bossdata Documentation

Release 0.2.8dev

David Kirkby

September 28, 2015

Contents

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 bossdata
```

To upgrade to the latest version:

```
% pip install bossdata --upgrade
```

1.1 Requirements

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

- requests
- progressbar
- astropy
- fitsio
- numpy

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	Bossdata Class
lite	0.2Mb	1	0	Y	Y	lite	*bossdata.spec.SpecFile*
spec	1.7Mb	1	ALL	Y	Y	spec	*bossdata.spec.SpecFile*
plate	110Mb	1000	0	Y	Y	plate	*bossdata.plate.PlateFile*
cframe	75Mb	500	1	N	Y	cframe	*bossdata.plate.FrameFile*
frame	30Mb	500	1	N	N	frame	*bossdata.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.

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:

```
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, *MODELFUX(5)*, is included in the lite format using names *MODELFLUX0...MODELFUX4*. 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
```

Fetched 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 signficantly (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 *`bossdata.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 `bossdata` package and therefore not automatically installed, but is included in scientific python distributions like anaconda.

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/
```

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 for details if you are an SDSS-IV collaborator.

API Usage

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

```
import bossdata.path
import bossdata.remote

try:
    finder = bossdata.path.Finder()
    mirror = bossdata.remote.Manager()
except ValueError as e:
    print(e)
    return -1
```

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, mdj=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:

```
import bossdata.spec

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

CHAPTER 7

Contributing

Contributions are welcome, and they are greatly appreciated! Every little bit helps, and credit will always be given.

You can contribute in many ways:

7.1 Types of Contributions

7.1.1 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.

7.1.2 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 :)

7.1.3 Work on Iusses

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.

7.1.4 Write Documentation

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

We use the sphinx napolean 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:`bossdata.module.Class`
- methods :meth:`bosdata.module.Class.method`
- functions :func:`bossdata.module.func`
- You can override the default link text by changing :role:`target` to :role:`text <target>`.

7.2 Get Started!

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

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

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

3. Install your local copy for local development:

```
cd bossdata/
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 bossdata/bits.py --max-line-length 95 bossdata
py.test --doctest-modules --verbose bossdata
```

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.

7.3 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.

7.4 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
```

7.5 New External Depencency 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*

History

8.1 0.2.8 (unreleased)

- No changes yet.

8.2 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".

8.3 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`.

8.4 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.

8.5 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.

8.6 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`.

8.7 0.2.2 (2015-06-15)

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

8.8 0.2.1 (2015-06-13)

- Fix issues #9 #12 #13

8.9 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 compatiblity 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.

8.10 0.1.0 (2015-05-24)

- First release on PyPI.

CHAPTER 9

Modules API Reference

9.1 bossdata package

See also API Usage. Use the links below to browse the API docs for each sub-module.

9.1.1 bossdata.path module

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 bossdata.path
finder = bossdata.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 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*)

Bases: `object`

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.

default_quasar_catalog_name = 'DR12Q'
Default quasar catalog name.

For more info about the BOSS quasar catalog, see http://www.sdss.org/dr12/algorithms/boss-dr12-quasar-catalog/

default_redux_version = 'v5_7_0'
Default to use when $BOSS_REDUX_VERSION is not set.

See Executable scripts and API Usage for details.

default_sas_path = '/sas/dr12/boss'
Default to use when $BOSS_SAS_PATH is not set.

See Executable scripts and API Usage for details.

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_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*)

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/sp 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.

9.1.2 bossdata.remote module

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 bossdata.remote
mirror = bossdata.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 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 `bossdata.path` module to build the paths of frequently used data files. See API Usage for recommendations on using the `bossdata.path` and `bossdata.remote` modules together.

class `bossdata.remote.`**`Manager`**(*data_url=None*, *local_root=None*, *verbose=True*)

Bases: `object`

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 we 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.

`default_data_url` **= 'http://dr12.sdss3.org'**

Default to use when $BOSS_DATA_URL is not set.

See Executable scripts and API Usage for details.

`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 than 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. Tere 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 `bossdata.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,iterable*) – This arg will normally be a single string but can optionally be an iterable over strings for some advanced functionality. Strings give the full path to a remote file and should normally be obtained using `bossdata.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 ocurred:

```
mirror = bossdata.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*)

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 `bossdata.path` methods.

Returns Absolute local path of the local file that mirrors the remote file.

Return type str

Raises `RuntimeError` – No local_root specified when this manager was created.

9.1.3 bossdata.meta module

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*)

Bases: `object`

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 *astropy.table.Table*: 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 :class

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.

bossdata.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.

bossdata.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.

bossdata.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** (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 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

bossdata.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.

9.1.4 bossdata.spec module

Access spectroscopic data for a single BOSS target.

class bossdata.spec.**Exposures**(*header*)

Bases: object

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_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.

class `bossdata.spec.`**`SpecFile`**(*path*)

Bases: `object`

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 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 *`bossdata.path.Finder.get_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_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*, *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. This method is analogous to `bossdata.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'.
- **ftype** (*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.

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_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.

`bossdata.spec.`**fiducial_loglam**

Array of fiducial log10(wavelength in Angstroms) covering all spectra.

Lookup the log10(wavelength) or wavelength corresponding to a particular integral pixel index using:

```
>>> fiducial_loglam[100]
3.554100305027835
>>> 10**fiducial_loglam[100]
3581.7915291606305
```

The bounding wavelengths of this range are:

