bits.BOSS_TARGET1 attribute), 46
QSO_BAD_BOSS (bossdata.bits.EBOSS_TARGET0 attribute), 48	QSO_KNOWN_SUPPZ (bossdata.bits.BOSS_TARGET1 attribute), 46
QSO_BONUS (bossdata.bits.BOSS_TARGET1 attribute), 46	QSO_LIKE (bossdata.bits.BOSS_TARGET1 attribute), 46
QSO_BONUS_MAIN (bossdata.bits.BOSS_TARGET1 attribute), 46	QSO_M31 (bossdata.bits.SPECIAL_TARGET1 attribute), 67
QSO_BOSS_TARGET (bossdata.bits.EBOSS_TARGET0 attribute), 48	QSO_MAG_OUTLIER (bossdata.bits.TARGET attribute), 69
QSO_BOSS_TARGET (bossdata.bits.EBOSS_TARGET1 attribute), 49	QSO_NN (bossdata.bits.BOSS_TARGET1 attribute), 45
QSO_CAP (bossdata.bits.TARGET attribute), 71	QSO_NOAALS (bossdata.bits.ANCILLARY_TARGET1 attribute), 31
QSO_CORE (bossdata.bits.BOSS_TARGET1 attribute), 45	QSO_ON_GALAXY (bossdata.bits.Q_EYEBALL attribute), 63
QSO_CORE_ED (bossdata.bits.BOSS_TARGET1 attribute), 45	QSO_PTF (bossdata.bits.EBOSS_TARGET0 attribute), 47
QSO_CORE_LIKE (bossdata.bits.BOSS_TARGET1 attribute), 46	QSO_RADIO (bossdata.bits.ANCILLARY_TARGET1 attribute), 30
QSO_CORE_MAIN (bossdata.bits.BOSS_TARGET1 attribute), 45	QSO_RADIO_AAL (bossdata.bits.ANCILLARY_TARGET1 attribute), 30
QSO_DEEP (bossdata.bits.ANCILLARY_TARGET2 attribute), 34	QSO_RADIO_IAL (bossdata.bits.ANCILLARY_TARGET1 attribute), 32
QSO_EBOSS_CORE (bossdata.bits.EBOSS_TARGET0 attribute), 47	QSO_REJECT (bossdata.bits.TARGET attribute), 70
QSO_EBOSS_FIRST (bossdata.bits.EBOSS_TARGET0 attribute), 47	QSO_REOBS (bossdata.bits.EBOSS_TARGET0 attribute), 48
QSO_EBOSS_KDE (bossdata.bits.EBOSS_TARGET0 attribute), 47	QSO_RIZ (bossdata.bits.ANCILLARY_TARGET1 attribute), 31
QSO_EBOSS_W3_ADM (bossdata.bits.ANCILLARY_TARGET2 attribute), 33	QSO_SDSS_TARGET (bossdata.bits.EBOSS_TARGET0 attribute), 48
QSO_FIRST_BOSS (bossdata.bits.BOSS_TARGET1 attribute), 45	QSO_SDSS_TARGET (bossdata.bits.EBOSS_TARGET1 attribute), 49
QSO_FIRST_CAP (bossdata.bits.TARGET attribute), 70	QSO_SKIRT (bossdata.bits.TARGET attribute), 70
QSO_FIRST_SKIRT (bossdata.bits.TARGET attribute), 70	QSO_STD (bossdata.bits.ANCILLARY_TARGET2 attribute), 32
QSO_GRI (bossdata.bits.ANCILLARY_TARGET1 attribute), 29	QSO_SUPPZ (bossdata.bits.ANCILLARY_TARGET2 attribute), 34
QSO_HIZ (bossdata.bits.ANCILLARY_TARGET1 attribute), 31	QSO_UKIDSS (bossdata.bits.BOSS_TARGET1 attribute), 46
QSO_HIZ (bossdata.bits.TARGET attribute), 70	QSO_VAR (bossdata.bits.ANCILLARY_TARGET2 attribute), 35
QSO_IAL (bossdata.bits.ANCILLARY_TARGET1 attribute), 32	QSO_VAR_FPG (bossdata.bits.ANCILLARY_TARGET2 attribute), 34
QSO_KDE (bossdata.bits.BOSS_TARGET1 attribute), 45	QSO_VAR_LF (bossdata.bits.ANCILLARY_TARGET2 attribute), 35
QSO_KDE_COADD (bossdata.bits.BOSS_TARGET1 attribute), 45	