```
>>> 10**fiducial_loglam[[0,-1]]
array([  3500.26      ,  10568.18251472])
```

The *SpecFile.get_valid_data()* and *PlateFile.get_valid_data()* methods provide a `fiducial_grid` option that returns data using this grid.

`bossdata.spec.`**`fiducial_pixel_index_range`** **= (0, 4800)**
Range of fiducial pixel indices that covers all spectra.

Use *get_fiducial_pixel_index()* to calculate fiducial pixel indices.

`bossdata.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

9.1.5 bossdata.bits module

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`**

Bases: `object`

BOSS survey target flags for ancillary programs

`RQSS_STMC`
int

(1<<35) defined in rqss090630.descr

`BLAZGXQSO`
int

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

`BLAZR`
int

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

`SPOKE`
int

(1<<41) defined in BOSS_slowpokes_v2.descr

`VARS`
int

(1<<5) defined in blake_boss_v2.descr

`QSO_RADIO_AAL`
int

(1<<26) defined in qsoals_v2.descr

`FAINTERM`
int

(1<<47) defined in sd3targets_final.descr

`RQSS_SFC`
int

(1<<33) defined in rqss090630.descr

`CHANDRAV1`
int

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

`BRIGHTGAL`
int

(1<<21) defined in bright_gal_v3.descr

`CXOBRIGHT`
int

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

`QSO_HIZ`
int

(1<<30) defined in sdss3_fan.descr

RVTEST
int

(1<<49) defined in redkg.descr

GAL_NEAR_QSO
int

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

MTEMP
int

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

BLAZGXR
int

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

SPEC_SN
int

(1<<40) defined in ancillary_supernova_hosts_v5.descr

CXOGRIZ
int

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

QSO_AAL
int

(1<<22) defined in qsoals_v2.descr

AMC
int

(1<<0) defined in blake_boss_v2.descr

BLAZXRSAM
int

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

QSO_RIZ
int

(1<<31) defined in sdss3_fan.descr

FBQSBAL
int

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

BLAZXR
int

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

RQSS_SF
int

(1<<32) defined in rqss090630.descr

BLUE_RADIO
int

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

RED_KG
int

(1<<48) defined in redkg.descr

BLAZGVAR
int

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

QSO_AALS
int

(1<<23) defined in qsoals_v2.descr

PREVBAL
int

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

LBQSBAL
int

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

QSO_RADIO
int

(1<<25) defined in qsoals_v2.descr

QSO_NOAALS
int

(1<<28) defined in qsoals_v2.descr

XMMBRIGHT
int

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

ELG
int

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

QSO_GRI
int

(1<<29) defined in sdss3_fan.descr

FLARE2
int

(1<<2) defined in blake_boss_v2.descr

SN_GAL3
int

(1<<38) defined in ancillary_supernova_hosts_v5.descr

FLARE1
int

(1<<1) defined in blake_boss_v2.descr

QSO_RADIO_IAL
int

(1<<27) defined in qsoals_v2.descr

HPM
int

(1<<3) defined in blake_boss_v2.descr

SN_GAL2
int

(1<<37) defined in ancillary_supernova_hosts_v5.descr

FAINTERL
int

(1<<46) defined in sd3targets_final.descr

BLAZGRFLAT
int

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

SN_GAL1
int

(1<<36) defined in ancillary_supernova_hosts_v5.descr

VARBAL
int

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

RQSS_STM
int

(1<<34) defined in rqss090630.descr

OTBAL
int

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

BLAZGX
int

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

XMMRED
int

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

BLAZGRQSO
int

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

CXORED
int

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

BRIGHTERL
int

(1<<44) defined in sd3targets_final.descr

BRIGHTERM
int

(1<<45) defined in sd3targets_final.descr

LOW_MET
int

(1<<4) defined in blake_boss_v2.descr

XMMGRIZ
int

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

SN_LOC
int

(1<<39) defined in ancillary_supernova_hosts_v5.descr

WHITEDWARF_NEW
int

(1<<42) defined in WDv5_eisenste_fixed.descr

QSO_IAL
int

(1<<24) defined in qsoals_v2.descr

WHITEDWARF_SDSS
int

(1<<43) defined in WDv5_eisenste_fixed.descr

XMMHR
int

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

ODDBAL
int

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

BLAZXRVAR
int

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

AMC = 1

BLAZGRFLAT = 1125899906842624

BLAZGRQSO = 2251799813685248

BLAZGVAR = 64

BLAZGX = 4503599627370496

BLAZGXQSO = 9007199254740992

BLAZGXR = 18014398509481984

BLAZR = 128

BLAZXR = 256

BLAZXRSAM = 512

BLAZXRVAR = 1024

BLUE_RADIO = 72057594037927936

BRIGHTERL = 17592186044416

BRIGHTERM = 35184372088832

BRIGHTGAL = 2097152

CHANDRAV1 = 144115188075855872

CXOBRIGHT = 288230376151711744

CXOGRIZ = 576460752303423488

CXORED = 1152921504606846976

ELG = 2305843009213693952

FAINTERL = 70368744177664

FAINTERM = 140737488355328

FBQSBAL = 32768

FLARE1 = 2

FLARE2 = 4

GAL_NEAR_QSO = 4611686018427387904

HPM = 8

LBQSBAL = 65536

LOW_MET = 16

MTEMP = 9223372036854775808L

ODDBAL = 131072

OTBAL = 262144

PREVBAL = 524288

QSO_AAL = 4194304

QSO_AALS = 8388608

QSO_GRI = 536870912

QSO_HIZ = 1073741824

QSO_IAL = 16777216

QSO_NOAALS = 268435456

QSO_RADIO = 33554432

QSO_RADIO_AAL = 67108864

QSO_RADIO_IAL = 134217728

QSO_RIZ = 2147483648

RED_KG = 281474976710656

RQSS_SF = 4294967296

RQSS_SFC = 8589934592

RQSS_STM = 17179869184

RQSS_STMC = 34359738368

RVTEST = 562949953421312

SN_GAL1 = 68719476736

SN_GAL2 = 137438953472

SN_GAL3 = 274877906944

SN_LOC = 549755813888

SPEC_SN = 1099511627776

SPOKE = 2199023255552

VARBAL = 1048576

VARS = 32

WHITEDWARF_NEW = 4398046511104

WHITEDWARF_SDSS = 8796093022208

XMMBRIGHT = 2048

XMMGRIZ = 4096

XMMHR = 8192

XMMRED = 16384

class bossdata.bits.**ANCILLARY_TARGET2**

Bases: object

additional BOSS survey target flags for ancillary programs

QSO_WISE_FULL_SKY

int

(1<<10) defined in none

KQSO_BOSS

int

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

TAU_STAR

int

(1<<52) defined in knapp_taurus.descr

_2MASSFILL

int

(1<<51) defined in rocksi_ges_segue.descr

LRG_ROUND3
int

(1<<22) defined in newman.descr

QSO_WISE_SUPP
int

(1<<9) defined in BOSS_QSO_targets_July_WISE.descr

DISKEMITTER_REPEAT
int

(1<<13) defined in shen.descr

SEQUELS_ELG
int

(1<<34) defined in sequels_elg.descr

ELAIS_N1_GMRT_GARN
int

(1<<60) LOFAR-selected target

QSO_VAR_SDSS
int

(1<<8) defined in VARQSO.descr

SDSSFILLER
int

(1<<38) defined in rockosi_ges_segue.descr

QSO_EBOSS_W3_ADM
int

(1<<31) defined in myers_eboss_qso_w3.descr

HIZ_LRG
int

(1<<21) defined in newman.descr

RADIO_2LOBE_QSO
int

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

COROTGES
int

(1<<49) defined in rocksi_ges_segue.descr

COROTGESAPOG
int

(1<<48) defined in rocksi_ges_segue.descr

HIZQSO82
int

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

STRIPE82BCG
int

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

KOE2068_STAR
int

(1<<44) defined in knapp_ngc2068.descr

QSO_VAR_LF
int

(1<<27) defined in palanque_str82.descr

SPIDERS_PILOT
int

(1<<25) defined in GreenMerloni_MD01.descr

ELAIS_N1_LOFAR
int

(1<<58) LOFAR-selected target

HIZQSOIR
int

(1<<1) defined in mcgreer-hizqso.fits

ELAIS_N1_JVLA
int

(1<<62) LOFAR-selected target

QSO_DEEP
int

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

XMM_PRIME
int

(1<<32) defined in georgekaksi_xmmxll.descr

TDSS_SPIDERS_PILOT
int

(1<<26) defined in GreenMerloni_MD01.descr

XMMSDSS
int

(1<<11) defined in georgakakis.descr

KOE2068BSTAR
int

(1<<46) defined in knapp_ngc2068.descr

SPOKE2
int

(1<<17) defined in dhital.descr

LBG
int

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

RM_TILE1
int

(1<<54) reverberation mapping, high priority

CLUSTER_MEMBER
int

(1<<16) defined in finoguenov_auxBOSS.descr

GES
int

(1<<35) defined in rockosi_ges_segue.descr

ELAIS_N1_FIRST
int

(1<<59) LOFAR-selected target

FAINT_ELG
int

(1<<18) defined in comparat.descr

SEQUELS_ELG_LOWP
int

(1<<39) defined in sequels_elg.descr

WISE_BOSS_QSO
int

(1<<14) defined in ross_wisebossqso.descr

PTF_GAL
int

(1<<19) defined in kasliwal.descr

_25ORI_WISE_W3
int

(1<<41) defined in knapp_25ori.descr

SEGUE1
int

(1<<36) defined in rockosi_ges_segue.descr

SEGUE2
int

(1<<37) defined in rockosi_ges_segue.descr

_25ORI_WISE
int

(1<<40) defined in knapp_25ori.descr

KOE2023_STAR
int

(1<<43) defined in knapp_ngc2023.descr

KOEKAP_STAR
int

(1<<42) defined in knapp_kappaori.descr

TDSS_PILOT_SNHOST
int

(1<<29) defined in TDSS_SPIDERS_MD03.descr

IAMASERS
int

(1<<12) defined in zaw.descr

XMM_SECOND
int

(1<<33) defined in georgekaksi_xmmxll.descr

KOEKAPBSTAR
int

(1<<47) defined in knapp_kappaori.descr

KOE2023BSTAR
int

(1<<45) defined in knapp_ngc2023.descr

ELAIS_N1_GMRT_TAYLOR
int

(1<<61) LOFAR-selected target

QSO_XD_KDE_PAIR
int

(1<<15) defined in myers.descr

QSO_SUPPZ
int

(1<<7) defined in qso_suppz.descr

QSO_VAR_FPG
int

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

RM_TILE2
int

(1<<55) reverberation mapping, low priority

QSO_STD
int

(1<<20) defined in margala.descr

WISE_COMPLETE
int

(1<<23) defined in weiner_wise.descr

QSO_VAR
int

(1<<3) defined in butler-variable.fits.gz

TDSS_PILOT
int

(1<<24) defined in GreenMerloni_MD01.descr

TDSS_PILOT_PM
int

(1<<28) defined in TDSS_SPIDERS_MD03.descr

SEQUELS_TARGET
int

(1<<53) any target in SEQUELS darktime program

FAINT_HIZ_LRG
int

(1<<30) defined in newman_lrg_w3.descr

APOGEE
int

(1<<50) defined in rocksi_ges_segue.descr

APOGEE = 1125899906842624

CLUSTER_MEMBER = 65536

COROTGES = 562949953421312

COROTGESAPOG = 281474976710656

DISKEMITTER_REPEAT = 8192

ELAIS_N1_FIRST = 576460752303423488

ELAIS_N1_GMRT_GARN = 1152921504606846976

ELAIS_N1_GMRT_TAYLOR = 2305843009213693952

ELAIS_N1_JVLA = 4611686018427387904

ELAIS_N1_LOFAR = 288230376151711744

FAINT_ELG = 262144

FAINT_HIZ_LRG = 1073741824

GES = 34359738368

HIZQSO82 = 1

HIZQSOIR = 2

HIZ_LRG = 2097152

IAMASERS = 4096

KOE2023BSTAR = 35184372088832

KOE2023_STAR = 8796093022208

KOE2068BSTAR = 70368744177664

KOE2068_STAR = 17592186044416

KOEKAPBSTAR = 140737488355328

KOEKAP_STAR = 4398046511104

KQSO_BOSS = 4

LBG = 144115188075855872

LRG_ROUND3 = 4194304

PTF_GAL = 524288

QSO_DEEP = 72057594037927936

QSO_EBOSS_W3_ADM = 2147483648

QSO_STD = 1048576

QSO_SUPPZ = 128

QSO_VAR = 8

QSO_VAR_FPG = 16

QSO_VAR_LF = 134217728

QSO_VAR_SDSS = 256

QSO_WISE_FULL_SKY = 1024

QSO_WISE_SUPP = 512

QSO_XD_KDE_PAIR = 32768

RADIO_2LOBE_QSO = 32

RM_TILE1 = 18014398509481984

RM_TILE2 = 36028797018963968

SDSSFILLER = 274877906944

SEGUE1 = 68719476736

SEGUE2 = 137438953472

SEQUELS_ELG = 17179869184

SEQUELS_ELG_LOWP = 549755813888