- QSO_VAR_SDSS (bossdata.bits.ANCILLARY_TARGET2 attribute), 34
- QSO_WISE_FULL_SKY (bossdata.bits.ANCILLARY_TARGET2 attribute), 33
- QSO_WISE_SUPP (bossdata.bits.ANCILLARY_TARGET2 attribute), 32
- QSO_XD_KDE_PAIR (bossdata.bits.ANCILLARY_TARGET2 attribute), 33
- QUALITY_HOLE (bossdata.bits.SEGUE2_TARGET2 attribute), 65
- QUALITY_HOLE (bossdata.bits.TTARGET attribute), 71
- QUESTIONABLE (bossdata.bits.M_EYEBALL attribute), 57
- ## R
- RADIO_2LOBE_QSO (bossdata.bits.ANCILLARY_TARGET2 attribute), 34
- RADIO_JETS (bossdata.bits.MANGA_TARGET3 attribute), 56
- RAMPAGINGBUNNY (bossdata.bits.MANGA_DRP2QUAL attribute), 53
- RawImageFile (class in bossdata.raw), 87
- read_plug_map() (bossdata.raw.RawImageFile method), 89
- RED_KG (bossdata.bits.ANCILLARY_TARGET1 attribute), 29
- REDDEN_STD (bossdata.bits.SEGUE1_TARGET2 attribute), 64
- REDDEN_STD (bossdata.bits.TTARGET attribute), 71
- REDMONSTER (bossdata.bits.MANGA_DRP2PIXMASK attribute), 52
- REDMONSTER (bossdata.bits.SPPIXMASK attribute), 68
- REPEAT (bossdata.bits.BOSSTILE_STATUS attribute), 44
- REPEAT (bossdata.bits.M_EYEBALL attribute), 57
- RESOLVE_STATUS (class in bossdata.bits), 63
- RING (bossdata.bits.M_EYEBALL attribute), 57
- RING (bossdata.bits.T_EYEBALL attribute), 72
- RM_TILE1 (bossdata.bits.ANCILLARY_TARGET2 attribute), 34
- RM_TILE2 (bossdata.bits.ANCILLARY_TARGET2 attribute), 34
- ROSAT_A (bossdata.bits.TARGET attribute), 71
- ROSAT_B (bossdata.bits.TARGET attribute), 70
- ROSAT_C (bossdata.bits.TARGET attribute), 70
- ROSAT_D (bossdata.bits.TARGET attribute), 70
- ROSAT_E (bossdata.bits.TARGET attribute), 69
- ROTATION_BAD (bossdata.bits.APOGEE_ASPCAPFLAG attribute), 38
- ROTATION_WARN (bossdata.bits.APOGEE_ASPCAPFLAG attribute), 39
- RQSS_SF (bossdata.bits.ANCILLARY_TARGET1 attribute), 29
- RQSS_SFC (bossdata.bits.ANCILLARY_TARGET1 attribute), 31
- RQSS_STM (bossdata.bits.ANCILLARY_TARGET1 attribute), 30
- RQSS_STMC (bossdata.bits.ANCILLARY_TARGET1 attribute), 30
- RUN_DUPLICATE (bossdata.bits.RESOLVE_STATUS attribute), 63
- RUN_EDGE (bossdata.bits.RESOLVE_STATUS attribute), 63
- RUN_IGNORE (bossdata.bits.RESOLVE_STATUS attribute), 63
- RUN_OVERLAPONLY (bossdata.bits.RESOLVE_STATUS attribute), 63
- RUN_PRIMARY (bossdata.bits.RESOLVE_STATUS attribute), 63
- RUN_RAMP (bossdata.bits.RESOLVE_STATUS attribute), 63
- RVTEST (bossdata.bits.ANCILLARY_TARGET1 attribute), 30
- ## S
- S0 (bossdata.bits.T_EYEBALL attribute), 72
- SATELLITE (bossdata.bits.Q_EYEBALL attribute), 62
- SATPIX (bossdata.bits.APOGEE_PIXMASK attribute), 40
- SATUR (bossdata.bits.OBJECT1 attribute), 58
- SATUR_CENTER (bossdata.bits.OBJECT2 attribute), 60
- SCATFAIL (bossdata.bits.MANGA_DRP2QUAL attribute), 53
- SCATTEREDLIGHT (bossdata.bits.MANGA_DRP2PIXMASK attribute), 52
- SCATTEREDLIGHT (bossdata.bits.SPPIXMASK attribute), 68
- SDSS_KNOWN (bossdata.bits.BOSS_TARGET1 attribute), 46
- SDSSFILLER (bossdata.bits.ANCILLARY_TARGET2 attribute), 35
- SECONDARY_COM (bossdata.bits.MANGA_TARGET1 attribute), 54

SECONDARY_COM2	(boss-	tribute), 64
data.bits.MANGA_TARGET1	attribute),	SEGUE2 (bossdata.bits.ANCILLARY_TARGET2
55		attribute), 34
SECONDARY_v1_1_0	(boss-	SEGUE2_BHB (bossdata.bits.SEGUE2_TARGET1 at-
data.bits.MANGA_TARGET1	attribute),	tribute), 65
55		SEGUE2_CHECKED (boss-
SECONDARY_v1_2_0	(boss-	data.bits.SEGUE2_TARGET1 attribute),
data.bits.MANGA_TARGET1	attribute),	65
55		SEGUE2_CHECKED (boss-
SEG1LOW_AGB (bossdata.bits.SEGUE1_TARGET at-		data.bits.SEGUE2_TARGET2 attribute),
tribute), 64		66
SEG1LOW_KG (bossdata.bits.SEGUE1_TARGET at-		SEGUE2_CLUSTER (boss-
tribute), 64		data.bits.SEGUE2_TARGET2 attribute),
SEG1LOW_TO (bossdata.bits.SEGUE1_TARGET at-		66
tribute), 64		SEGUE2_CWD (bossdata.bits.SEGUE2_TARGET1 at-
SEGUE1 (bossdata.bits.ANCILLARY_TARGET2		tribute), 65
attribute), 33		SEGUE2_HHV (bossdata.bits.SEGUE2_TARGET1 at-
SEGUE1_AGB (bossdata.bits.SEGUE1_TARGET		tribute), 65
attribute), 64		SEGUE2_HVS (bossdata.bits.SEGUE2_TARGET1 at-
SEGUE1_BD (bossdata.bits.SEGUE1_TARGET at-		tribute), 65
tribute), 64		SEGUE2_LKG (bossdata.bits.SEGUE2_TARGET1 at-
SEGUE1_BHB (bossdata.bits.SEGUE1_TARGET		tribute), 65
attribute), 64		SEGUE2_LM (bossdata.bits.SEGUE2_TARGET1
SEGUE1_CHECKED (bossdata.bits.SEGUE1_TARGET		attribute), 65
attribute), 64		SEGUE2_MII (bossdata.bits.SEGUE2_TARGET1
SEGUE1_CWD (bossdata.bits.SEGUE1_TARGET at-		attribute), 65
tribute), 64		SEGUE2_MSTO (bossdata.bits.SEGUE2_TARGET1 at-
SEGUE1_FG (bossdata.bits.SEGUE1_TARGET at-		tribute), 65
tribute), 64		SEGUE2_PMKG (bossdata.bits.SEGUE2_TARGET1 at-
SEGUE1_GD (bossdata.bits.SEGUE1_TARGET at-		tribute), 65
tribute), 64		SEGUE2_QA (bossdata.bits.SEGUE2_TARGET2
SEGUE1_KD (bossdata.bits.SEGUE1_TARGET at-		attribute), 66
tribute), 64		SEGUE2_REDDENING (boss-
SEGUE1_KG (bossdata.bits.SEGUE1_TARGET at-		data.bits.SEGUE2_TARGET2 attribute),
tribute), 64		66
SEGUE1_LM (bossdata.bits.SEGUE1_TARGET at-		SEGUE2_REDKG (bossdata.bits.SEGUE2_TARGET1
tribute), 64		attribute), 65
SEGUE1_MAN (bossdata.bits.SEGUE1_TARGET at-		SEGUE2_SPECPHOTO (boss-
tribute), 64		data.bits.SEGUE2_TARGET2 attribute),
SEGUE1_MPMSTO (bossdata.bits.SEGUE1_TARGET		66
attribute), 64		SEGUE2_STETSON (boss-
SEGUE1_MSWD (bossdata.bits.SEGUE1_TARGET at-		data.bits.SEGUE2_TARGET2 attribute),
tribute), 64		66
SEGUE1_QA (bossdata.bits.SEGUE1_TARGET2		SEGUE2_TARGET1 (class in bossdata.bits), 65
attribute), 64		SEGUE2_TARGET2 (class in bossdata.bits), 65
SEGUE1_SCIENCE (bossdata.bits.SEGUE1_TARGET2		SEGUE2_TEST (bossdata.bits.SEGUE2_TARGET2 at-
attribute), 65		tribute), 65
SEGUE1_SDM (bossdata.bits.SEGUE1_TARGET at-		SEGUE2_XDM (bossdata.bits.SEGUE2_TARGET1 at-
tribute), 64		tribute), 65
SEGUE1_TARGET (class in bossdata.bits), 63		select_all() (bossdata.meta.Database method), 76
SEGUE1_TARGET2 (class in bossdata.bits), 64		select_each() (bossdata.meta.Database method), 77
SEGUE1_TEST (bossdata.bits.SEGUE1_TARGET2 at-		SEQUELS_COLLIDED (boss-
tribute), 65		data.bits.EBOSS_TARGET0 attribute), 47
SEGUE1_WD (bossdata.bits.SEGUE1_TARGET at-		SEQUELS_ELG (boss-