SEQUELS_TARGET = 9007199254740992

SPIDERS_PILOT = 33554432

SPOKE2 = 131072

STRIPE82BCG = 64

TAU_STAR = 4503599627370496

TDSS_PILOT = 16777216

TDSS_PILOT_PM = 268435456

TDSS_PILOT_SNHOST = 536870912

TDSS_SPIDERS_PILOT = 67108864

WISE_BOSS_QSO = 16384

WISE_COMPLETE = 8388608

XMMSDSS = 2048

XMM_PRIME = 4294967296

XMM_SECOND = 8589934592

class bossdata.bits.**APOGEE2_TARGET1**

Bases: object

APOGEE2 primary target bits

APOGEE2_FAINT_EXTRA

int

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

APOGEE2_SCI_CLUSTER

int

(1<<9) Science cluster candidate member

APOGEE2_TWOBIN_0_5_TO_0_8

int

(1<<1) Selected in blue 0.5 < (J-Ks)o < 0.8 color bin

APOGEE2_NO_DERED

int

(1<<6) Selected with no dereddening

APOGEE2_MAGCLOUD_CANDIDATE

int

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

APOGEE2_DSPH_CANDIDATE

int

(1<<21) Selected as potential dSph member (non Sgr) (based on photometry)

APOGEE2_TWOBIN_GT_0_8

int

(1<<2) Selected in red (J-Ks)o > 0.8 color bin

APOGEE2_DSPH_MEMBER

int

(1<<20) Selected as confirmed dSph member (non Sgr)

APOGEE2_APOKASC_DWARF

int

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

APOGEE2_NORMAL_SAMPLE

int

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

APOGEE2_APOKASC_GIANT
int

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

APOGEE2_SGR_DSPH
int

(1<<26) Selected as confirmed Sgr core/stream member

APOGEE2_WASH_NOCLASS
int

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

APOGEE2_MAGCLOUD_MEMBER
int

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

APOGEE2_RRLYR
int

(1<<24) Selected as a bulge RR Lyrae star

APOGEE2_WASH_DWARF
int

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

APOGEE2_MANGA_LED
int

(1<<15) Star on a shared MaNGA-led design

APOGEE2_WASH_GIANT
int

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

APOGEE2_SFD_DERED
int

(1<<5) Selected with SFD_EBV dereddening

APOGEE2_APOKASC
int

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

APOGEE2_STREAM_CANDIDATE
int

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

APOGEE2_IRAC_DERED
int

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

APOGEE2_SHORT
int

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

APOGEE2_MEDIUM
int

(1<<12) Selected as part of a medium cohort

APOGEE2_LONG
int

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

APOGEE2_STREAM_MEMBER
int

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

APOGEE2_ONEBIN_GT_0_5
int

(1<<0) Selected in single (J-Ks)o > 0.5 color bin

APOGEE2_WISE_DERED
int

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

APOGEE2_ONEBIN_GT_0_3
int

(1<<16) Selected in single (J-Ks)o > 0.3 color bin

APOGEE2_APOKASC = 1073741824

APOGEE2_APOKASC_DWARF = 268435456

APOGEE2_APOKASC_GIANT = 134217728

APOGEE2_DSPH_CANDIDATE = 2097152

APOGEE2_DSPH_MEMBER = 1048576

APOGEE2_FAINT_EXTRA = 536870912

APOGEE2_IRAC_DERED = 8

APOGEE2_LONG = 8192

APOGEE2_MAGCLOUD_CANDIDATE = 8388608

APOGEE2_MAGCLOUD_MEMBER = 4194304

APOGEE2_MANGA_LED = 32768

APOGEE2_MEDIUM = 4096

APOGEE2_NORMAL_SAMPLE = 16384

APOGEE2_NO_DERED = 64

APOGEE2_ONEBIN_GT_0_3 = 65536

APOGEE2_ONEBIN_GT_0_5 = 1

APOGEE2_RRLYR = 16777216

APOGEE2_SCI_CLUSTER = 512

APOGEE2_SFD_DERED = 32

APOGEE2_SGR_DSPH = 67108864

APOGEE2_SHORT = 2048

APOGEE2_STREAM_CANDIDATE = 524288

APOGEE2_STREAM_MEMBER = 262144

APOGEE2_TWOBIN_0_5_TO_0_8 = 2

APOGEE2_TWOBIN_GT_0_8 = 4

APOGEE2_WASH_DWARF = 256

APOGEE2_WASH_GIANT = 128

APOGEE2_WASH_NOCLASS = 131072

APOGEE2_WISE_DERED = 16

class bossdata.bits.**APOGEE2_TARGET2**

Bases: object

APOGEE2 secondary target bits

APOGEE2_STANDARD_STAR

int

(1<<2) Stellar parameters/abundance standard

APOGEE2_GES_OVERLAP

int

(1<<14) Overlap with Gaia-ESO

APOGEE2_1M_TARGET

int

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

APOGEE2_EXTERNAL_CALIB

int

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

APOGEE2_GALAH_OVERLAP

int

(1<<17) Overlap with GALAH

APOGEE2_ARGOS_OVERLAP

int

(1<<15) Overlap with ARGOS

APOGEE2_INTERNAL_CALIB

int

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

APOGEE2_GAIA_OVERLAP

int

(1<<16) Overlap with Gaia

APOGEE2_OBJECT

int

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

APOGEE2_TELLURIC
int

(1<<9) Telluric calibrator target

APOGEE2_LITERATURE_CALIB
int

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

APOGEE2_RAVE_OVERLAP
int

(1<<18) Overlap with RAVE

APOGEE2_SKY
int

(1<<4) Sky fiber

APOGEE2_CALIB_CLUSTER
int

(1<<10) Selected as calibration cluster member

APOGEE2_RV_STANDARD
int

(1<<3) Stellar RV standard

APOGEE2_1M_TARGET = 4194304

APOGEE2_ARGOS_OVERLAP = 32768

APOGEE2_CALIB_CLUSTER = 1024

APOGEE2_EXTERNAL_CALIB = 32

APOGEE2_GAIA_OVERLAP = 65536

APOGEE2_GALAH_OVERLAP = 131072

APOGEE2_GES_OVERLAP = 16384

APOGEE2_INTERNAL_CALIB = 64

APOGEE2_LITERATURE_CALIB = 8192

APOGEE2_OBJECT = 1073741824

APOGEE2_RAVE_OVERLAP = 262144

APOGEE2_RV_STANDARD = 8

APOGEE2_SKY = 16

APOGEE2_STANDARD_STAR = 4

APOGEE2_TELLURIC = 512

class bossdata.bits.**APOGEE2_TARGET3**
Bases: object

APOGEE2 trinary target bits

APOGEE2_YOUNG_CLUSTER
int

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

APOGEE2_SUBSTELLAR_COMPANIONS
int

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

APOGEE2_KOI_CONTROL
int

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

APOGEE2_ANCILLARY
int

(1<<8) Selected as an ancillary target

APOGEE2_MASSIVE_STAR
int

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

APOGEE2_EB
int

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

APOGEE2_KOI
int

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

APOGEE2_MDWARF
int

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

APOGEE2_ANCILLARY = 256

APOGEE2_EB = 2

APOGEE2_KOI = 1

APOGEE2_KOI_CONTROL = 4

APOGEE2_MASSIVE_STAR = 512

APOGEE2_MDWARF = 8

APOGEE2_SUBSTELLAR_COMPANIONS = 16

APOGEE2_YOUNG_CLUSTER = 32

class `bossdata.bits.`**APOGEE_ASPCAPFLAG**
Bases: `object`

APOGEE ASPCAP mask bits

METALS_BAD
int

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

TEFF_BAD
int

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

CHI2_WARN
int

(1<<8) high chi^2 (> 2*median at ASPCAP temperature (WARN)

NFE_WARN
int

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

VMICRO_WARN
int

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

COLORTE_WARN
int

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

COLORTE_BAD
int

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

TEFF_WARN
int

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

CHI2_BAD
int

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

STAR_WARN
int

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

CFE_BAD
int

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

VMICRO_BAD
int

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

CFE_WARN
int

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

LOGG_BAD
int

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

ALPHAFE_BAD
int

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

NFE_BAD
int

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

ALPHAFE_WARN
int

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

SN_WARN
int

(1<<11) S/N<70 (WARN)

SN_BAD
int

(1<<27) S/N<50 (BAD)

METALS_WARN
int

(1<<3) WARNING on metals (see PARAMFLAG[3] 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-correltion of template > 1.5 (WARN)

STAR_BAD
int

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

NO_ASPCAP_RESULT
int

(1<<31) No result

LOGG_WARN
int

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

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-correltion of template > 2 (BAD)

ALPHAFE_BAD = 1048576

ALPHAFE_WARN = 16

CFE_BAD = 2097152

CFE_WARN = 32

CHI2_BAD = 16777216

CHI2_WARN = 256

COLORTE_BAD = 33554432

COLORTE_WARN = 512

LOGG_BAD = 131072

LOGG_WARN = 2

METALS_BAD = 524288

METALS_WARN = 8

NFE_BAD = 4194304

NFE_WARN = 64

NO_ASPCAP_RESULT = 2147483648

ROTATION_BAD = 67108864

ROTATION_WARN = 1024

SN_BAD = 134217728

SN_WARN = 2048

STAR_BAD = 8388608

STAR_WARN = 128

TEFF_BAD = 65536

TEFF_WARN = 1

VMICRO_BAD = 262144

VMICRO_WARN = 4

class bossdata.bits.**APOGEE_EXTRATARG**

Bases: object

APOGEE pixel level mask bits

COMMISSIONING

int

(1<<1) Commissioning data

NOT_MAIN

int

(1<<0) Not main survey target

TELLURIC

int

(1<<2) Telluric target

DUPLICATE

int

(1<<4) Duplicate observation of star

APO1M

int

(1<<3) APO1M + APOGEE observation

APO1M = 8

COMMISSIONING = 2

DUPLICATE = 16

NOT_MAIN = 1

TELLURIC = 4

class `bossdata.bits.`**APOGEE_PARAMFLAG**

Bases: `object`

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

CALRANGE_BAD

int

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

OTHER_WARN

int

(1<<10) Other warning condition

OTHER_BAD

int

(1<<2) Other error condition

CALRANGE_WARN

int

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

GRIDEDGE_WARN

int

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

PARAM_FIXED

int

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

GRIDEDGE_BAD

int

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

CALRANGE_BAD = 2

CALRANGE_WARN = 512

GRIDEDGE_BAD = 1

GRIDEDGE_WARN = 256

OTHER_BAD = 4

OTHER_WARN = 1024

PARAM_FIXED = 65536

class `bossdata.bits.`**APOGEE_PIXMASK**

Bases: `object`

APOGEE extra targeting bits

PERSIST_MED

int

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

NOSKY
int

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

CRPIX
int

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

BADERR
int

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

PERSIST_HIGH
int

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

SIG_TELLURIC
int

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

SIG_SKYLINE
int

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

UNFIXABLE
int

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

SATPIX
int

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

BADDARK
int

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

PERSIST_LOW
int

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

LITTROW_GHOST
int

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

BADFLAT
int

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

BADPIX
int

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

BADDARK = 16

BADERR = 64

BADFLAT = 32

BADPIX = 1

CRPIX = 2

LITTROW_GHOST = 256

NOSKY = 128

PERSIST_HIGH = 512

PERSIST_LOW = 2048

PERSIST_MED = 1024

SATPIX = 4

SIG_SKYLINE = 4096

SIG_TELLURIC = 8192

UNFIXABLE = 8

class `bossdata.bits.`**APOGEE_STARFLAG**

Bases: `object`

APOGEE star-level mask bits

PERSIST_MED

int

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

SUSPECT_RV_COMBINATION

int

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

BRIGHT_NEIGHBOR

int

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