data.bits.ANCILLARY_TARGET2 attribute), 34	52
SEQUELS_ELG_LOWP (boss- data.bits.ANCILLARY_TARGET2 attribute), 33	SMEARMEDSN (bossdata.bits.SPPIXMASK attribute), 68
SEQUELS_PTF_VARIABLE (boss- data.bits.EBOSS_TARGET0 attribute), 48	SN_BAD (bossdata.bits.APOGEE_ASPCAPFLAG at- tribute), 39
SEQUELS_TARGET (boss- data.bits.ANCILLARY_TARGET2 attribute), 35	SN_GAL1 (bossdata.bits.ANCILLARY_TARGET1 at- tribute), 30
SERENDIP_BLUE (bossdata.bits.TARGET attribute), 70	SN_GAL2 (bossdata.bits.ANCILLARY_TARGET1 at- tribute), 31
SERENDIP_DISTANT (bossdata.bits.TARGET at- tribute), 69	SN_GAL3 (bossdata.bits.ANCILLARY_TARGET1 at- tribute), 30
SERENDIP_FIRST (bossdata.bits.TARGET attribute), 69	SN_LOC (bossdata.bits.ANCILLARY_TARGET1 attribute), 31
SERENDIP_MANUAL (bossdata.bits.TARGET at- tribute), 70	SN_WARN (bossdata.bits.APOGEE_ASPCAPFLAG at- tribute), 38
SERENDIP_RED (bossdata.bits.TARGET attribute), 70	SOUTHERN_COMPLETE (boss- data.bits.SPECIAL_TARGET1 attribute), 67
SHELLS (bossdata.bits.M_EYEBALL attribute), 57	SOUTHERN_EXTENDED (boss- data.bits.SPECIAL_TARGET1 attribute), 66
SHELLS (bossdata.bits.T_EYEBALL attribute), 72	SOUTHERN_SURVEY (bossdata.bits.TARGET at- tribute), 69
SHUTTERS (bossdata.bits.IMAGE_STATUS attribute), 51	SOUTHERN_SURVEY (bossdata.bits.TTARGET attribute), 71
SIG_SKYLINE (bossdata.bits.APOGEE_PIXMASK at- tribute), 40	SPARE1 (bossdata.bits.OBJECT2 attribute), 61
SIG_TELLURIC (bossdata.bits.APOGEE_PIXMASK attribute), 40	SPARE2 (bossdata.bits.OBJECT2 attribute), 61
SKY (bossdata.bits.APOGEE_TARGET2 attribute), 43	SPARE3 (bossdata.bits.OBJECT2 attribute), 61
SKY (bossdata.bits.MANGA_TARGET2 attribute), 55	SPEC_SN (bossdata.bits.ANCILLARY_TARGET1 at- tribute), 29
SKY (bossdata.bits.SEGUE1_TARGET2 attribute), 65	SpecFile (class in bossdata.spec), 93
SKY (bossdata.bits.SEGUE2_TARGET2 attribute), 66	SPECIAL_FILLER (bossdata.bits.SPECIAL_TARGET1 attribute), 67
SKY (bossdata.bits.TTARGET attribute), 71	SPECIAL_TARGET1 (class in bossdata.bits), 66
SKY (bossdata.bits.ZWARNING attribute), 73	SPECTROPHOTO_STD (boss- data.bits.SEGUE1_TARGET2 attribute), 65
SKY_BAD (bossdata.bits.APOGEE_TARGET2 at- tribute), 43	SPECTROPHOTO_STD (bossdata.bits.TTARGET at- tribute), 71
SKYSUBBAD (bossdata.bits.MANGA_DRP2QUAL at- tribute), 53	SPIDERS_ERASS_AGN (boss- data.bits.EBOSS_TARGET0 attribute), 48
SKYSUBBAD (bossdata.bits.MANGA_DRP3QUAL at- tribute), 54	SPIDERS_ERASS_AGN (boss- data.bits.EBOSS_TARGET2 attribute), 50
SKYSUBFAIL (bossdata.bits.MANGA_DRP2QUAL at- tribute), 53	SPIDERS_ERASS_CLUS (boss- data.bits.EBOSS_TARGET0 attribute), 47
SMALL_DELTA_CHI2 (bossdata.bits.ZWARNING at- tribute), 73	SPIDERS_ERASS_CLUS (boss- data.bits.EBOSS_TARGET2 attribute), 49
SMEARHIGHSN (boss- data.bits.MANGA_DRP2PIXMASK attribute), 51	SPIDERS_PILOT (boss- data.bits.ANCILLARY_TARGET2 attribute), 32
SMEARHIGHSN (bossdata.bits.SPPIXMASK attribute), 68	SPIDERS_RASS_AGN (boss- data.bits.EBOSS_TARGET0 attribute), 48
SMEARIMAGE (boss- data.bits.MANGA_DRP2PIXMASK attribute), 51	SPIDERS_RASS_AGN (boss- data.bits.EBOSS_TARGET2 attribute), 50
SMEARIMAGE (bossdata.bits.SPPIXMASK attribute), 68	
SMEARMEDSN (boss- data.bits.MANGA_DRP2PIXMASK attribute),	