PERSIST_JUMP_NEG

int

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

BAD_PIXELS

int

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

PERSIST_LOW

int

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

LOW_SNR

int

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

VERY_BRIGHT_NEIGHBOR
int

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

SUSPECT_BROAD_LINES
int

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

COMMISSIONING
int

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

PERSIST_HIGH
int

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

PERSIST_JUMP_POS
int

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

BAD_PIXELS = 1

BRIGHT_NEIGHBOR = 4

COMMISSIONING = 2

LOW_SNR = 16

PERSIST_HIGH = 512

PERSIST_JUMP_NEG = 8192

PERSIST_JUMP_POS = 4096

PERSIST_LOW = 2048

PERSIST_MED = 1024

SUSPECT_BROAD_LINES = 131072

SUSPECT_RV_COMBINATION = 65536

VERY_BRIGHT_NEIGHBOR = 8

class bossdata.bits.**APOGEE_TARGET1**
Bases: object

APOGEE primary target bits

APOGEE_INTERMEDIATE
int

(1<<12) Intermediate cohort target

APOGEE_KEPLER_EB
int

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

APOGEE_FAINT_EXTRA
int

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

APOGEE_OLD_STAR
int

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

APOGEE_IRAC_DERED
int

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

APOGEE_CHECKED
int

(1<<31) This target has been checked

APOGEE_WISE_DERED
int

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

APOGEE_KEPLER_SEISMO
int

(1<<27) Kepler asteroseismology program target

APOGEE_LONG
int

(1<<13) Long cohort target

APOGEE_MDWARF
int

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

APOGEE_ANCILLARY
int

(1<<17) An ancillary program

APOGEE_SCI_CLUSTER
int

(1<<9) Probable cluster member

APOGEE_GC_PAL1
int

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

APOGEE_M31_CLUSTER
int

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

APOGEE_SERENDIPITOUS
int

(1<<15) Serendipitously interesting target to reobserve

APOGEE_NO_DERED
int

(1<<6) Selected using no dereddening

APOGEE_WASH_DWARF
int

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

APOGEE_WASH_GIANT
int

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

APOGEE_HIRES
int

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

APOGEE_FIRST_LIGHT
int

(1<<16) First list plate target

APOGEE_MASSIVE_STAR
int

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

APOGEE_DO_NOT_OBSERVE
int

(1<<14) Do not observe (again)

APOGEE_SHORT
int

(1<<11) Short cohort target

APOGEE_EXTENDED
int

(1<<10) Extended object

APOGEE_KEPLER_HOST
int

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

APOGEE_SEGUE_OVERLAP
int

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

APOGEE_FAINT
int

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

APOGEE_SGR_DSPH
int

(1<<26) Sagittarius dwarf spheroidal member

APOGEE_BRIGHT
int

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

APOGEE_SFD_DERED
int

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

APOGEE_MEDIUM
int

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

APOGEE_DISK_RED_GIANT
int

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

APOGEE_ANCILLARY = 131072

APOGEE_BRIGHT = 4

APOGEE_CHECKED = 2147483648

APOGEE_DISK_RED_GIANT = 4194304

APOGEE_DO_NOT_OBSERVE = 16384

APOGEE_EXTENDED = 1024

APOGEE_FAINT = 1

APOGEE_FAINT_EXTRA = 536870912

APOGEE_FIRST_LIGHT = 65536

APOGEE_GC_PAL1 = 16777216

APOGEE_HIRES = 1048576

APOGEE_INTERMEDIATE = 4096

APOGEE_IRAC_DERED = 8

APOGEE_KEPLER_EB = 8388608

APOGEE_KEPLER_HOST = 268435456

APOGEE_KEPLER_SEISMO = 134217728

APOGEE_LONG = 8192

APOGEE_M31_CLUSTER = 262144

APOGEE_MASSIVE_STAR = 33554432

APOGEE_MDWARF = 524288

APOGEE_MEDIUM = 2

APOGEE_NO_DERED = 64

APOGEE_OLD_STAR = 2097152

APOGEE_SCI_CLUSTER = 512

APOGEE_SEGUE_OVERLAP = 1073741824

APOGEE_SERENDIPITOUS = 32768

APOGEE_SFD_DERED = 32

APOGEE_SGR_DSPH = 67108864

APOGEE_SHORT = 2048

APOGEE_WASH_DWARF = 256

APOGEE_WASH_GIANT = 128

APOGEE_WISE_DERED = 16

class bossdata.bits.**APOGEE_TARGET2**

Bases: object

APOGEE secondary target bits

BUNDLE_HOLE
int

(1<<7) Bundle hole

APOGEE_KEPLER_COOLDWARF
int

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

APOGEE_CALIB_CLUSTER
int

(1<<10) Known calibration cluster member

APOGEE_LONGBAR
int

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

GUIDE_STAR
int

(1<<6) Guide star

APOGEE_RV_STANDARD
int

(1<<3) Radial velocity standard

APOGEE_FLUX_STANDARD
int

(1<<1) Flux standard

APOGEE_GC_SUPER_GIANT
int

(1<<12) Probable supergiant in Galactic Center

SKY
int

(1<<4) Sky

APOGEE_EMISSION_STAR
int

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

APOGEE_STANDARD_STAR
int

(1<<2) Stellar abundance, parameters standard

APOGEE_EMBEDDEDCLUSTER_STAR
int

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

SKY_BAD
int

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

APOGEE_TELLURIC
int

(1<<9) Hot (telluric) standard

LIGHT_TRAP
int

(1<<0) Light trap

APOGEE_MIRCLUSTER_STAR
int

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

APOGEE_CHECKED
int

(1<<31) This target has been checked

APOGEE_TELLURIC_BAD
int

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

APOGEE_GC_GIANT
int

(1<<11) Probable giant in Galactic Center

APOGEE_CALIB_CLUSTER = 1024

APOGEE_CHECKED = 2147483648

APOGEE_EMBEDDEDCLUSTER_STAR = 8192

APOGEE_EMISSION_STAR = 32768

APOGEE_FLUX_STANDARD = 2

APOGEE_GC_GIANT = 2048

APOGEE_GC_SUPER_GIANT = 4096

APOGEE_KEPLER_COOLDWARF = 65536

APOGEE_LONGBAR = 16384

APOGEE_MIRCLUSTER_STAR = 131072

APOGEE_RV_STANDARD = 8

APOGEE_STANDARD_STAR = 4

APOGEE_TELLURIC = 512

APOGEE_TELLURIC_BAD = 256

BUNDLE_HOLE = 128

GUIDE_STAR = 64

LIGHT_TRAP = 1

SKY = 16

SKY_BAD = 32

class `bossdata.bits.`**BOSSTILE_STATUS**

Bases: `object`

BOSS tiling code status bits

MIDLEVEL_PRIORITY
int

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

NAKED
int

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

DECOLLIDED
int

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

DUPLICATE
int

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

DUPLICATE_PRIMARY
int

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

KNOWN_OBJECT
int

(1<<16) galaxy has known redshift

DUPLICATE_TILED
int

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

FILLER
int

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

POSSIBLE_KNOCKOUT
int

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

REPEAT
int

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

ANCILLARY_ROUND2
int

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

IGNORE_PRIORITY
int

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

BLUEFIBER
int

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

TOOFAINT
int

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

OUT_OF_BOUNDS
int

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

BAD_CALIB_STATUS
int

(1<<14) bad CALIB_STATUS

BOSSTARGET
int

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

PREVIOUS_CHUNK
int

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

ANCILLARY
int

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

CENTERPOST
int

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

TOOBRIGHT
int

(1<<7) fibermag too bright

SUPPLEMENTARY
int

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

NOT_TILED_TARGET
int

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

TILED
int

(1<<0) assigned a fiber

ANCILLARY = 16

ANCILLARY_ROUND2 = 4194304

BAD_CALIB_STATUS = 16384

BLUEFIBER = 256

BOSSTARGET = 4

CENTERPOST = 512

DECOLLIDED = 8

DUPLICATE = 131072

DUPLICATE_PRIMARY = 262144

DUPLICATE_TILED = 524288

FILLER = 2048

IGNORE_PRIORITY = 64

KNOWN_OBJECT = 65536

MIDLEVEL_PRIORITY = 8388608

NAKED = 2

NOT_TILED_TARGET = 4096

OUT_OF_BOUNDS = 8192

POSSIBLE_KNOCKOUT = 32

PREVIOUS_CHUNK = 32768

REPEAT = 1024

SUPPLEMENTARY = 2097152

TILED = 1

TOOBRIGHT = 128

TOOFAINT = 1048576

class `bossdata.bits.`**BOSS_TARGET1**

Bases: `object`

BOSS survey primary target selection flags

QSO_UKIDSS

int

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

QSO_CORE_MAIN

int

(1<<40) Main survey core sample

SDSS_KNOWN

int

(1<<6) Matches a known SDSS spectra

QSO_BONUS

int

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

QSO_BONUS_MAIN
int

(1<<41) Main survey bonus sample

QSO_KDE
int

(1<<19) selected by kde+chi2

TEMPLATE_GAL_PHOTO
int

(1<<32) galaxy templates

GAL_CMASS_ALL
int

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

QSO_KDE_COADD
int

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

QSO_CORE_ED
int

(1<<42) Extreme Deconvolution in Core

TEMPLATE_STAR_PHOTO
int

(1<<34) stellar templates

GAL_IFIBER2_FAINT
int

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

QSO_KNOWN_MIDZ
int

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

GAL_LODPERP_DEPRECATED
int

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

QSO_CORE_LIKE
int

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

STD_FSTAR
int

(1<<20) standard f-stars

GAL_CMASS
int

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

GAL_LOZ
int

(1<<0) low-z lrgs

QSO_KNOWN_LOHIZ
int

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

QSO_NN
int

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

TEMPLATE_STAR_SPECTRO
int

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

GAL_CMASS_COMM
int

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

QSO_CORE
int

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

GAL_CMASS_SPARSE
int

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

STD_QSO
int

(1<<22) qso

QSO_LIKE
int

(1<<17) likelihood method

QSO_KNOWN_SUPPZ
int

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

STD_WD
int

(1<<21) white dwarfs

QSO_FIRST_BOSS
int

(1<<18) FIRST radio match

TEMPLATE_QSO_SDSS1
int

(1<<33) QSO templates

GAL_CMASS = 2

GAL_CMASS_ALL = 128

GAL_CMASS_COMM = 4

GAL_CMASS_SPARSE = 8

GAL_IFIBER2_FAINT = 256

GAL_LODPERP_DEPRECATED = 32

GAL_LOZ = 1

QSO_BONUS = 2048

QSO_BONUS_MAIN = 2199023255552

QSO_CORE = 1024

QSO_CORE_ED = 4398046511104

QSO_CORE_LIKE = 8796093022208

QSO_CORE_MAIN = 1099511627776

QSO_FIRST_BOSS = 262144

QSO_KDE = 524288

QSO_KDE_COADD = 65536

QSO_KNOWN_LOHIZ = 8192

QSO_KNOWN_MIDZ = 4096

QSO_KNOWN_SUPPZ = 17592186044416

QSO_LIKE = 131072

QSO_NN = 16384

QSO_UKIDSS = 32768

SDSS_KNOWN = 64

STD_FSTAR = 1048576

STD_QSO = 4194304

STD_WD = 2097152

TEMPLATE_GAL_PHOTO = 4294967296

TEMPLATE_QSO_SDSS1 = 8589934592

TEMPLATE_STAR_PHOTO = 17179869184

TEMPLATE_STAR_SPECTRO = 34359738368

class bossdata.bits.**CALIB_STATUS**

Bases: `object`

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

UNPHOT_EXTRAP_CLOUDY
int

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

PT_CLEAR
int

(1<<8) PT calibration for clear data

UNPHOT_DISJOINT
int

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

PS1_CONTRAIL
int

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

DEFAULT
int

(1<<10) a default calibration used

PHOTOMETRIC
int

(1<<0) Photometric observations

NO_UBERCAL
int

(1<<11) not uber-calibrated

PS1_PCOMP_MODEL
int

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

PS1_LOW_RMS
int

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

PS1_UNPHOT
int

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

PT_CLOUDY
int

(1<<9) PT calibration for cloudy data

INCREMENT_CALIB
int

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

ASTROMBAD
int

(1<<12) catastrophically bad astrometry

UNPHOT_OVERLAP
int

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

ASTROMBAD = 4096

DEFAULT = 1024

INCREMENT_CALIB = 32

NO_UBERCAL = 2048

PHOTOMETRIC = 1

PS1_CONTRAIL = 128

PS1_LOW_RMS = 16384

PS1_PCOMP_MODEL = 8192

PS1_UNPHOT = 64

PT_CLEAR = 256

PT_CLOUDY = 512

UNPHOT_DISJOINT = 16

UNPHOT_EXTRAP_CLEAR = 4

UNPHOT_EXTRAP_CLOUDY = 8

UNPHOT_OVERLAP = 2

class `bossdata.bits.`**EBOSS_TARGET0**
Bases: `object`

targeting bitmask for SEQUELS (eBOSS precursor)

QSO_REOBS
int

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

SEQUELS_COLLIDED
int

(1<<41) Collided galaxies from BOSS

TDSS_FES_VARBAL
int

(1<<35) TDSS Few epoch spectroscopy

SPIDERS_ERASS_AGN
int

(1<<22) ERASS AGN sources

QSO_EBOSS_KDE
int

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

SPIDERS_RASS_AGN
int

(1<<20) RASS AGN sources

QSO_EBOSS_CORE
int

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

SPIDERS_RASS_CLUS
int

(1<<21) RASS Cluster sources

TDSS_FES_DE
int

(1<<31) TDSS Few epoch spectroscopy

LRG_IZW
int

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

QSO_EBOSS_FIRST
int

(1<<14) Objects with FIRST radio matches

QSO_PTF
int

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

QSO_KNOWN
int

(1<<18) Known QSOs from previous surveys

LRG_RIW
int

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

TDSS_FES_NQHISN
int

(1<<33) TDSS Few epoch spectroscopy

TDSS_FES_DWARFC
int

(1<<32) TDSS Few epoch spectroscopy

SPIDERS_ERASS_CLUS
int

(1<<23) ERASS Cluster sources

DR9_CALIB_TARGET
int

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

DO_NOT_OBSERVE
int

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

TDSS_FES_MGII
int

(1<<34) TDSS Few epoch spectroscopy

TDSS_A
int

(1<<30) Main PanSTARRS selection for TDSS

QSO_BAD_BOSS
int

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

QSO_BOSS_TARGET
int

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

QSO_SDSS_TARGET
int

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

SEQUELS_PTF_VARIABLE
int

(1<<40) Variability objects from PTF

DO_NOT_OBSERVE = 1

DR9_CALIB_TARGET = 524288

LRG_IZW = 2

LRG_RIW = 4

QSO_BAD_BOSS = 32768

QSO_BOSS_TARGET = 65536

QSO_EBOSS_CORE = 1024

QSO_EBOSS_FIRST = 16384

QSO_EBOSS_KDE = 8192

QSO_KNOWN = 262144

QSO_PTF = 2048

QSO_REOBS = 4096

QSO_SDSS_TARGET = 131072

SEQUELS_COLLIDED = 2199023255552

SEQUELS_PTF_VARIABLE = 1099511627776

SPIDERS_ERASS_AGN = 4194304

SPIDERS_ERASS_CLUS = 8388608

SPIDERS_RASS_AGN = 1048576

SPIDERS_RASS_CLUS = 2097152

TDSS_A = 1073741824

TDSS_FES_DE = 2147483648

TDSS_FES_DWARFC = 4294967296

TDSS_FES_MGII = 17179869184

TDSS_FES_NQHISN = 8589934592

TDSS_FES_VARBAL = 34359738368

class `bossdata.bits.`**`EBOSS_TARGET1`**

Bases: `object`

targeting bitmask for eBOSS

QSO1_REOBS
int

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

QSO_SDSS_TARGET
int

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

ELG_TEST1
int

(1<<40) Test targets for ELG selection

LRG_KNOWN
int

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

STD_QSO
int

(1<<52) qso

LRG1_IDROP
int

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

QSO1_EBOSS_FIRST
int

(1<<14) Ojbects with FIRST radio matches

STD_WD
int

(1<<51) white dwarfs

QSO1_BAD_BOSS
int

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

QSO1_EBOSS_KDE
int

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

TDSS_TARGET
int

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

QSO_BOSS_TARGET
int

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

STD_FSTAR
int

(1<<50) standard f-stars

QSO_KNOWN
int

(1<<18) Known QSOs from previous surveys

QSO1_EBOSS_CORE
int

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

DO_NOT_OBSERVE
int

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

QSO1_PTF
int

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

LRG1_WISE
int

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

QSO1_VAR_S82
int

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

SPIDERS_TARGET
int

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

DO_NOT_OBSERVE = 1

ELG_TEST1 = 1099511627776

LRG1_IDROP = 4

LRG1_WISE = 2

LRG_KNOWN = 8

QSO1_BAD_BOSS = 32768

QSO1_EBOSS_CORE = 1024

QSO1_EBOSS_FIRST = 16384

QSO1_EBOSS_KDE = 8192

QSO1_PTF = 2048

QSO1_REOBS = 4096

QSO1_VAR_S82 = 512

QSO_BOSS_TARGET = 65536

QSO_KNOWN = 262144

QSO_SDSS_TARGET = 131072

SPIDERS_TARGET = 2147483648

STD_FSTAR = 1125899906842624

STD_QSO = 4503599627370496

STD_WD = 2251799813685248

TDSS_TARGET = 1073741824

class `bossdata.bits.`**`EBOSS_TARGET2`**

Bases: `object`

targeting bitmask for eBOSS

ELG_UGRIZWbright_TEST1

int

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

TDSS_FES_VARBAL

int

(1<<25) TDSS Few epoch spectroscopy

SPIDERS_ERASS_AGN

int

(1<<2) ERASS AGN sources

SPIDERS_RASS_AGN

int

(1<<0) RASS AGN sources

TDSS_CP

int

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

TDSS_FES_HYPSTAR

int

(1<<28) TDSS Few epoch spectroscopy

TDSS_FES_HYPQSO

int

(1<<27) TDSS Few epoch spectroscopy

TDSS_FES_DE
int

(1<<21) TDSS Few epoch spectroscopy

SPIDERS_RASS_CLUS
int

(1<<1) RASS Cluster sources

TDSS_FES_WDDM
int

(1<<29) TDSS Few epoch spectroscopy

ELG_SCUSS_TEST1
int

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

TDSS_FES_ACTSTAR
int

(1<<30) TDSS Few epoch spectroscopy

TDSS_FES_NQHISN
int

(1<<23) TDSS Few epoch spectroscopy

SPIDERS_XMMSL_AGN
int

(1<<4) XMM Slew survey

TDSS_FES_DWARFC
int

(1<<22) TDSS Few epoch spectroscopy

SPIDERS_ERASS_CLUS
int

(1<<3) ERASS Cluster sources

ELG_DES_TEST1
int

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

TDSS_FES_MGII
int

(1<<24) TDSS Few epoch spectroscopy

TDSS_A
int

(1<<20) Main PanSTARRS selection for TDSS

TDSS_B
int

(1<<26) Main TDSS SES version B

ELG_GRIW_TEST1
int

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

ELG_UGRIZW_TEST1
int

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

ELG_SDSS_TEST1
int

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

SPIDERS_XCLASS_CLUS
int

(1<<5) XMM serendipitous clusters

ELG_DESI_TEST1
int

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

ELG_DESI_TEST1 = 4398046511104

ELG_DES_TEST1 = 2199023255552

ELG_GRIW_TEST1 = 70368744177664

ELG_SCUSS_TEST1 = 1099511627776

ELG_SDSS_TEST1 = 8796093022208

ELG_UGRIZW_TEST1 = 17592186044416

ELG_UGRIZWbright_TEST1 = 35184372088832

SPIDERS_ERASS_AGN = 4

SPIDERS_ERASS_CLUS = 8

SPIDERS_RASS_AGN = 1

SPIDERS_RASS_CLUS = 2

SPIDERS_XCLASS_CLUS = 32

SPIDERS_XMMSL_AGN = 16

TDSS_A = 1048576

TDSS_B = 67108864

TDSS_CP = 2147483648

TDSS_FES_ACTSTAR = 1073741824

TDSS_FES_DE = 2097152

TDSS_FES_DWARFC = 4194304

TDSS_FES_HYPQSO = 134217728

TDSS_FES_HYPSTAR = 268435456

TDSS_FES_MGII = 16777216

TDSS_FES_NQHISN = 8388608

TDSS_FES_VARBAL = 33554432

TDSS_FES_WDDM = 536870912

class bossdata.bits.**FLUXMATCH_STATUS**

Bases: object

Flags from flux-based matching to SDSS photometry

FIBER_FLUXMATCH

int

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

BRIGHTEST_FLUXMATCH

int

(1<<5) picked the brightest child

NONMATCH_FLUXMATCH

int

(1<<2) flagged due to non-match

ORIGINAL_FLUXMATCH

int

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

PARENT_FLUXMATCH

int

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

NOPARENT_FLUXMATCH

int

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

BRIGHTEST_FLUXMATCH = 32

FIBER_FLUXMATCH = 2

NONMATCH_FLUXMATCH = 4

NOPARENT_FLUXMATCH = 8

ORIGINAL_FLUXMATCH = 1

PARENT_FLUXMATCH = 16

class bossdata.bits.**IMAGE_STATUS**

Bases: object

Sky and instrument conditions of SDSS image

DEAD_CCD

int

(1<<8) CCD bad (unphotometric)

BAD_ROTATOR

int

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

UNKNOWN
int

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

CLEAR
int

(1<<0) Clear skies

BAD_ASTROM
int

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

BAD_FOCUS
int

(1<<5) Focus bad (set score=0)

CLOUDY
int

(1<<1) Cloudy skies (unphotometric)

FF_PETALS
int

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

NOISY_CCD
int

(1<<9) CCD noisy (unphotometric)

SHUTTERS
int

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

BAD_ASTROM = 16

BAD_FOCUS = 32

BAD_ROTATOR = 8

CLEAR = 1

CLOUDY = 2

DEAD_CCD = 256

FF_PETALS = 128

NOISY_CCD = 512

SHUTTERS = 64

UNKNOWN = 4

class bossdata.bits.**MANGA_DAPQUAL**
Bases: object

Mask bits for MaNGA DAP quality flags

VALIDFILE
int

(1<<0) File is valid

VALIDFILE = 1

class `bossdata.bits.`**MANGA_DRP2PIXMASK**

Bases: `object`

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

BRIGHTSKY

int

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

SMEARHIGHSN

int

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

MANYBADCOLUMNS

int

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

BADARC

int

(1<<3) Bad arc solution

REDMONSTER

int

(1<<29) Contiguous region of bad chi^2 in sky residuals (with threshhold of relative chi^2 > 3).

BADTRACE

int

(1<<1) Bad trace

FULLREJECT

int

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

NOPLUG

int

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

NEARWHOPPER

int

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

MANYREJECTED

int

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

LARGESHIFT

int

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

NOSKY

int

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

WHOPPER
int

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

BADFLUXFACTOR
int

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

CROSSTALK
int

(1<<22) Cross-talk significant

COSMIC
int

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

DEADFIBER
int

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

SMEARIMAGE
int

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

COMBINEREJ
int

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

BADSKYFIBER
int

(1<<7) Sky fiber shows extreme residuals

PARTIALREJECT
int

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

LOWFLAT
int