SPIDERS_RASS_CLUS (bossdata.bits.EBOSS_TARGET0 attribute), 48
 SPIDERS_RASS_CLUS (bossdata.bits.EBOSS_TARGET2 attribute), 50
 SPIDERS_TARGET (bossdata.bits.EBOSS_TARGET1 attribute), 48
 SPIDERS_XCLASS_CLUS (bossdata.bits.EBOSS_TARGET2 attribute), 49
 SPIDERS_XMMSL_AGN (bossdata.bits.EBOSS_TARGET2 attribute), 49
 SPIRAL_STRUCTURE (bossdata.bits.T_EYEBALL attribute), 72
 SPOKE (bossdata.bits.ANCILLARY_TARGET1 attribute), 31
 SPOKE2 (bossdata.bits.ANCILLARY_TARGET2 attribute), 32
 SPPIXMASK (class in bossdata.bits), 68
 sql_create_table() (in module bossdata.meta), 78
 STAR_BAD (bossdata.bits.APOGEE_ASPCAPFLAG attribute), 38
 STAR_BHB (bossdata.bits.TARGET attribute), 70
 STAR_BROWN_DWARF (bossdata.bits.TARGET attribute), 69
 STAR_CARBON (bossdata.bits.TARGET attribute), 70
 STAR_CATY_VAR (bossdata.bits.TARGET attribute), 70
 STAR_ON_GALAXY (bossdata.bits.Q_EYEBALL attribute), 62
 STAR_PN (bossdata.bits.TARGET attribute), 70
 STAR_RED_DWARF (bossdata.bits.TARGET attribute), 71
 STAR_SUB_DWARF (bossdata.bits.TARGET attribute), 71
 STAR_WARN (bossdata.bits.APOGEE_ASPCAPFLAG attribute), 39
 STAR_WHITE_DWARF (bossdata.bits.TARGET attribute), 70
 STATIONARY (bossdata.bits.OBJECT2 attribute), 60
 STD_APASS_COM (bossdata.bits.MANGA_TARGET2 attribute), 55
 STD_FSTAR (bossdata.bits.BOSS_TARGET1 attribute), 45
 STD_FSTAR (bossdata.bits.EBOSS_TARGET1 attribute), 49
 STD_FSTAR (bossdata.bits.MANGA_TARGET2 attribute), 56
 STD_FSTAR_COM (bossdata.bits.MANGA_TARGET2 attribute), 56
 STD_QSO (bossdata.bits.BOSS_TARGET1 attribute), 45
 STD_QSO (bossdata.bits.EBOSS_TARGET1 attribute), 49
 STD_STD_COM (bossdata.bits.MANGA_TARGET2 attribute), 55
 STD_WD (bossdata.bits.BOSS_TARGET1 attribute), 46
 STD_WD (bossdata.bits.EBOSS_TARGET1 attribute), 49
 STD_WD (bossdata.bits.MANGA_TARGET2 attribute), 55
 STD_WD_COM (bossdata.bits.MANGA_TARGET2 attribute), 56
 STELLIB_2MASS_COM (bossdata.bits.MANGA_TARGET2 attribute), 55
 STELLIB_COM_mar2015 (bossdata.bits.MANGA_TARGET2 attribute), 55
 STELLIB_KNOWN_COM (bossdata.bits.MANGA_TARGET2 attribute), 55
 STELLIB_SDSS_COM (bossdata.bits.MANGA_TARGET2 attribute), 55
 STRAIGHT_RADIO (bossdata.bits.SPECIAL_TARGET1 attribute), 67
 STRIPE82BCG (bossdata.bits.ANCILLARY_TARGET2 attribute), 34
 SUBTRACTED (bossdata.bits.OBJECT1 attribute), 59
 summarize_bitmask_values() (in module bossdata.bits), 75
 SUPPLEMENTARY (bossdata.bits.BOSSTILE_STATUS attribute), 44
 SURVEY_BADFIELD (bossdata.bits.RESOLVE_STATUS attribute), 63
 SURVEY_BEST (bossdata.bits.RESOLVE_STATUS attribute), 63
 SURVEY_EDGE (bossdata.bits.RESOLVE_STATUS attribute), 63
 SURVEY_PRIMARY (bossdata.bits.RESOLVE_STATUS attribute), 63
 SURVEY_SECONDARY (bossdata.bits.RESOLVE_STATUS attribute), 63
 SUSPECT_BROAD_LINES (bossdata.bits.APOGEE_STARFLAG attribute), 41
 SUSPECT_RV_COMBINATION (bossdata.bits.APOGEE_STARFLAG attribute), 41

T

T_EYEBALL (class in bossdata.bits), 71
 TARGET (class in bossdata.bits), 69
 TAU_STAR (bossdata.bits.ANCILLARY_TARGET2 attribute), 34

- TAURUS_GALAXY (bossdata.bits.SPECIAL_TARGET1 attribute), 67
- TAURUS_STAR (bossdata.bits.SPECIAL_TARGET1 attribute), 67
- TDSS_A (bossdata.bits.EBOSS_TARGET0 attribute), 47
- TDSS_A (bossdata.bits.EBOSS_TARGET2 attribute), 50
- TDSS_B (bossdata.bits.EBOSS_TARGET2 attribute), 49
- TDSS_CP (bossdata.bits.EBOSS_TARGET2 attribute), 50
- TDSS_FES_ACTSTAR (bossdata.bits.EBOSS_TARGET2 attribute), 50
- TDSS_FES_DE (bossdata.bits.EBOSS_TARGET0 attribute), 48
- TDSS_FES_DE (bossdata.bits.EBOSS_TARGET2 attribute), 50
- TDSS_FES_DWARFC (bossdata.bits.EBOSS_TARGET0 attribute), 48
- TDSS_FES_DWARFC (bossdata.bits.EBOSS_TARGET2 attribute), 50
- TDSS_FES_HYPQSO (bossdata.bits.EBOSS_TARGET2 attribute), 50
- TDSS_FES_HYPSTAR (bossdata.bits.EBOSS_TARGET2 attribute), 49
- TDSS_FES_MGII (bossdata.bits.EBOSS_TARGET0 attribute), 47
- TDSS_FES_MGII (bossdata.bits.EBOSS_TARGET2 attribute), 50
- TDSS_FES_NQHISN (bossdata.bits.EBOSS_TARGET0 attribute), 48
- TDSS_FES_NQHISN (bossdata.bits.EBOSS_TARGET2 attribute), 50
- TDSS_FES_VARBAL (bossdata.bits.EBOSS_TARGET0 attribute), 48
- TDSS_FES_VARBAL (bossdata.bits.EBOSS_TARGET2 attribute), 50
- TDSS_FES_WDDM (bossdata.bits.EBOSS_TARGET2 attribute), 50
- TDSS_PILOT (bossdata.bits.ANCILLARY_TARGET2 attribute), 34
- TDSS_PILOT_PM (bossdata.bits.ANCILLARY_TARGET2 attribute), 33
- TDSS_PILOT_SNHOST (bossdata.bits.ANCILLARY_TARGET2 attribute), 35
- TDSS_SPIDERS_PILOT (bossdata.bits.ANCILLARY_TARGET2 attribute), 35
- TDSS_TARGET (bossdata.bits.EBOSS_TARGET1 attribute), 48
- TEFF_BAD (bossdata.bits.APOGEE_ASPCAPFLAG attribute), 38