(1<<18) Flat field less than 0.5

NODATA
int

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

BADSKYCHI
int

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

SCATTEREDLIGHT
int

(1<<21) Scattered light significant

NEARBADPIXEL
int

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

_3DREJECT
int

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

BADFLAT
int

(1<<2) Low counts in fiberflat

BADPIX
int

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

SMEARMEDSN
int

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

BADARC = 8

BADFLAT = 4

BADFLUXFACTOR = 134217728

BADPIX = 32768

BADSKYCHI = 268435456

BADSKYFIBER = 128

BADTRACE = 2

BRIGHTSKY = 16777216

COMBINEREJ = 67108864

COSMIC = 65536

CROSSTALK = 4194304

DEADFIBER = 8192

FULLREJECT = 524288

LARGESHIFT = 64

LOWFLAT = 262144

MANYBADCOLUMNS = 16

MANYREJECTED = 32

NEARBADPIXEL = 131072

NEARWHOPPER = 256

NODATA = 33554432

NOPLUG = 1

NOSKY = 8388608

PARTIALREJECT = 1048576

REDMONSTER = 536870912

SCATTEREDLIGHT = 2097152

SMEARHIGHSN = 2048

SMEARIMAGE = 1024

SMEARMEDSN = 4096

WHOPPER = 512

class `bossdata.bits.`**MANGA_DRP2QUAL**

Bases: `object`

Mask bits for MaNGA DRP-2d quality flags

FULLCLOUD

int

(1<<12) Completely cloudy exposure

LOWEXPTIME

int

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

EXTRACTBRIGHT

int

(1<<2) Extracted spectra abnormally bright

SKYSUBBAD

int

(1<<10) Bad sky subtraction

BADIFU

int

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

SKYSUBFAIL

int

(1<<11) Failed sky subtraction

ARCFOCUS

int

(1<<8) Bad focus on arc frames

EXTRACTBAD

int

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

SCATFAIL

int

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

RAMPAGINGBUNNY

int

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

BADDITHER
int

(1<<7) Bad dither location information

HIGHSCAT
int

(1<<5) High scattered light levels

VALIDFILE
int

(1<<0) File is valid

ARCFOCUS = 256

BADDITHER = 128

BADIFU = 16

EXTRACTBAD = 2

EXTRACTBRIGHT = 4

FULLCLOUD = 4096

HIGHSCAT = 32

LOWEXPTIME = 8

RAMPAGINGBUNNY = 512

SCATFAIL = 64

SKYSUBBAD = 1024

SKYSUBFAIL = 2048

VALIDFILE = 1

class `bossdata.bits.`**MANGA_DRP3PIXMASK**
Bases: `object`

Mask bits per spaxel for a MaNGA data cube.

DEADFIBER
int

(1<<2) Major contributing fiber is dead

NOCOV
int

(1<<0) No coverage in cube

DONOTUSE
int

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

LOWCOV
int

(1<<1) Low coverage depth in cube

FORESTAR
int

(1<<3) Foreground star

DEADFIBER = 4

DONOTUSE = 1024

FORESTAR = 8

LOWCOV = 2

NOCOV = 1

class `bossdata.bits.`**MANGA_DRP3QUAL**
Bases: `object`

Mask bits for MaNGA DRP-3d quality flags

VALIDFILE
int

(1<<0) File is valid

SKYSUBBAD
int

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

BADDEPTH
int

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

BADOMEGA
int

(1<<6) Omega greater than threshhold in one or more sets

BADSET
int

(1<<7) One or more sets are bad

CRITICAL
int

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

BADASTROM
int

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

VARIABLELSF
int

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

BADFLUX
int

(1<<8) Bad flux calibration

HIGHSCAT
int

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

BADASTROM = 16

BADDEPTH = 2

BADFLUX = 256

BADOMEGA = 64

BADSET = 128

CRITICAL = 1073741824

HIGHSCAT = 8

SKYSUBBAD = 4

VALIDFILE = 1

VARIABLELSF = 32

class `bossdata.bits.`**MANGA_TARGET1**
Bases: `object`

Mask bits identifying galaxy samples.

SECONDARY_COM2
int

(1<<8) July 2014 commissioning

PRIMARY_v1_1_0
int

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

NONE
int

(1<<0)

SECONDARY_v1_1_0
int

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

COLOR_ENHANCED_v1_1_0
int

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

PRIMARY_v1_2_0
int

(1<<10)

FILLER
int

(1<<13) Filler targets

SECONDARY_v1_2_0
int

(1<<11)

COLOR_ENHANCED_COM2
int

(1<<9) July 2014 commissioning

COLOR_ENHANCED_COM
int

(1<<3) March 2014 commissioning

PRIMARY_PLUS_COM
int

(1<<1) March 2014 commissioning

ANCILLARY
int

(1<<14) Ancillary program targets

SECONDARY_COM
int

(1<<2) March 2014 commissioning

COLOR_ENHANCED_v1_2_0
int

(1<<12)

PRIMARY_COM2
int

(1<<7) July 2014 commissioning

ANCILLARY = 16384

COLOR_ENHANCED_COM = 8

COLOR_ENHANCED_COM2 = 512

COLOR_ENHANCED_v1_1_0 = 64

COLOR_ENHANCED_v1_2_0 = 4096

FILLER = 8192

NONE = 1

PRIMARY_COM2 = 128

PRIMARY_PLUS_COM = 2

PRIMARY_v1_1_0 = 16

PRIMARY_v1_2_0 = 1024

SECONDARY_COM = 4

SECONDARY_COM2 = 256

SECONDARY_v1_1_0 = 32

SECONDARY_v1_2_0 = 2048

class `bossdata.bits.`**`MANGA_TARGET2`**

Bases: `object`

Mask bits identifying non-galaxy samples.

`NONE`

int

(1<<0)

`STD_WD_COM`

int

(1<<21)

`STELLIB_2MASS_COM`

int

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

`STD_STD_COM`

int

(1<<22)

`SKY`

int

(1<<1)

`STELLIB_COM_mar2015`

int

(1<<5) Commissioning selection in March 2015

`STELLIB_KNOWN_COM`

int

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

`STELLIB_SDSS_COM`

int

(1<<2) Commissioning selection using SDSS photometry

`STD_APASS_COM`

int

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

`STD_FSTAR`

int

(1<<23)

`STD_FSTAR_COM`

int

(1<<20)

`STD_WD`

int

(1<<24)

`NONE` **= 1**

SKY = 2

STD_APASS_COM = 33554432

STD_FSTAR = 8388608

STD_FSTAR_COM = 1048576

STD_STD_COM = 4194304

STD_WD = 16777216

STD_WD_COM = 2097152

STELLIB_2MASS_COM = 8

STELLIB_COM_mar2015 = 32

STELLIB_KNOWN_COM = 16

STELLIB_SDSS_COM = 4

class bossdata.bits.**MANGA_TARGET3**

Bases: object

Mask bits identifying ancillary samples.

AGN_WISE

int

(1<<3)

NONE

int

(1<<0)

DEEP_COMA

int

(1<<19)

LETTERS

int

(1<<11)

DWARF

int

(1<<14)

AGN_PALOMAR

int

(1<<4)

EDGE_ON_WINDS

int

(1<<6)

VOID

int

(1<<5)

BCG
int

(1<<17)

PAIR_RECENTER
int

(1<<8)

RADIO_JETS
int

(1<<15)

ANGST
int

(1<<18)

MWA
int

(1<<13)

AGN_OIII
int

(1<<2)

MASSIVE
int

(1<<12)

PAIR_ENLARGE
int

(1<<7)

PAIR_SIM
int

(1<<9)

DISKMASS
int

(1<<16)

AGN_BAT
int

(1<<1)

PAIR_2IFU
int

(1<<10)

AGN_BAT = 2

AGN_OIII = 4

AGN_PALOMAR = 16

AGN_WISE = 8

ANGST = 262144

BCG = 131072

DEEP_COMA = 524288

DISKMASS = 65536

DWARF = 16384

EDGE_ON_WINDS = 64

LETTERS = 2048

MASSIVE = 4096

MWA = 8192

NONE = 1

PAIR_2IFU = 1024

PAIR_ENLARGE = 128

PAIR_RECENTER = 256

PAIR_SIM = 512

RADIO_JETS = 32768

VOID = 32

class bossdata.bits.**M_EYEBALL**

Bases: object

Eyeball flags for mergers in VAGC

QUESTIONABLE

int

(1<<2)

DRY

int

(1<<3)

MAJOR

int

(1<<7)

REPEAT

int

(1<<12)

MULTIPLE

int

(1<<8)

ALL_RED

int

(1<<9)

SHELLS
int

(1<<5)

NOT_MERGER
int

(1<<1)

TIDAL_TAILS
int

(1<<4)

ALL_BLUE
int

(1<<10)

AFTER
int

(1<<15)

DONE
int

(1<<0)

DURING
int

(1<<14)

MIXED_REDBLUE
int

(1<<11)

RING
int

(1<<6)

BEFORE
int

(1<<13)

AFTER = 32768

ALL_BLUE = 1024

ALL_RED = 512

BEFORE = 8192

DONE = 1

DRY = 8

DURING = 16384

MAJOR = 128

MIXED_REDBLUE = 2048

MULTIPLE = 256

NOT_MERGER = 2

QUESTIONABLE = 4

REPEAT = 4096

RING = 64

SHELLS = 32

TIDAL_TAILS = 16

class bossdata.bits.**OBJECT1**

Bases: object

Object flags from photo reductions for SDSS (first 32)

NOPROFILE

int

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

INTERP

int

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

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.

DEBLENDED_AS_PSF

int

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

NOSTOKES

int

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

BAD_RADIAL

int

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

PEAKCENTER

int

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

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.

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.

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).

SATUR
int

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

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.

BINNED2
int

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

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 to close to the edge to extract a profile at the local center, and OBJECT1_EDGE is also set.

MANYPETRO
int

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

BINNED1
int

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

CHILD
int

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

NODEBLEND
int

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

INCOMPLETE_PROFILE
int

(1<<16) A circle, centerd 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.

PETROFAINT

int

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

MOVED

int

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

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.

SUBTRACTED

int

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

ELLIPFAINT

int

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

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).

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).

EDGE

int

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

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.

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.

NOPETRO
int

(1<<8) No Petrosian radius or other Petrosian quanties could be measured.

BADSKY = 4194304

BAD_RADIAL = 32768

BINNED1 = 268435456

BINNED2 = 536870912

BINNED4 = 1073741824

BLENDED = 8

BRIGHT = 2

CANONICAL_CENTER = 1

CHILD = 16

CR = 4096

DEBLENDED_AS_PSF = 33554432

DEBLEND_PRUNED = 67108864

DEBLEND_TOO_MANY_PEAKS = 2048

EDGE = 4

ELLIPFAINT = 134217728

INCOMPLETE_PROFILE = 65536

INTERP = 131072

MANYPETRO = 512

MANYR50 = 8192

MANYR90 = 16384

MOVED = 2147483648

NODEBLEND = 64

NOPETRO = 256

NOPETRO_BIG = 1024

NOPROFILE = 128

NOSTOKES = 2097152

NOTCHECKED = 524288

PEAKCENTER = 32

PETROFAINT = 8388608

SATUR = 262144

SUBTRACTED = 1048576

TOO_LARGE = 16777216

class `bossdata.bits.`**`OBJECT2`**

Bases: `object`

Object flags from photo reductions for SDSS (second 32)

`AMOMENT_UNWEIGHTED`

int

(1<<21) 'Adaptive' moments are actually unweighted.

`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.

`DEBLEND_PEEPHOLE`

int

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

`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

`NOTCHECKED_CENTER`

int

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

`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.

`DEBLENDED_AS_MOVING`

int

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

`INTERP_CENTER`

int

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

`AMOMENT_MAXITER`

int

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

`BAD_MOVING_FIT`

int

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

`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.

DEBLEND_NOPEAK