- TEFF_WARN (bossdata.bits.APOGEE_ASPCAPFLAG attribute), 38
- TELLURIC (bossdata.bits.APOGEE_EXTRATARG attribute), 39
- TEMPLATE_GAL_PHOTO (bossdata.bits.BOSS_TARGET1 attribute), 46
- TEMPLATE_QSO_SDSS1 (bossdata.bits.BOSS_TARGET1 attribute), 46
- TEMPLATE_STAR_PHOTO (bossdata.bits.BOSS_TARGET1 attribute), 45
- TEMPLATE_STAR_SPECTRO (bossdata.bits.BOSS_TARGET1 attribute), 46
- TEST_TARGET (bossdata.bits.TTARGET attribute), 71
- TIDAL_TAILS (bossdata.bits.M_EYEBALL attribute), 57
- TIDAL_TAILS (bossdata.bits.T_EYEBALL attribute), 72
- TILED (bossdata.bits.BOSSTILE_STATUS attribute), 44
- TILED (bossdata.bits.VAGC_SELECT attribute), 73
- TOO_FEW_DETECTIONS (bossdata.bits.OBJECT2 attribute), 60
- TOO_FEW_GOOD_DETECTIONS (bossdata.bits.OBJECT2 attribute), 60
- TOO_LARGE (bossdata.bits.OBJECT1 attribute), 59
- TOOBRIGHT (bossdata.bits.BOSSTILE_STATUS attribute), 44
- TOOFAINT (bossdata.bits.BOSSTILE_STATUS attribute), 44
- TraceSet (class in bossdata.plate), 85
- TTARGET (class in bossdata.bits), 71
- ## U
- U_EXTRA (bossdata.bits.SPECIAL_TARGET1 attribute), 68
- U_EXTRA2 (bossdata.bits.SPECIAL_TARGET1 attribute), 67
- U_PRIORITY (bossdata.bits.SPECIAL_TARGET1 attribute), 68
- UNCLASSIFIABLE (bossdata.bits.Q_EYEBALL attribute), 62
- UNCLASSIFIABLE (bossdata.bits.T_EYEBALL attribute), 72
- UNFIXABLE (bossdata.bits.APOGEE_PIXMASK attribute), 40
- UNKNOWN (bossdata.bits.IMAGE_STATUS attribute), 51
- UNPHOT_DISJOINT (bossdata.bits.CALIB_STATUS attribute), 47
- UNPHOT_EXTRAP_CLEAR (bossdata.bits.CALIB_STATUS attribute), 46
- UNPHOT_EXTRAP_CLOUDY (bossdata.bits.CALIB_STATUS attribute), 47
- UNPHOT_OVERLAP (bossdata.bits.CALIB_STATUS attribute), 46
- UNPLUGGED (bossdata.bits.ZWARNING attribute), 73