int

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

STATIONARY
int

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

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.

MAYBE_EGHOST
int

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

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.

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.

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.

NODEBLEND_MOVING
int

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

BRIGHTEST_GALAXY_CHILD
int

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

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.

TOO_FEW_GOOD_DETECTIONS
int

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

BINNED_CENTER
int

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

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).

BAD_COUNTS_ERROR
int

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

SPARE1
int

(1<<31)

PEAKS_TOO_CLOSE
int

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

SPARE3
int

(1<<29)

SPARE2
int

(1<<30)

TOO_FEW_DETECTIONS
int

(1<<2) The object has the MOVED flag set, but has too few detection 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.

AMOMENT_MAXITER = 8388608

AMOMENT_SHIFT = 4194304

AMOMENT_UNWEIGHTED = 2097152

BAD_COUNTS_ERROR = 256

BAD_MOVING_FIT = 8

BAD_MOVING_FIT_CHILD = 512

BINNED_CENTER = 64

BRIGHTEST_GALAXY_CHILD = 524288

CANONICAL_BAND = 1048576

CENTER_OFF_AIMAGE = 131072

DEBLENDED_AS_MOVING = 1

DEBLENDED_AT_EDGE = 8192

DEBLEND_DEGENERATE = 262144

DEBLEND_NOPEAK = 16384

DEBLEND_PEEPHOLE = 268435456

DEBLEND_UNASSIGNED_FLUX = 1024

HAS_SATUR_DN = 134217728

INTERP_CENTER = 4096

LOCAL_EDGE = 128

MAYBE_CR = 16777216

MAYBE_EGHOST = 33554432

NODEBLEND_MOVING = 2

NOTCHECKED_CENTER = 67108864

PEAKS_TOO_CLOSE = 32

PSF_FLUX_INTERP = 32768

SATUR_CENTER = 2048

SPARE1 = 2147483648

SPARE2 = 1073741824

SPARE3 = 536870912

STATIONARY = 16

TOO_FEW_DETECTIONS = 4

TOO_FEW_GOOD_DETECTIONS = 65536

class `bossdata.bits.`**`Q_EYEBALL`**

Bases: `object`

Quality eyeball flags from VAGC

GOOD_Z

int

(1<<26)

QSO_ON_GALAXY

int

(1<<18)

BAD_DEBLEND

int

(1<<4)

UNCLASSIFIABLE

int

(1<<2)

DONE
int

(1<<0)

USE_PARENT
int

(1<<15)

SATELLITE
int

(1<<10)

DOUBLE_Z
int

(1<<23)

USE_CHILD_IMAGE
int

(1<<27)

IS_STAR
int

(1<<22)

USE_CHILD_SPECTRUM
int

(1<<28)

IN_HUGE_OBJECT
int

(1<<16)

BAD_SPEC_CLASS
int

(1<<14)

DOUBLE_STAR
int

(1<<6)

HII
int

(1<<7)

BAD_SPECTRUM
int

(1<<20)

BAD_PARENT_CENTER
int

(1<<25)

INTERNAL_REFLECTION
int

(1<<13)

EDGE
int

(1<<9)

OTHER
int

(1<<1)

FLECK
int

(1<<5)

BAD_Z
int

(1<<12)

POSSIBLE_LENS
int

(1<<21)

USE_ANYWAY
int

(1<<8)

NEED_BIGGER_IMAGE
int

(1<<3)

STAR_ON_GALAXY
int

(1<<17)

PLANETARY_NEBULA
int

(1<<24)

PLANE
int

(1<<11)

NEGATIVE_QSO_FIT
int

(1<<19)

BAD_DEBLEND = 16

BAD_PARENT_CENTER = 33554432

BAD_SPECTRUM = 1048576

BAD_SPEC_CLASS = 16384

BAD_Z = 4096

DONE = 1

DOUBLE_STAR = 64

DOUBLE_Z = 8388608

EDGE = 512

FLECK = 32

GOOD_Z = 67108864

HII = 128

INTERNAL_REFLECTION = 8192

IN_HUGE_OBJECT = 65536

IS_STAR = 4194304

NEED_BIGGER_IMAGE = 8

NEGATIVE_QSO_FIT = 524288

OTHER = 2

PLANE = 2048

PLANETARY_NEBULA = 16777216

POSSIBLE_LENS = 2097152

QSO_ON_GALAXY = 262144

SATELLITE = 1024

STAR_ON_GALAXY = 131072

UNCLASSIFIABLE = 4

USE_ANYWAY = 256

USE_CHILD_IMAGE = 134217728

USE_CHILD_SPECTRUM = 268435456

USE_PARENT = 32768

class `bossdata.bits.`**RESOLVE_STATUS**

Bases: `object`

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_EDGE

int

(1<<4) near lowest or highest column

RUN_OVERLAPONLY
int

(1<<2) only appears in the overlap between two fields

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_IGNORE
int

(1<<3) bright or parent object that should be ignored

RUN_PRIMARY
int

(1<<0) primary within the objects own run (but not necessarily for the survey as a whole)

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

RUN_DUPLICATE
int

(1<<5) duplicate measurement of same pixels in two different fields

SURVEY_EDGE
int

(1<<12) Not kept as secondary because it is RUN_RAMP or RUN_EDGE object

RUN_RAMP
int

(1<<1) in what would be the overlap area of a field, but with no neighboring field

RUN_DUPLICATE = 32

RUN_EDGE = 16

RUN_IGNORE = 8

RUN_OVERLAPONLY = 4

RUN_PRIMARY = 1

RUN_RAMP = 2

SURVEY_BADFIELD = 2048

SURVEY_BEST = 512

SURVEY_EDGE = 4096

SURVEY_PRIMARY = 256

SURVEY_SECONDARY = 1024

class `bossdata.bits.`**`SEGUE1_TARGET`**

Bases: `object`

SEGUE-1 primary target bits

`SEGUE1_BHB`

int

(1<<13) blue horizontal branch star

`SEGUE1_AGB`

int

(1<<23) asympototic giant branch stars

`SEGUE1_MSWD`

int

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

`SEG1LOW_TO`

int

(1<<11) low latitude selection of bluetip stars

`SEGUE1_SDM`

int

(1<<22) M sub-dwarfs

`SEGUE1_BD`

int

(1<<21) brown dwarfs

`SEGUE1_LM`

int

(1<<16) low metallicity star

`SEGUE1_WD`

int

(1<<19) white dwarf

`SEGUE1_MAN`

int

(1<<24) manual selection

`SEG1LOW_KG`

int

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

`SEGUE1_CHECKED`

int

(1<<31) was a checked object

`SEGUE1_CWD`

int

(1<<17) cool white dwarf

SEGUE1_MPMSTO
int

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

SEG1LOW_AGB
int

(1<<27) low latitude selection of AGB stars

SEGUE1_GD
int

(1<<18) G-dwarf

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_KD
int

(1<<15) K-dwarfs

SEGUE1_KG
int

(1<<14) K-giants (l and red)

SEG1LOW_AGB = 134217728

SEG1LOW_KG = 1024

SEG1LOW_TO = 2048

SEGUE1_AGB = 8388608

SEGUE1_BD = 2097152

SEGUE1_BHB = 8192

SEGUE1_CHECKED = 2147483648

SEGUE1_CWD = 131072

SEGUE1_FG = 512

SEGUE1_GD = 262144

SEGUE1_KD = 32768

SEGUE1_KG = 16384

SEGUE1_LM = 65536

SEGUE1_MAN = 16777216

SEGUE1_MPMSTO = 1048576

SEGUE1_MSWD = 4096

SEGUE1_SDM = 4194304

SEGUE1_WD = 524288

class `bossdata.bits.`**`SEGUE1_TARGET2`**

Bases: `object`

SEGUE-1 secondary target bits

`SEGUE1_SCIENCE`

int

(1<<30) SEGUE-1 science target

`SPECTROPHOTO_STD`

int

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

`SKY`

int

(1<<4) sky target

`REDDEN_STD`

int

(1<<1) reddening standard star

`SEGUE1_QA`

int

(1<<3) QA Duplicate Observations (unused)

`SEGUE1_TEST`

int

(1<<31) SEGUE-1 test target

`REDDEN_STD` = 2

`SEGUE1_QA` = 8

`SEGUE1_SCIENCE` = 1073741824

`SEGUE1_TEST` = 2147483648

`SKY` = 16

`SPECTROPHOTO_STD` = 32

class `bossdata.bits.`**`SEGUE2_TARGET1`**

Bases: `object`

SEGUE-2 primary target bits

`SEGUE2_MSTO`

int

(1<<0) Main-sequence turnoff

`SEGUE2_CHECKED`

int

(1<<31) was a checked object

`SEGUE2_REDKG`

int

(1<<1) Red K-giant stars

SEGUE2_PMKG
int

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

SEGUE2_BHB
int

(1<<13) Blue horizontal branch star

SEGUE2_CWD
int

(1<<17) Cool white dwarf

SEGUE2_HVS
int

(1<<5) hyper velocity candidate

SEGUE2_LKG
int

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

SEGUE2_MII
int

(1<<7) M giant

SEGUE2_XDM
int

(1<<6) extreme sdM star

SEGUE2_HHV
int

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

SEGUE2_LM
int

(1<<4) Low metallicity

SEGUE2_BHB = 8192

SEGUE2_CHECKED = 2147483648

SEGUE2_CWD = 131072

SEGUE2_HHV = 256

SEGUE2_HVS = 32

SEGUE2_LKG = 4

SEGUE2_LM = 16

SEGUE2_MII = 128

SEGUE2_MSTO = 1

SEGUE2_PMKG = 8

SEGUE2_REDKG = 2

SEGUE2_XDM = 64

class bossdata.bits.**SEGUE2_TARGET2**

Bases: object

SEGUE-2 secondary target bits

SEGUE2_CHECKED

int

(1<<31) was a checked object

SEGUE2_REDDENING

int

(1<<1) reddening standard

SEGUE2_SPECPHOTO

int

(1<<5) spectrophotometric star

SEGUE2_CLUSTER

int

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

QUALITY_HOLE

int

(1<<8) quality hole

SKY

int

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

BUNDLE_HOLE

int

(1<<7) bundle hole

HOT_STD

int

(1<<9) hot standard

GUIDE_STAR

int

(1<<6) guide star

SEGUE2_STETSON

int

(1<<11) Stetson standard target

SEGUE2_QA

int

(1<<3) repeat target across plates

LIGHT_TRAP

int

(1<<0) light trap hole

SEGUE2_TEST
int

(1<<2) test target

BUNDLE_HOLE = 128

GUIDE_STAR = 64

HOT_STD = 512

LIGHT_TRAP = 1

QUALITY_HOLE = 256

SEGUE2_CHECKED = 2147483648

SEGUE2_CLUSTER = 1024

SEGUE2_QA = 8

SEGUE2_REDDENING = 2

SEGUE2_SPECPHOTO = 32

SEGUE2_STETSON = 2048

SEGUE2_TEST = 4

SKY = 16

class `bossdata.bits.`**SPECIAL_TARGET1**
Bases: `object`

SDSS special program target bits

ALLPSF_NONSTELLAR
int

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

U_EXTRA
int

(1<<23) extra u-band target

FAINT_QSO
int

(1<<26) faint QSO in south

ORION_BD
int

(1<<12) Brown dwarf in Orion

BCG
int

(1<<10) brightest cluster galaxy

ORION_MSTAR_LATE
int

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

FAINT_LRG
int

(1<<25) faint LRG in south

COMMISSIONING_STAR
int

(1<<3) star in commissioning

FSTAR
int

(1<<5) F-stars

DEEP_GALAXY_RED
int

(1<<8) deep LRG

ALLPSF_STELLAR
int

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

PREBOSS_QSO
int

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

LOWZ_ANNIS
int

(1<<1) low-redshift cluster galaxy

VARIABLE_LOPRI
int

(1<<30) low priority variable

TAURUS_GALAXY
int

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

PREMARVELS
int

(1<<19) pre-MARVELS stellar target

U_EXTRA2
int

(1<<24) extra u-band target

BENT_RADIO
int

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

LOWZ_GALAXY
int

(1<<7) low-redshift galaxy

TAURUS_STAR
int

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

SOUTHERN_COMPLETE
int

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

PERSEUS
int

(1<<37) galaxy in perseus-pisces

MSTURNOFF
int

(1<<11) main sequence turnoff

LOWZ_LOVEDAY
int

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

ALLPSF
int

(1<<31) i<19.1 point sources

HIPM
int

(1<<34) high proper motion

DISKSTAR
int

(1<<4) thin/thick disk star

SPECIAL_FILLER
int

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

U_PRIORITY
int

(1<<22) priority u-band target

APBIAS
int

(1<<0) aperture bias target

QSO_M31
int

(1<<2) QSO in M31

STRAIGHT_RADIO
int

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

SOUTHERN_EXTENDED
int

(1<<20) simple extension of southern targets

PREBOSS_LRG
int

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

PHOTOZ_GALAXY
int

(1<<16) test galaxy for photometric redshifts

HYADES_MSTAR
int

(1<<6) M-star in Hyades

VARIABLE_HIPRI
int

(1<<29) high priority variable

DEEP_GALAXY_RED_II
int

(1<<9) deep LRG

ORION_MSTAR_EARLY
int

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

ALLPSF = 2147483648

ALLPSF_NONSTELLAR = 4294967296

ALLPSF_STELLAR = 8589934592

APBIAS = 1

BCG = 1024

BENT_RADIO = 134217728

COMMISSIONING_STAR = 8

DEEP_GALAXY_RED = 256

DEEP_GALAXY_RED_II = 512

DISKSTAR = 16

FAINT_LRG = 33554432

FAINT_QSO = 67108864

FSTAR = 32

HIPM = 17179869184

HYADES_MSTAR = 64

LOWZ_ANNIS = 2

LOWZ_GALAXY = 128

LOWZ_LOVEDAY = 274877906944

MSTURNOFF = 2048

ORION_BD = 4096

ORION_MSTAR_EARLY = 8192

ORION_MSTAR_LATE = 16384

PERSEUS = 137438953472

PHOTOZ_GALAXY = 65536

PREBOSS_LRG = 262144

PREBOSS_QSO = 131072

PREMARVELS = 524288

QSO_M31 = 4

SOUTHERN_COMPLETE = 2097152

SOUTHERN_EXTENDED = 1048576

SPECIAL_FILLER = 32768

STRAIGHT_RADIO = 268435456

TAURUS_GALAXY = 68719476736

TAURUS_STAR = 34359738368

U_EXTRA = 8388608

U_EXTRA2 = 16777216

U_PRIORITY = 4194304

VARIABLE_HIPRI = 536870912

VARIABLE_LOPRI = 1073741824

class `bossdata.bits.`**SPPIXMASK**

Bases: `object`

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

BRIGHTSKY

int

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

SMEARHIGHSN

int

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

MANYBADCOLUMNS

int

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

BADARC

int

(1<<3) Bad arc solution

REDMONSTER
int

(1<<28) Contiguous region of bad chi^2 in sky residuals (with threshhold of relative chi^2 > 3).

BADTRACE
int

(1<<1) Bad trace from routine TRACE320CRUDE

FULLREJECT
int

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

NOPLUG
int

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

NEARWHOPPER
int

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

MANYREJECTED
int

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

LARGESHIFT
int

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

NOSKY
int

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

WHOPPER
int

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

BADFLUXFACTOR
int

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

CROSSTALK
int

(1<<21) Cross-talk significant

SMEARIMAGE
int

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

COMBINEREJ
int

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

BADSKYFIBER
int

(1<<7) Sky fiber shows extreme residuals

PARTIALREJECT
int

(1<<19) Some pixels rejected in extraction

LOWFLAT
int

(1<<17) Flat field less than 0.5

NODATA
int

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

BADSKYCHI
int

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

SCATTEREDLIGHT
int

(1<<20) Scattered light significant

NEARBADPIXEL
int

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

BADFLAT
int

(1<<2) Low counts in fiberflat

SMEARMEDSN
int

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

BADARC = 8

BADFLAT = 4

BADFLUXFACTOR = 67108864

BADSKYCHI = 134217728

BADSKYFIBER = 128

BADTRACE = 2

BRIGHTSKY = 8388608

COMBINEREJ = 33554432

CROSSTALK = 2097152

FULLREJECT = 262144

LARGESHIFT = 64

LOWFLAT = 131072

MANYBADCOLUMNS = 16

MANYREJECTED = 32

NEARBADPIXEL = 65536

NEARWHOPPER = 256

NODATA = 16777216

NOPLUG = 1

NOSKY = 4194304

PARTIALREJECT = 524288

REDMONSTER = 268435456

SCATTEREDLIGHT = 1048576

SMEARHIGHSN = 2048

SMEARIMAGE = 1024

SMEARMEDSN = 4096

WHOPPER = 512

class `bossdata.bits.`**TARGET**

Bases: `object`

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

QSO_FIRST_SKIRT

int

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

QSO_CAP

int

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

GALAXY_RED

int

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

STAR_CARBON

int

(1<<14) dwarf and giant carbon stars

STAR_WHITE_DWARF

int

(1<<19) hot white dwarfs

GALAXY_RED_II

int

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

GALAXY_BIG

int

(1<<7) Low-surface brightness main sample galaxy (mu50>23 in r-band)

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_MANUAL
int

(1<<24) manual serendipity flag

STAR_SUB_DWARF
int

(1<<16) low-luminosity subdwarfs

QSO_FIRST_CAP
int

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

QSO_SKIRT
int

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

STAR_PN
int

(1<<28) central stars of planetary nebulae

STAR_BHB
int

(1<<13) blue horizontal-branch stars

QSO_HIZ
int

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

STAR_BROWN_DWARF
int

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

SERENDIP_FIRST
int

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

SOUTHERN_SURVEY
int

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

STAR_RED_DWARF
int

(1<<18) red dwarfs

STAR_CATY_VAR
int

(1<<17) cataclysmic variables

QSO_REJECT
int

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

GALAXY
int

(1<<6) Main sample galaxy

SERENDIP_RED
int

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

SERENDIP_DISTANT
int

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

QSO_MAG_OUTLIER
int

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

ROSAT_A
int

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

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

ROSAT_B
int

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

ROSAT_E
int

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

ROSAT_D
int

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

SERENDIP_BLUE
int

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

GALAXY = 64

GALAXY_BIG = 128

GALAXY_BRIGHT_CORE = 256

GALAXY_RED = 32

GALAXY_RED_II = 67108864

QSO_CAP = 2

QSO_FIRST_CAP = 8

QSO_FIRST_SKIRT = 16

QSO_HIZ = 1

QSO_MAG_OUTLIER = 33554432

QSO_REJECT = 536870912

QSO_SKIRT = 4

ROSAT_A = 512

ROSAT_B = 1024

ROSAT_C = 2048

ROSAT_D = 4096

ROSAT_E = 134217728

SERENDIP_BLUE = 1048576

SERENDIP_DISTANT = 8388608

SERENDIP_FIRST = 2097152

SERENDIP_MANUAL = 16777216

SERENDIP_RED = 4194304

SOUTHERN_SURVEY = 2147483648

STAR_BHB = 8192

STAR_BROWN_DWARF = 32768

STAR_CARBON = 16384

STAR_CATY_VAR = 131072

STAR_PN = 268435456

STAR_RED_DWARF = 262144

STAR_SUB_DWARF = 65536

STAR_WHITE_DWARF = 524288

class bossdata.bits.**TTARGET**

Bases: object

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

HOT_STD

int

(1<<9) hot standard star

BUNDLE_HOLE

int

(1<<7) fiber bundle hole

SPECTROPHOTO_STD

int

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

QUALITY_HOLE
int

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

SKY
int

(1<<4) sky target

SOUTHERN_SURVEY
int

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

QA
int

(1<<3) quality assurance target

GUIDE_STAR
int

(1<<6) guide star hole

REDDEN_STD
int

(1<<1) reddening standard star

TEST_TARGET
int

(1<<2) a test target

LIGHT_TRAP
int

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

BUNDLE_HOLE = 128

GUIDE_STAR = 64

HOT_STD = 512

LIGHT_TRAP = 1

QA = 8

QUALITY_HOLE = 256

REDDEN_STD = 2

SKY = 16

SOUTHERN_SURVEY = 2147483648

SPECTROPHOTO_STD = 32

TEST_TARGET = 4

class `bossdata.bits.`**T_EYEBALL**
Bases: `object`

Type eyeball flags from VAGC

BAR
int

(1<<18)

PSF
int

(1<<13)

S0
int

(1<<7)

UNCLASSIFIABLE
int

(1<<2)

OUTFLOW
int

(1<<27)

DONE
int

(1<<0)

DUST_LANE
int

(1<<17)

RING
int

(1<<19)

UNUSED_0
int

(1<<6)

DISK
int

(1<<4)

TIDAL_TAILS
int

(1<<20)

MERGER
int

(1<<26)

HII_REGIONS
int

(1<<15)

NEAR_NEIGHBORS
int

(1<<25)

IRREGULAR
int

(1<<5)

ASYMMETRIC
int

(1<<14)

OTHER
int

(1<<1)

DUST_ASYMMETRY
int

(1<<24)

SPIRAL_STRUCTURE
int

(1<<16)

PITCH_4
int

(1<<12) openly wound

SHELLS
int

(1<<21)

PITCH_0
int

(1<<8) tightly wound

PITCH_1
int

(1<<9)

PITCH_2
int

(1<<10)

PITCH_3
int

(1<<11)

WARPED_DISK
int

(1<<23)

ELLIPTICAL
int

(1<<3)

BLUE_CORE
int

(1<<22)

ASYMMETRIC = 16384

BAR = 262144

BLUE_CORE = 4194304

DISK = 16

DONE = 1

DUST_ASYMMETRY = 16777216

DUST_LANE = 131072

ELLIPTICAL = 8

HII_REGIONS = 32768

IRREGULAR = 32

MERGER = 67108864

NEAR_NEIGHBORS = 33554432

OTHER = 2

OUTFLOW = 134217728

PITCH_0 = 256

PITCH_1 = 512

PITCH_2 = 1024

PITCH_3 = 2048

PITCH_4 = 4096

PSF = 8192

RING = 524288

S0 = 128

SHELLS = 2097152

SPIRAL_STRUCTURE = 65536

TIDAL_TAILS = 1048576

UNCLASSIFIABLE = 4

UNUSED_0 = 64

WARPED_DISK = 8388608

class `bossdata.bits.`**VAGC_SELECT**
Bases: `object`

Selection flags for Main VAGC sample

MAIN
int

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

PLATEHOLE
int

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

TILED
int

(1<<0) selected because near a tiled target

MAIN = 4

PLATEHOLE = 2

TILED = 1

class `bossdata.bits.`**ZWARNING**
Bases: `object`

Warnings for SDSS spectra.

SMALL_DELTA_CHI2
int

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

LITTLE_COVERAGE
int

(1<<1) too little wavelength coverage (WCOVERAGE < 0.18)

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

BAD_TARGET
int

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

NEGATIVE_MODEL
int

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

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)

MANY_OUTLIERS
int

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

NODATA
int

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

UNPLUGGED
int

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

BAD_TARGET = 256

LITTLE_COVERAGE = 2

MANY_OUTLIERS = 16

NEGATIVE_EMISSION = 64

NEGATIVE_MODEL = 8

NODATA = 512

SKY = 1

SMALL_DELTA_CHI2 = 4

UNPLUGGED = 128

Z_FITLIMIT = 32

`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.

`bossdata.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.

`bossdata.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 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

9.1.6 bossdata.plate module

Access BOSS plate data products.

class bossdata.plate.**FrameFile**(*path*, *index=None*, *calibrated=None*)

Bases: object

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 spectograph. 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 spectograph. 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*)

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 spectograph. 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.
- **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 `bossdata.plate.`**`Plan`**(*path*)

Bases: `object`

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 *`bossdata.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'.
- **`ftype`** (*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*)

Bases: `object`

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*)

Bases: `object`

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

`bossdata.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

9.1.7 bossdata.plot module

Support for plotting BOSS spectscopic 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.

`bossdata.plot.`**`by_fiber`**(*data*, *mask=None*, *subsets={}*, *percentile_cut=0.0*, *plot_label=None*, *data_label=None*, *default_options={'marker': 'o', 'lw': 0.5, 's': 60}*)

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.

`bossdata.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={'marker': 'o', 'lw': 0.5, 's': 60}*, *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.

Python Module Index

b

Symbols

A

B

C

D

E

F

G

H

I

K

N

O

P

Q

R

S

T

U

V

W

X

Z