UNUSED_0 (bossdata.bits.T_EYEBALL attribute), 72
 USE_ANYWAY (bossdata.bits.Q_EYEBALL attribute), 63

USE_CHILD_IMAGE (bossdata.bits.Q_EYEBALL attribute), 62

USE_CHILD_SPECTRUM (bossdata.bits.Q_EYEBALL attribute), 63

USE_PARENT (bossdata.bits.Q_EYEBALL attribute), 62

V

VAGC_SELECT (class in bossdata.bits), 73

VALIDFILE (bossdata.bits.MANGA_DAPQUAL attribute), 51

VALIDFILE (bossdata.bits.MANGA_DRP2QUAL attribute), 53

VALIDFILE (bossdata.bits.MANGA_DRP3QUAL attribute), 54

VARBAL (bossdata.bits.ANCILLARY_TARGET1 attribute), 30

VARIABLE_HIPRI (bossdata.bits.SPECIAL_TARGET1 attribute), 67

VARIABLE_LOPRI (bossdata.bits.SPECIAL_TARGET1 attribute), 67

VARIABLELSF (bossdata.bits.MANGA_DRP3QUAL attribute), 54

VARS (bossdata.bits.ANCILLARY_TARGET1 attribute), 31

VERY_BRIGHT_NEIGHBOR (bossdata.bits.APOGEE_STARFLAG attribute), 40

VMICRO_BAD (bossdata.bits.APOGEE_ASPCAPFLAG attribute), 39

VMICRO_WARN (bossdata.bits.APOGEE_ASPCAPFLAG attribute), 38

VOID (bossdata.bits.MANGA_TARGET3 attribute), 56

W

WARPED_DISK (bossdata.bits.T_EYEBALL attribute), 73

WHITEDWARF_NEW (bossdata.bits.ANCILLARY_TARGET1 attribute), 29

WHITEDWARF_SDSS (bossdata.bits.ANCILLARY_TARGET1 attribute), 31

WHOPPER (bossdata.bits.MANGA_DRP2PIXMASK attribute), 53

WHOPPER (bossdata.bits.SPPIXMASK attribute), 69

WISE_BOSS_QSO (bossdata.bits.ANCILLARY_TARGET2 attribute), 34

WISE_COMPLETE (bossdata.bits.ANCILLARY_TARGET2 attribute), 34

X

XMM_PRIME (bossdata.bits.ANCILLARY_TARGET2 attribute), 33

XMM_SECOND (bossdata.bits.ANCILLARY_TARGET2 attribute), 33

XMMBRIGHT (bossdata.bits.ANCILLARY_TARGET1 attribute), 32

XMMGRIZ (bossdata.bits.ANCILLARY_TARGET1 attribute), 30

XMMHR (bossdata.bits.ANCILLARY_TARGET1 attribute), 31

XMMRED (bossdata.bits.ANCILLARY_TARGET1 attribute), 31

XMMSDSS (bossdata.bits.ANCILLARY_TARGET2 attribute), 34

Z

Z_FITLIMIT (bossdata.bits.ZWARNING attribute), 73

ZWARNING (class in bossdata.bits), 73